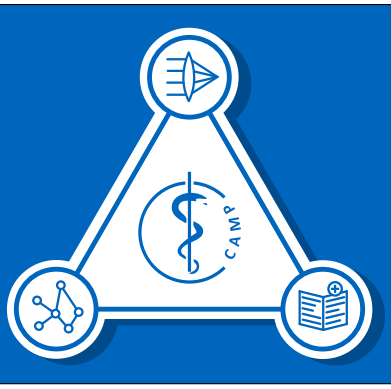
Computer Aided Medical Procedures
Prof. Dr. Nassir Navab



Dissertation

Deep Learning based Clinical Decision Support through Strong Differentiable Domain Priors

Hendrik Burwinkel

TUM School of Computation, Information and Technology
Technische Universität München

Technische Universität München
TUM School of Computation, Information and Technology

Deep Learning based Clinical Decision Support through Strong Differentiable Domain Priors

Hendrik Burwinkel

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Stephan Günnemann

Prüfer der Dissertation: 1. Prof. Dr. Nassir Navab

2. Prof. Dr. Raphael Sznitman

Die Dissertation wurde am 27.06.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 14.11.2022 angenommen.

Hendrik Burwinkel
Deep Learning based Clinical Decision Support through Strong Differentiable Domain Priors
Dissertation

Technische Universität München
TUM School of Computation, Information and Technology
Lehrstuhl für Informatikanwendungen in der Medizin
Boltzmannstraße 3
85748 Garching bei München

Abstract

Medical decision support based on machine learning and especially deep learning has greatly advanced in recent years. Providing faster, more reliable, and accurate diagnosis, treatment planning and assistance can significantly change the quality of patient care, treatment outcome and the daily medical routine. With increasing data complexity and digitalization of medical procedures, there is high potential but also rising challenges in the field. Especially for very complex tasks and only limited amount of data (a ubiquitous situation in medical context), domain prior information can play a crucial role in improving model performance by providing a structural understanding of the data. This results in the need for effective methods to incorporate the various forms of domain knowledge into the learning process. This work provides a conceptual analysis, develops methods for the effective consideration of both empirical and model-based priors and evaluates them on suitable medical applications.

First, the thesis will focus on empirical priors. Here, we will target relational knowledge in form of sample connectivity and meta information and logical knowledge in the form of rules and expert information. Graph Convolutional Networks (GCNs) have proven very effective in the processing and incorporation of relational information encoded as a graph structure within the data and will be focused on in this part of the thesis. While for numerical data, interesting applications have been developed in recent years, the direct application on imaging data, a medical domain with prominently only limited amount of data, is less pronounced. We will therefore motivate, develop and evaluate methodologies that leverage graph domain information for image processing within the challenging task of disease classification for chest X-ray images. In a second step, we will combine the powerful usage of relational knowledge with logical knowledge and developed a decision support method within the important medical field of symptom-based toxin classification, a domain with extensive literature-based knowledge about symptom to poison matching.

The second part of the thesis will focus on model-based priors in the form of scientific theories that are expressible in mathematical terms. Here, the thesis will focus on scientific knowledge in the form of physics-based priors, which allow full domain-covering mathematical descriptions. The thesis will give a detailed background on methodologies to incorporate such scientific priors. Following principles of physically motivated machine learning, the work shows the high efficiency of physical prior information and develops a methodology within the scope of cataract surgery for improved intraocular lens calculation, an exemplary chosen ophthalmic procedure that allows an inclusion of precise optical modelling.

Zusammenfassung

Die medizinische Entscheidungsunterstützung auf Basis von maschinellem Lernen und insbesondere Deep Learning hat in den letzten Jahren große Fortschritte gemacht. Die Bereitstellung einer schnelleren, zuverlässigeren und präziseren Diagnose, Behandlungsplanung und -unterstützung kann die Qualität der Patientenversorgung, das Behandlungsergebnis und die tägliche medizinische Routine erheblich verbessern. Mit zunehmender Datenkomplexität und Digitalisierung medizinischer Abläufe besteht ein hohes Potenzial mit gleichzeitig steigenden Herausforderungen in diesem Bereich. Besonders bei sehr komplexen Aufgaben und nur begrenzten Datenmengen (eine allgegenwärtige Situation im medizinischen Kontext) können Domain-Prior-Informationen eine entscheidende Rolle bei der Verbesserung der Modellleistung spielen, indem sie ein strukturelles Verständnis der Daten liefern. Daraus ergibt sich der Bedarf an effektiven Methoden, um die verschiedenen Formen des Domänenwissens in den Lernprozess einzubeziehen. Diese Arbeit liefert eine konzeptionelle Analyse und entwickelt Methoden zur effektiven Berücksichtigung von sowohl empirischen als auch modellbasierten Priors.

