Computer Aided Medical Procedures
Prof. Dr. Nassir Navab

Dissertation

Graph Deep Learning for Healthcare Applications

Anees Babasaheb Kazi

Fakultät für Informatik
Technische Universität München

Technische Universität München
Fakultät für Informatik

Graph Deep Learning for Healthcare Applications

Anees Babasaheb Kazi

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende(r):	Prof. Dr. Stephan Günnemann
Prüfer der Dissertation:	1. Prof. Dr. Nassir Navab
	2. Prof. Dr. Daniel Rückert
	3. Prof. Michael Bronstein, Ph.D.

Die Dissertation wurde am 24.11.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 08.04.2021 angenommen.

Anees Babasaheb Kazi
Graph Deep Learning for Healthcare Applications
Dissertation, Version 0.0

Technische Universität München
Fakultät für Informatik

Boltzmannstraße 3
85748 and Garching bei München

Part I

abstract

Abstract

In recent times, Graph deep learning has emerged as a powerful machine learning technique that provides a generalized application of deep neural architectures to non-Euclidean structured data successfully. Graph-based learning models have found numerous applications in social sciences, computer vision and graphics, basic sciences and biological sciences. This thesis focuses on successfully integrating graph deep learning with medical applications such as disease prediction for Alzheimer's, Autism, Parkinson's and Brain imaging. The key objective is not only to solve clinical problems but also to address a wide variety of technical challenges still open in the field.

Graph Convolutional Networks (GCNs) can be used in a medical applications by setting the patients in relation to each other with a neighborhood graph, often by associating them semantically through non-imaging medical data. On this graph, patients are considered as nodes, patient similarities are represented as edge weights, and features from e.g., imaging modalities are incorporated through graph signal processing. GCNs then provide a principled manner for learning optimal graph parameters that minimize an objective, for example, disease prediction.

During the whole journey of the thesis, we pose several technical and clinical challenges and solve them with graph deep learning. This thesis starts by investigating the clinical relevance in having GCNs for medical applications such as disease prediction and brain imaging. It is divided into four main chapters dealing with technical challenges such as multiple graph scenario, graph heterogeneity, graph attention for personalized treatment and graph learning. Along with the four primary problems mentioned above, certain secondary problems in the field, such as out of sample extension and dealing with multi-modal datasets are also solved. Medical applications such as disease prediction, brain imaging, finding the relevance of meta factors are shown, and non-medical applications such as point cloud segmentation and zero-shot learning on ImageNet dataset are also considered.

Towards the end, important open questions such as interpretability and robustness of GCNs are discussed as future directions. This thesis is one of the pioneer works in introducing graph deep learning to the medical field and demonstrates that GDL has a great potential in the medical applications due to its high capability to integrate multi-modal complementary data. In the future, further research areas such as mesh analysis, brain graphs connectomes, drug-to-drug interaction etc. could be explored.

Zusammenfassung

In jüngster Zeit hat sich Graph Deep Learning als leistungsstarke maschinelle Lerntechnik herausgestellt, die eine Verallgemeinerung erfolgreicher tiefer neuronaler Architekturen auf nichteuklidische strukturierte Daten ermöglicht. Graphbasierte Lernmodelle haben Anwendung in den Sozialwissenschaften, Computer Vision und Grafik, Grundlagenwissenschaften und Biowissenschaften gezeigt.

Diese Arbeit konzentriert sich auf die Integration von Graph Deep Learning in medizinischer Anwendung wie der Vorhersage von Krankheiten bei Alzheimer, Autismus und Parkinson sowie dem Neuro-Imaging. Ziel ist es nicht nur, klinische Probleme zu lösen, sondern eine Vielzahl von technischen Herausforderungen zu bewältigen, die auf diesem Gebiet ausstehen.

Graph Convolutional Networks (GCNs) können in einfachen Worten in einer medizinischen Anwendung verwendet werden, indem die Patienten mit einem Nachbarschaftsgraphen in Beziehung zueinander gesetzt werden, häufig indem sie semantisch durch nicht bildgebende medizinische Daten verknüpft werden. In diesem Diagramm können Patienten als Knoten betrachtet werden, Patientenähnlichkeiten werden als Kantengewichte dargestellt und Merkmale von z. B. Bildgebungsmodalitäten werden durch Diagrammsignalverarbeitung einbezogen. GCNs bieten dann eine prinzipielle Methode zum Lernen optimaler Diagrammparameter, die ein Ziel, beispielsweise die Vorhersage von Krankheiten, minimieren.

Während des gesamten Prozesses der Dissertation stellen wir uns verschiedenen technischen und klinischen Herausforderungen und lösen sie mit graphentiefem Lernen. Beginnend mit der Untersuchung der klinischen Relevanz von GCNs für medizinische Anwendungen wie Krankheitsvorhersage und dem Neuro-Imaging ist die Arbeit in vier Hauptkapitel unterteilt, die sich mit technischen Herausforderungen wie dem Szenario mit mehreren Graphen, der Heterogenität der Graphen, der Aufmerksamkeit der Graphen für die personalisierte Behandlung und dem Lernen der Graphen befassen.

Zusammen mit den oben genannten vier Hauptproblemen zeigen wir Lösungen für einige Nebenprobleme auf dem Gebiet. Zum Beispiel werden medizinische Anwendungsgebiete wie Krankheitsvorhersage, Neuro-Imaging und die Ermittlung der Relevanz von Metafaktoren gezeigt und nichtmedizinische Anwendungen wie Punktwolkensegmentierung und Zero-Shot-Lernen im ImageNet-Datensatz berücksichtigt.

Gegen Ende werden offene Fragen wie Interpretierbarkeit und Robustheit für GCNs als zukünftige Forschungsrichtungen diskutiert. Diese Dissertation ist eine Pionierarbeit für die Einführung von Graph Deep Learning im medizinischen Bereich und zeigt, dass GDL ein großes Potenzial in medizinischen Anwendungen hat, da es einen besseren Weg zur

Integration multimodaler komplementärer Daten aufweist. In Zukunft könnten Netzanalysen, Gehirngraphen-Konnektome und Wechselwirkungen zwischen Medikamenten untersucht werden.

Acknowledgments

I would like to express my gratitude firstly to my supervisor prof. Dr. Nassir Navab for the extreme support and really extraordinary supervision in my overall development as a Ph.D. candidate as well as a better human being. I am really happy and content with the person I have become over this period. I would like to also thank Dr. Seyed Ahmad Ahmadi for all his support and for teaching me how to be ethical in my research and write exceptional papers. This cannot end without a proper thank you to Dr. Shadi Albarqouni and Prof. Diana Mateus for being patient with me during the early days of my Ph.D. and helping me get started in this wonderful journey I've been on for the past 4 years.

Baba, Nariza, and Arif, thank you for your immense support. Your love and belief in me have been the backbone for my survival during the rough days and made it easy for me to concentrate on my work without distractions. The journey of my Ph.D. would not have been fun enough without the most wonderful friends I made at CAMP. Thank you, Mai, Leslie, Shun-cheng, Ashkan, David, Gerome, Hendrik, and many more. I thoroughly enjoyed all the discussions and deadlines we targeted and achieved together. Last but not the least, thank you to all my friends out of CAMP for being there with me throughout my journey.

Contents

Part II

Introduction

1 Introduction

> *Art is nothing without knowledge.*
>
> — **Jean Mignot**
>
> ...

1.1 Overview

Multi-modal data in health care generally comprises of imaging (Magnetic Resonance Imaging (MRI), function-MRI, Positron Emission Tomography (PET),X-ray etc.) and non-imaging (clinical test, demographics, reports, baseline first visit report etc.) information. Such data is collected together and used by the clinical experts for disease diagnosis and treatment planning. This diverse nature of the data provides complementary information about the patient's condition to make an informed decision.

One brilliant way to incorporate the complementary data into a single pipeline for better clinical outcome, is by using Graph Convolutional network (GCN). An important component in GCN models is to build a population graph, where the graph represents pairwise patient similarities. Very conveniently, either of the complementary data can be used to generate the graph. Such graph will incorporate the hidden relationships between the patients following the complementary angles from different data elements. In recent times, GCNs are employed as a powerful machine learning tool for Computer Aided Diagnosis(CADx) and disease prediction. This setting was first introduced in [Par+17] for a healthcare application in order to address the binary disease classification task. There are other methods in the literature that are used to combine the multi-modal data. Few described in this thesis are early, late and intermediate fusion. These methods use either conventional machine learning techniques or deep learning models for training.

In all the methods mentioned above, the features are not considered at an individual level. For example, in the case of Alzheimer's disease prediction, the age of the patient plays a more important role compared to gender. The method that uses GCNs [Par+17] for disease prediction proposes to combine graphs generated from each non-imaging data elements to generate one graph. During the analysis of GCNs, it was found that GCNs are sensitive towards the input graph structure. Furthermore, in the clinical sense, a graph coming from different features will inherit distinct structure. Eventually, each graph may have a different clinical relevance towards the task.

Therefore, a technique capable of exploiting the individual characteristics of each multi-modal data element is required for better disease prediction. We propose a GC based deep model that

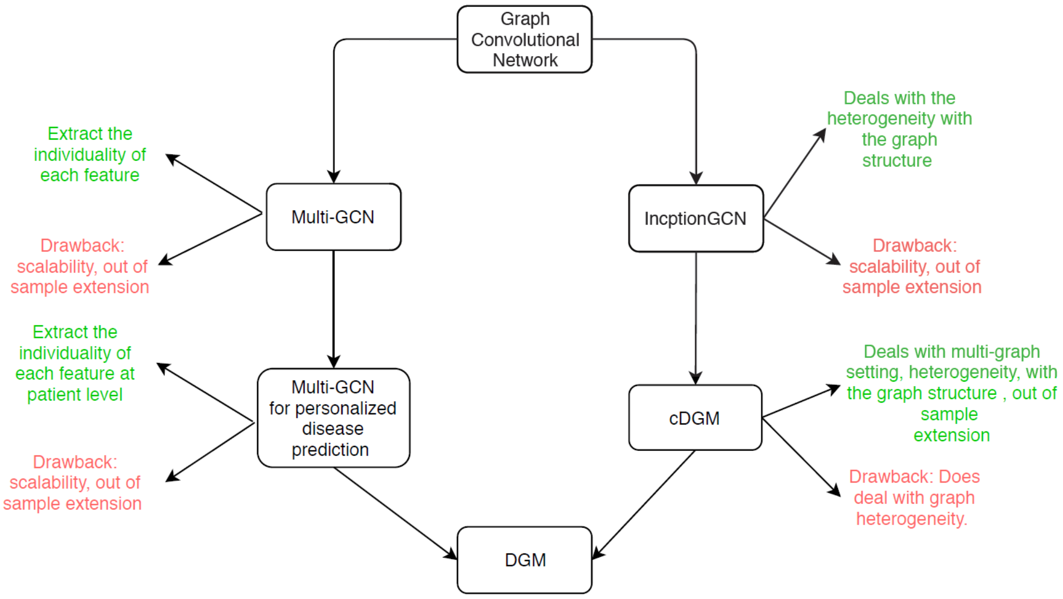

Graph Convolutional Network
Multi-GCN
IncptionGCN
Extract the individuality of each feature
Drawback: scalability, out of sample extension
Deals with the heterogeneity with the graph structure
Drawback: scalability, out of sample extension
Multi-GCN for personalized disease prediction
Extract the individuality of each feature at patient level
Drawback: scalability, out of sample extension
cDGM
Deals with multi-graph setting, heterogeneity, with the graph structure , out of sample extension
Drawback: Does deal with graph heterogeneity.
DGM

Fig. 1.1. The entire thesis at one glance

considers the distinctiveness of each element of the multi-modal data. Further, we propose a novel self-attention layer that weights each element of the non-imaging data by exploring its semantic relation to the underlying disease. The proposed method is superior in comparison to other state-of-the-art methods, in terms of computational speed and performance. The weights learned by the self-attention layer show a clinical co-relation with the disease considered. Do the population level weights for each element hold good at patient level? Clinical experts inherently include and infer from the complementarity nature of multi-modal data during diagnosis and treatment planning. They often consider a varied order of importance for this heterogeneous data for patient level personalized decisions.

Current learning-based methods, including the method mentioned in the previous paragraph, have achieved better a performance by paying equal attention to all of the individual information. Only a few methods have focused on patient-specific attention learning schemes for each modality. Towards this, we extend our multi-graph model which focuses on learning patient-specific order of importance for the multi-modal data elements. In this technique, an LSTM-based attention mechanism is incorporated with graph convolutions to learn patient level attention. This method has two advantages, 1) GC learns the class specific representation for the nodes and 2) LSTM based attention mechanism optimally weights each non-imaging elements at patient level. This combination provides better patient level disease prediction as output.

While exploring the multiple graph scenario and analysing the impact of graph structure, it is interesting to notice that graphs are not necessarily uniform in terms of edge weights, degree of nodes etc. In other words, graphs are super heterogeneous structures. In this thesis, such non-uniformity is termed as ’intra-graph heterogeneity’. In the medical domain, GCNs provide a principled and a versatile technique to integrate multi-modal data. For the scenario mentioned above, a graph based deep learning method for disease prediction is introduced. In order to handle both inter and intra-graph structural heterogeneity during training, new spectral domain based geometric ’inception modules’ are defined. The architecture developed this way is named ’InceptionGCN’. In these modules, filters with different kernel sizes are designed. Further, analyses on the behaviour of conventional GCNs and InceptionGCN is provided for varying input scenarios on simulated data.

Although some key challenges in employing Graph Deep Learning in healthcare applications are solved by using the methods mentioned above, there still remain a few open questions associated with their technical aspects. Some of these open questions are:

- **Scalability**: What if the number of graphs in a multiple-graph scenario increases to a computationally untrackable number?

- **Inductive setting**: All the methods above use transductive setting which means including training and testing samples during model training. How to handle out of sample extension?

- All the methods above, use predefined graphs, which are computed during the preprocessing of the data. However, such graphs might not be optimal, especially in the medical domain as there is no gold standard procedure to construct a graph. Can the model learn the latent graph relevant to the task at hand?

The comprehensive solution to the open questions mentioned above is provided in the final model. A vital aspect of this model is to learn a latent graph for a given population, where this graph represents the pair-wise similarities among patients. As mentioned in the earlier section, the graphs have been defined manually based on non-imaging features. Such way of graph construction lacks a clear definition and requires careful tuning. In the final work, a novel way to learn a semantic and a clinically relevant graph is detailed. The output graph is proven to be optimal for a primary task such as disease prediction. The proposed model trains in an end-to-end fashion along with learning the latent graph dynamically.

In contrast to commonly employed GCN techniques, the proposed method uses a spatial approach and is also capable of incorporating inductive setting. Furthermore, this method is generic and scalable. Significant improvements over conventional models are demonstrated using this approach, thus emphasizing the importance of graph learning for more precise and robust inferences in medical applications using GCNs.

1.2 Motivation

Deep Learning (DL) is a sub-type of machine learning that leverages a layered algorithmic architecture to analyze data. Data is filtered or passed through a cascade of multiple layers, with each subsequent layer using the output obtained from the previous layer for generating its output. DL models can consequently get more precise as the data grows, progressively learning from previous outputs to improve their ability to make connections and correlations. A simple example of a deep learning architecture is shown in the Fig. 1.2. Deep learning is loosely based on how biological neurons communicate with each other in the brains of animals to process information. In basic deep learning models, each layer may be assigned a specific portion of a transformation task and data might traverse the layers multiple times to refine and optimize the final output. These "hidden" layers serve to perform the mathematical translation tasks that turn raw input into meaningful output. In recent literature, deep learning has achieved enormous success in a large variety of applications. This emerging area of machine learning has been quickly developing and has been introduced to most conventional fields of application, as well as to several new fields with more possibilities. The state-of-the art performance is achieved by using deep learning when compared to traditional machine learning approaches in the fields of image processing, computer vision, speech recognition, machine translation, art, medical imaging, medical information processing, robotics and control, bioinformatics, natural language processing, cyber security, and many others.

Machine Learning (ML) is a subset of AI that has revolutionized several fields over the last few decades. In recent times, Neural Networks (NN), which is a sub-field of ML has found the presence of Deep Learning (DL) ubiquitous. DL has been exhibiting remarkable performance in almost every area of application. Learning is a process consisting of estimating parameters of the model such that a given task can be performed by the trained model (algorithm). For example, in Artificial Neural Networks (ANN), the parameters are the weight matrices. DL, on the other hand, consists of several layers in between the input and the output layers. This allows for multiple stages of non-linear information processing units to extract representations for feature learning and pattern classification.

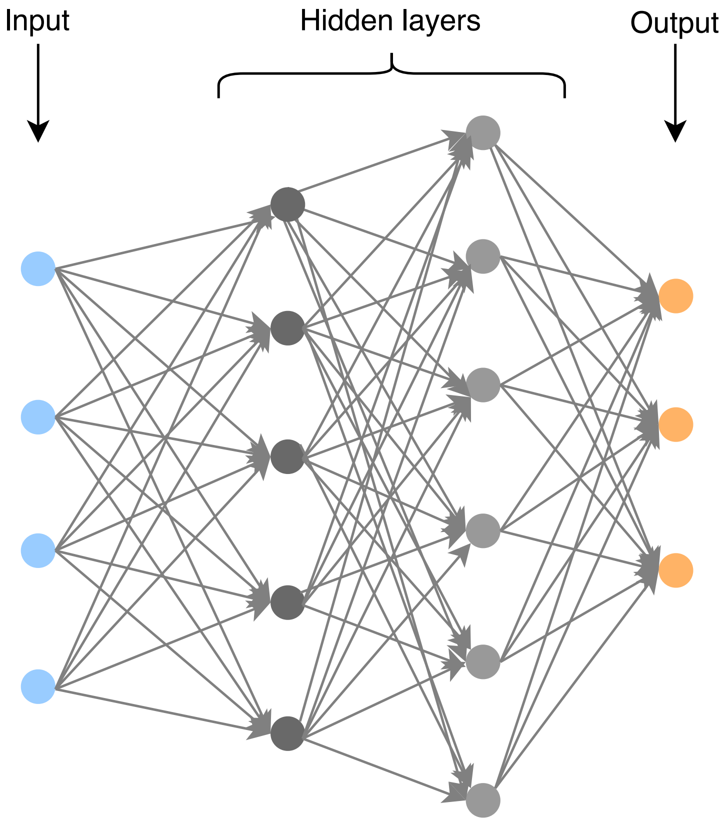


Fig. 1.2. A simple neural network that shows the message passing technique used in deep learning models. The information is transformed and passed forward at each hidden layer (shown in gray).

1.2.1 Impact of AI on research today

AI can be vaguely defined as computational capability of machines to perform intellectual processes, typical of human cognitive functions such as learning, reasoning and problem solving. Since the advent of AI, its application has vastly impacted almost all areas of medicine and healthcare and is bound to continue in the coming years as well. It has also resulted in significantly transforming the manner in which medicine is practiced in this day and age of technology; revolutionizing conventional methods for diagnostic , therapeutic decision-making and treatment-response evaluation. It will progressively play a key role in preventive medicine by providing more precise and effective clinical decisions due to which it will become an integral component of healthcare systems worldwide. Perhaps the most promising role for AI would be to augment the role of human experts and to contribute towards enhanced efforts for precision medicine. [He+19].

In medicine and healthcare, AI is poised to play an increasingly influential role. This is due to advancements in computing resources, learning algorithms, and the availability of huge datasets from wearable health devices and medical records. AI's healthcare industry is rising at a rate of 40% and is projected to hit $6.6 billion by 2021 [SM20]. As learning algorithms interact with training data, they are getting more reliable and precise, allowing newer insights into diagnostics, care choices, and patient outcomes.[Est+19]. Deep learning models are achieving expert-level accuracy at a broad variety of medical tasks such as identifying moles from melanomas [Est+17; Hae+18], diabetic retinopathy, cardiovascular risk, and referrals from fundus [Gul+16; Pop+18; De +18]. Furthermore, they are also being used for abnormality detection in Optical Coherence Tomography (OCT) eye images **??**, lesion detection in mammograms [Koo+17], and spinal analysis with Magnetic Resonance Imaging (MRI) [JKZ16]. It has also been demonstrated that a single deep-learning model is effective in diagnoses across multiple medical modalities [JKZ16; Ker+18].

Natural Language Processing (NLP) is another sub-branch of DL that focuses on analyzing and inferring information from text and speech. Recurrent Neural Networks (RNN) based algorithms play an important role for NLP based applications[SVL14]. NLP based methods have been widely applied in healthcare for processing the text such as surgical reports [Sto+14], [Raj+12] , radiology reports [Men+05], narrative reports [Hri+03], [RGH08], [RGH08], [Byr+14]. [FJD13] clinical reports of severe conditions, [And+15] progress notes and [Haz+05] patient questionnaire [Wag+12]. Some of the works also use laboratory reports [FM08], chest X-ray reports[Wag+12], baseline first visit reports [Fis+00]. Treatment plans and patient summaries are used in [JJB12], short free text [Hun+08]; pathology reports [Iml+13] and electronic health records are used in [Wat+11] and [Por+09].

1.2.2 Importance of multi-modal data in healthcare

A key limitation of the methods mentioned above is that, they use single modality of data such as either images or text. However, their algorithmic performance is compared to that of human experts. This often could be considered as a shortcoming for Computer Aided Diagnosis(CADx) systems for the diagnostic tasks as the human expert in the real-world clinical settings has access to both the medical imagery and supplemental data, including the patient history, health record, additional tests, patient testimony, etc [**esteva2019guid**]. Such data is called *multi-modal data*. A lot of research has been done [Cai+19] towards such multi-modal data. In order to learn more desirable feature representations for each patient, it is essential to analyse multi-modal data. Many DL based algorithms have been been successfully applied to multi-modal data for a wide variety of tasks such as disease diagnosis [Ma+18a], clinical prediction [Xu+18b], treatment and planning [Rie+08]. Recurrent Attentive and Intensive Model (RAIM) [Xu+18b] analyzed both continuous monitoring data and discrete clinical events to predict physiological decomposition and length of stay in the hospital. In order to investigate the complex correlations between the features and labels of Alzheimer's disease diagnosis, ML-MVC [Qia+19] proposed to model multi-view inputs and generate a latent representation. [Qia+19]. Furthermore, some of the other conventional ways to fuse multi-modal data are described below.

Early fusion or data-level fusion: Data-level fusion is a common method of combining multiple modalities of data before conducting the analysis. This method is known as early fusion or data-level fusion. Two possible approaches of early fusion technique are proposed in [Kha+13]. The first approach is combining the data by removing the correlation between two sensors. The second approach is to fuse data at its lower dimensional common space. There are several mathematical solutions including Principal Component Analysis (PCA), Canonical Correlation Analysis (CCA) and Independent Component Analysis (ICA) that can be used to achieve either or both approaches. Early fusion techniques are more applicable to data obtained from sensors. Poria et al [PCG15] implemented early stage data fusion by concatenating the features in multi-modal data. A vital drawback of early stage data fusion is that a significant amount of data could get deducted from the modalities when projecting the data to a lower dimensional embedding.

Late fusion or decision level fusion: In this technique, each modality is independently processed and then fused at the decision-making stage. Late data fusion is inspired by ensemble classifiers [Kun14]. This technique is much simpler than the early fusion method, particularly when the data sources are significantly diverse from each other in terms of the units of measurement. Late fusion often performs better compared to early fusion because the errors of each modality are handled independently.

Intermediate fusion: Among the fusion methods, this method is the most flexible as it allows the data fusion at various stages of the model training. In intermediate fusion, the input data is changed into a higher level representation (feature) through multiple layers. Although it is possible to fuse several modalities at different stages, the training may lead to model over-fitting. Unlike early level fusion and late fusion, intermediate fusion offers flexibility to fuse features at different depths. **Other fusion techniques:** [Kar+14] provides a "slow-fusion" network. In this technique, the video features are slowly fused across multiple fusion layers during training. This approach shows a better performance in large-scale video stream classification problem. Along the same line, other research [Nev+15] shows a gradual fusion method which involves fusing highly correlated input modalities initially and then less correlated ones progressively. This work showed a state-of-the-art performance in communicative gesture recognition.

Further, with the richness of imaging and non-imaging data, it is essential to have models that are capable of representing a potentially large population and exploiting the heterogeneous information associated with it, along with the correlation that may exist between patients and the heterogeneous data. A novel way has been recently introduced which is based on graphs representing the population. Graphs provide a natural way of representing populations and their similarities. In such a graph setting, the information of each patient (features from imaging and non-imaging modalities) is represented by a node. Pairwise similarities between the patient's information is represented as edges between these nodes. Such graph based models provide a powerful and a clinically semantic setup for population-level analysis of multi-modal data. Unlike linear classifier which relies solely on imaging feature vectors, Graph Convolutional Networks (GCNs) model the interaction and similarities between the patients through a graph that encodes these pairwise similarities.

The primary objective of this thesis is to facilitate handling of multi-modal data using GCNs. Apart from showing how GCNs can be leveraged for the multi-modal data analysis, multiple technical challenges from the field of Graph Deep Learning (GDL) - a super set of GCNs, are also solved. [Par+17; DBV16] describes how GCN can be used for population level analysis using a graph consisting of each patient as a node. First question that is answered in this thesis is 'What if we have possibility of multiple graphs?' Answering this question and analysing the solution is crucial, because having more than one graph for the same set of population helps gather information under a different light of each graph. This problem is solved in chapter 3. The thorough analysis for the need of multiple graph is demonstrated, followed by the model which intelligently merges the multiple graphs together for personalized disease prediction is demonstrated in chapter 4. The chapter that follows this deals with intra-graph heterogeneity. The main question is 'Given a graph, do the conventional GCN based methods undertake the heterogeneity with a graph?' Here, heterogeneity means the density of edges between the nodes. The efforts in this chapter are primarily invested on building a model that focuses on the structure of the graph. In all the chapters, multiple technical and clinical problem with GCNs are solved. All the methods presented here work on predefined graphs. The core challenge that the last chapter addresses is specific to graph learning. The detailed outline of the thesis is given the section below.

1.3 Thesis outline

For the next chapters, we provide an overview and the related literature survey associated to them. Most of the methods and analysis of this thesis are published or are under review for major conferences or journal of the suitable field. A complete list of papers is listed in Appendix A.

Chapter 2. We build the required mathematical foundation inspired from graph signal processing. This is required to understand basic working of graph convolutional networks. We detail the spatial and spectral approach followed by the relevant literature survey.

Chapter 3. Based on [Par+17], this chapter discusses 1) influence of graph structure on training of GCN and its output and 2) a novel model that handles multiple Graph Scenarios for disease prediction. An attention mechanism that learns the relevance of each graph towards the task is also explained. The related works are:

- Parisot, S., Ktena, S.I., Ferrante, E., Lee, M., Moreno, R.G., Glocker, B. and Rueckert, D., 2017, September. Spectral graph convolutions for population-based disease prediction. In International conference on medical image computing and computer-assisted intervention (pp. 177-185). Springer, Cham.

- Kazi, A., Shekarforoush, S., Kortuem, K., Albarqouni, S. and Navab, N., 2019, April. Self-attention equipped graph convolutions for disease prediction. In 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019) (pp. 1896-1899). IEEE.

Chapter 4. Although multi-GCN is capable of employing the distinctiveness of each features and learning the semantic and hierarchical weighting of each graph, the patient level relevance of each feature might be different. In this chapter we introduce an RNN based attention mechanism that evaluate the non-imaging features at patient level. This method is applied to personalised disease prediction task. The related works are:

- Kazi, A., Shekarforoush, S., Krishna, S.A., Burwinkel, H., Vivar, G., Wiestler, B., Kortüm, K., Ahmadi, S.A., Albarqouni, S. and Navab, N., 2019, October. Graph convolution based attention model for personalized disease prediction. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 122-130). Springer, Cham.

Chapter 5. Another interesting aspect regarding the heterogeneous structure within the graph and its influence towards the performance of GCNs is explored in this chapter. We provide a robust solution for learning better node level representations which boost the model performance. The related works are:

- Kazi, A., Shekarforoush, S., Krishna, S.A., Burwinkel, H., Vivar, G., Kortüm, K., Ahmadi, S.A., Albarqouni, S. and Navab, N., 2019, June. InceptionGCN: receptive field aware graph convolutional network for disease prediction. In International Conference on Information Processing in Medical Imaging (pp. 73-85). Springer, Cham.

Chapter 6. All the methods above still face some common limitations such as 1) scalability: the models may computationally explode if the number of nodes and number of graphs exceed certain limit, 2) out of sample extension: all the methods work in transductive setting, making it difficult to apply the model to a previously unknown node. 3) compulsion on having pretrained graph: the methods so far have a constraint of having the graph(s) defined during the preprocessing. In this work, we propose a model that accomodates all the three limitations in one model and shows superior performance. The related works are:

- Cosmo, L., Kazi, A., Ahmadi, S.A., Navab, N. and Bronstein, M., 2020, October. Latent-Graph Learning for Disease Prediction. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 643-653). Springer, Cham. Kazi, A., Cosmo, L., Navab, N. and Bronstein, M., 2020. Differentiable Graph Module (DGM) Graph Convolutional Networks. arXiv preprint arXiv:2002.04999.

Chapted 7. Finally we discussion some open questions in the field and conclude the thesis with future directions.

Part III

Background and Challenges

Background and Challenges 2

> *If we knew what it was we were doing, it would not be called research, would it?*
>
> — **Albert Einstein**
>
> ...

2.1 Background

The tremendous success of deep learning has boosted research on pattern recognition and data mining. Deep learning aims at learning complicated semantic concepts by fetching them from simpler ones in a hierarchical or a multi-layer manner. Different deep learning paradigms cover a broad range of applications such as object detection [Zha+19; Hu+18], Segmentation [LXL19; Yao+19], speech recognition [EKK11] etc. and have revolutionized the approach that was used earlier, wherein several handcrafted features were used to extract semantic concepts [Wu+20]. As shown in Fig. 2.3 (a) a regular CNN takes an image as input, applies a filter on it and produces an activation/ feature map as output. A peculiar quality of all the three components mentioned above are 1) **Images** are represented as regular grid in Euclidean space where each pixel to be processed by the filter has a fixed number of equidistant neighbors 2) **Filters** are represented in the form of grid and are shared over the input dataset. Filters run over the image in a zigzag manner depending upon the different strides. Output of such a convolution is also a grid. Such a setup is shown in Fig. 2.1. Deep learning models have been objectively successful particularly for speech, image, and video signals where data lies on a regular grid. More recent efforts are focused on trying to apply learning on non-Euclidean geometric data. Many applications require such data for example, applications to point clouds [Wan+19b], [LS18] and [Te+18], human action recognition [Jai+16], [YXL18], text classification [KW16a], [HYL17a], [Vel+17] etc.

Non-euclidean based data appears in numerous applications. For example, in social networks each user can be considered as a node on a population based social graph. In genetics, gene expression data are modeled as signals defined on the regulatory network. In neuroscience, anatomical and functional structures of the brain can be represented as a graph for connectome analysis. In computer graphics and vision, three-dimensional (3-D) objects are modeled as Riemannian manifolds. [Bro+17]

In the following section, the significance of a graph neural networks is illustrated. The conventional artificial neural networks like CNNs are incapable of handling the graph input because CNNs need specific ordering of nodes (pixels) which is absent in graph inputs. As graphs do not have specific ordering, all possible orders have to be considered for obtaining

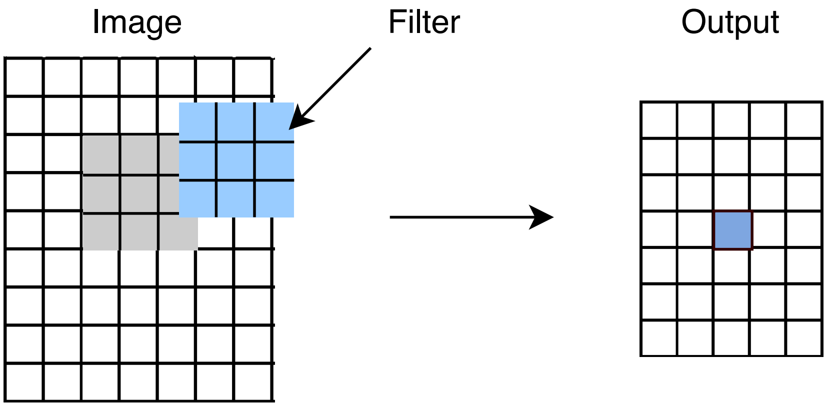


Fig. 2.1. In this figure a conventional way of convolution on an image is shown. Image (input), filer and the output all regular grids.

the optimal results, which could either be redundant or computationally expensive. GNNs solve this problem by propagating on each node respectively, ignoring the input order of nodes. Therefore, the output of GNNs does not depend on the node order on the input. Additionally, an edge in a graph represents a certain relationship between two nodes which could either be prior knowledge or which could be discovered. In the CNNs, this relationship is just the euclidean distance based pixel neighborhood. Furthermore, incorporating cognitive like intelligence to artificial neural networks is a very important research topic for future AI. The reasoning process that occurs in a human brain is almost based on the relationship based graph which could be mapped by the diverse relations of knowledge and experiences a human brain has accumulated. Conventional neural networks such as generative models have shown the ability to generate synthetic data such as images and documents. This is achieved by learning the distribution of the data. However, these networks still cannot learn the reasoning graph from large experimental data. GNN on the other hand explores to generate the graph from unstructured data like scene images or the demographic in a social network which could be a potential possibility for future high-level AI. Recently, it has been proven that an untrained GNN with a simple architecture also performs well [Zho+18].

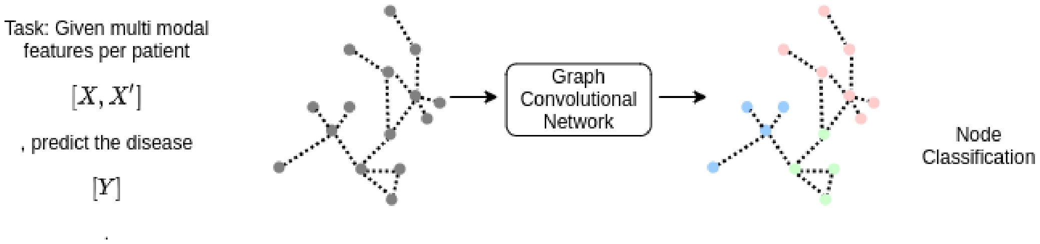

Task: Given multi modal features per patient
$[X, X']$
, predict the disease
$[Y]$
Graph Convolutional Network
Node Classification

Fig. 2.2. In this figure a basic setup of Graph Convolutional Network is shown. Where input is a multi-modal dataset. X and X' are the different modalities. Y is the label to be predicted for the test nodes.

2.2 A Brief History of Graph Deep Learning (GDL)

Geometric Deep Learning (GDL) is a sub-field of deep learning with techniques that are able to generalize (structured) deep neural models to non-Euclidean domains, such as graphs and manifolds [Bro+17].

One of the very first works in this direction was done by Sperduti et al. (1997) [97] first applied neural networks to Directed Acyclic Graphs (DAGs). This concept of graph neural networks was started by Gori et al. (2005) [GMS05] and further elaborated in Scarselli et al. (2009) [Sca+08] and Gallicchio et al. (2010) [GM10]. Message passing technique within the neighborhood is used for node' representation learning in the above works.
GNNs can be categorised mainly into two types the spectral-based and the spatial-based approaches. One of the first works that focused on spectral-based ConvGNNs was done by Bruna et al. [Bru+13] which leveraged the spectral graph theory to perform graph convolution. Following this research, many other advanced works were presented in the spectral-based ConvGNNs such as [DBV16; KW16a; Lev+18]. The research of spatial-based ConvGNNs started much earlier than spectralbased ConvGNNs.
Micheli et al. 2019 [24] first addressed graph mutual dependency by architecturally composite nonrecursive layers while inheriting ideas of message passing from RecGNNs.
However, this work did not gain enough attention until recently after which many other spatial-based ConvGNNs (e.g., [AT16; NAK16; Gil+17]) emerged. In addition to that, many architectural variants have also been developed in the past few years such as graph auto-encoders (GAEs) [KW16b], Spatial-Temporal Graph Neural Networks (STGNNs)[Wan+20] and Graph Attention Networks [Wan+19a; Wu+20]. In the next section the mathematical details of both spectral and spatial GCN are provided.

2.2.1 Mathematics of Graph Convolutional Networks (GCNs)

Notation

As a convention in this thesis, the bold uppercase characters denote matrices and bold lower-case characters denote vectors. Unless particularly specified, the notations used in this thesis are illustrated in Table I. The minimal set of definitions required to understand GCNs are defined below. A graph is defined as $G = (V, E)$ where V is the set of nodes (vertices) and E is the set of edges. Let $v_i \in V$ denote a node and $e_{ij} = (v_i, v_j) \in E$ to denote an edge between v_i and v_j. The neighborhood of a node v is defined as $\mathbb{N}(v) = \{u \in V \,|(u, v) \in E\}$. The adjacency matrix $\mathbf{A}$ is a $N \times N$ matrix with $\mathbf{A}_{ij} = 1$ if $e_{ij} \in E$ and $\mathbf{A}_{ij} = 0$ if $e_{ij} \notin E$. A graph may have node attributes X, where $X \in R^{N \times d}$ is a node feature matrix with $x_i \in R^d$ representing the feature vector of a node i.

2.2.2 Affinity graph construction:

The construction of an affinity graph is crucial to accurately model the interactions among the patients and should be designed carefully. The affinity graph $G = (V, E, W)$ is constructed on

Tab. 2.1. All the notation to explain the spatial and spectral version of GCN.

$\mathbf{G}$	Graph
$\mathbf{V}$	Set of Nodes
$\mathbf{E}$	Set of Edges
v_i, v_j	$i_t h$ and $j_t h$ node in the population
e_{ij}	Edge between i_{th} and j_{th} node
$\mathbb{N}(v)$	Neighborhood of node v
$\mathbf{A}$	Adjacency matrix
$\mathbf{N}$	Total number of nodes
$\mathbf{x_i}$	Feature vector belonging to node v_i
$\mathbf{W}$	Weight matrix representing the weights of each edge
η_i	Non-imaging feature element for node i
β	Threshold to select the edge
ρ	Correlation distance
σ	Kernel representing width of the Gaussian
$\mathbf{L}$	Graph Laplacian matrix
$\mathbf{I}$	Identity matrix
$\mathbf{D}$	Degree matrix
$\mathbf{U}$	Eigen vector matrix of the graph Laplacian
$\mathbf{\Lambda}$	Eigen values matrix
$\mathbf{F}$	Fourier transform
g_θ	Learnable function with parameters θ
δ_i	Kronecker delta function
$\mathbf{H}^l$	Output activation matrix for l^{th} GC layer
w_{ij}	element of the weight matrix W representing edge weight between node i and j
s_v	Concatenated features for personalised attention for node v
ω, b	The weights and biases of the dense layer
$d_G(i,j)$	The computed shortest path distance between v_i and v_j
Ω	The sum of all edge weights on the shortest path from x_i to x_j
k	k nearest neighbors

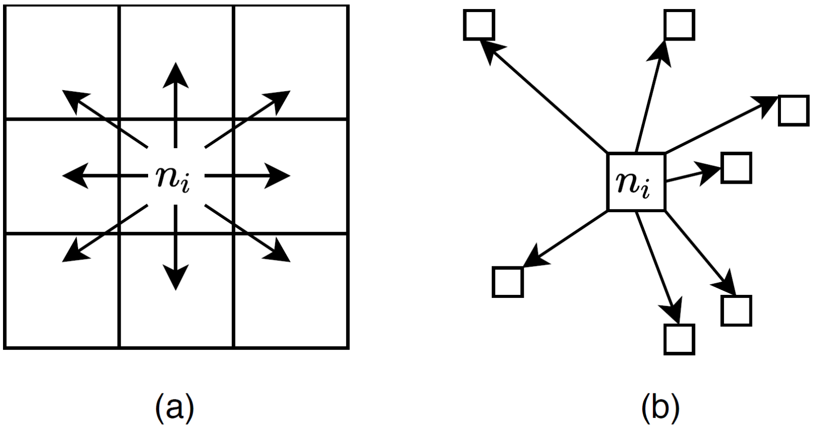


Fig. 2.3. (a) A small 3 × 3 part of an image with central pixel as node n_i. Each pixel in an image can be treated as n_i with fixed number of equidistant neighbors. (b) A similar scenario in the graph data where each node may or may not be connected to equidistant points.

the entire population (including training and testing samples) of the patients, where $|V|=$ N vertices, E are the edge connections of the graph and W are the weights of the edges. Considering each patient as a node n_i in G, this graph incorporates the similarities between the patients with respect to the non-imaging data η. The features $x_i \in \mathbb{R}^N$ at every node n_i are fetched from imaging data.
First, a binarized edge graph $E \in \mathbb{R}^{N\times N}$ is constructed representing the connections. Mathematically, E can be defined as

$$E_{i,j} = \begin{cases} 1 & if\,|\eta_i - \eta_j| < \beta \\ 0 & otherwise \end{cases} \tag{2.1}$$

where η_i and η_j are the values of the non-imaging element for nodes i and j respectively, and β is the threshold for the corresponding element. The weight matrix $W \in \mathbb{R}^{N\times N}$ weights the edges based on the correlation distance between the features at every node. The weight matrix elements are defined as $w_{i,j} = Sim\,(x_i, x_j)$,where $Sim(x_i, x_j) = exp(-\frac{[\rho(x_i,x_j)]^2}{2\sigma^2})$ with ρ being the 'correlation distance' and σ being the width of the kernel. This weight computation is identical to the procedure described in [Par+17] for a fair comparison. The final affinity matrix A is constructed as $A = W \circ E$ with $\circ$ being the Hadamard product.

In the next section, the mathematical details of the spatial approach for computing graph convolution is provided.

2.3 Spatial approach

Similar to the convolutional operation of a conventional CNN on an image, spatial Graph Convolutional (GC) methods define graph convolutions based on spatial relations that may exist between the nodes. Consider image as a special case of graph with each pixel being a node. Each pixel is directly connected to its one-hop neighboring pixels as shown in Fig. 2.3 (a). During convolution a learnable filter is generally applied, for instance, to such a 3×3 patch shown in the Fig. 2.3 (a), by taking the average of the weighted values of n_i which is the central pixel and its neighbors. Similarly, in the spatial approach of graph convolutions the node feature of the central node is convoluted with its neighbor's features to get the output representation of the central node. Similarly, in the spatial graph convolution approaches the central node representation is convoluted with its neighbors' representations to derive the updated representation for the central node, as illustrated in Fig. 2.3 (b). The message passing in the spatial graph convolutional operation is done by propagating the node information along the edges.

Neural Network for Graphs (NN4G) [Mic09], is one of the first works to present spatial GNN. It learns the mutual dependency between the graph and the features through a neural architecture with independent learnable parameters at each layer. In this work, graph convolutions are performed by directly summing up a node's neighborhood information. It also leverages residual and skip connections to retain the information over each layer. Hence, NN4G obtains its next layer node representations by,

$$h_v^{(k)} = f(W^{(l)^T} x_v) + \sum_{i=1}^{l-1} \sum_{u \in N(u)} \Theta^{(l)^T} h_u^{(l-1)} \tag{2.2}$$

where $f()$ is an activation function and $h(0) = 0$. Equation 2.2 can also be written in a matrix form:

$$H^l = f(XW^l + \sum_{i=1}^{l-1} AH^{l-1}\Theta^l) \tag{2.3}$$

One drawback of NN4G is that, it uses unnormalized adjacency matrix which can cause high variance in the output representations of the nodes.

Two other important works that define spatial graph convolutions is Graph Attention Network (GAT) [Vel+17] and GraphSage [HYL17a]. Unlike GraphSage [HYL17a], GAT [Vel+17] claims that relevance of neighboring nodes to the central node for a particular task are neither identical nor pre-determined like GCN [KW16a]. GAT proposes an attention mechanism to learn the relative weights between two connected nodes. The spatial graph convolutional operation presented here is defined as,

$$h_v^{(l)} = \sigma(\sum_{u \in \mathbb{N} \cup v} \alpha_{u,v}^{(l)} W^{(l)} h_u^{(l-1)}) \tag{2.4}$$

where $h_v^{(0)} = x_v$. The attention weight $\alpha_{u,v}^l$ measures the edge weight between the node v and its neighbor u:

$$\alpha_{u,v}^{(l)} = softmax(g(a^T \left[W^l h_v^{l-1} \| W^l h_u^{l-1} \|\right])) \tag{2.5}$$

where g(·) is a LeakyReLU activation function and a is a vector of learnable parameters. Finally, a softmax function is applied to the output of the GAT layer that sums up the attention weights to one over all neighbors of the node v. A multi-head attention is also proposed in this work to increase the model's expressive capability. GAT shows an impressive improvement over GraphSage on node classification tasks.

2.3.1 Literature survey on spatial approaches for GCN

. NN4G [Mic09] is the first and GAT [Vel+17] is the most important spatial graph convolution technique. In this sub section a small literature survey on other spatial graph convolutional approaches is provided. Diffusion Convolutional Neural Network (DCNN) [AT16] considers graph convolutions as a diffusion process. In this method, it is assumed that the information transfer between the connecting nodes happens with a certain transitional probability. Therefore, the information distribution can reach equilibrium after several layers. Similar to NN4G, Contextual Graph Markov Model (CGMM) [BEM18] proposes a probabilistic model while maintaining spatial locality. Thus, CGMM has the advantage of probabilistic interpretability. Diffusion Graph Convolution (DGC) [Li+17] sums up outputs instead of concatenating them at each diffusion step. Another shortest path based PGC-DGCNN [TNS18] has been proposed that increases the contributions of distant neighbors. In this method a shortest path adjacency matrix A is defined where, if the shortest path between a node v and a node u is of length j, then $A_{uv} = 1$ otherwise 0. Partition Graph Convolution (PGC) [YXL18] partitions a node's neighbors into Q groups. Later Q adjacency matrices are constructed according to the defined neighborhood by each group. Subsequently, PGC applies GCN [Kip+18] for the graph convolution operation with a different parameter matrix to each neighbor group and sums up these results. Next, Message Passing Neural Network (MPNN) [Gil+17] gives a generic framework of spatial-based ConvGNNs. Simlar to NN4G, the message passing process between the nodes is done along the edges directly. MPNN is used to generate many existing GNNs just by assuming different forms of learnable parameters, such as [Kip+18], [Duv+15], [Kea+16], [ACT+17]. Gated Attention Network (GAAN)[Zha+18b] proposes a self-attention mechanism which computes an additional attention score for each attention head, unlike GAT that assumes the contributions of all attention heads to be equal. On top of applying graph attention spatially, GeniePath [Liu+19b] proposes an LSTM like gating mechanism to control information flow across the graph convolutional layers.

Mixture Model Network (MoNet) [Mon+17] proposes node pseudo-coordinates to determine the relative position between a node and its neighbor. In this way, different weights are assigned to a node's neighbors. A weight function is then used to map this relative position to a relative weight between the two nodes. MoNet is a very generic framework which can be used to derive other existing approaches for manifold learning such as Geodesic CNN (GCNN) [Mas+15], Anisotropic CNN (ACNN) [Bos+16], Spline CNN [Fey+18]. Further, the graph is constructed using a non-parametric weight functions. MoNet can be used to derive other methods such as GCN [Kip+18], DCNN[AT16]. MoNet also proposes a Gaussian kernel with learnable parameters to learn the weight function adaptively. In PATCHY-SAN [NAK16] the neighbors are ordered according to their graph labels and the top q neighbors are selected. Since each node now will have a fixed number of ordered neighbors, graph-structured data can be converted into grid-structured data easily. PATCHY-SAN applies a standard 1-D convolutional filter to aggregate neighborhood feature information where the order of the filter's weights corresponds to the order of a node's neighbors. One of the drawback of this method is that the ranking criterion only considers graph structures, which requires heavy computation for data processing. Another method called Large scale Graph Convolutional Network (LGCN) [GWJ18] ranks the neighbors of a node based on node feature information.

Now that the mathematical description and the literature survey of the spatial based method has been detailed, the spectral approach for graph convolution is provided in the next subsection which will then be followed by a similar literature survey.

2.4 Spectral approach

Spectral-based methods have a solid mathematical foundation in graph signal processing [Shu+13; SM13]. In such methods, graphs are considered undirected. An undirected graph is mathematically represented as the normalised graph Laplacian matrix. It can be defined as $\mathbf{L} = \mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{\frac{1}{2}}$ where, $\mathbf{D}$ is a diagonal matrix of node degrees $\mathbf{D}_{ii} = \sum_j (\mathbf{A}_{i,j})$. The normalized graph Laplacian matrix being real symmetric positive semi-definite can be factored as $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U} = [\mathbf{u_0}, \mathbf{u_1}, \mathbf{u_2}, ..., \mathbf{u_{n-1}}] \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ is the matrix of eigenvectors and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues (spectrum), $\mathbf{\Lambda}$ is the the diagonal matrix of eigenvalues (spectrum), $\mathbf{\Lambda}_{ii} = \lambda_i$. The eigenvectors of the normalized Laplacian matrix form an orthonormal space, that is $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. In the graph signal processing, a graph signal $\mathbf{x}_i \in \mathbb{R}^d$ is a feature vector of i^{th} node of a graph where $\mathbf{x}_i$ is the value of the i^{th} node. The Fourier transform to a signal $\mathbf{x}$ in graph signal processing is defined as $F(x) = \mathbf{U}^T\mathbf{x}$, and the inverse graph Fourier transform is defined as $F^{-1}(\hat{\mathbf{x}}) = \mathbf{U}\hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ represents the resulted signal from the graph Fourier transform. Elements of the transformed signal $\hat{\mathbf{x}}$ are the coordinates of the graph signal in the new space, such that the input signal can be represented as $\mathbf{x} = \sum_i \hat{\mathbf{x}}_\mathbf{i}\mathbf{u}_i$ which is exactly the inverse graph Fourier transform. Now the graph convolution of the input signal $\mathbf{x}$ with learnable function $g_\theta = diag(\theta)$ is defined as the product of the signal x with $g_\theta = diag(\theta)$ in the Fourier domain. This results in $y = Ug_\theta\left(\Lambda\right)U^T x = g_\theta\left(U\Lambda U^T\right)x = g_\theta\left(L\right)x$ interpreting g_θ as a function of the eigenvalues Λ [KW16a]. In order to prevent the computationally prohibitive matrix multiplication necessary to perform the Fourier Transform of signal $\mathbf{x}$, g_θ can be reformulated using the Chebyshev polynomial parameterization of the filters $g_\theta\left(\Lambda\right) = \sum_{r=0}^{k}\theta_r T_r(\Lambda)$, where $\theta \in \mathbb{R}^k$ is a vector of Chebyshev coefficients with degree k [KW16a; DBV16]. Since $L^k = (U\Lambda U^T)^k = U\Lambda^k U^T$, $g_\theta\left(\Lambda\right)$ could be written as a function of $g_\theta\left(L\right)$. Therefore, the spectral filtering on a signal x can be performed with $g_\theta * x = \sum_{r=0}^{k}\theta_r T_r(L)x$. The value of vertex j of the filter g_θ centered at vertex i is given by

$$(g_\theta(L)\delta_i)_j = (g_\theta(L))_{ij} = \sum_k \theta_k(L^k)_{ij} \tag{2.6}$$

where δ_i is Kronecker delta function.

2.4.1 Literature survey on spectral approaches:

As mentioned in the earlier section, spectral-based methods get their mathematical foundation mainly from graph signal processing [Shu+13], [SM13], [Wu+20]. The first spectral domain formulation of CNNs on graphs was done by [Bru+13]. Although, this paper was of significant importance conceptually, it had computational drawbacks. Some of these drawbacks were later addressed in the followup works of [HBL15] and [DBV16].

2.5 Impact of GCNs in non-medical doamin

GCNs have shown a broad spectrum of applications across different fields and domains. Apart from solving generic tasks such as node classification, graph classification, network embedding and graph generation, other graph related tasks such as node clustering [Wan+17], link prediction [ZC18], and graph partitioning [KTO18] have also been addressed by GCNs. Few of the important applications are described below.
In Computer vision GCNs are mainly applied to scene graph generation, point clouds classification /segmentation, and action recognition. The main goal of scene graph generation models is to parse an image into a semantic graph that consists of objects and their possible semantic relationships [Xu+17], [TM20], [Li+18b]. Given the scene graphs, the process of generating realistic images is done in [JGF18].

In a textual description of the scene, if each word is considered a node, this can become a promising solution in order to synthesize images from the given text. Classifying and segmenting the nodes of such graphs allows a device called LiDAR to understand the surrounding environment. A point cloud is a set of 3-D points recorded by LiDAR scans. In the following three works [Wan+19b], [LS18] and [Te+18], point clouds are converted into k-nearest neighbor graphs and ConvGNNs are used to exploit the topological structure.

In the field of human action recognition, human joints which are linked by skeletons naturally form a graph. [Jai+16], [YXL18] use STGNNs to learn human action patterns from the time series of human joint locations. Further applications of GNNs includes human-object interaction [Qi+18], few-shot image classification [GB17], [Guo+18], [Liu+19a], semantic segmentation [Qi+17], [Yi+17], visual reasoning [Che+18] and question answering [NLS18].