Zunächst wird sich die Arbeit auf empirische Priors konzentrieren. Hier fokussieren wir uns auf relationales Wissen in Form von Stichprobenkonnektivität und Metainformationen als auch logisches Wissen in Form von Regeln und Expertenwissen. Graph Convolutional Networks haben sich bei der Verarbeitung und Einbeziehung von relationalen Informationen, die als Graphstruktur in den Daten kodiert sind, als sehr effektiv erwiesen und werden in diesem Teil der Arbeit näher betrachtet. Während für numerische Daten in den letzten Jahren interessante Anwendungen entwickelt wurden, ist die direkte Anwendung auf bildgebende Daten, eine medizinische Domäne mit prominent nur begrenzten Datenmengen, weniger ausgeprägt. Wir werden daher Methoden entwickeln und evaluieren, die Graphdomäneninformationen für die Bildverarbeitung im Rahmen der anspruchsvollen Aufgabe der Krankheitsklassifikation für Röntgenbilder des Brustkorbs nutzen. In einem zweiten Schritt werden wir die leistungsfähige Nutzung von relationalem Wissen mit logischem Wissen kombinieren und eine Entscheidungsunterstützungsmethode innerhalb des wichtigen medizinischen Bereichs der symptombasierten Giftklassifikation entwickeln, einer Domäne mit umfangreichem literaturbasiertem Wissen über Symptom-Gift-Matching.

Der zweite Teil der Arbeit beschäftigt sich mit modellbasierten Priors in Form von wissenschaftlichen Theorien, die in mathematischen Termen ausdrückbar sind. Hier wird sich die Arbeit auf Wissen in Form von physikbasierten Priors konzentrieren, die vollständige domänenübergreifende mathematische Beschreibungen ermöglichen. Die Arbeit wird einen detaillierten Hintergrund zu den Methoden geben, um solche wissenschaftlichen Priors einzubinden. In Anlehnung an Prinzipien des physikalisch motivierten maschinellen Lernens zeigt die Arbeit die hohe Effizienz physikalischer Vorabinformationen und entwickelt eine Metho-

dik im Rahmen der Kataraktchirurgie zur verbesserten Berechnung von Intraokularlinsen, einem ophthalmologischen Verfahren, das eine Einbeziehung präziser optischer Modellierung erlaubt.

Acknowledgments

First, I would like to thank my PhD supervisor, Prof. Dr. Nassir Navab for his constant trust in my work and guidance during my PhD time, which was invaluable to the completion of this thesis. The CAMP chair is a special place due to his vast knowledge and incredible kindness, which creates a productive and fruitful environment for everyone who is lucky enough to be part of this amazing team. I really enjoyed counting me as one of them. Second, I would like to express my deep gratitude to Dr. Seyed-Ahmad Ahmadi, who acted as my mentor through most of my PhD time. The very enriching discussions about ideas as well as the mental support he provided were essential to me in this journey. It has been a great pleasure for me.

Furthermore, I would like to thank my colleagues and friends at the CAMP chair, especially my former office mates Johanna and Fabian, and also Christoph, Helisa, Matthias, David, Gerome, Anees, Tobias, Magda, Walter, Ashkan, Maria, Thomas, Chrissi, Xavier, Guillaume and many others. The PhD time would have not been the same without you and I am happy that we could make this experience together. We shared pain and joy in several exciting research projects and became more than colleagues over the years. A special thanks also to Martina for always being there in the time of need. Another thank you goes to Prof Dr. Florian Eyer and Dr. Tobias Zellner for their great support in our "AI in toxicology" project. The input and effort that you gave to our project was amazing and of great help for us.

Another big thanks goes to the company Carl Zeiss Meditec AG and there especially to Dr. Holger Matz, Dr. Stefan Saur and Dr. Christoph Hauger. A few years ago, they gave me the chance to pursue my PhD in a cooperation project with them and I am very thankful for the trust, freedom and support they gave me during this time and the warm welcome I received now as a fixed member of their team. Further, I would like to thank Michael Trost for the many fruitful discussions that I hope will continue in the future.

Finally, I would like to thank all the great people, who make my life a good one. Thank you to my friends for all the support and great times, I am very happy and blessed to have you. A very big thank you to my parents and my brother Dominik for their constant support and faith in me, it is difficult to express how much this means to me. And Franzi, thank you for being in my life, for your support, for the open ear you had for me during the PhD time and the patience you showed me during long working weekends.

Contents

Part I

Introduction

1 Introduction

1.1 Motivation

Of all technologies rising in the 21^{st} century, the field of machine learning and especially deep learning based on neural networks is one of the most prominent and promising. The learning capabilities of neural networks have shown tremendous success in various fields from computer vision tasks over reinforcement learning to medical image analysis and decision support [91, 192]. The computational power of these models results from thousands up to millions of learnable parameters [87], which allow the automatized learning of complex tasks from the provided data. While being powerful, the performance of neural networks is strongly dependent on this data. If the required quality, quantity or informativeness is not sufficient, the networks tend to overfit or learn an unstable and insufficient prediction behavior, making it infeasible for a real-life application.

Within the medical domain, the collection of larger amounts of high-quality data is very challenging, especially for imaging data, but also for other modalities. Many of the publicly available data sets only contain a few thousand and sometimes even only a few hundred samples [250]. At the same time, medical decision support has a high demand in prediction performance and stability. Techniques have been developed to improve the neural network performance on insufficient amounts and quality of medical data in form of the reduction of model complexity, the introduction of strong data augmentation and regularization procedures. While all these approaches help to improve model performance and generalizability, they are focusing on the prevention of harmful learning behavior. They are however not providing additional information about the underlying medical task that goes beyond the provided data [250]. This property is a substantial drawback of the described techniques, since instead of understanding more about the domain the network should operate in, it is simply prohibited from learning something wrong. Therefore, methodologies that integrate new information about the data domain into the learning task have the potential to boost the performance of neural networks significantly further.

Prior domain knowledge exists in various forms. It can be given by relational information about different patient groups, medical expert information about certain disease conditions and corresponding symptoms or underlying physical concepts of a measurement modality and corresponding patterns within the data. Depending on the medical field, one or multiple of these domain priors might be available. For e.g. clinical image diagnosis, additional information about the patient (age, clinical sex, medical condition) are intuitively and subconsciously used by clinicians for their diagnosis. Depending on if the physician faces a young or old patient, man or woman, a patient with or without preexisting clinical conditions, he or she will look at the same image with a different mind set and connect findings in the image differently.

This process results from experience the physician has gained over time. It is not hard to deduce that a neural network can benefit in its performance from this additional information in the same way. An effective incorporation of the various forms of prior knowledge is however far from trivial. This thesis will address the topic of effective domain knowledge integration into the learning process of neural networks.

1.2 Challenges and Contributions

Depending on the prediction task and the form of domain knowledge, different approaches and strategies need to be developed in order to allow for a proper integration into the learning process. Challenges of domain knowledge incorporation are given by e.g. the required differentiability of the inclusion process and the potentially highly unstructured form in which it is presented. This thesis will target the effective integration of domain prior knowledge into the learning process of neural networks. Three major categories of domain prior knowledge are given by relational knowledge (e.g. connections and relations within the data structure), logical knowledge (e.g. data-based if-then rules and expert information) and scientific knowledge (e.g. physical theories and differentiable modeling) [201]. Since both relational and logical knowledge priors are mostly empirically motivated, within the thesis we will group them into empirical priors, scientific knowledge is defined as model priors. This definition choice will be described in detail in Sec. 2.2.

For empirical priors, we will focus on the usage of graph convolutional neural networks, a form of neural network which is capable of effectively processing relational and structural data information expressed as a graph. Within related work, graph neural networks have been very effectively used for the processing of numerical information like pre-extracted image features. So far, little focus is placed however on the inclusion of relational knowledge via graph-based processing of imaging data itself. Here, we hypothesize the potential loss of important feature information relevant in graph context. We will develop a methodology for graph-based image processing and demonstrate the effectiveness of our approach on the challenging task of disease prediction on chest X-ray images. Then, we will extend the domain knowledge inclusion to logical knowledge in form of expert literature information provided as if-then data pairs. The developed approach will be based on an explicit encoding of the expert knowledge within the neural network and will be applied to the important field of intoxication prediction for poison control, a field with a rich literature in form of symptom to poison matching.