Text classification is a common application of GNNs in natural language processing. Inference of document labels is done using the inter-relations that exist between the documents using GNNs [KW16a], [HYL17a], [Vel+17]. Natural language data can also contain an internal graph structure, such as a syntactic dependency tree, considering the fact that such data exhibits a sequential order. [MT17] proposes the Syntactic GCN that runs on top of a CNN/RNN sentence encoder. Furthermore, [Bas+17] and [MBT18] apply the Syntactic GCN to the task of neural machine translation.

Graph-based recommender systems take items and users as nodes. By defining the relations between items, users graph-based recommender [BKW17], [Yin+18] and [Mon+17] are able to provide quality recommendations. GNNs are also applied in the field of Chemistry to study the graph structure of molecules/compounds or to learn molecular fingerprints [Szl+05], [Kea+16], to predict molecular properties [Gil+17], to infer protein interfaces [Fou+17] and to synthesize chemical compounds [Li+18c], [DK18], [You+18].

Forecasting traffic speed, volume or the density of roads is handled in [Zha+18b], [Li+17], [YYZ17]. The applications of GNNs cover a wide variety of other problems such as program verification [Li+15], program reasoning [ABK17], social influence prediction [Qiu+18], adversarial attacks prevention [ZAG18], event detection [NG18] and combinatorial optimization [LCK18].

2.6 GCN for clinical problems

Some applications of GCNs in the medical domain are electrical health records modeling [Cho+17], [Cho+18a], brain networks [Kaw+17]. Application in medical test migration [Ren+20], Autism classification [Liu+20], analyze functional magnetic resonance images (fMRI) and discover neurological biomarkers [LD20], for handling an incomplete data for disease prediction [Viv+20]. Metric learning for brain signal analysis [Kte+18], modeling neuropathophysiological heterogeneity [Dvo+20], brain age prediction [Sta+20].
In the chapters that follow, Some of the challenges faced in the field of healthcare and their solutions provided using GCNs are illustrated.

The main objective of the GCN based methods described in the upcoming chapters is mathematically defined below. The general task solved in the entire thesis is given as follows. Let the dataset $\mathcal{D} = \{X, Y, \delta\}$. Here, $X \in \mathbb{R}^{N \times d}$ represents the feature matrix for N patients and each patient is provided with d-dimensional features. Let Y be the corresponding label matrix (one-hot encoded) and δ the demographic data matrix. The task is to predict the class label $\hat{Y}$ for test subjects for K classes. $\delta \in \mathbb{R}^{N \times M}$ represents that for each patient, M-dimensional demographic data is provided. The m^{th} affinity graphs $G^{(m)} \in \mathbb{R}^{N \times N}$ are computed from the respective δ^m demographic element. The model $f(\cdot)$ to solve the task is derived by

$$\hat{Y} = f(X, G^{(m)}; \theta). \tag{2.7}$$

The model takes X and $G^{(m)}$ as input to train the parameters θ and generates discriminative features for tasks such as disease classification, age prediction etc.

Part IV

GCN and Multiple Graph Scenarios

3 GCN and Multiple Graph Scenarios

> *Science is the acceptance of what works and the rejection of what does not. That needs more courage than we might think.*
>
> **— Jacob Bronowsk**
>
> ..

In the previous chapter the theoretical and mathematical description of a spatial and spectral GCN is provided. A forward propagation model is also detailed as well. One peculiar characteristics about the simple GCN model is that it undertakes only one graph.In this chapter, a scenario is analyzed where multiple graphs are processed simultaneously. In clinical routines, multi-modal data is collected comprising of imaging and non-imaging data which is utilized together for treatment and planning. Each piece of complementary information about the patients is distinct and important. A model which takes into account the distinctiveness of each element of the multi-modal data for better disease prediction is been explored in this part.

3.1 Introduction

Experts investigate patient's heterogeneous multi-modal data collected by imaging sources and non-imaging demographics (age, gender, weight, body-mass index, clinical test reports, health records etc) to take an reasoned decision for disease diagnosis and treatment planning. Computer Aided Diagnosis systems (CADs) exploit such rich data as complementary information. Generally, CAD systems combine all the complementary features often by regressing them [**lorenzi2016multimodal**] or by using some feature selection techniques [**memarian2015multimodal**], or by reducing the dimensionality with an auto encoder [**calhoun2016multimodal**; **tiwari2011multi**; **ngiam2011multimodal**] or by simply concatenating all the features to use deep learning based models [**Xu2016**]. The methods mentioned above use the complementary information from all the different modalities at a global level, however such varied information can be optimally combine further. For example, the output features are generally biased towards modalities with dominant features where the non-dominant but may be important features might get neglected [Bis20]. This way the models surely do not exploit the individuality of each modality. Plus, each demographic element carries different importance and relevance towards the diagnosis of a disease.
In the first part of this thesis, a model is developed that is capable of evaluating the significance of every element of the demographic data while performing the prediction task based on the selective weighting procedure for elements of demographic data. This method boosts the

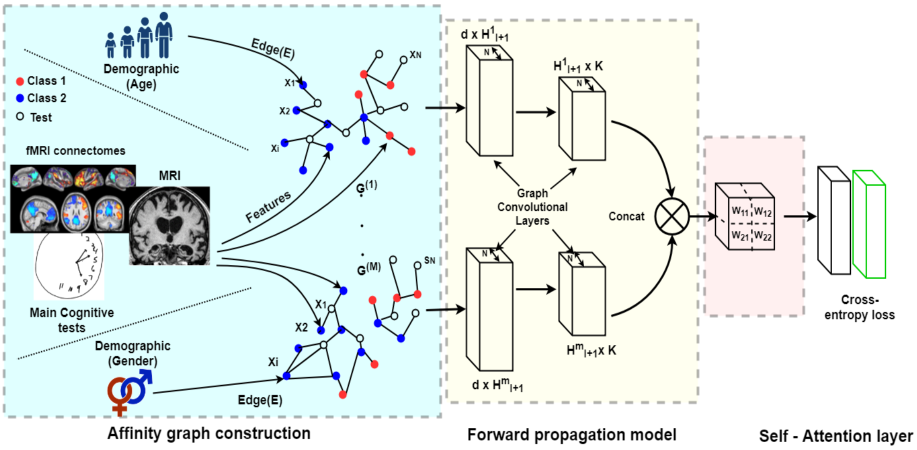


Fig. 3.1. Figure describes the Multi-Layered Parallel Graph Convolutional Network with $M = 2$. Two branches have same input features but input affinity matrix.

original task of disease prediction task to incorporate more clinical semantics.

In order to achieve this, graph based deep method is designed. Graphs provides more elegant way to increase the impact of the modalities which could get neglected due to the dominant features in representation learning[Par+17; KW16a]. GCNs as is mentioned in the previous section leverages the similarities between the patients and form an affinity graph for training. One of the first work on GCNs with medical application [Par+17] presents an intelligent and novel use case of Graph Convolutional Networks (GCN) for the binary classification task. The method proposes to use each meta data (demographic information) separately to construct a neighborhood graph and finally combine all the neighborhood graphs (by averaging) to get the final affinity graph, unlike the conventional non-graph based methods, which fuses the information for the prediction task. Inspite of the elegant way of combing the meta-data in this method, it yields varied results for distinct input neighborhood graphs. Each of these affinity graphs and indirectly each element of the demographic data carries distinct neighborhood relationships (based on element dependent criteria) and statistical properties with respect to the entire population. In this part of thesis we focus on analyzing the impact and relevance of the neighborhood definitions on the final task of disease prediction. Further, this part of the thesis is dedicated towards developing a model that automatically learns relative weighting of meta-data. In this chapter, a model capable of incorporating the information of each graph separately is proposed which incorporates a 'Self-Attention layer' which automatically learns the weighting for each meta-data with respect to its relevance to the prediction task. This model outperforms the state-of-the-art method.

3.2 Methodology

As described earlier in equation 3.1, the task is the following:

$$\hat{Y} = f(X, G; \theta) \qquad (3.1)$$

In the multiple graph scenario, the proposed task can be mathematically described as:

$$\hat{Y} = f(X, G^{(m)}; \theta). \tag{3.2}$$

Here, model intakes feature matrix X and m graphs $G^{(m)}$ as input to train the parameters θ and outputs discriminative features for classification. The full pipeline is shown in fig 3.1. It can be divided into three main parts: a) affinity matrix W^m construction, b) the forward propagation model: this is used to produce class-specific output features, and 3) the self attention layer for weighting output activations of each branch.

Affinity Matrix $W^{(m)}$ Construction: For the multiple graph scenario, we construct M different affinity graphs each corresponding to a demographic element. Let's say we have M -dimensional non-imaging feature vector for each patient. Then for each m^{th} element, an undirected and unweighted graph $G^m = \left\{X, E^{(m)}\right\}$ is defined. For fair comparison we follow affinity graph construction technique same as the state of the art [Par+17] and all the M graphs have a common vertex set X. $E^{(m)} \in \mathbb{R}^{N\times N}$ is a demographic element specific to the edge matrix. Each graph $G^{(m)}$ reveals distinct intrinsic relationships between the vertices. Based on the given demographic element, the edges between the nodes can be defined as:

$$E_{i,j}^{(m)} = \begin{cases} 1 & if\ |\eta_{i,m} - \eta_{j,m}| < \beta_m \\ 0 & otherwise \end{cases}, \tag{3.3}$$

where η_m is the corresponding demographic element and β_m is a threshold. By weighting the edges, affinity matrices are generated from these binarized graphs. A similarity metric between the subjects $Sim(X_i, X_j)$, *e.g.* correlation coefficient, is incorporated to weight the edges as

$$W_{i,j}^{(m)} = Sim(X_i, X_j) \circ E_{i,j}^{(m)}\left(X_i, X_j\right), \tag{3.4}$$

where $\circ$ is the Hadamard product.

Forward propagation model: The proposed model $f(\cdot)$ is designed such that it trains for each affinity graph separately. The model bears the parallel setting of M branches as shown in Fig. 3.1. Each branch is equipped with GC layers based on spectral graph theory. Unlike grid-based convolutions [KW16a; DBV16], these layers help adopt convolutions on graphs. The m^{th} branch of forward propagation model is given by:

$$H_{l+1}^{(m)} = \sigma\left(D^{(m)^{-\frac{1}{2}}} W^{(m)} D^{(m)^{-\frac{1}{2}}} H_l^{(m)} \Theta_l^{(m)}\right) \tag{3.5}$$

where, D is the diagonal matrix with $D_{ii}^{(m)} = \sum_j W_{ij}^{(m)}$. $\Theta_l^{(m)}$ are the trainable layer-specific filters, which can be derived from a first-order approximation of localized spectral filters on graphs [KW16a], and $H_{l+1}^{(m)}$ is the feature output of the l^{th} layer ($H_0^{(m)} = X$). $D^{(m)^{-\frac{1}{2}}} W^{(m)} D^{(m)^{-\frac{1}{2}}}$ is the normalized graph Laplacian, and $\sigma(\cdot)$ is the rectified linear unit function. Each m^{th} branch outputs $H_{logits} \in \mathbb{R}^{N\times K}$. H_{logits} which are fed to the self-attention layer described below.

Self-Attention Layer: Because of the graphs, the logits for M branches vary with respect to each other, although features on each vertex are common. In order to rank the demographic data elements, a linear combination layer is designed to rank the logits from the last hidden layer as

$$\hat{Y} = Softmax\left(\sum_{m=1}^{M} \omega_m H_{logits}^{(m)}\right), \tag{3.6}$$

where ω_m is the trainable scalar weight associated with the demographic element. Each element of ω_m array is branch specific, weighting the logits of each branch. and $\hat{Y}$ are the normalized log probabilities. We define our objective function as binary weighted cross entropy loss on the labeled data to train the model parameter.

3.3 Experiments

The experiments in this chapter are designed to (1) investigate the impact of each affinity graph on the performance of the models, (2) investigate the performance of the predictive model with multi-graph setting approaches [Par+17], (3) compare our proposed model with 3 methods, linear classifier, two-layered dense neural network, and baseline GCN [Par+17] and (4) investigate the clinical insights of self-attention layer with multi-graph setting.

Dataset: A publicly available dataset Tadpole [Mar+18] is used for the prediction of Alzheimer's disease and our investigation of multiple graph scenario. Tadpole is a subset of ADNI[**jack2008alzheimer**] with a cohort of 564 patients. The primary task is to classify each patient into either of the three classes, Normal, Mild Cognitive Impairment (MCI) and Alzheimer's disease (AD). Each patient comes with a set of multi-modal features collected from various biomarkers (MR, PET imaging, cognitive tests, CSF biomarkers, etc) together with the risk factors such as APOE genotyping status and FDG PET imaging which measures the cell metabolism, where cells affected by AD show reduced metabolism. Further demographics such as age and gender is provided. Entire data is pre-processed with ADNI's standard data-processing pipeline. According to medical literature [**shaffer2013predicting**], age and gender are the most important risk factors for the prediction of the disease followed by APOE and FDG. The data is split into training and validation sets (90%, and 10%, respectively).

Implementation: Number of features d = 354, dropout rate: 0.3, ℓ_2- regularisation: 5×10^{-4}. All the experiments are implemented in Tensorflow[1] and performed with Nvidia GeForce GTX 1080 Ti 10 GB GPU.The ratio of computational time for the proposed model vs the GCN is 4.51. We use early stopping criteria to decide the number of epochs for each setting. The model is evaluated based on the mean classification accuracy (ACC) for 10-fold Cross-Validation.

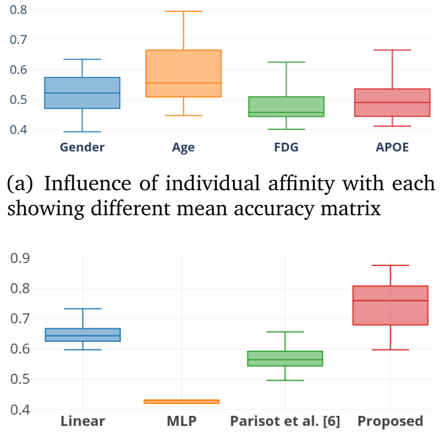
3.4 Results and Discussion

All the results of the above experiments are discussed in detail below.

Influence of individual affinity matrix: In this set of experiments all the factors such as node features, model architecture and model parameters are kept constant. The affinity matrix is the only changing factor. It can be seen from fig. 3.2 that, a) The results vary with change in affinity graph. We can interpret that each input affinity matrix has unequal relevance to the task at hand. For instance, the best performance is shown by the age graph where as the model performs the worst with FDG as the input graph. b) It can be noted that the performance is decreased when all the graphs are averaged. Such a setting is used in the baseline method [Par+17]. This shows that an average of the affinity graphs degrades the performance that otherwise could have been achieved.

Performance with different combinations of graphs: Another experiment is performed

[1] www.tensorflow.org



(a) Influence of individual affinity with each showing different mean accuracy matrix

(b) The results over all the comparative methods with proposed model outperforming.

Fig. 3.2. All the three subfigures show the boxplots of accuracy over 10-folds cross validation.

using all the various affinity matrices by varying the combination as input. This validates that if the combination of affinity matrices changes, the performance varies. In the clinical literature for the diagnosis of AD [**perrin2009multimodal**], it is stated that age and gender are the most important factors compared to other risk factors (APOE, FDG). Results are presented in terms of accuracy boxplots, as seen in fig. 3.3 confirming that different combinations show heterogeneity in the results. In comparison, the mixture of gender and age reveals the highest performance, with most combinations using FDG with APOE lower the performance. This shows the proposed model upholds the same clinical semantic as [**perrin2009multimodal**]. This experiment also reconfirms that the overall performance decreases when all affinity graphs are weighted equally and the positive influence of other affinity matrices is deteriorated by average due to the loss of neighborhood structure for individual graphs. The proposed self-attention model outperforms all combinations as it captures the proper weighting needed for

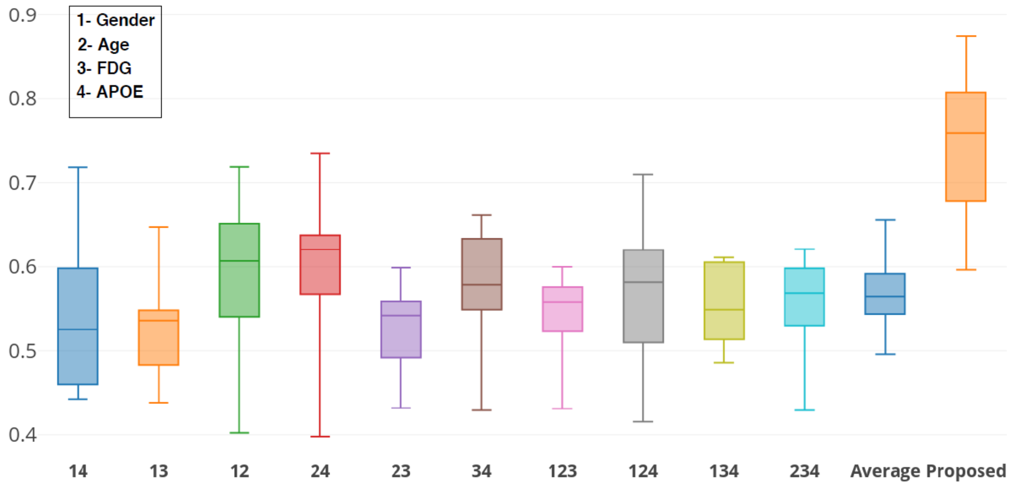


Fig. 3.3. The box plot represents the accuracy obtained with different combinations of the four affinity matrices. X-axis show the combination of 2, 3 or all affinity matrices where 1,2,3 and 4 stand for Gender, Age, FDG and APOE respectively (as shown in the box at the top left).

optimum performance.

Performance in comparison to other methods: As shown in figure 3.2, the proposed method is compared with three state-of-the-art methods, namely linear classifier[HK70], neural network and GCN [Par+17]. These methods are chosen to investigate 1) how the features at each node are class separable? 2) What is the model's output when the features are concatenated? 3) What is the significance of incorporating the graph? and 4) how essential is it to weight the graphs?

One can interpret that the features are class separable since the linear classifier performs very well in contrast with two other approaches. For fair comparison, the NN model is designed such that it has same number of hidden layers (2) and hidden unties (16, 3 respectively). In case of the neural network (NN), the features are concatenated and this becomes the problem. NN fails to perform well with this architecture, where as the graph based baseline approach [Par+17] shows better results compared to the NN. This shows the strength of incorporating the graph. As can be seen, the baseline [Par+17] enhances the performance of the GCN's power with respect to NN, but it performs lower than linear and proposed method. This is attributable to the corrupted combination of the neighborhood. Finally, with the right weighted combination of neighborhood and H_{logits}, the proposed method outperforms other methods. The GC layers are first trained for 150 epochs and the self-attention layer is trained further. This helps channelize the learning of weights of GC layers as well as the self-attention layer. The features at every node are kept simpler to gain more insights about the effect of the graphs.

Effect of self attention: The weights learned by each branch are also investigated in our experiments. The self-attention layer learned maximum weight for gender and age (0.35 and 0.27 respectively) and lower weight for FDG and APOE (0.09 and 0.29 respectively). From [**perrin2009multimodal**] it is confirmed that age and gender are significant factor for predicting AD.

3.5 Conclusion

The experiments demonstrated in this chapter go inline with our hypothesis that affinity graphs influence the performance of GCN models and eventually disease prediction differently. The combination of affinities changes the possible neighborhood between the subjects. In comparison, our proposed self-attention approach explicitly integrates the unequal contributions of the graphs and outperforms all the setups with large margins. The order of complexity for the proposed model compared to the baseline model [Par+17] is almost equal as $O(n) \approx O(2n)$, making it scalable for a larger number of demographic elements.

Clinical statistics provided by the literature further obey the choice of thresholds for generating the graphs. One might argue that the performance might decrease by splitting a single graph into many graphs as some connections are lost in the thresholding process. However, the aggregation of graphs from various sources of information can contribute to the loss of individual structure and unequal relevance cannot be considered.

Part V

GCN and Personalised Disease Prediction

4 GCN and Personalised Disease Prediction

> *A man who dares to waste one hour of time has not discovered the value of life.*
>
> — **Charles Darwin**
>
> ..

Clinical experts consider the complementary multi-modal data for disease diagnosis and decision making. Often a varied order of importance for this heterogeneous data is considered for personalized decisions. Learning patient-specific attention for individual modality has not been explored in the current literature. In this chapter, a model is introduced that improves the disease prediction and learns personalised patient-specific order of importance for the multi-modal data elements simultaneously. The model achieves this by the novel combination of LSTM-based attention mechanism and graph convolutional networks (GCNs). In this process, class-specific features are learned by GC layers and the attention mechanism integrates the multi-modal features into the final outcome, individually for each patient. In this chapter, the proposed technique is leveraged for the task of disease prediction for Parkinson's and Alzheimer's dataset.

4.1 Introduction

Recent deep learning methods have focused on learning the class-specific representation for various tasks, in particular disease prediction. However, medical data generally bear high heterogeneity due to the diverse condition of patients. For example, patients with same disease stage may have different diagnosis and require personalised treatment planning based on demographic, clinical and imaging data. In case of Alzheimer's disease, especially APOE-targeting ones,drugs may act differently in patients with different APOE genotypes but same disease condition. Moreover, other than the demographic and genetic factors, brain imaging and other biomarkers may also be used for patient-specific diagnosis. The importance of such personalized diagnosis of the disease has been shown in [**bu2016toward**; **peng2016towards**] respectively. A CAD system capable of learning such patient-specific decision is required to improve the clinical outcome.

Recent literature [**ng2015personalized**; **suo2017personalized**] has shown that personalized models can improve the model performance compared to the general conventional models [TDG16]. Usually such a pipeline consists of two stages: 1) to measure the similarity among the patients, and 2) to design a module that evaluates patient-specific disease condition based on corresponding non-medical features. Such a framework, if build would mimic the clinical workflow where experts collect and scrutinize the patients with similar condition before making any patient specific decision. In the previous chapters and the corresponding literature GCNs have already shown their superiority over other conventional methods for extracting the similarity among the patients for disease prediction [**kazi2018self**; Par+17].

As mentioned in chapter 2, GCNs incorporate the relationships among patients based on non-imaging features such as demographic, clinical reports, medical history etc. This is achieved through a neighborhood graph. The nodes (patients) are represented with features from, e.g., imaging modalities. Finally, GCNs provide a sophisticated framework to learn the model parameters in order to optimize the objective. Many methods [**zhang2018multi**; **ma2018multi**; Ma+18b] have provided efficient GCNs based models to deal with multi-modal data (in a multi-graph scenario), by consolidating the heterogeneous information using techniques such as pooling, concatenation, or averaging at the end. Apart from these, [Vel+17; Kaz+19c; Fou+17] have proposed a node-level attention mechanism to weigh the neighbors during training. In comparison to alternative traditional methods, GCNs have been proven to be far more superior. In this chapter, an end-to-end pipeline that aims to merge multi-graph setting and node level attention mechanism has been proposed, which is unlike the methods mentioned above. Personalized diagnosis is achieved by firstly clustering all similar patients based on certain criteria, and then by learning the patient-specific traits. Multiple representations that are modality specific are learned for each patient. LSTM based attention scheme is leveraged to obtain patient-specific weights, for the non-imaging data. This scheme optimally integrates the multi-modal data to arrive at a final patient specific decision.

The methodological contributions here are:

- Clustering the patients based on similarity by employing GCNs with multi-graph setting.
- Learning the weights for each non-imaging factor by leveraging LSTM based attention mechanism to achieve personalized disease prediction.

This method could be extended to other multi-modal datasets for tasks such as disease prediction etc. In this chapter, two applications are presented, which are: 1) Alzheimer's disease prediction, and 2) Parkinson's disease prediction, using publicly available datasets in both predictions. The superiority of the proposed model is demonstrated in terms of accuracy, f1 score, sensitivity, and PPV.

4.2 Methodology

Consider, X denoting the population of N patients. Imaging feature for each patient n_i is denoted by $x_i \in \mathbb{R}^D$. The corresponding non-imaging features are denoted by $\eta_i \in \mathbb{R}^M$. Overall, the imaging and non-imaging feature matrix for the entire population is denoted by $X \in \mathbb{R}^{N \times D}$ and $\eta \in \mathbb{R}^{N \times M}$. The class labels $Y \in \mathbb{R}^{N \times C}$ for C classes (one-hot encoded) are available only for the training set Y_{tr}. Given the above information, the final task is to predict the classes for the test set Y_t. m non-imaging elements are employed to integrate the similarity between the patients. m affinity graphs each denoted by $G^m \in \mathbb{R}^{N \times N}$ are defined

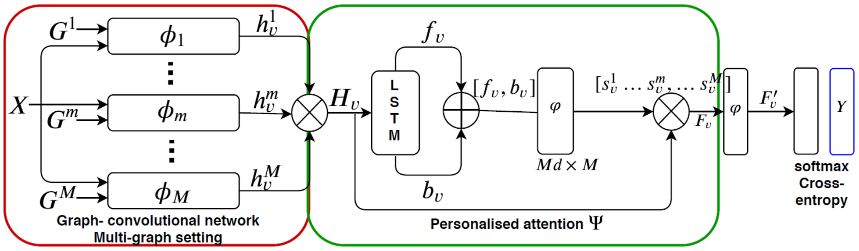


Fig. 4.1. The end to end pipeline of the proposed method. The red box indicates the GC layers extracting similarity among patients. The green box produces personalized attention scores later used for combining representations. In the end, logits and cross-entropy loss are calculated.

corresponding to each element m in η. Thus, $G^1, ..., G^m, ..., G^M$ different graphs are obtained from distinct elements of non-imaging modality. The graph construction procedure is entirely described in the chapter 2.2.2.

Given the graph information, the task of disease prediction for test set Y_t is redefined as,

$$\widehat{Y_t} = \Psi(\Phi_m(X, G^m, \Theta_1^m), \Theta_2). \tag{4.1}$$

where, $\widehat{Y_t}$ are the predicted labels for the test set, Φ_m is a function specific to η^m modality that learns feature representation for each patient as shown in Fig. 4.1, Ψ is the personalized attention function which weights the representations corresponding to each η^m. $[\Theta_1^1, ..., \Theta_1^m, ..., \Theta_1^M]$ and Θ_2 represents the set of learnable parameters of both the functions Φ_m and Ψ respectively.

Step 1 of the end-to-end pipeline of the personalized prediction is to obtain the similarity among patients. GCNs are the rightful choice for achieving the same. Therefore, each Φ_m is defined using a graph convolution operation. In the proposed model, each branch Φ_m consists of two GC layers. Bidirectional LSTM that learns the attention scores is incorporated to build the personalized attention mechanism Ψ. This LSTM based mechanism learns attention scores for the representations obtained from each Φ_m. The inherent property of LSTM is to incorporate the inter-relation between each of the inputs. This property of LSTM allows the proposed model to learn the final representations F_v for each patient individually, retaining the inter-relationships between the outputs of each Φ_m. The entire end to end pipeline is illustrated in Fig. 4.1.

In the next subsection, mathematical details of both the functions Φ_m and Ψ are provided. Further, the process of building an end to end model using Φ_m and Ψ is detailed. The optimization function used for this model is weighted cross entropy loss to learn the filters.

4.2.1 Definition of Φ_m

Spectral approach is chosen to define Φ_m corresponding to each m^{th} branch. Therefore, the Chebyshev polynomial approximated version of graph convolution is defined for each graph G^m. For each node v feature vector x_v, where $x_v \in X$, is intialized. Mathematically this spectral operation is defined as $h_v^m = g_\phi(L^m)x_v = \sum_{r=0}^{K-1} \phi_r T_r(L^m) x_v$, where $h_v^m \in \mathbb{R}^d$ is the output activation corresponding to η^m, $\phi_r \in \mathbb{R}^K$ is the vector of learnable Chebyshev coefficients, $T_r(L^m)$ is the Chebyshev polynomial of order k for L^m and $x \in \mathbb{R}^{N \times N}$ are the input features. Using the transductive setup, the graph G^m incorporates the entire population including training and testing patients. The graph G^m plays an important role in spectral convolution to define the neighborhood for each patient in order to perform convolution. This information is incorporated in each of the normalized graph Laplacian L^m corresponding to G^m. Further mathematical details on GCNs can be found in chapter 2.

The input to attention obtaining function Ψ is derived by concatenating the outputs of each branch. This input is denoted by H_v and can be represented as $H_v = [h_v^1, ..., h_v^m, ..., h_v^M]$ for node v. The activation or the node representation h_v^m learnt from m^{th} branch of GC layers is specific to the m^{th} element of non-imaging risk factor. H_v then becomes personalized feature sequence of learnt multi-modal representation for each patient. The required affinity graph $G = (V, E, W)$ where $|V|=$ N vertices, $E \in \mathbb{R}^{N \times N}$ is defined by the binary edge connections matrix of the graph and $W \in \mathbb{R}^{N \times N}$ is the weight matrix. $Sim(x_i, x_j) \circ E^m_{i,j} = \begin{cases} 1 \quad if\,|\eta^m_i - \eta^m_j| < \beta^m \\ 0 \qquad otherwise \end{cases}$,where $Sim(x_i, x_j) = exp(-\frac{[\rho(x_i,x_j)]^2}{2\sigma^2})$ with ρ being the 'correlation distance, σ being the width of the kernel and β^m being the threshold for edge construction.

4.2.2 Personalized attention mechanism

In this section, the LSTM based attention scheme used for personalized attention is mathematically detailed. For $H_v = [h_v^1, ..., h_v^m, ..., h_v^M]$, a scalar attention score s_v^m is learnt for each representation h_v^m such that $\sum_m s_v^m = 1$. Learning an attention for multiple representations of the same patient ensures that the attention mechanism defined here outputs a personalized weighted representation for the corresponding patient. The obtained attention scores $[s_v^1, ..., s_v^m, ..., s_v^M]$ represents the importance of the m^{th} branch, for node v. The step-by-step details of the proposed node-level attention mechanism is provided below.

Step 1: The concatenated output H_v from GCN represents a sequence which is later fed onto the LSTM attention cell with d units. Let f_v and b_v represent the forward and backward LSTM hidden activation for each node v, h_m be the feature representation of the m^{th} branch, $r_{m \pm 1}$ be the hidden state of bi-directional LSTM and α be the tanh activation function.

Step 2: Node level attention scores for each graph is generated by applying a linear mapping φ and a softmax to s_v as, $s_v = softmax(\varphi\,\{[f_v, b_v]\,\omega + b\})$ where, $s_v \in \mathbb{R}^M$, $[.,.]$ stands for concatenation. φ is applied as a dense layer without any non-linear activation, ω and b are the weights and biases of the dense layer respectively.

Step 3: Final representation for node v is fetched by taking the weighted sum of the representation as $F_v = \sum_m s_v^m \cdot h_v^m$, where $F_v \in \mathbb{R}^d$ is the personalized final representation obtained from weighted aggregation over all graphs. F_v is applied to a dense layer in order to output logits for each class F'_v. Thus, the weighted cross-entropy loss is minimized. The proposed node-level attention based method is described in Fig. 4.1

4.3 Experiments and results

Two publicly available datasets Parkinson's Progression Markers Initiative (PPMI)(www.ppmi-info.org/data) and TADPOLE [Mar+18] are used to analyze and evaluate the proposed model.
Dataset description:
The multi-modal dataset PPMI consists of 324 patients out of which 75 are healthy and 249 are diseased cases. Each patient in this dataset comes with brain MRI volume and non-imaging meta data such as clincal score of Unified Parkinson's Disease Rating Scale (UPDRS), Montreal Cognitive Assessment (MoCA) scores and demographics (age and gender). Pre-processing of the dataset involves co-registering each 2-D image of the MRI volume to the SRI24 ATLAS, followed by skull stripping using ROBEX and eventually normalizing each volume into the range [0,1]. 1-dimensional feature representation for each patient is obtained by feeding the MRI volumes onto a 3D auto-encoder which is pre-trained for anomaly detection. This pre-processing pipeline is followed from [**baur2018deep**]. The bottleneck feature vector from the 3D auto-encoder is obtained for each volume and the size of this feature vector is empirically chosen to be 320. TADPOLE dataset consists of data associated with 564 patients. It comprises of 160 normal, 320 Mild Cognitive Impairment (MCI) and 84 Alzheimer's Disease (AD) patients. In this dataset, each patient is provided with a 354 dimensional feature vector pre-extracted from brain MR and PET imaging, CSF, cognitive test and non-imaging risk factors such as age, gender, APOE genotype, average FDG and PET imaging values.

For both the datasets non-imaging features are leveraged in the affinity graph construction. Next, the imaging features are used as input feature matrix X. The values of the thresholds β^m (from equation 2.1) is chosen same as [**kazi2018self**] for the fair comparison. The task for both the datasets at hand is to predict the disease for each patient. The datasets are divided into a spilt of 90% train and 10% test.

The experiments are designed to prove that, 1) LSTM based attention mechanism is a better fusing scheme in comparison to other representation, as shown in the Table. 4.2, and 2) the proposed end to end pipeline works better than single/multi-graph global attention mechanism shown in Table. 4.1. Baseline methods include concatenation, maxpool and average pooling techniques applied to H_v. Further, the proposed method is compared to five state of the art methods which are categorized into three types as shown in Table. 4.1. The five comparative methods are briefly described below.

- [**kazi2018self**] has multi-graph setting with global attention scheme
- [Vel+17] uses personalized attention scheme for node level classification task.
- [Par+17] is a GCN based method using single affinity graph. This single graph is computed by averaging all the graphs constructed by the technique in [Par+17].

The performance comparison of the proposed model with these two methods prove the superiority of the multi-graph setting over the single-graph setting. The advantage of graph based method over conventional non-graph based method is shown by the results of multi-layered perceptron (MLP).

Tab. 4.1. Performance comparison of proposed method to state of the art methods, in terms of five metrics for PPMI and TADPOLE dataset. The results obtained are statistically significant in the setting marked by Asterisk symbol.

PPMI						
		Accuracy	**F1score**	**Sensitivity**	**Specificity**	**PPV**
	Proposed	**91.04±04.68**	**76.12±14.74**	79.28±20.57	**91.93±05.76**	**75.97±13.18**
(a)	Kazi et al. [**kazi2018self**]	86.72±06.37*	53.25±19.77*	52.85±24.07*	88.30±07.83	58.72±19.80*
	Ma et al. [Ma+18b]	45.06±22.86*	39.57±02.73*	**94.28±09.98***	14.18±17.36*	25.18±02.47*
(b)	Kipf et al. [KW16a]	28.39±03.01*	33.49±08.98*	43.03±17.96*	66.71±16.37*	28.82±08.62*
	Parisot et al.[Par+17]	86.72± 05.00	73.27±10.66	79.64±16.73*	89.11±04.44	69.13±09.95
(c)	MLP	50.30±08.50*	27.53±09.77*	42.14±18.40*	52.16±06.75*	20.63±06.76*
TADPOLE						
		Accuracy	**F1score**	**Sensitivity**	**Specificity**	**PPV**
	Proposed	**83.33±03.89**	**72.55±11.97**	**76.87±18.64**	**86.64±09.06**	**72.06 ±13.48**
(a)	Kazi et al. [**kazi2018self**]	82.26±07.75	68.13±14.89	72.50±15.08*	83.11±11.48	66.47±20.59
	Ma et al.[Ma+18b]	49.46±06.79*	44.70±05.58*	75.00±15.30*	36.48±16.16*	32.22±04.47*
(b)	Kipf et al. [KW16a]	50.88±07.31*	51.88±06.75*	76.38±10.22*	52.84±09.46*	39.45±05.70*
	Parisot et al.[Par+17]	72.69±08.00*	61.95±14.59*	68.75±15.65*	78.18±09.85*	57.11±15.31*
(c)	MLP	79.60±07.27	69.63±09.82	80.00±10.12*	79.89± 07.13	61.90±10.30

Interpretation: All attention based techniques gain better results in comparison to other methods with no attention, as shown in Table. 4.1 (a) vs. (b) and (c) respectively. The proposed method achieves 4.32% and 1.07% improvement in accuracy for PPMI and TADPOLE, respectively. The reason for this is the personalized attention technique that can weight different graphs for each patient separately. The ablation study of the proposed attention mechanism shows its performance in comparison with other fusing mechanisms. Here, an improvement of 1.01 % and 0.89% over the other methods is obtained which is shown in the Table. 4.2. A significance test using Kolmogorov-Smirnov test (K-S test) is performed and is shown by $*$ in all the tables. This means that the results obtained by the proposed methods are statistically significant compared to the methods mentioned above ($p < 0.05$). The stability of the proposed method is depicted in the small variance over the results of 10 folds. For the thorough evaluation of the method, F1 score, sensitivity, specificity and PPV are reported for both the datasets as they are highly imbalanced. For most of the reported metrics, the proposed method outperforms others.

4.4 Discussion and Conclusion

In this chapter, a model for personalized disease prediction is presented and its application in Alzheimer's and Parkinson's disease prediction is shown using two publicly available datasets.

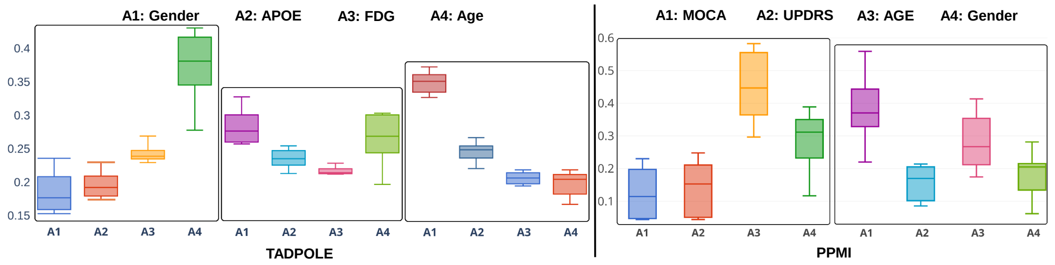


Fig. 4.2. The figure illustrates the boxplot of the attention scores learned for each patient, for 10 folds for TADPOLE. The variance of each boxplot is computed for 10 folds. Each boxplot shows weights learnt for each class normal (left), MCI (center) and Alzheimer's (right) per patient (TADPOLE dataset). For PPMI dataset Normal (left) and abnormal (right) for each patient is shown.

Tab. 4.2. Performance comparison of the proposed personalized attention method to the baseline methods for both the datasets. This shows that LSTM mechanism improves the overall classification of the model. The results presented are statistically significant in the setting marked by Asterisk symbol.

PPMI					
	Accuracy	**F1score**	**Sensitivity**	**Specificity**	**PPV**
Proposed	**91.04±04.68**	**76.12±14.74**	79.28±20.57	**91.93±05.76**	**75.97±13.18**
Concat	90.53±07.01	74.08±08.44	**82.32±15.87***	88.33±04.06*	68.58±07.45
MaxPool	86.40±05.48	72.80±10.43	78.75±14.61*	88.73±06.25	69.39±13.22
AvgPool	87.97±04.62	74.75±10.59	78.92±16.27*	90.75±04.27	72.65±09.91
TADPOLE					
	Accuracy	**F1score**	**Sensitivity**	**Specificity**	**PPV**
Proposed	**83.33±03.89**	72.55±11.97	76.87±18.64	**86.64±09.06**	**72.06±13.48**
Concat	82.44±08.95	**77.37±08.55**	**92.36±08.22***	80.96±08.82	67.13±10.11
MaxPool	68.41±08.02*	56.58±15.52*	55.62±20.29	85.40*±06.54	60.40±13.02
AvgPool	70.36±06.36*	56.82±10.72*	50.62±11.94*	88.78±09.97	70.65±21.21

The proposed model is built with graph convolutional layers. Each branch with GC layer intakes modality-specific affinity graph and patient-specific features to perform convolution on features vectors. The spectral approach is used for this model. Further, the personalized LSTM based attention mechanism integrates these graph-specific feature representations for each patient separately. The learned attentions for each graph almost matches the clinical order of importance and is well suited for personalized disease prediction. Comparison with the baseline methods (ref. Table 4.2) along with various fusing approaches shows the superiority of the proposed attention mechanism. The proposed method is compared to five different state of the art methods (ref. Table 4.1). The superior performance of the proposed model, in terms of different metrics necessary to analyze the performance on a much harder data setting including class imbalance is demonstrated in Table 4.1. Fig. 4.2 shows the attention weights obtained individually for all patients in both the datasets. High heterogeneity in the weights for a given task is required. The high variance of each boxplot means that the attentions required for each patient are different. The boxplots also depict a particular pattern for each class for both the datasets. This shows that the proposed model is able to learn a global pattern for each class. For Alzheimer's disease diagnosis, clinicians follow a general order of importance that is age, followed by gender, APOE and FDG [**bu2016toward**; **peng2016towards**]. The proposed method follows similar order of importance. For the PPMI dataset, a different trend is observed for the attentions. Here, the proposed model is slightly incorrect in following the order of importance. This is because the size of both of our datasets is 564 and 324 patients, which represents only a sub-sample of the entire population.

The bottleneck aspects of this method are: 1) Scalability to the value of m and N which means a larger number of affinity graphs and more number of patients. The proposed model might become computationally bulky if number of parallel branches corresponding to each meta element increases. Similarly, if the size of graph increases (with higher value of N), the memory limit of computing machine could be reached, and 2) Out-of-sample extension: the proposed method uses the transductive approach, which limits its capability to produce results for a completely unknown patient. However, the spectral convolution designed using Chebyshev parameterization is computationally inexpensive [DBV16]. In order to overcome the problem of out-of-sample extension, a spatial graph convolutional approach can used to design the layers. Such a method will allow graph modifications during training.

Such a technique is not only suitable for out-of-sample extension, but also facilitates learning from the graphs. In such an approach one graph can be learned from multi-graph setting along with better representation learning. In this work, the primary focus is to learn patient-specific attention for each graph using spectral convolutions, in order to make it scalable and convergent enough to accommodate multiple graphs. The proposed method could be generalized to other multi-modal datasets. An immediate future work for this method could be the development of an inductive version of this model in order to address the out-of-sample extension. Also, the graph learning technique for the spectral domain can be explored.

During the process of analyzing the input data, modelling and training, a new challenge was faced related to graph structure. As shown in Fig.

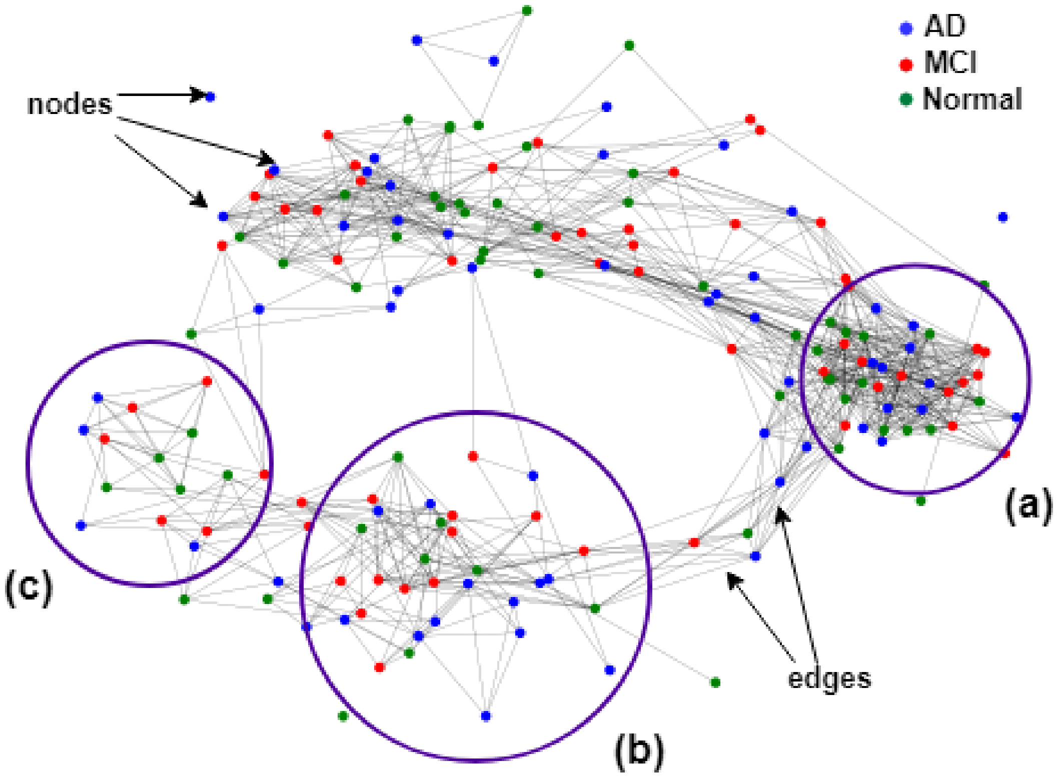


Fig. 4.3. The figure illustrates the heterogeneity within the graph. The graph shown here is obtained from TADPOLE dataset using the non-imaging feature 'age' for each patient. Each node represents a patient and the color represents the class of each patient.

Part VI

GCN and Intra-Graph Heterogeneity

5 GCNs and Intra-Graph Heterogeneity

> *Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less.*
>
> — **Marie Curie**
>
> ...

In the recent literature and from the previous chapters it can be seen that Geometric deep learning provides a principled and dynamic way to integrate imaging and non-imaging modalities in the medical domain. So far, GCNs have been explored for wide variety of applications. In particular, the Graph Convolutional Networks (GCNs) were investigated on a broad range of topics such as disease prediction, segmentation, and matrix completion by using massive, multi-modal datasets. In this chapter a unique spectral domain architecture is introduced for the application of disease prediction. The innovation lies in the concept of geometric 'inception modules' which can catch structural heterogeneity within the graph-termed as 'intra' - graph and between different graphs termed as 'inter'- graph heterogeneity during convolutions. To build the proposed architecture, we design filters that have different kernel sizes. We present results of our disease prediction on two publicly available databases. In addition , we offer insight into the behavior of normal GCNs and our proposed model on simulated data under differing input scenarios.

5.1 Introduction

Graph Convolutional Networks (GCNs)[DBV16] is increasingly focused on for applying deep learning to unstructured data in the medical domain. Multiple applications have been demonstrated to date, including prediction of Autism Spectrum Disorder with multiple learning to distinguish between diseased and healthy brains [Kte+18], matrix completion to predict missing values in medical data [Viv+18], and finding similarity, the primary task is disease prediction with complementary imaging and non-imaging multi-modal data. In all the mentioned works, GCNs had a significant effect on the application of multi-modal medical data. The key difference to previous learning-based approaches is placing patients with a neighborhood graph in relation to each other, often by associating them with non-imaging data such as gender, age, clinical scores or other meta-information. Patients can be viewed as nodes on this graph, patient correlations are defined as edge weights, and characteristics from e.g. image modalities are integrated by graph signal processing. Then GCNs have a principled way to learn optimal graph filters that optimize a task. Here, node-level classification is used for the

primary task of disease prediction.
A simple analogy to node-based population classification is image segmentation with CNNs, where each pixel is a node, and the graph is the image grid. In such domains, filters with a constant size can gain semantic features over the entire grid domain, provided a constant number of neighbors that are equidistant. The number of neighbors and their distance from one another contributes to heterogeneous density and spatial structure in the case of irregular graphs. Applying filters over the entire grid domain with a constant kernel size can not produce semantic and comparable features.

Graphs based on patient data identify similar variation in medical datasets, as each patient may have a distinct combination of non-imaging data and different number of neighbors. It gives a clear example in Fig. **??** (left) representing a population graph of 150 Alzheimer's disease classification subjects. These subjects are arranged in clusters of varying density and local topology (regions a, b and c). In order to learn the cluster specific features, the heterogeneity in the graph structure should be taken into account. By applying multi-sized kernels on the same input, a model capable of producing similar intra-cluster and different inter-cluster features can be designed. To this end, we propose InceptionGCN, motivated by the popular implementation of CNNs architecture [Sze+15].The proposed model leverages spectral convolutions of various kernel sizes and selects optimal features to solve the classification problem.Recently, there are only few works have focused on receptive field size of GCN filter such as [Yu+19], [Ros+20], [HC20] .However, the work in this chapter is the first one to identify the challenge of graph heterogeneity and receptive field size of the kernel. Many previous works [DBV16; KW16a] use GCNs with constant filter size for the node-classification task. Even though these methods GCNs superiority, however none of the method consider the graph's heterogeneity issue. In [Liu+18], a method is proposed that defines a receptive path for each node, instead of the entire receptive field for performing the representation learning convolution operations. A DenseNet-like architecture [Hua+17], in which outputs from consecutive layers are concatenated, is proposed in [Xu+18a]. The receptive field is addressed indirectly here, since the output activations of successive layers depend on multiple preceding layers through skip connections. Another work [HYL17a] uses either preset, hand-designed, or aggregator-based functions. In addition , the method requires a predefined node number, which is hard to achieve. In this chapter, we show 1) InceptionGCN is superior than other method in terms of performance and convergence, 2) Analysis of the effect of graph-structure and filter size on the performance of the model. This motivates the needs of multiple kernel sizes and 3) A novel InceptionGCN model is proposed with multiple kernel sizes. InceptionGCN is validated on synthetic data and clinical data. Further, we show the robustness of model towards different approaches for constructing graph.

5.2 Methodology

Standard models [Par+17] use a constant filter size for all layers, which requires each node 's features to be learned with neighbors at a fixed number of hops away without understanding the scale and shape of the clusters. The proposed InceptionGCN model overcomes this constraint by adjusting the size of the filters across the GC-layers to generate separable output features for the class. This property of the proposed model is particularly desirable where there is a distinct difference in class distribution and/or where the classes heavily overlap.

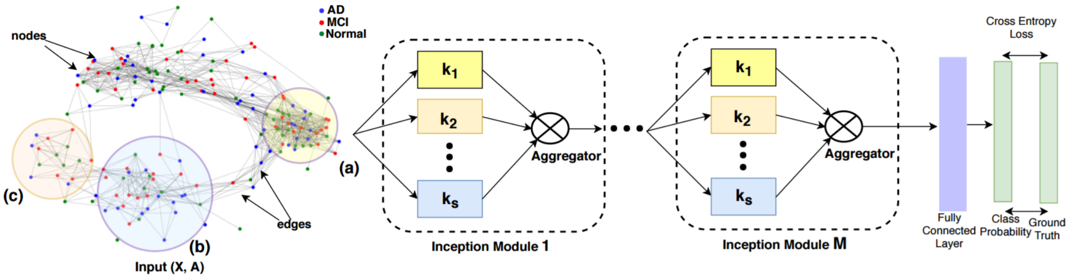


Fig. 5.1. Left: Affinity graph with clusters for TADPOLE dataset, different cluster sizes are depicted at points (a), (b) and (c). Right: Setup of InceptionGCN , feature matrix X is processed by several GC-layers with considered neighborhood $k_1, \cdots, k_S$ in each inception module. The output of each layer is used in the aggregator function.

Using this setup, we attempt to solve the task of disease classification by integrating semantics of varied correlations arising from various graphs within the population. We provide a detailed description of the model beginning with the construction of the affinity graph followed by the theoretical concept and a discussion on the proposed architecture.

5.2.1 Mathematical concept for inception modules

In chapter 2, the mathematical details of the spectral convolution is defined. To recap, the spectral convolution on a signal x can be defined as with $g_\theta * x = \sum_{r=0}^{k} \theta_r T_r(L)x$ here, L is the graph Laplacian (normalized or unnormalized), and $k > 0$ be an integer (here k stands for the k^{th} hop neighbor). Given such setting, for any two vertices i and j, we can define:

$$\left(L^k\right)_{ij} = \begin{cases} \Omega & d_G(i,j) \leq k \\ 0 & otherwise \end{cases} \tag{5.1}$$

,where $d_G(i,j)$ is the computed shortest path distance between x_i and x_j and Ω is the sum of all edge weights on the shortest path from x_i to x_j. Therefore from eq. 2 the spectral filters represented by k^{th} order polynomial of the Laplacian are exactly k-hop localized.

5.2.2 Inception modules

For spectral convolution with signal x, the localization of a filter is defined by taking all the neighbors at a distance of k hops into account. A filter s with a fixed k_s used on the full dataset X can be defined as $y_s = \sum_{r=1}^{k_s} T_r(L)X\theta_{r,s}$. Here y_s represents the output of a filter in a neighborhood distance of k_s-hop. Instead of using one filter, we now use S filters with differing neighborhood k_s to account for the different sizes and variances of clusters and structure in the results. The combination of these s filters is the nucleus of the inception module as they consider the close proximity of a signal x and the broader neighborhood scenarios at the same time. Every filter of the inceptionGCN module has its own parameter vector θ_s and they perform a convolution on the dataset X and give out the output y_s. The final output of each filter is combined together with an aggregator-function Ψ to get the output y of the inception module as $y = \Psi\left(y_1, \cdots, y_S\right)$, where every $\theta_s \in \mathbb{R}^{k_s}$ with entries $\theta_{r,s}$ is a vector of learnable parameter for each filter of the inception module. Further, to merge the output activations of each filter two aggregators Ψ are proposed,(1) concatenation and (2) max-pooling. The proposed model architecture is shown in Fig. 5.1 which accommodates M inception modules. Each inception module consists of S_m GC-layers. All the S_m layer are structured in parallel fashion with filters of different $k_{s,m}$. ReLU is used at the output of each GC-layer. A labelled subset of graph nodes are selected for the training. The loss is computed and gradients are backpropagated only for these training samples. Such setting is considered as transductive. We chose weighted cross-entropy as our loss function. Due to the graph connections, the training process on the labelled data is transferred to the unlabeled data by signal diffusion which corresponds to the behavior of a standard GCN.