In the second part of the thesis, we will focus on the incorporation of model priors in form of physical formulas. This form of prior knowledge in principal provides a fully covering mathematical description of the data domain. For the integration of this form of prior knowledge, we will design a physically motivated domain-specific loss function as well as a network pre-training procedure based on vast amounts of artificial data. We evaluate it on the challenging task of intraocular lens calculation, a medical field with detailed physical priors of the human eye optics.

The thesis tries to address central aspects and challenges related to the three considered domain knowledge priors and motivate a direction to solve these problems based on the

developed methodologies. For every direction, a broader picture is provided to visualize how methodologies applied to the specific medical tasks in this thesis can be used more widely. The thesis therefore tries to paint a picture demonstrating the importance of domain knowledge, its high effectiveness when used properly and the potential it can have in various medical applications.

1.3 Organization

The thesis is structured into four parts. Within the first part, the introduction and motivation of the thesis are given in chapter 1. In the following chapter 2, we will give a condensed overview about the concepts of machine learning in the form of artificial neural networks and the topic of domain prior knowledge incorporation that will be addressed in more detail in the corresponding chapters.

The second part of the thesis will focus on the area of empirical priors and present graph convolutional networks as a very effective tool to incorporate them into the training process. The chapters are defined as follows:

- Chapter 3 will give a profound overview on graph neural networks (GNNs) and a background about the current state of the art of their applications within the medical field.

- Chapter 4 will present the usage of graph neural networks (GNNs) in the context of incorporation of relational information. For purely numerical data sets, there is already related work that has shown the high potential of this approach. We will therefore focus on the development of a direct application of GNNs on images, as discussed above a domain with typically only a limited amount of available data. The direct approach will allow the system to adapt the individual feature representations to the requirements of the graph, which is in itself stabilizing the prediction. The approach is evaluated on the challenging task of chest X-ray based disease classification.

- Within chapter 5, we will present how empirical priors in the form of logical or expert knowledge can be incorporated into the training as well. Here the expert knowledge is presented in the form of literature information for the challenging task of symptom based poison prediction, an important part of medical decision support for poison control centers. A GNN-based methodology is developed that is able to incorporate empirical information in both the form of patient meta information and expert literature knowledge in a fully differentiable way.

The third part of the thesis will focus on the second important form of domain priors: model priors in the form of scientific differentiable formulas. Compared to the empirically driven approaches discussed in the previous part, these approaches have the great advantage of an universal underlying theory that allows to completely cover the full data domain in a differentiable way. Here, the thesis will focus on the application of physical models:

- Chapter 6 will provide a detailed background about the incorporation of physical domain priors within the learning task and the application of physical constraints for training control and stabilization.

- Chapter 7 shows in detail, how this form of physical prior can be incorporated into the medical domain. We demonstrate the effectiveness of the approach with an application in ophthalmic surgery, more specifically the intraocular lens (IOL) refractive power calculation of cataract surgery, where the optical physics of the eye provide a strong and informative physical prior. The resulting optical network is analyzed in detail and compared against state of the art approaches in the field of IOL calculation.

In the fourth and final part, within chapter 8 we draw a conclusion about the outcomes and implications of the developed methodologies and address directions for potential future works.

The work presented in this PhD thesis is my own work, if not explicitly stated otherwise, nevertheless the *we*-form will be used throughout the thesis. This has partly stylistic purposes, but also reflects that research and the scientific process lives from the interaction with the scientific community and is always influenced by this interaction with colleagues and other researchers.

2 Domain prior knowledge in Machine Learning

The term domain prior knowledge refers to any kind of information that adds value and information to a data domain on which a learning task should be performed [201, 250]. Within this thesis, we will distinguish between the two forms of 1) empirical prior knowledge that is directly derived from data relations or following data-based rules, as well as 2) model priors that have an underlying scientific theory describing the domain as a whole. We want to leverage this kind of prior information explicitly within the learning process of neural network approaches. Artificial neural networks and their various architectural realizations are a form of machine learning, which has performed tremendous achievements in a vast amount of applications in recent years. Within the following section, we will give a brief introduction into this type of machine learning.

2.1 Artificial neural networks

2.1.1 Basic concepts of neural networks

Before starting deeper into the topic of this thesis, a first overview about the general concept of neural networks is given. Although it is assumed that the reader is familiar with the basic idea of this type of machine learning concepts, for completeness we will recollected its fundamental ideas and mathematical assumptions. The general idea of neural networks is based on the encoding of information within a hierarchical setup of neuron layers with an increasing complexity of encoded features from layer to layer [87]. This consecutive setup allows a neural network to learn very difficult tasks by a distribution of the level of understanding. The basic building block of the neural network is the neuron. Its initial idea is motivated from the setup of thought processes within the human brain, although compared to the brain the concept is substantially simplified. A brain neuron is an electrically excitable cell within the brain which is connected to other neurons via synapses. From these neurons, it can receive electrochemical signals. When the neuron experiences a sufficient voltage change over a short period of time, the neuron is activated, generates an electrochemical potential and sends it to the next connected neuron. This potential can be exciting or inhibiting to the activation of the following neuron [263].