5.3 Experiments and Results

In the previous section, the mathematical explanation for the need of the InceptionGCN model is provided. Here, two main experimental setups are presented main to show firstly, the influence of spectral convolutions from different graphs and kernel sizes of the filters, and secondly, the comparison of proposed InceptionGCN to other methods based on accuracy. Two publicly available multi-modal datasets are thoroughly analyzed for both the baseline [Par+17] and the proposed method. At last, the insights into generalized design choices for building a data and task-specific model is provided.

5.3.1 Datasets:

TADPOLE [Mar+18]: This dataset is a subset of the Alzheimer's Disease Neuroimaging Initiative (adni.loni.usc.edu), consisting of 557 patients with 354 multi-modal features per patient. The target is to classify each patient into one of the three classes (Cognitively Normal (CN), Mild Cognitive Impairments (MCI) or Alzheimer's Disease (AD). Features are extracted from MR and PET imaging, cognitive tests, CSF and clinical assessments. The protein class APOE constitutes another factor assisting in patient classification. Testing this gene status provides a risk factor of developing AD. FDG-PET imaging measures the brain cell metabolism, where cells affected by AD show reduced metabolism. Furthermore, demographics are provided (age, gender).

Graph construction: A binarized graph is built with every demographic data element (age, gender), APOE status, and FDG PET measures. For the rest of the three we chose $\beta = 0$ and for age we choose $\beta = 2$ respectively. The edges are based on the $Sim(x_i, x_j)$ i.e. the feature similarity measure. The 'Mixed' affinity graph is constructed by averaging all the graphs weighted with W and 'Mixed (noSim)' without weighting. The weight matrix W imposes the neighborhood shown in Fig. 5.4 on the binarized graph.

ABIDE [Abr+17]: The Autism Brain Imaging Data Exchange (ABIDE) aggregates data from 20 different sites and openly shares 1112 existing resting-state functional magnetic resonance imaging (R-fMRI) datasets with corresponding phenotypic elements (gender) for 2 classes normal and with Autism Spectrum Disorder (ASD). We choose 871 subjects divided into normal(468) and ASD diseased (403) subjects. For fair comparison, we follow the same pre-

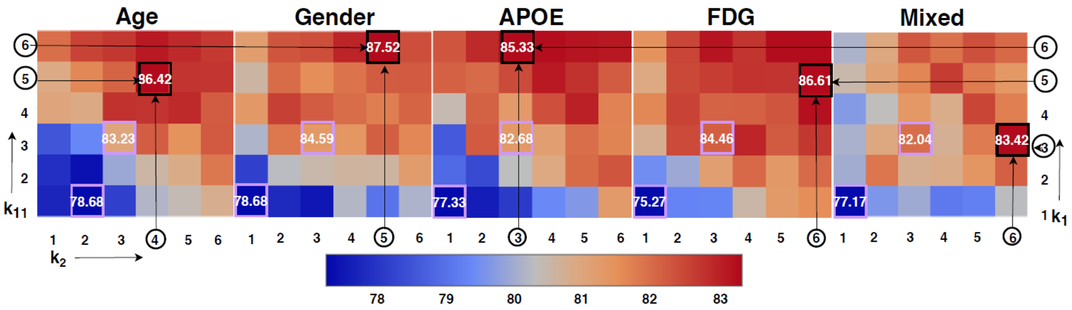


Fig. 5.2. Heat maps for representing the performance of GCN on the TADPOLE dataset with varying kernel size of the filters. Each heat map comes from the distinct graph mentioned above. The highest and lowest performing combination of k_1 and k_2 are highlighted with a black box and the corresponding k-values are shown.

processing step as performed in the baseline method [Par+17]. We choose the non-imaging elements gender and site, to construct two affinity graphs by choosing $\beta = 0$ for both graphs.

5.3.2 Experiments on medical datasets

Here, both the experimental setups mentioned above are shown and the subsequent findings on the medical datasets are dicussed.

Effect of different kernel size on spectral convolution This set of experiment is designed to investigate the optimal kernel size of the filter required for each graph. Each graph may contain multiple clusters with different variance and structure. The baseline model [Par+17] with two GC-layers in sequence is used to find out the required graph specific filter sizes (i.e. value of k). Keeping the input constant (features and graph) we vary k_1 and $k_2 \in [1, 6]$. Here, k=1 and k=6 indicate the kernel size of one-hop (smallest) and six-hop neighbors (largest) respectively. From obtained heatmaps the best performing combination of k_1 and k_2 are chosen for the InceptionGCN model as different kernel sizes. In this way, the sequential GCN is expected to work at its peak when compared with the proposed model. The validity of such setting is discussed in the later section.

Results: Fig. 5.2 shows the corresponding results in terms of heatmaps. The local and global features are learnt by the filters with smaller and larger value of k respectively. The results vary drastically with an average of 8% with the change in k_1 and k_2. It demonstrates models of spectral convolution are susceptible to k range. All the heatmaps show that the accuracy increases with the value of k, but saturates after certain value of k. It was observed that in most of the cases $k_1 > k_2$ is the best combination. This way initial layer filters look at global features in a same way as conventional CNNs. The accuracy shows variation as the graph structure and the value of k is changed. A similar trend is seen for ABIDE, which reassures the concept of sensitivity towards k.

InceptionGCN vs sequential GCN approaches A comparison with four baselines is shown here. Parisot et al. [Par+17] is the traditional GCN with $k_1 = k_2 = 3$. The same architecture of [Par+17] is modified with the best combination of the two ks mentioned as baseline $[k_1, k_2]$.The aggregator-function Ψ is also evaluated for a proper selection of activations from all the individual GC-layers of the inception module by comparing them to the baseline $[k_1, k_1]$ and $[k_2, k_2]$. The comparison shows that Ψ is not biased towards any particular kernel size.

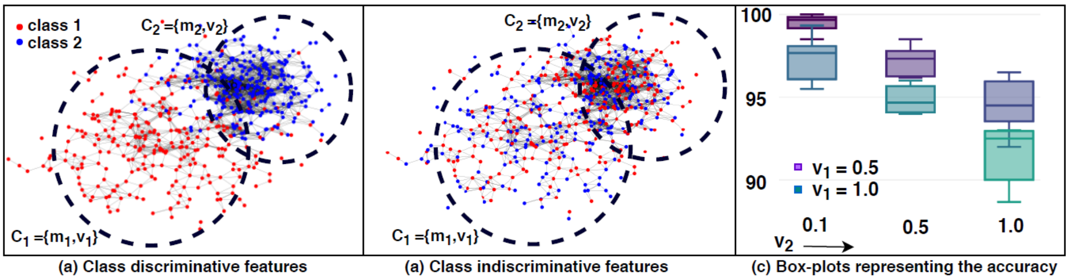


Fig. 5.3. (a) represents the scenario of simulated data, where we change variances v_1 and v_2, (b) shows the scenario where the features are sampled from random distribution, (c) shows the variation in the performance in terms of accuracy for all the combinations of v_1 and v_2 for scenario (a)

Each graph yields a different performance, with such a setting. This shows the effect of different neighborhood graph on the performance as shown in Tab. 5.2. The proposed model outperforms the baselines [Par+17] by an average margin of 4.12 % for TADPOLE dataset. The comparative results for ABIDE are given in Tab. 5.3. The proposed model performs comparable to the [Par+17] baseline, but can not surpass it. In case of feature-based edge weighting the performance decreases compared to the other weighting case. This shows the non-discriminative nature of the features. The images acquired from multiple sites make learning of the class-discriminative features more challenging for the model.

5.3.3 Experiments on simulated data

Seeing the contradictory performance on the two datasets, we investigate the model in detail for better understanding of the spectral model and to interpret better design choices for user-specific tasks. These experiments are specifically designed to investigate only the choice of the kernel size of the filters.

Two 2-dimensional clusters C_1 and C_2 having normal Gaussian distributions are created. Each cluster distribution with 300 points each in Euclidean domain representing one class. The graph is constructed based on the Euclidean distance between the features. The graph is sparsifed using $\beta = 0.5$. Such a setting shows that the graph is highly correlated to the labels. In order to keep the experiment easy to interpret, we set means $[m_1, m_2]$=$[-1, 1]$ for C_1 and C_2 respectively and vary the corresponding variances v_1 and v_2. For the features, two settings are shown: **class-discriminative**, here the (x,y)-values of the location of each point are considered as features and **class-indiscriminative**, here features are randomly sampled from a uniform distribution for both the classes. Fig. 5.3 (a) and (b) show both the setting. For the model architecture, M is kept equal to 1 for both the baseline models [Par+17] and InceptionGCN. Both the networks are trained for 200 epochs, with learning rate=0.2.

Results and interpretation Boxplots are used to demonstrate the results of this experiment in Fig. 5.3(c). Each box displays the accuracy of the classification for various k values, ranging from 1 to 10 for the class-discriminative features baseline model. Keeping v_1 = 0.5, v_2 is varied for [0.1, 0.5, 1.0]. This experiments is repeated with v_1=1.0. It can be perceived that the model is less sensitive to the value of k when two clusters are explicitly separable. It can also be seen from the last two box plots that the model becomes adaptive to k with

Tab. 5.1. The performance of the model in terms of accuracy is represented in the table. v_1 and v_2 represent the variances of 2 classes of the simulated 2D Gaussian data. (a) In these cases the graph and corresponding features are highly correlated to the classes, whereas in (b) only the graph is correlated to the classes.

		$v_1 = 0.5$		$v_1 = 1.0$	
	k	$v_2 = 0.1$	$v_2 = 1.0$	v_2=0.1	v_2=1.0
(a)	1	**98.50 ± 01.38**	94.50 ± 01.83	95.67 ± 02.49	92.50 ± 02.61
	10	**99.00 ± 01.11**	93.67 ± 04.93	95.50 ± 07.98	91.00 ± 04.42
	Inception-GCN (1 layer [k1,k2]=[1,10])	94.83 ± 03.02	**97.00 ± 02.56**	92.00 ± 03.56	**94.33 ± 03.56**
(b)	1	49.33 ± 06.84	50.33 ± 07.48	49.50 ± 04.60	50.00 ± 06.28
	10	60.33 ± 16.78	53.50 ± 10.99	**50.83 ± 06.02**	55.33 ± 14.79
	Inception-GCN (1 layer [k1,k2]=[1,10])	**66.50± 17.12**	**64.00 ± 17.95**	48.00 ± 07.88	**69.00 ± 24.79**

Tab. 5.2. Depicts the mean accuracies from stratified k-fold cross validation for all the setups of experiments for TADPOLE. The values of the chosen $[k_1, k_2]$ for the graphs are highlighted in the Fig. 5.2.

Affinity	Age	Gender	APOE	FDG	Mixed	Mixed (noSim)
Parisot et al. [Par+17]	82.55 ± 04.78	84.59 ± 04.82	82.68 ± 05.70	84.46 ± 0 5.46	82.04 ± 05.71	82.11 ± 04.94
Baselines						
$[k_1, k_2]$	86.42 ± 03.95	87.52 ± 03.51	85.33 ± 04.75	86.61 ± 04.53	83.42 ± 05.93	81.95 ± 05.92
$[k_1, k_1]$	85.46 ± 05.60	86.19 ± 04.91	85.08 ± 05.21	86.55 ± 04.55	81.85 ± 06.28	81.36 ± 05.98
$[k_2, k_2]$	86.42 ± 03.98	84.59 ± 04.82	78.75 ± 04.45	84.46 ± 05.46	80.86 ± 05.69	80.99 ± 04.71
InceptionGCN						
concat	**88.35 ± 03.03**	**88.06 ± 04.39**	**88.14 ± 03.20**	**86.99 ± 03.98**	**84.35 ± 06.97**	**83.62 ± 06.09**
max-pool	**88.53 ± 03.27**	**88.19 ± 03.83**	**88.49 ± 03.05**	**87.65 ± 05.11**	**84.11 ± 04.50**	**83.87 ± 05.07**

higher variance. Similar patterns are found as the value of v_1 is changed to 1.0, but $v_1 =$ 1.0 indicates a clear decrease in accuracy. Filters with a larger receptive field would produce generalized global features if there is a significant variation in the data.

In addition, we apply our model to simulated data with just one Inception module integrating two separate $[k1, k2]$=[1,10] GC layers. For four different settings, the results are compared of a single-layered GCN with k=[1,5,10] with one layered inception module. The superiority of our model is seen mainly in the challenging scenarios, where the variance of both classes is quite high (i.e. $v_1 = 1.0$ and $v_2 = 1.0$, cf. Tab. 5.1). Here, we report the results for class indiscriminative features, where the performance drastically drops when features are totally random for all the models. InceptionGCN outperforms the baseline in most of the cases.

5.4 Discussion and Conclusion

Our results show that both the spectral convolution and the proposed model obtained high classification accuracies for TADPOLE (cf. Tab. 5.2), with a clear margin of InceptionGCN over the baselines. In the case of the ABIDE dataset, however, both methods had comparable

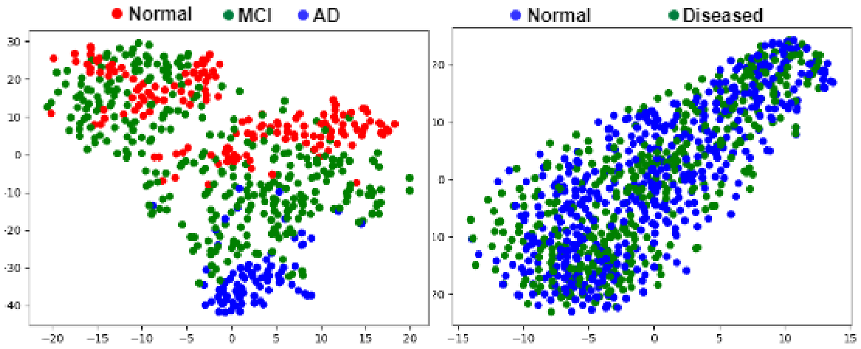


Fig. 5.4. TSNE embedding in 2-dimensional space visualized on raw features for TADPOLE(left) and ABIDE(right) datasets.

performance, which was considerably lower than on TADPOLE (cf. Tab. 5.3). To investigate the different performances of both models, we utilized simulated data with i) different degrees of class overlap in the feature space and ii) entirely random features, forcing the GCN models to rely on connectivity alone (Tab. 5.1). It can be concluded that while both GCN models are very sensitive to variance of data, our model shows the superiority in case of having large variances and overlapping of clusters from different classes. The main factors affecting the performance of GCN are features, graph and filters. With all the experiments we discuss all the factors in detail.

Influence of the graph: For the ABIDE dataset, images are collected from 20 different sites and imaging conditions, which adds considerable heterogeneity to the data. Consequently, the affinity graph based on site information consists of 20 disjoint clusters. Building a graph based on site information allows only the neighbors (i.e. samples from the same site) to contribute to the feature learning. This has less clinical relevance to the classification task, whereas for TADPOLE, the risk factors and demographics are clinically relevant. Such relevance of the graph can be determined using the graphs' energy function provided in [GHK13]. This energy increases if nodes from different classes are connected with high affinities. When many of

Tab. 5.3. Depicts the mean accuracy from stratified k-fold Cross Validation for all the setups of experiments for ABIDE. The baseline values of $[k_1, k_2]$ are [4,5], [6,5] and [4,4] for Gender, Site and Mixed, Mixed(noSim)$respectively$.

Affinity	Gender	Site	Mixed	Mixed(noSim)
Parisot et al [Par+17]	67.39 ± 04.76	67.39 ± 01.49	67.85 ± 00.63	69.80 ± 04.35
Baselines				
$[k_1, k_2]$	**68.19 ± 05.38**	**69.00 ± 04.07**	**70.26 ± 03.70**	**70.26 ± 04.58**
$[k_1, k_1]$	66.70 ± 06.90	68.65 ± 04.31	69.91 ± 07.50	69.80 ± 03.90
$[k_2, k_2]$	65.78 ± 06.50	68.65 ± 04.31	69.00 ± 03.80	69.46 ± 04.69
Inception-GCN				
concat	66.36 ± 05.66	67.97 ± 04.43	66.70 ± 06.27	69.23 ± 06.66
max-pool	67.05 ± 05.47	67.39 ± 05.80	66.02 ± 05.92	69.11 ± 06.68

these wrong connections exist, the relevance of the graph for the classification task will be low. Next, the mixed affinity graph performs worst overall in terms of accuracy (cf. Tab. 5.2 and Tab. 5.3) and Standard Deviation (SD) (cf. Tab. 5.3). This indicates that a straightforward creation of the mixed affinity graph by averaging impairs the inherent structure of each graph, and important clinical semantics from individual graphs may get lost. This is confirmed by the unequal performance observed for each affinity graph, which may even indicate a ranking of relevance of each non-imaging element to the objective. A more elegant way to combine all the affinity graphs is by ranking them while training [Kaz+18].

Influence of the features: The importance of a proper feature choice becomes clear in the tests on simulated data. When using randomly sampled features for every node (cf. Tab. 5.1) the overall performance drops drastically. A large standard deviation in the performance shows that filters are not learned properly and the model does not converge. The same behavior can be seen for the TADPOLE and ABIDE dataset when comparing the mixed and mixed (noSim) (cf. Tab. 5.2 and 5.3). Since the features of the ABIDE dataset are not distinguishing the nodes into different clusters compared to the TADPOLE dataset (Fig. 5.4), the performance of the models drops for ABIDE when using the feature similarity (Sim), which is used for graph construction. At the same time, the models receive a performance boost when the meaningful features of TADPOLE are included into the graph generation process.

Influence of the kernel size: We investigated the effect of features and heterogeneity of the graph towards the choice of k. Our results show that in case of class separable features, a larger value of k will give more compact features. From Tab. 5.1, it is clear that InceptionGCN performs better in case that the classes have large and different variances. In such a case, InceptionGCN with multiple k_s manages to capture the class discriminative features for the nodes. If the clusters are compact (v=0.1) the choice of k does not matter. From Fig. 5.3 (c), we see that the model is not sensitive to k if the clusters are compact, whereas it becomes sensitive when the variance increases. In case of class indiscriminative features and a less relevant graph (as is the case of ABIDE) a larger kernel size helps to learn global class discriminative feature.

Sequential model vs. InceptionGCN : Choosing the values of the two k from sequential model (GCN) for a parallel setting might seem ambiguous. In Tab. 5.2, the role of the aggregator-function is clearly visible in the performance, since the baselines are all the possible combinations that the final output of our model can get. Furthermore, our proposed model converges 1.63 times faster in terms of epochs compared to the baseline method when trained with early stopping criteria with window size of 25 due to a better feature learning process.

Future scope: Potential improvements of the InceptionGCN model include out-of-sample inference (i.e. inductive learning), which will highly improve the usability of the model. Another area of investigation is the integration of multiple affinity graphs into one model. Furthermore, the InceptionGCN model structure itself can also be optimized, first by using a learnable pre-processing step to obtain the neighborhood values k, and second, by analyzing the number of hidden units in each GC-layer and the overall number of inception modules necessary.

Part VII

GCNs and Graph Learning

GCN and Graph Learning 6

> *When we strive to become better than we are, everything around us becomes better too.*
>
> — **Paulo Coelho**
>
> ...

Graph deep learning has recently emerged as a promising ML tool enabling the generalization of effective deep neural architectures to structured data that are not Euclidean. These approaches have demonstrated promising results across a wide variety of applications from social science, particle physics, computer vision, graphics, and chemistry. One of the limitations of most current graph neural network architectures is that they are often confined to the transductive setting and rely on a precomputed graph. In other words these methods assume that the underlying graph is fixed and known. This presumption is not inherently valid in certain settings, such as those in medical and healthcare applications, because the graph may be noisy, partly- or even completely unknown, and one is therefore interested in inferring it from the data. This is particularly important in inductive settings when handling nodes that are not present in the graph at the time of training. In addition, such a graph may often itself communicate insights that are far more important than the downstream task. In this part, Differentiable Graph Module (DGM) is introduced, a learnable function that predicts the edge probability in the task-relevant graph, which can be combined with convolutional graph neural network layers and finally trained in an end-to-end fashion. An extensive evaluation for applications in healthcare (disease prediction), brain imaging (age prediction), computer graphics (3D dot cloud segmentation), and computer vision (zero-shot learning) are demonstrated in this chapter. The proposed model offers substantial improvement in both transductive and inductive settings over baselines, and achieves state-of-the-art performance.

6.1 Introduction

Geometric deep learning (GDL) is a new, emerging deep learning branch that attempts to generalize deep neural networks to non-Euclidean structured data such as graphs and manifolds [Bro+17; HYL17b; Bat+18]. Graphs are ubiquitous in various branches of science, in particular, being general abstract descriptions of relationships and interaction systems. Graph-based learning models have been successfully applied in social sciences [ZC18; Qi+18], computer vision and graphics [Qi+17; Mon+17; Wan+19b], physical [Cho+18b; Duv+15; Gil+17; Li+18c], medical, and biological [**Zitnik19**; Par+18; Par+17; Mel+19; Kaz+19c; ZAL18; Gai+19] sciences. The Graph Neural Networks (GNNs) are a common approach to graph learning. While dating back to at least [Sca+08], GNNs has become a useful and common tool mainly due to the recent development. Today's wide variety of GNN architectures includes spectral [Bru+13] and spectral-like [DBV16; KW16a; Lev+18; Bia+19] methods, local charting [Mon+17], and attention [Vel+17; Kon18; BL17; Mon+18]. Battaglia et al. [Bat+18] showed that most GNNs can be formulated in terms of message passing [Gil+17]. Assuming that the underlying graph is *given* and *fixed* is a notable downside to most GNN architectures, while graph-like operations usually amount to modifying the node-wise functionality. Architectures such as message passing neural networks [Gil+17] or primal-dual convolutions [Mon+18] also require updating of the edge features, but the graph *topology* still remains the same. This is also a limiting presumption. For some problems, the data can be assumed to have some underlying graph structure [LWC12], which is called *latent graph*, may not have the graph itself. For example, this is the case in medical and healthcare applications where the graph could be noisy, partially- or even entirely unknown, and one is therefore interested in infering it from the data. This is particularly important in inductive settings where some nodes in the graph may be present at testing but not at training. In addition, the graph can often be much more valuable than the downstream function, as it conveys some model interpretability. Graph topology inference is a longstanding problem that has been solved using the [**Dong19**; **Mateos19**] signal processing techniques. Several models dealing with latent graphs have recently been introduced in the machine-learning literature. Kipf et al. [Kip+18] used a variational autoencoder, where the latent code represents the interaction graph underlying a physical system, and the reconstruction is based on graph neural networks. Wang et al. [Wan+19b] proposed dynamic graph CNNs (DGCNN) for point cloud analysis, where a KNN graph is built on the fly inside the neural network's feature space. Zhan et al. [Zha+18a] proposed the construction and combination of multiple Laplacians with learnable weights. Li et al. [Li+18a] similarly proposed a spectral graph convolution method, in which a residual Laplacian defined on the feature output from each layer and the input Laplacian is modified for each layer. With the Laplacians, both the approaches learn the graph but still require an initial graph. Huang et al. [Hua+18] proposed a further variant of spectral filters which would parameterize the Laplacian instead of the filter coefficient. Jiang et al. [**jiang2019semi**] proposed a model where graph learning and graph convolution are integrated into a unified network architecture, but it is computationally expensive to learn attention coefficients on node distances. Franceschi et al. [Fra+19] formulated graph learning as a problem of bilevel optimization, by modeling and optimizing the graph as a hyper-parameter with a separate loss. This method is transductive and does not increase in scale. **Main contributions** In this part a model is proposed that simultaneously learns the graph and graph convolutional filters on it. Several of the proposed model 's settings are demonstrated, including a continuous and discrete differentiable graph construction, and show how it can be optimized. It is also shown in the preceding graph learning methods can be considered as our model's particular settings. A detailed ablation analyses of the proposed model and evaluation on healthcare and brain imaging applications (disease and age prediction), computer graphics (3D point cloud segmentation), and computer vision (zero shot learning) is shown. The proposed model demonstrates substantial changes over baselines and produces cutting-edge performance in both transductive and inductive environments.

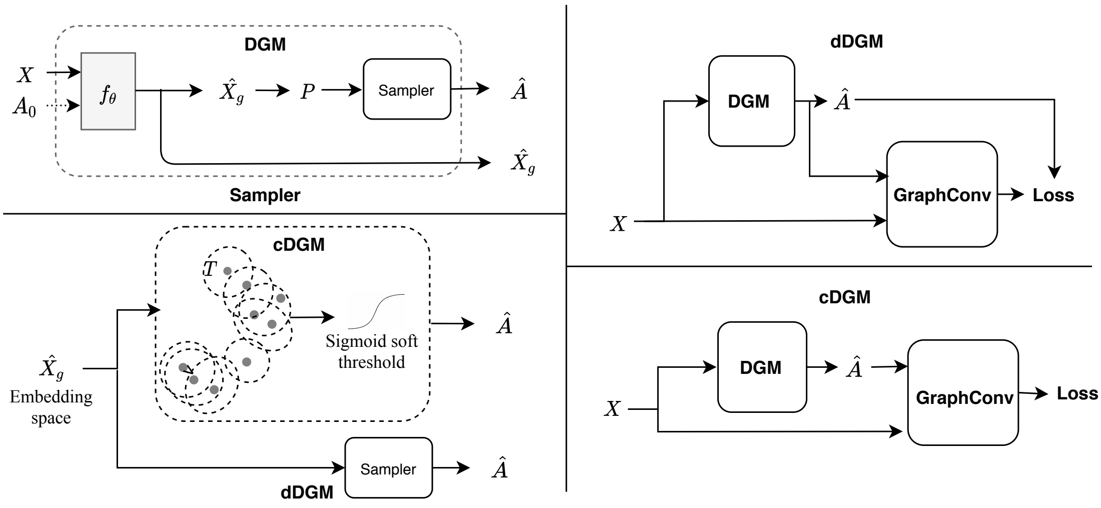


Fig. 6.1. *Left:* Two-layered architecture including Differentiable Graph Module (DGM) that learns the graph, and Diffusion Module that uses the graph convolutional filters. *Right:* Details of DGM in its two variants, cDGM and dDGM.

6.2 Background

Given a set of N data points with dimension d, denoted as $\mathbf{X} \in R^{N \times d}$, a common challenge in machine learning is to generate a representation that is conscious of the underlying structure of the data. This structure may be represented as a graph $\mathcal{G} = (\mathcal{V}, \mathbf{A})$ where $\mathcal{V} = \{1, \dots, n\}$ is the set of vertices and $\mathbf{A} = (a_{ij})$ is a matrix of (weighted) adjacency. $\mathbf{A}$ is used to define the edges of the graph $\mathcal{E} = \{(i, j) : a_{ij} > 0, \, Ij \in \mathcal{V}\}$; the edge weight $a_{ij} \geq 0$is the affinity between $\mathbf{x}i$ and $\mathbf{x}_j$ scales.

Assuming this structure is given with the details, we have a graph attributed to the nodes denoted as $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{X})$ on which a graph neural network (GNN) can be implemented. GNN attempts to find an *embedding* $\mathbf{Z} = f_{\mathbf{\Theta}}(\mathbf{X}, \mathbf{A})$ by doing message passing [Gil+17; Bat+18]

$$\mathbf{z}_i = \sum_{j \in \mathcal{N}_i} h_{\mathbf{\Theta}}(\mathbf{x}_i, \mathbf{x}_j, a_{ij}) \tag{6.1}$$

in a local neighborhood $\mathcal{N}_i = \{j : (i, j) \in \mathcal{E}\}$ of the node. Here $h_{\mathbf{\Theta}}$ denotes a learnable function shared across nodes, the parameters $\mathbf{\Theta}$ of which are chosen to minimize a downstream loss. Equation (6.1) is often referred to as *edge convolution* (EC) [Wan+19b] owing to the extensive usage of the conventional convolution operation on grid. A special case of (6.1) of node-wise linear transformation $h_{\mathbf{\Theta}} = a_{ij}\mathbf{\Theta}\mathbf{x}_j$ by matrix $\mathbf{\Theta}$ is called *graph convolution* (GC) [DBV16; KW16a]. The node embeddings can be used for node-wise classification, or pooled for graph-wise classification tasks.

Latent graphs In this part of the thesis, we are interested in learning the unknown graph. Knowing the graph has two purposes: First, to represent structure of the data. Second, it is used to achieve the embedding of the data points as support for graph-based convolutions. Wang et al. [Wan+19b], who proposed the edge convolution (6.1) with $h(\mathbf{x}_i, \mathbf{x}_j - \mathbf{x}_i)$ on a KNN graph constructed from the data points $\mathbf{X}$, is the most related approach to this paper. Dynamic Graph CNN (DGCNN) allows the graph to be updated on the fly between the layers of the network, hence the name of the method. The biggest obstacle to incorporate graph building as part of the deep learning framework is that it is a discrete, non-differentiable structure. DGCNN optimizes the graph convolution filters and layer activations for the downstream tasks of classification and segmentation. However, the graph is constructed ad-hoc using a KNN rule on the input activation of each layer, without a dedicated loss for the graph to be learned. As such, the graph is dynamically constructed but not learned, and the underlying latent graph of the domain is not recovered. The proposed method, described in the next section, aims at addressing these issues.

6.3 Method

6.3.1 Architecture

In this chapter, a general graph learning strategy is proposed based on the output features of each layer and the graphs are optimized during training together with the network parameters. The architecture consists of two major components, the **Differentiable Graph Module (DGM)** and **Diffusion Module**, which are shown in Figure 6.1 and described below.

Differentiable Graph Module: The DGM is dedicated to build a (weighted) graph that represents the input space. DGM takes as input the feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ (and optionally an initial graph $\mathcal{G}_0$) and gives the output graph $\mathcal{G}$. Since the $\mathcal{V}$ node set is fixed, its adjacency matrices $\mathbf{A_0}$, and $\mathbf{A}$ can represent the two graphs. The input features $\mathbf{X} \in \mathbb{R}^{N \times d}$ are first transformed into *auxiliary features* $\hat{\mathbf{X}} = f_{\mathbf{\Theta}}(\mathbf{X}) \in \mathbb{R}^{N \times \hat{d}}$ by means of a parametric function $f_{\mathbf{\Theta}}$, which typically reduces the input dimension ($\hat{d} \ll d$). If the initial graph $\mathcal{G}_0$is given, the general form $f_{\mathbf{\Theta}}$(6.1) can be used, where new $\hat{\mathbf{X}}$ features are computed by edge- or graph-convolution on $\mathcal{G}_0$. Otherwise, $f_{\mathbf{\Theta}}$ is applied to each node feature independently, acting row-wise on the matrix $\mathbf{X}$. Second, the auxiliary features $\hat{\mathbf{X}}$ are used for graph construction. The edge probability are thus defined as $p_{ij}(\mathbf{X}; \mathbf{\Theta}, t) = e^{-t\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|_2^2} = e^{-t\|f_{\mathbf{\Theta}}(\mathbf{x}_i) - f_{\mathbf{\Theta}}(\mathbf{x}_j)\|_2^2}$, where t is also a learnable parameter. For the sake of simplicity, Euclidean metric for defining the edge probability is chosen and other metrics, e.g. hyperbolic [**poincare_embeddings**; **hyperbolic_fisheye**; Kri+10], could also be used. A straightforward way to derive a graph $\mathcal{G}$ is to transform the probability matrix $\mathbf{P}(\mathbf{X}; \mathbf{\Theta}, t)$ into a weighted adjacency matrix, e.g. by soft-thresholding the distances $\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|$ using the sigmoid function $a_{ij} = 1/(1+p_{ij}e^{tT})$, where T denotes the threshold. Thus, the adjacency matrix $\mathbf{A}(\mathbf{X}; \mathbf{\Theta}, t, T)$ represents the graph and is parametrized through $\mathbf{\Theta}, t$ and the additional parameter T, and is differentiable w.r.t. these parameters. This version of the proposed architecture, is refered to as *continuous DGM (cDGM)*. One of the potential drawbacks of cDGM is that it can generate a dense matrix of adjacence, i.e. a fully connected graph with several edges having near-zero weight. As an effective alternative, the *Gumbel-Top-k trick*[KHW19] can be used to sample edges from the probability $\mathbf{P}(\mathbf{X}; \mathbf{\Theta}, t)$ to generate a sparse k-degree graph. Such sampling can be regarded as a stochastic relaxation of the KNN rule. For each node i, k edges $(i, j_{i,1}), \ldots, (i, j_{i,k})$ are extracted as the first k elements of $\mathrm{argsort}(\log(\mathbf{p}_i) - \log(-\log(\mathbf{q}))$, where $\mathbf{q} \in \mathbb{R}^N$ is uniform i.i.d. in the interval $[0, 1]$. The extracted samples thus follow the categorical distribution $p_{ij}/\sum_r p_{ir}$ [KHW19]. Now, the edge set of this sparse graph $\mathcal{G}$ can be constructed as $\mathcal{E}(\mathbf{X}; \mathbf{\Theta}, t) = \{(i, j_{i,1}), \ldots, (i, j_{i,k}) : i = 1, \ldots, N\}$ and represent it by the unweighted adjacency matrix

$\mathbf{A}(\mathbf{X}; \mathbf{\Theta}, t)$. The main benefit of this matrix is its sparse nature. Next we demonstrate how to learn the parameters $\mathbf{\Theta}, t$ efficiently. This version of the proposed architecture will henceforth be referred to as *discrete DGM (dDGM)*. Note that since the sampling scheme for the dDGM graph is stochastic, the network's prediction at inference time is not deterministic. Taking advantage of this, a consensus scheme is implemented. The classification task is run for 8 times in our experiments, and maximum of the cumulative soft predictions is selected as the predicted class.

Diffusion Module: This module takes as input the $\mathbf{G}$ graph generated by the DGM and the $\mathbf{X}$ features, and gives a new set of $\mathbf{X}' = g_{\mathbf{\Phi}}(\mathbf{X})$ output features. Here, $g_{\mathbf{\Phi}}$ represents a general function of the form (6.1); it is either edge- or graph-convolution on $\mathcal{G}$ in the following experiments.

Combined model: A multi-layer network is used in which each node, numbered as $l = 1, \ldots, L$, contains a **DGM** and **Diffusion Module**, as shown in Figure 6.1. The output of lth layer is given by,

$$\hat{\mathbf{X}}^{(l+1)} = f_{\mathbf{\Theta}}^{(l+1)}([\mathbf{X}^{(l)} \,|\, \hat{\mathbf{X}}^{(l)}], \mathbf{A}^{(l)}) \qquad \mathbf{A}^{(l+1)} \sim \mathbf{P}^{(l)}(\hat{\mathbf{X}}^{(l+1)}) \qquad \mathbf{X}^{(l+1)} = g_{\mathbf{\Phi}}(\mathbf{A}^{(l+1)}, \mathbf{X}^{(l)})$$

Assumptions:

- $\mathbf{X}^{(0)} = \mathbf{X}$
- $\mathbf{A}^{(0)} = \mathbf{I}$, if no intial graph is given (i.e., the initial graph is $\mathcal{G}^{(0)} = (\mathcal{V}, \emptyset)$)
- $f_{\mathbf{\Theta}}^{(0)}$ is a node-wise function (MLP).

Further, based on the task at hand, the output activations $\mathbf{X}^{(L)}$ of the last layer L can be fed as input to an MLP to obtain the final node predictions. DGCNN can be obtained as one of the basic setting of our model $f\mathbf{\Theta} = \mathrm{id}$ in the **DGM** module and using the **Diffusion** module for edge convolution.

6.3.2 Training

The sampling scheme followed in dDGM prevents the gradient of the downstream classification loss function to flow through our network's graph prediction branch. Hence the graph prediction branch only involves graph features $\hat{\mathbf{X}}$. Here, for the graph optimization we define a loss inspired from reinforcement learning [**Ronald1992**], that rewards edges involved in correct classification and penalizing the ones that led to misclassification. Let the node-wise labels predicted by the proposed model be $\mathbf{y}$ and $\tilde{\mathbf{y}}$ the groundtruth labels. The reward function $\delta(y_i, \tilde{y}_i)$ can then be given by taking value -1 if $y_i = \tilde{y}_i$ and 1 otherwise. We derive the graph loss as

$$L_{\text{graph}}(\mathbf{\Theta}^{(1)}, \ldots, \mathbf{\Theta}^{(L)}) = \sum_{i=1}^{N} \delta(y_i, \tilde{y}_i) \sum_{l=1}^{L} \sum_{j:(i,j)\in\mathcal{E}^{(l)}} \log\, p_{ij}^{(l)}(\mathbf{\Theta}^{(l)}) \tag{6.2}$$

The gradient of the above graph loss approximates the gradient of the expectation $\mathbb{E}_{(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(L)})\sim(\mathbf{P}(\mathbf{\Theta}^{(1)}),\ldots,\mathbf{P}(\mathbf{\Theta}^{(L)}))} \sum_i \delta(y_i, \tilde{y}_i)$ with respect to the parameters of the graphs in all the layers. The samples from different classes are weighted unevenly by the Equation (**??**)

Tab. 6.1. Classification accuracy in % on Tadpole in the transductive setting. The top three performance scores are highlighted in color as: **First**, **Second**, **Third**.

Method	Accuracy
Linear classifier	70.22±6.32
Multi-GCN [Kaz+19a]	76.06±0.72
Spectral-GCN [Par+17]	81.00±6.40
InceptionGCN [Kaz+19c]	84.11±4.50
DGCNN [Wan+19b]	84.59±4.33
LDS [Fra+19]	**87.06±3.67**
cDGM	**92.91±2.50**
dDGM	**94.14±2.12**

in the early stages of the training especially when the classification accuracy is poor. This leads the network to support an uniform estimation of low probabilities for all the edges. The model weighs positive and negative samples according to the current per-class precision to avoid this behavior:

$$L_{\text{graph}}(\mathbf{\Theta}^{(1)},\ldots,\mathbf{\Theta}^{(L)}) = \sum_{c}^{\#\text{Classes}} \sum_{i\in c} \delta_c(y_i,\tilde{y}_i) \sum_{l=1}^{L} \sum_{j:(i,j)\in\mathcal{E}^{(l)}} \log\, p_{ij}^{(l)}(\mathbf{\Theta}^{(l)}) \tag{6.3}$$

with $\delta_c(y_i,\tilde{y}_i)$ being the class accuracy acc_c computed on the current prediction if $y_i \neq \tilde{y}_i$, or $\delta_c(y_i,\tilde{y}_i) = 1 - \text{acc}_c$ otherwise. Uneven distribution of the samples between the different classes in the dataset are dealt with a per-class accuracy rather than a global accuracy. The addition of Graph loss L_{graph} and classification loss are then optimised together.

6.4 Experiments and Results

The proposed method is tested on four applications from different domains (healthcare, computer graphics, computer vision). The four applications specifically are disease prediction, age prediction, 3D point cloud segmentation and, zero-shot learning.

Healthcare and Brain imaging applications Here two datasets are used. 1) *Tadpole* [Mar+18] consists of 564 patients. Each provided with 354 dimensional representation vector derived from imaging (MRI, fMRI, PET) and non-imaging (demographics and genotypes) features. The task is to classify each patient as 'Normal Control', 'Alzheimer's Disease' and 'Mild Cognitive Impairment'. *UK Biobank*[Mil+16] is selected for the second dataset. This consists of 14,503 individuals, each with a 440 dimensional feature derived from brain MRI and fMRI imaging. The task here is to classify the age group of the patient (50-59,60-69, 70-79, and 80-89). All tasks are either transductive or inductive, where all nodes are provided during training in the former setting but the labels of the test nodes are kept, while in the latter setting test nodes are completely removed during training and reintroduced only during testing. Previous methods [Par+17; Kaz+19a; Kaz+19c; Kaz+19b] used GNNs with hand-crafted population graphs focused on non-imaging meta-functions such as patient age and sex. The proposed method

Tab. 6.2. Scalability: training and test iteration times for different number of nodes.

Training iteration			
Method	n =564	5k	10k
DGCNN	6.99ms	28.2ms	104ms
LDS [Fra+19]	1.84s	>30m	>30m
cDGN	7.35ms	47.8ms	211ms
dDGN	8.29ms	37.0ms	141ms

Test iteration			
Method	n =564	5k	10k
DGCNN	4.60ms	25.2ms	102ms
LDS [Fra+19]	1.84s	>30m	>30m
cDGN	3.02ms	15.1ms	51ms
dDGN	3.97ms	24.6ms	104ms

Tab. 6.3. Classification accuracy in % for disease and age prediction tasks in the *transductive* and *inductive* settings on the Tadpole (left) and UK Biobank datasets (right). †Does not support inductive setting.

Method	TADPOLE		UK Biobank	
	Transductive	Inductive	Transductive	Inductive
DGCNN	84.59±4.33	**82.99±4.91**	58.35±0.91	**51.84±8.16**
LDS	**87.06±3.67**	†	OOM	†
cDGM	**92.91±2.50**	**91.85±2.62**	**61.32±1.51**	**55.91±3.49**
dDGM	**94.10±2.12**	**92.17±3.65**	**63.22±1.12**	**57.34±5.32**

allows the graph to be learned directly from the input features of the patients, without any pre-computed graph. The following methods are used as baselines: simple linear classifier as a non-graph method. Multi-GCN [Kaz+19a], Spectral-GCN [Par+17], InceptionGCN [Kaz+19c] as graph methods with hand-crafted graph; and DGCNN [Wan+19b] and LDS [Fra+19] as methods that learn the graph. We note that LDS does not support inductive learning. In Tables 6.1 and 6.3 shows results using 10-fold cross validation. As can be seen the proposed method shows significantly better results. In terms of training ans testing times, our model is on par with DGCNN and about three orders of magnitude faster than LDS as shown in Table 6.2.

Ablation study

The functions f and g are generic and could be chosen according to the task. In this section, an ablation study of different configurations of our architecture, in particular, the choice of functions f and g (identity, node-wise (MLP), graph convolution (GC), edge convolution (EC)) and the graph construction strategy (cDGM and dDGM) is shown for Tadpole dataset in transductive setting. The architecture for the ablation study consists of two convolutional layers with output size of 16. Results are shown in Table 6.4. It was observed that with $f = \mathrm{id}$

DGCNN is obtained, here a graph based on kNN selection is dynamically computed only on the space of node representation at each epoch; this setting is significantly inferior to the use of graph based convolution. The best found is using graph convolution for both f and g.

Computer graphics DGCNN architecture from [Wan+19b] is mimiced for this task. However, their graph kNN sampling scheme is replaced by our DGM with the feature depth of 16. Further $k = 20$ and other training parameters are kept same as DGCNN. During inference time, given the stochastic nature of the predicted graph, the classification of each point is repeated for 8 times and the $argmax$ of the cumulative soft predictions is chosen.
The mean Intersection-over-Union (mIoU) values are reported in Table 6.5. These value are calculated by averaging the IoUs of all testing shapes. The proposed approach makes performance improvements on almost all shape groups over the original kNN sampling scheme, which is considered very difficult. Figure 6.2 shows the sampling probabilities of some points (denoted by red) on different shapes for two layers of the network respectively. It can be noticed that the probability of the connecting two points is not related to the point feature space which is used for classification. However it retains certain spatial information and seems to be inspecting symmetries of the shape.

Zero-shot learning (ZSL) in computer vision In ZSL, the task is to learn classifiers for the unseen classes and solve the classification problem for samples belonging to unseen classes based on training data of only seen classes. The most popular approach is to train a network to predict a vector representation for a class starting from some implicit prior knowledge, i.e. semantic embedding [Xia+18]. Recent works showed that using additional explicit relations between classes in the form of knowledge graphs can help to significantly improve the learning of classifier for the unknown classes and hence the classification accuracy for unseen data samples.

Formally, let $\mathbf{X} \in \mathbb{R}^{N \times S}$ be the semantic embeddings (i.e. word vectors) associated with each category class. The Zero-Shot task loss is defined as the summation over all the $M < N$ training classes of $\sum_{i=1}^{M} \|\mathbf{w}_i - \tilde{\mathbf{w}}_i\|_2^2$, where $\mathbf{w}_i$ and $\tilde{\mathbf{w}}_i$ are the predicted and ground-truth vector representation of the ith class, respectively. Note that, even if in ZSL we deal with a regression problem, it is straightforward to adapt it to deal with our graph loss defined in equation 6.3, considering $\operatorname{argmin}_j \|\mathbf{w}_i - \mathbf{w}_j\|_2$ as the predicted category for sample $\mathbf{x}_i$.

Mimicking [Kam+19], our model consists of two graph convolution layers with hidden and output layer of dimension 2048 and 2049, paired with two DGM layers of dimension 16 for

Tab. 6.4. Ablation study on Tadpole transductive task. Shown is classification accuracy for different architectural choices. Notation $f+g$ refers to the choice of the DGM and Diffusion modules (GC: graph convolution, EC: edge convolution; I: identity, MLP: multilayer perceptron). *Configuration equivalent to DGCNN.

	I + EC	MLP + GC	GC + GC	MLP + EC	GC + EC
cDGM	—	92.42±3.82	90.68±4.58	92.29±4.18	91.78±3.21
dDGM	84.27±4.20*	93.47±3.82	**94.09±1.81**	93.27±3.20	**94.14±2.12**

Tab. 6.5. Comparison of mIoU(%) score in ShapeNet part segmentation task.

	# Shapes	DGCNN	dDGM
Airplane	2690	84.0	**84.1**
Bag	76	**83.4**	82.5
Cap	55	**86.7**	84.6
Car	898	77.8	**77.9**
Chair	3758	90.6	**91.3**
Earphone	69	74.7	**79.0**
Guitar	787	91.2	**92.5**
Knife	392	87.5	**87.7**
Lamp	1547	82.8	**83.7**
Laptop	451	95.7	**96.5**
Motorbike	202	66.3	**66.8**
Mug	184	94.9	**95.1**
Pistol	283	81.1	**83.1**
Rocket	66	**63.5**	62.3
Skateboard	152	74.5	**77.8**
Table	5271	**82.6**	82.2
MEAN		85.2	**85.6**

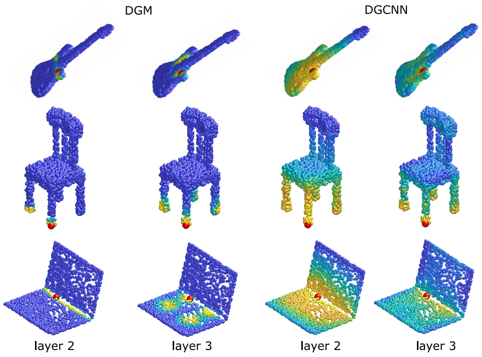


Fig. 6.2. Comparison between our DGM and DGCNN sampling on the feature space in the last two convolutional layers of the network. In DGM the colormap encodes the probability of each point to be connected to the red point. For DGCNN we plot the exponential of the negative Euclidean distance on feature space.

graph representation and $k = 3$. Each layer is composed by the following convolution on graphs:

$$\mathbf{X}^{(l+1)} = \sigma\left((\mathbf{D}^{(l)})^{-1}\mathbf{A}^{(l)}\mathbf{X}^{(l)}\mathbf{\Theta}^{(l)}\right) \tag{6.4}$$

where $\sigma(\cdot)$ is a *LeakyReLU* non linearity, $\mathbf{\Theta}^{(l)}$ are the learned weights and $(\mathbf{D}^{(l)})^{-1}\mathbf{A}^{(l)}$, with $d_{ii}^{(l)} = \sum_j a_{ij}^{(l)}$ is the non-symmetric normalization of the adjacency matrix $\mathbf{A}^{(l)}$. Essentially, our model is the same as SGCN [Kam+19], where we replace the input knowledge graph by our DGM module for learning $\mathbf{A}$.

Following [Kam+19], we use weights of the last fully connected layer of a ResNet-50 [He+16] pre-trained on ImageNet 2012 dataset [Den+09] as our target vector representation $\tilde{\mathbf{y}}_i \in \mathbb{R}^{2049}$. Input semantic features $\mathbf{x}_i \in \mathbb{R}^{300}$ are extracted with GloVe text model [PSM14] trained on Wikipedia dataset.

We train our model on the 21K ImageNet dataset classes, where we have as input the semantic embedding for all classes, but only the first 1K have a corresponding ground-truth vector representation. The model is trained for 5000 iterations on a randomly subsampled set of 7K categories containing all the 1K of training.

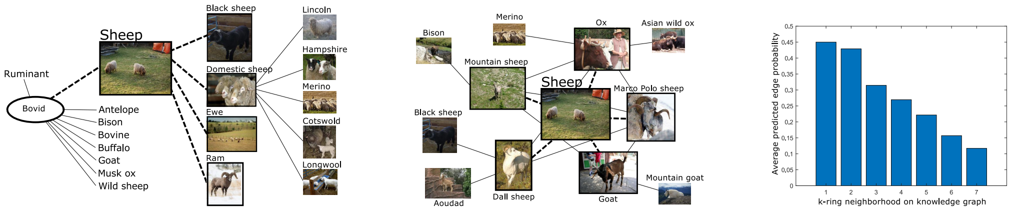


Fig. 6.3. Example of the 2-ring neighborhood of the "sheep" category on the knowledge graph (left) and on our predicted graph (center) sampled considering the 5 most probable edges. On the right the average predicted probability of edges belonging to the k-ring neighborhood (AwA2 test categories). Higher probabilities corresponding to nearest neighbors suggest that the predicted graph structure is loosely related to the knowledge graph.

Testing is performed on AWA2 dataset, composed of 37,322 images belonging to 50 different animal classes. We use the test split proposed in [WYG18] comprising images from 10 classes not present in the first 1K of ImageNet used for training. Top-1 accuracy for dDGM is 74.7%, greater than that of GCNZ [WYG18] (70.5%) but lower than DGP (77.3%). Note that, unlike the last two methods, we do not make use of the knowledge graph. As shown in Figure 6.3, the knowledge graph seems indeed a good graph representation for zero-shot task. Our predicted graph shows some similarity to it, however, fails to capture its hierarchical structure. We leave for future work imposing additional constraints on the graph structure, e.g. making it tree-like.

6.5 Discussion and conclusion

In this chapter, the challenge of graph learning in graph neural networks is addressed. The proposed Differentiable Graph Module (DGM) predicts a probabilistic graph which allows a discrete graph to be sampled accordingly. This sampled graph is used further in any graph convolutional operator. Further, a weighted loss inspired by reinforcement learning allowing for the optimization of edge probabilities is proposed.

The proposed DGM is very generic and adaptable to any graph convolution based method. The generic nature of DGM is shown with wide variety of applications mainly in healthcare (disease prediction), brain imaging (age and gender prediction), computer graphics (3D point cloud segmentation) and computer vision (zero-shot learning). In these applications we also deal with multi-modal datasets and inductive settings.
Further, we discuss some open problems with the proposed method. Even though, the proposed method is computationally more lightweight than the existing approaches (e.g. [JML19]), quadratic complexity with respect to the number of input nodes still exist. This is because it requires the computation of all pairwise distances. Limiting probability calculation in a node neighborhood and using a tree-based algorithm may help minimize the complexity to $\mathcal{O}(n \log n)$. In addition, our selection of sampling k neighbors does not take into account the graph's heterogeneity in terms of node degree distribution. Other sampling schemes such as (e.g. the threshold-based sampling [JML19]) may be explored. It would also be useful to consider prior knowledge of the graph, e.g. by imposing a node degree distribution, or by presenting an initial input graph that can be adapted for a particular task.
In a broader perspective, this work addresses a significant gap in graphical neural network re-

search, addressing the scenario when the graph is unknown. In addition to providing improved quantitative results on the downstream mission, our approach potentially provides a means of understanding what geometric deep models are learning. This knowledge exploration is presented in the form of a graph that matches the data, which may provide useful insights in certain applications and be even more useful than the downstream task itself.

Encouraging results on patient population databases offer potential use or method for computer-aided diagnostics, which could have a significant positive impact on healthcare. At the same time, it is not clear how our approach (and generally graph neural networks) copes with adversarial attacks that could pose such risks when used in sensitive environments such as healthcare.

Part VIII

Discussion and Conclusion

7 Discussion and Conclusion

> *An expert is a person who has made all the mistakes that can be made in a very narrow field.*
>
> — **Niels Bohr**
>
> ...

7.1 Discussion

In terms of effectiveness for solving numerous graph data based problems, GNNs have already been proven to be powerful, but some open challenges still exist which could be attributed to the complexity of graphs. Some of these challenges are discussed in this section along with possible future directions:

Model depth: It is proven in literature that the depth of deep neural models are pivotal for the success of deep learning [He+16]. In other words the number of layers in a model extract low to high level information from the input to learn a rich representation [He+16].

In the case of GCNs, [LHW18] demonstrates that as the number of graph convolutional layers increases the model performance drastically decreases. This behavior can be explained by analyzing the process of graph convolution. Theoretically, at the first GC layer, the central node transforms into the weighted summation of its neighbors. In the next layer, the same central node implicitly accumulates the information from its two-hop neighbors, and this process continues to occur in all subsequent layers. Eventually, if in theory, the number of GC layers increases to infinity, then each node will have the mean representation of the population. This means that all nodes representations will converge to a single point in the data embedding [LHW18]. Considering this limitation, the depth of the GCN should be assessed for optimal results on graph data. In particular, this choice could be governed by the dataset size, variance in the input features with respect to the class distribution, in case of classification and degree distribution. The larger and more heterogeneous the dataset, the more depth can be increased accordingly. The degree distribution factor can be used in case the graph is given. The denser the graph is in terms of the degree distribution, the chances of learning the average node representations are high. In such a scenario, shallow GCN architectures could be more suitable.