This idea of the collection of signals from previous neurons in order to activate or inhibit a following neuron describes the basic working principle of a neural network. A neuron within a standard feed-forward neural network is connected to previous neurons and receives signals in form of a numerical value from every connected neuron. The resulting signal typically consists of the signals received from the previous neurons weighted with the weight

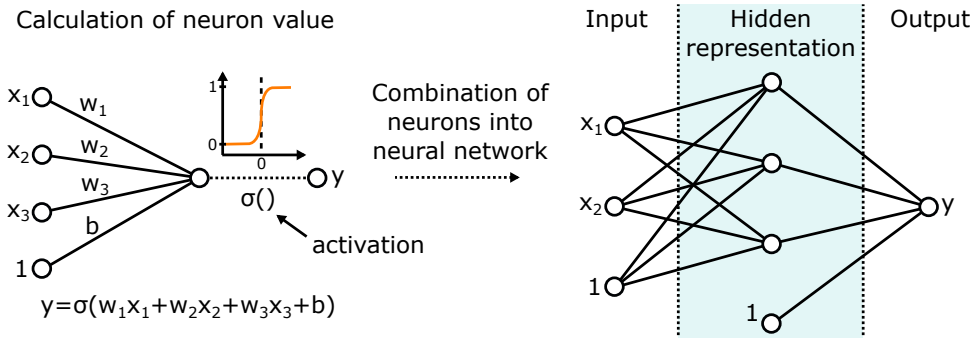


Fig. 2.1. On the left side, the basic building block of a standard feed-forward neural network is shown. A group of input signals x_i is weighted with corresponding weights w_i and a bias b is added. The resulting value is activated within an activation function $\sigma()$ and results in the output of the neuron. This setup is called a perceptron. Several of these neurons can be combined within a neural network layer, providing the input to the neurons of the following layers, shown on the right side.

belonging to the connection. The neuron accumulates the received signals into a new signal and processes it within an activation function. If the signal is strong enough to result in an activation, a signal is send from the neuron to the next neurons to which it is connected [87]. To visualize this in a simple example, we refer to Fig. 2.1. The neuron receives a signal from three connected neurons, which are weighted with three different weights. As activation, a non-linear function like the sigmoid or Rectified Linear Unit (ReLU) are used. Activation functions are in general functions, which perform a more or less steep non-linear value change when certain threshold values are surpassed. The sigmoid function is e.g. depicted in Fig. 2.1 and defined as $\sigma(x) = 1/(1 + \exp -x)$. Another very commonly used activation function is the Rectified Linear Unit (ReLU), which returns $\sigma(x) = 0$ for inputs x smaller than zero and corresponds to the linear function $\sigma(x) = x$ for values equal or larger than zero [87]. In order to shift the potential of these activation functions, a so called bias b is added to the signal. If the resulting signal is sufficiently strong (activated), the neuron provides its signal to the next neurons. One such processing cell is called a perceptron. Several layers of neurons can be placed consecutively to form a neural network (Fig. 2.1 right). The intermediate representations of the input, which are not presented to the user of the network, are referred to as hidden representations.

To simplify the mathematical expression of neural networks, the field relies on the concepts of linear algebra. Here, a group of inputs x_i is represented in form of a vector $\vec{x}$ and the weights w_{ij} applied to transform the vector $\vec{x}$ to the hidden representation $\vec{x'}$ is given by a matrix $\mathbf{W}$. The added bias is given in form of vector $\vec{b}$. The full network shown in Fig. 2.1 right can then easily be expressed in Eq. 2.1:

$$y = \mathbf{W}_2 \cdot \sigma(\mathbf{W}_1 \vec{x} + \vec{b}_1) + \vec{b}_2 \tag{2.1}$$

Here, $\mathbf{W}_i$ and $\vec{b}_i$ correspond to the weights and biases of the first and second layer correspondingly and the entry w_{ij} of $\mathbf{W}_i$ refers to the weight applied onto the entry x_j of vector $\vec{x}$ to receive x'_i of the new representation vector $\vec{x}'$. Within the activation function $\sigma()$, the hidden representation is calculated. Depending on the task, a function is applied onto the output y as well, here given by a scalar value. The concept of hidden representations as well as the non-linear activation via e.g. the sigmoid function or ReLU are essential aspects of neural