Scalability trade-off: For GCNs the scalability of models is a crucial factor during deployment. Scalability associated with GCNs can be measured in the following three ways:

- Number of nodes: The dataset size is particularly important in the case of transductive settings as the entire dataset together with the graph needs to be loaded onto the GPU. This can be solved by iteratively sampling from the entire dataset. This sampled data can be considered representative of the entire dataset. The mean performance of multiple such iterations can be then consider as the final performance of the model.

- Number of edges: Fully connected graphs are not a great choice to use during the model training, as they do not inherit any latent structure and the heat dissipation in the graph Laplacian would converge all the nodes to a single point. This mean all the nodes will be acquire the mean representation of all the points. Generally, graphs are pruned using various methods such as thresholding, pooling, sampling etc.

- Number of graphs: As stated in chapter 3, multiple graphs are essential for the better performance of the model. However, the increase in the number of graphs would explode the number of learnable parameters and eventually the computational complexity. One possible solution to such problem of scalability is solved in chapter 6, where the features selection process is designed such that the final graph is learned based on weighted input features, otherwise used to create separate graphs.

Directional and signed graphs: All the methods in this thesis are designed for bidirectional graphs. However, these graphs can either be directional or signed. Many applications such as brain signal analysis, cardiac mesh analysis require directional and signed graphs respectively. Both signed and directional graphs raise additional challenges to the existing methods [Deb+91]. Hyper-graphs on the other hand representing complex relationships among multiple objects are studied [DD03]. Only a few works like [LNK19] and [Gul+18] in the recent past have focused on designing deep learning models to handle such graphs.
Dynamic graphs: Even though in chapter 6, we illustrated a model that learns a graph during training, the number of nodes still stay constant. However, many real graphs are dynamic in nature which means nodes and edges may change with time. One such example could be a gigantic social network like Facebook, where the users (nodes) may develop multiple connections over time. Further, new users may join and the existing ones may leave. Handling of such graphs is out of scope of this thesis and could be a potential future direction. Some of the relevant works in this direction so far are [DMT18] and [Pha+16].

Interpretability: In the medical domain, interpretability is crucial irrespective of the type of model. For instance, in decision making and treatment planning procedures, interpretability is important while deploying the deep algorithms into the clinical usage system. Interpretability for GCNs is not straight forward as the outcome depends mainly on four factors, which are 1) input features, 2) input/ learned graph, 3) network parameters and 4) network training. Some of the methods in this direction are [Yin+19; Hua+20; Yua+20]. Interpretability of GCN has a huge scope in medical domain. One of the latest works in this regard is [Jau+20].

7.2 Conclusion

GCNs have been recently introduced in healthcare. This thesis successfully demonstrated applications of GCNs in healthcare by: 1) integrating GCN in medical domain for applications such as Alzheimer's classification, age and gender prediction, Autism prediction and Parkinson's prediction. On the technical front, four main questions were answered, which are 1) how to handle multiple graph scenario? 2) how to handle multiple graphs for personalised medicine? 3) how to handle intra-graph heterogeneity? and 4) how to learn a latent graph?

All the models proposed in this thesis exhibit state of the art results on various medical problems using four publicly available datasets, medical datasets and a few synthesized datasets. Furthermore, the graph learning method presented in this thesis is also applied to computer vision problems such as point cloud segmentation and zero-shot learning. All the models designed and proposed throughout this thesis are robust and computationally light-weight.

In conclusion, this thesis is one of the pioneer works which establishes that deep learning on graphs is a promising and a fast-developing area of research for achieving better outcomes in healthcare applications. Further research in this direction shall provide a critical building block in modeling CADx systems for both single-modal and multi-modal data. Graph deep learning is definitely an important step towards ushering in a better era of machine learning and artificial intelligence, especially in the field of healthcare.

Part IX

Appendix

A List of Authored and Co-authored Publications

2020

[Kaz+20] **Kazi, A.**, Cosmo, L., Navab, N. and Bronstein, M., 2020. Differentiable Graph Module (DGM) Graph Convolutional Networks. arXiv preprint arXiv:2002.04999.

[Cos+20] Cosmo, L., **Kazi, A.**, Ahmadi, S.A., Navab, N. and Bronstein, M., 2020. Latent Patient Network Learning for Automatic Diagnosis. arXiv preprint arXiv:2003.13620.

2019

[Kaz+19b] **Kazi, A.**, Shekarforoush, S., Krishna, S.A., Burwinkel, H., Vivar, G., Wiestler, B., Kortüm, K., Ahmadi, S.A., Albarqouni, S. and Navab, N., 2019, October. Graph Convolution Based Attention Model for Personalized Disease Prediction. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 122-130). Springer, Cham.

[Kaz+19c] **Kazi, A.**, Shekarforoush, S., Krishna, S.A., Burwinkel, H., Vivar, G., Kortüm, K., Ahmadi, S.A., Albarqouni, S. and Navab, N., 2019, June. InceptionGCN: receptive field aware graph convolutional network for disease prediction. In International Conference on Information Processing in Medical Imaging (pp. 73-85). Springer, Cham.*oral*

[Kaz+19a] **Kazi, A.**, Shekarforoush, S., Kortuem, K., Albarqouni, S. and Navab, N., 2019, April. Self-attention equipped graph convolutions for disease prediction. In 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019) (pp. 1896-1899). IEEE.*oral*

2017

[Kaz+17] **Kazi, A.**, Albarqouni, S., Sanchez, A.J., Kirchhoff, S., Biberthaler, P., Navab, N. and Mateus, D., 2017, September. Automatic classification of proximal femur fractures based on attention models. In International Workshop on Machine Learning in Medical Imaging (pp. 70-78). Springer, Cham..

Co-authored

[Viv+20] Vivar, G., **Kazi, A.**, Burwinkel, H., Zwergal, A., Navab, N. and Ahmadi, S.A., 2020. Simultaneous imputation and disease classification in incomplete medical datasets using Multigraph Geometric Matrix Completion (MGMC). arXiv preprint arXiv:2005.06935.

[Jim+20] Jiménez-Sánchez, A., **Kazi, A.**, Albarqouni, S., Kirchhoff, C., Biberthaler, P., Navab, N., Kirchhoff, S. and Mateus, D., 2020. Precise proximal femur fracture classification for interactive training and surgical planning. International Journal of Computer Assisted Radiology and Surgery.

[Viv+19] Vivar, G., Burwinkel, H., **Kazi, A.**, Zwergal, A., Navab, N. and Ahmadi, S.A., 2019. Multi-modal Graph Fusion for Inductive Disease Classification in Incomplete Datasets. arXiv preprint arXiv:1905.03053.

[Bur+19] Burwinkel, H., **Kazi, A.**, Vivar, G., Albarqouni, S., Zahnd, G., Navab, N. and Ahmadi, S.A., 2019, October. Adaptive image-feature learning for disease classification using inductive graph networks. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 640-648). Springer, Cham.

B

Abstracts of Publications not Discussed in this Thesis

Simultaneous imputation and disease classification in incomplete medical datasets using Multigraph Geometric Matrix Completion (MGMC)

Vivar, Gerome, Kazi, Anees, Burwinkel, Hendrik, Zwergal, Andreas, Navab, Nassir, Ahmadi, Seyed-Ahmad

Large-scale population-based studies in medicine are a key resource towards better diagnosis, monitoring, and treatment of diseases. They also serve as enablers of clinical decision support systems, in particular Computer Aided Diagnosis (CADx) using machine learning (ML). Numerous ML approaches for CADx have been proposed in literature. However, these approaches assume full data availability, which is not always feasible in clinical data. To account for missing data, incomplete data samples are either removed or imputed, which could lead to data bias and may negatively affect classification performance. As a solution, we propose an end-to-end learning of imputation and disease prediction of incomplete medical datasets via Multigraph Geometric Matrix Completion (MGMC). MGMC uses multiple recurrent graph convolutional networks, where each graph represents an independent population model based on a key clinical meta-feature like age, sex, or cognitive function. Graph signal aggregation from local patient neighborhoods, combined with multigraph signal fusion via self-attention, has a regularizing effect on both matrix reconstruction and classification performance. Our proposed approach is able to impute class relevant features as well as perform accurate classification on two publicly available medical datasets. We empirically show the superiority of our proposed approach in terms of classification and imputation performance when compared with state-of-the-art approaches. MGMC enables disease prediction in multimodal and incomplete medical datasets. These findings could serve as baseline for future CADx approaches which utilize incomplete datasets.

arXiv preprint arXiv:2005.06935 (2020)

Multi-modal Graph Fusion for Inductive Disease Classification in Incomplete Datasets

Vivar, Gerome, Burwinkel, Hendrik, Kazi, Anees, Zwergal, Andreas, Navab, Nassir, Ahmadi, Seyed-Ahmad

With the need for adequate analysis of blood flow dynamics, different imaging modalities have been developed to measure varying blood velocities over time. Due to its numerous advantages, Doppler ultrasound sonography remains one of the most widely used techniques in clinical routine, but requires additional preprocessing to recover 3D velocity information. Despite great progress in the last years, recent approaches do not jointly consider spatial and temporal variation in blood flow. In this work, we present a novel gating- and compounding-free method to simultaneously reconstruct a 3D velocity field and a temporal flow profile from arbitrarily sampled Doppler ultrasound measurements obtained from multiple directions. Based on a laminar flow assumption, a patch-wise B-spline formulation of blood velocity is coupled for the first time with a global waveform model acting as temporal regularization. We evaluated our method on three virtual phantom datasets, demonstrating robustness in terms of noise, angle between measurements and data sparsity, and applied it successfully to five real case datasets of carotid artery examination.

arXiv preprint arXiv:1905.03053(2019)

Precise proximal femur fracture classification for interactive training and surgical planning

Jiménez-Sánchez, Amelia*, Kazi, Anees*, Albarqouni, Shadi, Kirchhoff, Chlodwig, Biberthaler, Peter, Navab, Nassir, Kirchhoff, Sonja, Mateus, Diana

Purpose: Demonstrate the feasibility of a fully automatic computer-aided diagnosis (CAD) tool, based on deep learning, that localizes and classifies proximal femur fractures on X-ray images according to the AO classification. The proposed framework aims to improve patient treatment planning and provide support for the training of trauma surgeon residents.

Material and methods: A database of 1347 clinical radiographic studies was collected. Radiologists and trauma surgeons annotated all fractures with bounding boxes and provided a classification according to the AO standard. In all experiments, the dataset was split patient-wise in three with the ratio 70%:10%:20% to build the training, validation and test sets, respectively. ResNet-50 and AlexNet architectures were implemented as deep learning classification and localization models, respectively. Accuracy, precision, recall and [Formula: see text]-score were reported as classification metrics. Retrieval of similar cases was evaluated in terms of precision and recall.

Results: The proposed CAD tool for the classification of radiographs into types "A," "B" and "not-fractured" reaches a [Formula: see text]-score of 87% and AUC of 0.95. When classifying fractures versus not-fractured cases it improves up to 94% and 0.98. Prior localization of the fracture results in an improvement with respect to full-image classification. In total, 100% of the predicted centers of the region of interest are contained in the manually provided bounding boxes. The system retrieves on average 9 relevant images (from the same class) out of 10 cases.

Conclusion: Our CAD scheme localizes, detects and further classifies proximal femur fractures achieving results comparable to expert-level and state-of-the-art performance. Our auxiliary

localization model was highly accurate predicting the region of interest in the radiograph. We further investigated several strategies of verification for its adoption into the daily clinical routine. A sensitivity analysis of the size of the ROI and image retrieval as a clinical use case were presented.

International Journal of Computer Assisted Radiology and Surgery(2020))

Adaptive image-feature learning for disease classification using inductive graph networks. In International Conference on Medical Image Computing and Computer-Assisted InterventionInternational Conference on Medical Image Computing and Computer-Assisted Intervention(2019)

Burwinkel, H., **Kazi, A.**, Vivar, G., Albarqouni, S., Zahnd, G., Navab, N. and Ahmadi, S.A., 2019,

Recently, Geometric Deep Learning (GDL) has been introduced as a novel and versatile framework for computer-aided disease classification. GDL uses patient meta-information such as age and gender to model patient cohort relations in a graph structure. Concepts from graph signal processing are leveraged to learn the optimal mapping of multi-modal features, e.g. from images to disease classes. Related studies so far have considered image features that are extracted in a pre-processing step. We hypothesize that such an approach prevents the network from optimizing feature representations towards achieving the best performance in the graph network. We propose a new network architecture that exploits an inductive end-to-end learning approach for disease classification, where filters from both the CNN and the graph are trained jointly. We validate this architecture against state-of-the-art inductive graph networks and demonstrate significantly improved classification scores on a modified MNIST toy dataset, as well as comparable classification results with higher stability on a chest X-ray image dataset. Additionally, we explain how the structural information of the graph affects both the image filters and the feature learning.

International Conference on Medical Image Computing and Computer-Assisted Intervention(2019)

Bibliography

[Abr+17] Alexandre Abraham, Michael P Milham, Adriana Di Martino, et al. "Deriving reproducible biomarkers from multi-site resting-state data: An Autism-based example". In: *NeuroImage* 147 (2017), pp. 736–745 (cit. on p. 48).

[ABK17] Miltiadis Allamanis, Marc Brockschmidt, and Mahmoud Khademi. "Learning to represent programs with graphs". In: *arXiv preprint arXiv:1711.00740* (2017) (cit. on p. 21).

[And+15] Heather D Anderson, Wilson D Pace, Elias Brandt, et al. "Monitoring suicidal patients in primary care using electronic health records". In: *The Journal of the American Board of Family Medicine* 28.1 (2015), pp. 65–71 (cit. on p. 7).

[ACT+17] Farhad Arbabzadah, Stefan Chmiela, Alexandre Tkatchenko, et al. "Quantum-chemical insights from deep tensor neural networks". In: *Bulletin of the American Physical Society* 62 (2017) (cit. on p. 19).

[AT16] James Atwood and Don Towsley. "Diffusion-convolutional neural networks". In: *Advances in neural information processing systems*. 2016, pp. 1993–2001 (cit. on pp. 15, 19).

[BEM18] Davide Bacciu, Federico Errica, and Alessio Micheli. "Contextual graph markov model: A deep and generative approach to graph processing". In: *arXiv preprint arXiv:1805.10636* (2018) (cit. on p. 19).

[Bas+17] Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. "Graph convolutional encoders for syntax-aware neural machine translation". In: *arXiv preprint arXiv:1704.04675* (2017) (cit. on p. 21).

[Bat+18] Peter W Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv:1806.01261* (2018) (cit. on pp. 57–59).

[BKW17] Rianne van den Berg, Thomas N Kipf, and Max Welling. "Graph convolutional matrix completion". In: *arXiv preprint arXiv:1706.02263* (2017) (cit. on p. 21).

[Bia+19] Filippo Maria Bianchi, Daniele Grattarola, Cesare Alippi, and Lorenzo Livi. "Graph neural networks with convolutional ARMA filters". In: *arXiv:1901.01343* (2019) (cit. on p. 58).

[Bis20] John Mark Bishop. "Artificial Intelligence is stupid and causal reasoning won't fix it". In: *arXiv preprint arXiv:2008.07371* (2020) (cit. on p. 25).

[Bos+16] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. "Learning shape correspondence with anisotropic convolutional neural networks". In: *Advances in neural information processing systems*. 2016, pp. 3189–3197 (cit. on p. 19).

[Bro+17] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. "Geometric deep learning: going beyond euclidean data". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42 (cit. on pp. 13, 15, 57).

[BL17] Joan Bruna and X Li. "Community detection with graph neural networks". In: *Stat* 1050 (2017), p. 27 (cit. on p. 58).

[Bru+13] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. “Spectral networks and locally connected networks on graphs”. In: *arXiv preprint arXiv:1312.6203* (2013) (cit. on pp. 15, 20, 58).

[Bur+19] Hendrik Burwinkel, Anees Kazi, Gerome Vivar, et al. “Adaptive image-feature learning for disease classification using inductive graph networks”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 640–648 (cit. on p. 78).

[Byr+14] Roy J Byrd, Steven R Steinhubl, Jimeng Sun, Shahram Ebadollahi, and Walter F Stewart. “Automatic identification of heart failure diagnostic criteria, using text analysis of clinical notes from electronic health records”. In: *International journal of medical informatics* 83.12 (2014), pp. 983–992 (cit. on p. 7).

[Cai+19] Qiong Cai, Hao Wang, Zhenmin Li, and Xiao Liu. “A Survey on Multimodal Data-Driven Smart Healthcare Systems: Approaches and Applications”. In: *IEEE Access* 7 (2019), pp. 133583–133599 (cit. on p. 8).

[Che+18] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. “Iterative visual reasoning beyond convolutions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7239–7248 (cit. on p. 21).

[Cho+17] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. “GRAM: graph-based attention model for healthcare representation learning”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017, pp. 787–795 (cit. on p. 22).

[Cho+18a] Edward Choi, Cao Xiao, Walter Stewart, and Jimeng Sun. “Mime: Multilevel medical embedding of electronic health records for predictive healthcare”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 4547–4557 (cit. on p. 22).

[Cho+18b] N. Choma et al. “Graph neural networks for icecube signal classification”. In: *Proc. ICMLA*. 2018 (cit. on p. 57).

[Cos+20] Luca Cosmo, Anees Kazi, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein. “Latent Patient Network Learning for Automatic Diagnosis”. In: *arXiv preprint arXiv:2003.13620* (2020) (cit. on p. 77).

[DK18] Nicola De Cao and Thomas Kipf. “MolGAN: An implicit generative model for small molecular graphs”. In: *arXiv preprint arXiv:1805.11973* (2018) (cit. on p. 21).

[De +18] Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, et al. “Clinically applicable deep learning for diagnosis and referral in retinal disease”. In: *Nature medicine* 24.9 (2018), pp. 1342–1350 (cit. on p. 7).

[Deb+91] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. “Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity”. In: *Journal of medicinal chemistry* 34.2 (1991), pp. 786–797 (cit. on p. 72).

[DBV16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. “Convolutional neural networks on graphs with fast localized spectral filtering”. In: *Advances in neural information processing systems*. 2016, pp. 3844–3852 (cit. on pp. 9, 15, 20, 27, 40, 45, 46, 58, 59).

[Den+09] Jia Deng, Wei Dong, Richard Socher, et al. “Imagenet: A large-scale hierarchical image database”. In: *CVPR*. Ieee. 2009 (cit. on p. 65).

[DMT18] Tyler Derr, Yao Ma, and Jiliang Tang. “Signed graph convolutional networks”. In: *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2018, pp. 929–934 (cit. on p. 72).

[DD03] Paul D Dobson and Andrew J Doig. “Distinguishing enzyme structures from non-enzymes without alignments”. In: *Journal of molecular biology* 330.4 (2003), pp. 771–783 (cit. on p. 72).

[Duv+15] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, et al. “Convolutional networks on graphs for learning molecular fingerprints”. In: *NeurIPS*. 2015, pp. 2224–2232 (cit. on pp. 19, 57).

[Dvo+20] Nicha C Dvornek, Xiaoxiao Li, Juntang Zhuang, Pamela Ventola, and James S Duncan. “Demographic-Guided Attention in Recurrent Neural Networks for Modeling Neuropathophysiological Heterogeneity”. In: *International Workshop on Machine Learning in Medical Imaging*. Springer. 2020, pp. 363–372 (cit. on p. 22).

[EKK11] Moataz El Ayadi, Mohamed S Kamel, and Fakhri Karray. “Survey on speech emotion recognition: Features, classification schemes, and databases”. In: *Pattern Recognition* 44.3 (2011), pp. 572–587 (cit. on p. 13).

[Est+17] Andre Esteva, Brett Kuprel, Roberto A Novoa, et al. “Dermatologist-level classification of skin cancer with deep neural networks”. In: *nature* 542.7639 (2017), pp. 115–118 (cit. on p. 7).

[Est+19] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, et al. “A guide to deep learning in healthcare”. In: *Nature medicine* 25.1 (2019), pp. 24–29 (cit. on p. 7).

[Fey+18] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. “Splinecnn: Fast geometric deep learning with continuous b-spline kernels”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 869–877 (cit. on p. 19).

[Fis+00] Marcelo Fiszman, Wendy W Chapman, Dominik Aronsky, R Scott Evans, and Peter J Haug. “Automatic detection of acute bacterial pneumonia from chest X-ray reports”. In: *Journal of the American Medical Informatics Association* 7.6 (2000), pp. 593–604 (cit. on p. 7).

[Fou+17] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. “Protein interface prediction using graph convolutional networks”. In: *Advances in neural information processing systems*. 2017, pp. 6530–6539 (cit. on pp. 21, 34).

[Fra+19] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. “Learning discrete structures for graph neural networks”. In: (2019) (cit. on pp. 58, 62, 63).

[FM08] F Jeff Friedlin and Clement J McDonald. “A software tool for removing patient identifying information from clinical documents”. In: *Journal of the American Medical Informatics Association* 15.5 (2008), pp. 601–610 (cit. on p. 7).

[FJD13] Kin Wah Fung, Chiang S Jao, and Dina Demner-Fushman. “Extracting drug indication information from structured product labels using natural language processing”. In: *Journal of the American Medical Informatics Association* 20.3 (2013), pp. 482–488 (cit. on p. 7).

[Gai+19] Pablo Gainza, Freyr Sverrisson, Frederico Monti, et al. “Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning”. In: *Nature Methods* 17 (2019), pp. 184–192 (cit. on p. 57).

[GM10] Claudio Gallicchio and Alessio Micheli. “Graph echo state networks”. In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2010, pp. 1–8 (cit. on p. 15).

[GHK13] Emden R Gansner, Yifan Hu, and Shankar Krishnan. “Coast: A convex optimization approach to stress-based embedding”. In: *International Symposium on Graph Drawing*. Springer. 2013, pp. 268–279 (cit. on p. 52).

[GWJ18] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. “Large-scale learnable graph convolutional networks”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 1416–1424 (cit. on p. 19).

[GB17] Victor Garcia and Joan Bruna. “Few-shot learning with graph neural networks”. In: *arXiv preprint arXiv:1711.04043* (2017) (cit. on p. 21).

[Gil+17] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. “Neural message passing for quantum chemistry”. In: *arXiv preprint arXiv:1704.01212* (2017) (cit. on pp. 15, 19, 21, 57–59).

[GMS05] Marco Gori, Gabriele Monfardini, and Franco Scarselli. “A new model for learning in graph domains”. In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* Vol. 2. IEEE. 2005, pp. 729–734 (cit. on p. 15).

[Gul+18] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, et al. “Hyperbolic attention networks”. In: *arXiv preprint arXiv:1805.09786* (2018) (cit. on p. 72).

[Gul+16] Varun Gulshan, Lily Peng, Marc Coram, et al. “Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs”. In: *Jama* 316.22 (2016), pp. 2402–2410 (cit. on p. 7).

[Guo+18] Michelle Guo, Edward Chou, De-An Huang, et al. “Neural graph matching networks for fewshot 3d action recognition”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 653–669 (cit. on p. 21).

[Hae+18] Holger A Haenssle, Christine Fink, R Schneiderbauer, et al. “Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists”. In: *Annals of Oncology* 29.8 (2018), pp. 1836–1842 (cit. on p. 7).

[HYL17a] Will Hamilton, Zhitao Ying, and Jure Leskovec. “Inductive representation learning on large graphs”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 1024–1034 (cit. on pp. 13, 18, 21, 46).

[HYL17b] William L Hamilton, Rex Ying, and Jure Leskovec. “Representation learning on graphs: Methods and applications”. In: *arXiv:1709.05584* (2017) (cit. on p. 57).

[Haz+05] Brian Hazlehurst, H Robert Frost, Dean F Sittig, and Victor J Stevens. “MediClass: A system for detecting and classifying encounter-based clinical events in any electronic medical record”. In: *Journal of the American Medical Informatics Association* 12.5 (2005), pp. 517–529 (cit. on p. 7).

[He+19] Jianxing He, Sally L Baxter, Jie Xu, et al. “The practical implementation of artificial intelligence technologies in medicine”. In: *Nature medicine* 25.1 (2019), pp. 30–36 (cit. on p. 7).

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *CVPR*. 2016, pp. 770–778 (cit. on pp. 65, 71).

[HBL15] Mikael Henaff, Joan Bruna, and Yann LeCun. “Deep convolutional networks on graph-structured data”. In: *arXiv preprint arXiv:1506.05163* (2015) (cit. on p. 20).

[HK70] A. E. Hoerl and R. W. Kennard. “Ridge Regression: Biased Estimation for Nonorthogonal Problems”. In: *Technometrics* 12 (1970), pp. 55–67 (cit. on p. 30).

[Hri+03] George Hripcsak, Suzanne Bakken, Peter D Stetson, and Vimla L Patel. “Mining complex clinical data for patient safety research: a framework for event discovery”. In: *Journal of biomedical informatics* 36.1-2 (2003), pp. 120–130 (cit. on p. 7).

[Hu+18] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. “Relation networks for object detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3588–3597 (cit. on p. 13).

[Hua+17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. “Densely connected convolutional networks.” In: *CVPR*. Vol. 1. 2. 2017, p. 3 (cit. on p. 46).

[Hua+20] Qiang Huang, Makoto Yamada, Yuan Tian, et al. “GraphLIME: Local interpretable model explanations for graph neural networks”. In: *arXiv preprint arXiv:2001.06216* (2020) (cit. on p. 72).

[Hua+18] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. “Adaptive sampling towards fast graph representation learning”. In: *NeurIPS*. 2018, pp. 4558–4567 (cit. on p. 58).

[HC20] Yongxiang Huang and Albert CS Chung. “Edge-Variational Graph Convolutional Networks for Uncertainty-Aware Disease Prediction”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2020, pp. 562–572 (cit. on p. 46).

[Hun+08] James Hunter, Albert Gatt, François Portet, Ehud Reiter, and Somayajulu Sripada. “Using natural language generation technology to improve information flows in intensive care units”. In: (2008) (cit. on p. 7).

[Iml+13] Timothy D Imler, Justin Morea, Charles Kahi, and Thomas F Imperiale. “Natural language processing accurately categorizes findings from colonoscopy and pathology reports”. In: *Clinical Gastroenterology and Hepatology* 11.6 (2013), pp. 689–694 (cit. on p. 7).

[Jai+16] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. “Structural-rnn: Deep learning on spatio-temporal graphs”. In: *Proceedings of the ieee conference on computer vision and pattern recognition*. 2016, pp. 5308–5317 (cit. on pp. 13, 21).

[JKZ16] Amir Jamaludin, Timor Kadir, and Andrew Zisserman. “SpineNet: automatically pinpointing classification evidence in spinal MRIs”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2016, pp. 166–175 (cit. on p. 7).

[JML19] Soobeom Jang, Seong-Eun Moon, and Jong-Seok Lee. “Brain Signal Classification via Learning Connectivity Structure”. In: *CoRR* abs/1905.11678 (2019). arXiv: `1905.11678` (cit. on p. 66).

[Jau+20] Guillaume Jaume, Pushpak Pati, Antonio Foncubierta-Rodriguez, et al. “Towards explainable graph representations in digital pathology”. In: *arXiv preprint arXiv:2007.00311* (2020) (cit. on p. 72).

[JJB12] Peter B Jensen, Lars J Jensen, and Søren Brunak. “Mining electronic health records: towards better research applications and clinical care”. In: *Nature Reviews Genetics* 13.6 (2012), pp. 395–405 (cit. on p. 7).

[Jim+20] Amelia Jiménez-Sánchez, Anees Kazi, Shadi Albarqouni, et al. “Precise proximal femur fracture classification for interactive training and surgical planning”. In: *International Journal of Computer Assisted Radiology and Surgery* (2020) (cit. on p. 78).

[JGF18] Justin Johnson, Agrim Gupta, and Li Fei-Fei. “Image generation from scene graphs”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1219–1228 (cit. on p. 21).

[Kam+19] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, et al. “Rethinking knowledge graph propagation for zero-shot learning”. In: *CVPR*. 2019, pp. 11487–11496 (cit. on pp. 64, 65).

[Kar+14] Andrej Karpathy, George Toderici, Sanketh Shetty, et al. “Large-scale video classification with convolutional neural networks”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2014, pp. 1725–1732 (cit. on p. 8).

[Kaw+17] Jeremy Kawahara, Colin J Brown, Steven P Miller, et al. “BrainNetCNN: Convolutional neural networks for brain networks; towards predicting neurodevelopment”. In: *NeuroImage* 146 (2017), pp. 1038–1049 (cit. on p. 22).

[KTO18] Tatsuro Kawamoto, Masashi Tsubaki, and Tomoyuki Obuchi. “Mean-field theory of graph neural networks in graph partitioning”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 4361–4371 (cit. on p. 21).

[Kaz+18] Anees Kazi, Shadi Albarqouni, Karsten Kortuem, and Nassir Navab. “Multi Layered-Parallel Graph Convolutional Network (ML-PGCN) for Disease Prediction”. In: *arXiv preprint arXiv:1804.10776* (2018) (cit. on p. 53).

[Kaz+17] Anees Kazi, Shadi Albarqouni, Amelia Jimenez Sanchez, et al. “Automatic classification of proximal femur fractures based on attention models”. In: *International Workshop on Machine Learning in Medical Imaging*. Springer. 2017, pp. 70–78 (cit. on p. 77).

[Kaz+20] Anees Kazi, Luca Cosmo, Nassir Navab, and Michael Bronstein. “Differentiable Graph Module (DGM) Graph Convolutional Networks”. In: *arXiv preprint arXiv:2002.04999* (2020) (cit. on p. 77).

[Kaz+19a] Anees Kazi, Shayan Shekarforoush, Karsten Kortuem, Shadi Albarqouni, Nassir Navab, et al. “Self-attention equipped graph convolutions for disease prediction”. In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE. 2019, pp. 1896–1899 (cit. on pp. 62, 63, 77).

[Kaz+19b] Anees Kazi, Shayan Shekarforoush, S Arvind Krishna, et al. “Graph Convolution Based Attention Model for Personalized Disease Prediction”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 122–130 (cit. on pp. 62, 77).

[Kaz+19c] Anees Kazi, Shayan Shekarforoush, S Arvind Krishna, et al. “InceptionGCN: receptive field aware graph convolutional network for disease prediction”. In: *International Conference on Information Processing in Medical Imaging*. Springer. 2019, pp. 73–85 (cit. on pp. 34, 57, 62, 63, 77).

[Kea+16] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. “Molecular graph convolutions: moving beyond fingerprints”. In: *Journal of computer-aided molecular design* 30.8 (2016), pp. 595–608 (cit. on pp. 19, 21).

[Ker+18] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, et al. “Identifying medical diagnoses and treatable diseases by image-based deep learning”. In: *Cell* 172.5 (2018), pp. 1122–1131 (cit. on p. 7).

[Kha+13] Bahador Khaleghi, Alaa Khamis, Fakhreddine O Karray, and Saiedeh N Razavi. “Multisensor data fusion: A review of the state-of-the-art”. In: *Information fusion* 14.1 (2013), pp. 28–44 (cit. on p. 8).

[KW16a] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016) (cit. on pp. 13, 15, 18, 20, 21, 26, 27, 38, 46, 58, 59).

[KW16b] Thomas N Kipf and Max Welling. “Variational graph auto-encoders”. In: *arXiv preprint arXiv:1611.07308* (2016) (cit. on p. 15).

[Kip+18] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. “Neural relational inference for interacting systems”. In: *arXiv:1802.04687* (2018) (cit. on pp. 19, 58).

[Kon18] Risi Kondor. “N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials”. In: *arXiv:1803.01588* (2018) (cit. on p. 58).

[Koo+17] Thijs Kooi, Geert Litjens, Bram Van Ginneken, et al. “Large scale deep learning for computer aided detection of mammographic lesions”. In: *Medical image analysis* 35 (2017), pp. 303–312 (cit. on p. 7).

[KHW19] Wouter Kool, Herke van Hoof, and Max Welling. “Stochastic Beams and Where to Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement”. In: *CoRR* abs/1903.06059 (2019). arXiv: `1903.06059` (cit. on p. 60).

[Kri+10] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. “Hyperbolic geometry of complex networks”. In: *Physical Review E* 82.3 (2010), p. 036106 (cit. on p. 60).

[Kte+18] Sofia Ira Ktena, Sarah Parisot, Enzo Ferrante, et al. “Metric learning with spectral graph convolutions on brain connectivity networks”. In: *NeuroImage* 169 (2018), pp. 431–442 (cit. on pp. 22, 45).

[Kun14] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014 (cit. on p. 8).

[LS18] Loic Landrieu and Martin Simonovsky. “Large-scale point cloud semantic segmentation with superpoint graphs”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4558–4567 (cit. on pp. 13, 21).

[Lev+18] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. “Cayleynets: Graph convolutional neural networks with complex rational spectral filters”. In: *IEEE Transactions on Signal Processing* 67.1 (2018), pp. 97–109 (cit. on pp. 15, 58).

[LHW18] Qimai Li, Zhichao Han, and Xiao-Ming Wu. “Deeper insights into graph convolutional networks for semi-supervised learning”. In: *arXiv preprint arXiv:1801.07606* (2018) (cit. on p. 71).

[Li+18a] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. “Adaptive graph convolutional neural networks”. In: *AAAI*. 2018 (cit. on p. 58).

[LD20] Xiaoxiao Li and James Duncan. “BrainGNN: Interpretable Brain Graph Neural Network for fMRI Analysis”. In: *bioRxiv* (2020) (cit. on p. 22).

[Li+17] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting”. In: *arXiv preprint arXiv:1707.01926* (2017) (cit. on pp. 19, 21).

[Li+18b] Yikang Li, Wanli Ouyang, Bolei Zhou, et al. “Factorizable net: an efficient subgraph-based framework for scene graph generation”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 335–351 (cit. on p. 21).

[Li+15] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. “Gated graph sequence neural networks”. In: *arXiv preprint arXiv:1511.05493* (2015) (cit. on p. 21).

[Li+18c] Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. “Learning deep generative models of graphs”. In: *arXiv preprint arXiv:1803.03324* (2018) (cit. on pp. 21, 57).

[LCK18] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. “Combinatorial optimization with graph convolutional networks and guided tree search”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 539–548 (cit. on p. 22).

[Liu+20] Jin Liu, Yu Sheng, Wei Lan, et al. “Improved ASD classification using dynamic functional connectivity and multi-task feature selection”. In: *Pattern Recognition Letters* 138 (2020), pp. 82–87 (cit. on p. 22).

[Liu+19a] Lu Liu, Tianyi Zhou, Guodong Long, et al. “Prototype propagation networks (PPN) for weakly-supervised few-shot learning on category graph”. In: *arXiv preprint arXiv:1905.04042* (2019) (cit. on p. 21).

[LNK19] Qi Liu, Maximilian Nickel, and Douwe Kiela. “Hyperbolic graph neural networks”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 8230–8241 (cit. on p. 72).

[LWC12] Wei Liu, Jun Wang, and Shih-Fu Chang. “Robust and scalable graph-based semisupervised learning”. In: *Proc. IEEE* 100.9 (2012), pp. 2624–2638 (cit. on p. 58).

[Liu+18] Ziqi Liu, Chaochao Chen, Longfei Li, et al. “GeniePath: Graph Neural Networks with Adaptive Receptive Paths”. In: *arXiv preprint arXiv:1802.00910* (2018) (cit. on p. 46).

[Liu+19b] Ziqi Liu, Chaochao Chen, Longfei Li, et al. “Geniepath: Graph neural networks with adaptive receptive paths”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 4424–4431 (cit. on p. 19).

[LXL19] Zhoumin Lu, Haiping Xu, and Genggeng Liu. “A survey of object co-segmentation”. In: *IEEE Access* 7 (2019), pp. 62875–62893 (cit. on p. 13).

[Ma+18a] Fenglong Ma, Quanzeng You, Houping Xiao, et al. “Kame: Knowledge-based attention model for diagnosis prediction in healthcare”. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2018, pp. 743–752 (cit. on p. 8).

[Ma+18b] Tengfei Ma, Cao Xiao, Jiayu Zhou, and Fei Wang. “Drug Similarity Integration Through Attentive Multi-view Graph Auto-Encoders”. In: *arXiv preprint arXiv:1804.10850* (2018) (cit. on pp. 34, 38).

[MBT18] Diego Marcheggiani, Joost Bastings, and Ivan Titov. “Exploiting semantics in neural machine translation with graph convolutional networks”. In: *arXiv preprint arXiv:1804.08313* (2018) (cit. on p. 21).

[MT17] Diego Marcheggiani and Ivan Titov. “Encoding sentences with graph convolutional networks for semantic role labeling”. In: *arXiv preprint arXiv:1703.04826* (2017) (cit. on p. 21).

[Mar+18] Razvan V Marinescu, Neil P Oxtoby, Alexandra L Young, et al. “TADPOLE Challenge: Prediction of Longitudinal Evolution in Alzheimer’s Disease”. In: *arXiv preprint arXiv:1805.03909* (2018) (cit. on pp. 28, 37, 48, 62).

[Mas+15] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. “Geodesic convolutional neural networks on riemannian manifolds”. In: *Proceedings of the IEEE international conference on computer vision workshops*. 2015, pp. 37–45 (cit. on p. 19).

[Mel+19] Cooper Mellema, Alex Treacher, Kevin Nguyen, and Albert Montillo. “Multiple Deep Learning Architectures Achieve Superior Performance Diagnosing Autism Spectrum Disorder Using Features Previously Extracted From Structural And Functional Mri”. In: *2019 IEEE ISBI)*. IEEE. 2019, pp. 1891–1895 (cit. on p. 57).

[Men+05] Eneida A Mendonça, Janet Haas, Lyudmila Shagina, Elaine Larson, and Carol Friedman. “Extracting information on pneumonia in infants using natural language processing of radiology reports”. In: *Journal of biomedical informatics* 38.4 (2005), pp. 314–321 (cit. on p. 7).

[Mic09] Alessio Micheli. “Neural network for graphs: A contextual constructive approach”. In: *IEEE Trans. Neural Netw* 20.3 (2009), pp. 498–511 (cit. on pp. 18, 19).

[Mil+16] Karla L Miller, Fidel Alfaro-Almagro, Neal K Bangerter, et al. “Multimodal population brain imaging in the UK Biobank prospective epidemiological study”. In: *Nature neuroscience* 19.11 (2016), p. 1523 (cit. on p. 62).

[Mon+18] Federico Monti et al. “Dual-primal graph convolutional networks”. In: *arXiv:1806.00770* (2018) (cit. on p. 58).

[Mon+17] Federico Monti, Davide Boscaini, Jonathan Masci, et al. “Geometric deep learning on graphs and manifolds using mixture model cnns”. In: *Proc. CVPR*. 2017 (cit. on pp. 19, 21, 57, 58).

[NLS18] Medhini Narasimhan, Svetlana Lazebnik, and Alexander Schwing. “Out of the box: Reasoning with graph convolution nets for factual visual question answering”. In: *Advances in neural information processing systems*. 2018, pp. 2654–2665 (cit. on p. 21).

[Nev+15] Natalia Neverova, Christian Wolf, Graham Taylor, and Florian Nebout. “Moddrop: adaptive multi-modal gesture recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.8 (2015), pp. 1692–1706 (cit. on p. 8).

[NG18] Thien Huu Nguyen and Ralph Grishman. “Graph Convolutional Networks With Argument-Aware Pooling for Event Detection.” In: *AAAI*. Vol. 18. 2018, pp. 5900–5907 (cit. on p. 22).

[NAK16] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. “Learning convolutional neural networks for graphs”. In: *International conference on machine learning*. 2016, pp. 2014–2023 (cit. on pp. 15, 19).

[Par+18] Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, et al. “Disease Prediction using Graph Convolutional Networks: Application to Autism Spectrum Disorder and Alzheimer’s Disease”. In: *Medical image analysis* (2018) (cit. on p. 57).

[Par+17] Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, et al. “Spectral graph convolutions for population-based disease prediction”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2017, pp. 177–185 (cit. on pp. 3, 9, 17, 26–28, 30, 34, 37, 38, 46, 48–52, 57, 62, 63).

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global vectors for word representation”. In: *EMNLP*. 2014, pp. 1532–1543 (cit. on p. 65).

[Pha+16] Trang Pham, Truyen Tran, Dinh Phung, and Svetha Venkatesh. “Column networks for collective classification”. In: *arXiv preprint arXiv:1609.04508* (2016) (cit. on p. 72).

[Pop+18] Ryan Poplin, Avinash V Varadarajan, Katy Blumer, et al. “Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning”. In: *Nature Biomedical Engineering* 2.3 (2018), p. 158 (cit. on p. 7).

[PCG15] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. “Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis”. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2015, pp. 2539–2544 (cit. on p. 8).

[Por+09] François Portet, Ehud Reiter, Albert Gatt, et al. “Automatic generation of textual summaries from neonatal intensive care data”. In: *Artificial Intelligence* 173.7-8 (2009), pp. 789–816 (cit. on p. 7).

[Qi+18] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. “Learning human-object interactions by graph parsing neural networks”. In: *ECCV*. 2018, pp. 401–417 (cit. on pp. 21, 57).

[Qi+17] Xiaojuan Qi, Renjie Liao, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. “3d graph neural networks for rgbd semantic segmentation”. In: *ICCV*. 2017, pp. 5199–5208 (cit. on pp. 21, 57).

[Qia+19] Zhi Qiao, Xian Wu, Shen Ge, and Wei Fan. “Mnn: multimodal attentional neural networks for diagnosis prediction”. In: *Extraction* 1 (2019), A1 (cit. on p. 8).

[Qiu+18] Jiezhong Qiu, Jian Tang, Hao Ma, et al. “Deepinf: Social influence prediction with deep learning”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 2110–2119 (cit. on p. 21).

[Raj+12] Ali S Raja, Ivan K Ip, Luciano M Prevedello, et al. “Effect of computerized clinical decision support on the use and yield of CT pulmonary angiography in the emergency department”. In: *Radiology* 262.2 (2012), pp. 468–474 (cit. on p. 7).

[Ren+20] Yongjian Ren, Yuliang Shi, Kun Zhang, Zhiyong Chen, and Zhongmin Yan. “Medical Treatment Migration Prediction Based on GCN via Medical Insurance Data”. In: *IEEE Journal of Biomedical and Health Informatics* 24.9 (2020), pp. 2516–2522 (cit. on p. 22).

[Rie+08] Christian Rieder, Felix Ritter, Matthias Raspe, and Heinz-Otto Peitgen. “Interactive visualization of multimodal volume data for neurosurgical tumor treatment”. In: *Computer Graphics Forum*. Vol. 27. 3. Wiley Online Library. 2008, pp. 1055–1062 (cit. on p. 8).

[RGH08] Angus Roberts, Robert Gaizauskas, and Mark Hepple. “Extracting clinical relationships from patient narratives”. In: *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*. 2008, pp. 10–18 (cit. on p. 7).

[Ros+20] Emanuele Rossi, Fabrizio Frasca, Ben Chamberlain, et al. “SIGN: Scalable Inception Graph Neural Networks”. In: *arXiv preprint arXiv:2004.11198* (2020) (cit. on p. 46).

[SM13] Aliaksei Sandryhaila and José MF Moura. “Discrete signal processing on graphs”. In: *IEEE transactions on signal processing* 61.7 (2013), pp. 1644–1656 (cit. on p. 20).

[Sca+08] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. “The graph neural network model”. In: *IEEE Transactions on Neural Networks* 20.1 (2008), pp. 61–80 (cit. on pp. 15, 57).

[Shu+13] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains”. In: *IEEE signal processing magazine* 30.3 (2013), pp. 83–98 (cit. on p. 20).

[Sta+20] Kamilė Stankevičiūtė, Tiago Azevedo, Alexander Campbell, Richard AI Bethlehem, and Pietro Liò. “Population Graph GNNs for Brain Age Prediction”. In: *bioRxiv* (2020) (cit. on p. 22).

[Sto+14] Christof Stocker, Leopold-Michael Marzi, Christian Matula, et al. “Enhancing patient safety through human-computer information retrieval on the example of german-speaking surgical reports”. In: *2014 25th International Workshop on Database and Expert Systems Applications*. IEEE. 2014, pp. 216–220 (cit. on p. 7).

[SM20] Rodrigo Suarez-Ibarrola and Arkadiusz Miernik. *Prospects and Challenges of Artificial Intelligence and Computer Science for the Future of Urology*. 2020 (cit. on p. 7).

[97] “Supervised neural networks for the classification of structures”. In: *IEEE Transactions on Neural Networks* 8.3 (1997), pp. 714–735 (cit. on p. 15).

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. “Sequence to sequence learning with neural networks”. In: *Advances in neural information processing systems*. 2014, pp. 3104–3112 (cit. on p. 7).

[Sze+15] Christian Szegedy, Wei Liu, Yangqing Jia, et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on CVPR*. 2015, pp. 1–9 (cit. on p. 46).

[Szl+05] Arthur D Szlam, Mauro Maggioni, Ronald R Coifman, and James C Bremer Jr. “Diffusion-driven multiscale analysis on manifolds and graphs: top-down and bottom-up constructions”. In: *Wavelets XI*. Vol. 5914. International Society for Optics and Photonics. 2005, p. 59141D (cit. on p. 21).

[TM20] Ruixue Tang and Chao Ma. “Interpretable Neural Computation for Real-World Compositional Visual Question Answering”. In: *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer. 2020, pp. 89–101 (cit. on p. 21).

[Te+18] Gusi Te, Wei Hu, Amin Zheng, and Zongming Guo. “Rgcnn: Regularized graph cnn for point cloud segmentation”. In: *Proceedings of the 26th ACM international conference on Multimedia*. 2018, pp. 746–754 (cit. on pp. 13, 21).

[TDG16] Nicola Toschi, Andrea Duggento, and Maria Guerrisi. “Predictive disease modeling for personalized and preventive medicine”. In: (2016) (cit. on p. 33).

[TNS18] Dinh V Tran, Nicolò Navarin, and Alessandro Sperduti. “On filter size in graph convolutional networks”. In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2018, pp. 1534–1541 (cit. on p. 19).

[Vel+17] Petar Veličković, Guillem Cucurull, Arantxa Casanova, et al. “Graph attention networks”. In: *arXiv:1710.10903* (2017) (cit. on pp. 13, 18, 19, 21, 34, 37, 58).

[Viv+19] Gerome Vivar, Hendrik Burwinkel, Anees Kazi, et al. “Multi-modal Graph Fusion for Inductive Disease Classification in Incomplete Datasets”. In: *arXiv preprint arXiv:1905.03053* (2019) (cit. on p. 78).

[Viv+20] Gerome Vivar, Anees Kazi, Hendrik Burwinkel, et al. “Simultaneous imputation and disease classification in incomplete medical datasets using Multigraph Geometric Matrix Completion (MGMC)”. In: *arXiv preprint arXiv:2005.06935* (2020) (cit. on pp. 22, 77).

[Viv+18] Gerome Vivar, Andreas Zwergal, Nassir Navab, and Seyed-Ahmad Ahmadi. “Multi-modal Disease Classification in Incomplete Datasets Using Geometric Matrix Completion”. In: *Graphs in Biomedical Image Analysis and Integrating Medical Imaging and Non-Imaging Modalities* 1 (2018), pp. 24–31 (cit. on p. 45).

[Wag+12] Kavishwar B Wagholikar, Kathy L MacLaughlin, Michael R Henry, et al. “Clinical decision support with automated text processing for cervical cancer screening”. In: *Journal of the American Medical Informatics Association* 19.5 (2012), pp. 833–839 (cit. on p. 7).

[Wan+17] Chun Wang, Shirui Pan, Guodong Long, Xingquan Zhu, and Jing Jiang. “Mgae: Marginalized graph autoencoder for graph clustering”. In: *ACM*. ACM. 2017, pp. 889–898 (cit. on p. 21).

[Wan+19a] Xiao Wang, Houye Ji, Chuan Shi, et al. “Heterogeneous graph attention network”. In: *The World Wide Web Conference*. 2019, pp. 2022–2032 (cit. on p. 15).

[WYG18] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. “Zero-shot recognition via semantic embeddings and knowledge graphs”. In: *CVPR*. 2018, pp. 6857–6866 (cit. on p. 66).

[Wan+20] Xiaoyang Wang, Yao Ma, Yiqi Wang, et al. “Traffic Flow Prediction via Spatial Temporal Graph Neural Network”. In: *Proceedings of The Web Conference 2020*. 2020, pp. 1082–1092 (cit. on p. 15).

[Wan+19b] Yue Wang, Yongbin Sun, Ziwei Liu, et al. “Dynamic graph cnn for learning on point clouds”. In: *ACM TOG* 38.5 (2019), p. 146 (cit. on pp. 13, 21, 57–59, 62–64).

[Wat+11] Alice J Watson, Julia O’Rourke, Kamal Jethwani, et al. “Linking electronic health record-extracted psychosocial data in real-time to risk of readmission for heart failure”. In: *Psychosomatics* 52.4 (2011), pp. 319–327 (cit. on p. 7).

[Wu+20] Zonghan Wu, Shirui Pan, Fengwen Chen, et al. “A comprehensive survey on graph neural networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2020) (cit. on pp. 13, 15, 20).

[Xia+18] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. “Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly”. In: *IEEE PAMI* (2018) (cit. on p. 64).

[Xu+17] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. “Scene graph generation by iterative message passing”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5410–5419 (cit. on p. 21).

[Xu+18a] Keyulu Xu, Chengtao Li, Yonglong Tian, et al. “Representation Learning on Graphs with Jumping Knowledge Networks”. In: *arXiv preprint arXiv:1806.03536* (2018) (cit. on p. 46).

[Xu+18b] Yanbo Xu, Siddharth Biswal, Shriprasad R Deshpande, Kevin O Maher, and Jimeng Sun. “Raim: Recurrent attentive and intensive model of multimodal patient monitoring data”. In: *Proceedings of the 24th ACM SIGKDD international conference on Knowledge Discovery & Data Mining*. 2018, pp. 2565–2573 (cit. on p. 8).

[YXL18] Sijie Yan, Yuanjun Xiong, and Dahua Lin. “Spatial temporal graph convolutional networks for skeleton-based action recognition”. In: *arXiv preprint arXiv:1801.07455* (2018) (cit. on pp. 13, 19, 21).

[Yao+19] Rui Yao, Guosheng Lin, Shixiong Xia, Jiaqi Zhao, and Yong Zhou. "Video object segmentation and tracking: A survey". In: *arXiv preprint arXiv:1904.09172* (2019) (cit. on p. 13).

[Yi+17] Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas. "Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2282–2290 (cit. on p. 21).

[Yin+18] Rex Ying, Ruining He, Kaifeng Chen, et al. "Graph convolutional neural networks for web-scale recommender systems". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 974–983 (cit. on p. 21).

[Yin+19] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. "Gnnexplainer: Generating explanations for graph neural networks". In: *Advances in neural information processing systems*. 2019, pp. 9244–9255 (cit. on p. 72).

[You+18] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. "Graph convolutional policy network for goal-directed molecular graph generation". In: *Advances in neural information processing systems*. 2018, pp. 6410–6421 (cit. on p. 21).

[YYZ17] Bing Yu, Haoteng Yin, and Zhanxing Zhu. "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting". In: *arXiv preprint arXiv:1709.04875* (2017) (cit. on p. 21).

[Yu+19] Shuangzhi Yu, Guanghui Yue, Ahmed Elazab, et al. "Multi-scale Graph Convolutional Network for Mild Cognitive Impairment Detection". In: *International Workshop on Graph Learning in Medical Imaging*. Springer. 2019, pp. 79–87 (cit. on p. 46).

[Yua+20] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. "XGNN: Towards Model-Level Explanations of Graph Neural Networks". In: *arXiv preprint arXiv:2006.02587* (2020) (cit. on p. 72).

[Zha+18a] Kun Zhan, Xiaojun Chang, Junpeng Guan, et al. "Adaptive structure discovery for multimedia analysis using multiple features". In: *IEEE transactions on cybernetics* 49.5 (2018), pp. 1826–1834 (cit. on p. 58).

[Zha+18b] Jiani Zhang, Xingjian Shi, Junyuan Xie, et al. "Gaan: Gated attention networks for learning on large and spatiotemporal graphs". In: *arXiv preprint arXiv:1803.07294* (2018) (cit. on pp. 19, 21).

[ZC18] Muhan Zhang and Yixin Chen. "Link prediction based on graph neural networks". In: *Proc. NeurIPS*. 2018 (cit. on pp. 21, 57).

[Zha+19] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. "Object detection with deep learning: A review". In: *IEEE transactions on neural networks and learning systems* 30.11 (2019), pp. 3212–3232 (cit. on p. 13).

[Zho+18] Jie Zhou, Ganqu Cui, Zhengyan Zhang, et al. "Graph neural networks: A review of methods and applications". In: *arXiv preprint arXiv:1812.08434* (2018) (cit. on p. 14).

[ZAL18] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. "Modeling polypharmacy side effects with graph convolutional networks". In: *Bioinformatics* 34.13 (2018), pp. i457–i466 (cit. on p. 57).

[ZAG18] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. "Adversarial attacks on neural networks for graph data". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 2847–2856 (cit. on p. 22).

List of Figures

List of Tables