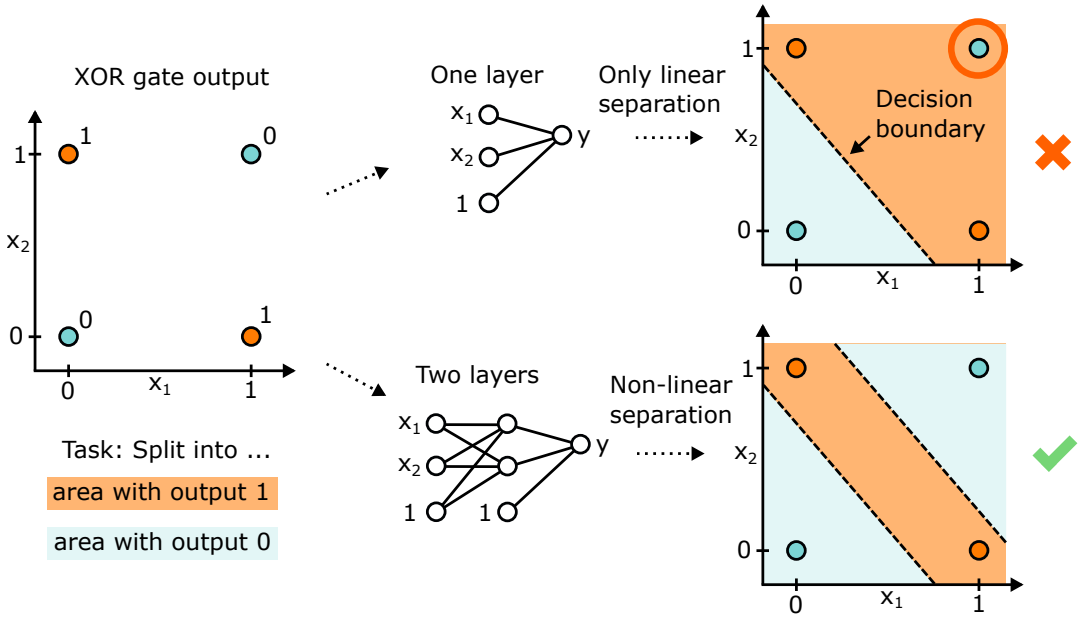


Fig. 2.2. As shown on the left side, representing the XOR gate requires the separation of output values $y = 0$ and $y = 1$ based on the input values x_1 and x_2. From the positioning of each data point in the data domain, it becomes clear that the problem is not linearly separable. Since a single-layer perceptron can however only solve linearly separable problems (linear dependency on x_1 and x_2 before applying activation), it is impossible to represent the XOR gate. Here, the Heaviside step function is used as activation. Using an additional hidden layer within a multilayer perceptron (MLP) with non-linear activations after each layer introduces the required non-linearity and provides the possibility to create a representation fulfilling the XOR requirement. The shown decision boundaries are exemplary solutions.

networks, since they enable the learning of more complex tasks in the first place [87]. We will explain this in more detail in the following.

We can ask the question, why it is beneficial to use multiple layers within a neural network instead of only one layer. To understand the benefit of multiple layers, we will look at a very simple problem that our neural network should represent, the so-called XOR gate. The XOR gate receives two binary inputs, resulting in the four possible input combinations (0, 0), (0, 1), (1, 0) and (1, 1). For (0, 0) and (1, 1), the output of the network should be 0, for (0, 1) and (1, 0), it should be 1. The value space is visualized in Fig. 2.2 left. If we have only one layer within the neural network, we would receive a direct connection from input to output, corresponding to two weighted connections to the output value. On this value, an activation function can be applied, e.g. the Heaviside step function. The Heaviside step function returns zero for an input value equal or below zero and returns one for a positive input value. As described above, this setup (input, output, activation) is called perceptron. Within the perceptron, a task as provided here can be solved, as long as the problem is linearly separable. For the XOR problem however this is unfortunately not the case, as can be easily seen from Fig. 2.2. No straight line within the data domain can successfully separate the 0 and 1 outputs (Fig. 2.2 top right). As a result, there is no combination of weight values, for which the given resulting line (boundary between area with outcome 1 and area with outcome 0) will correctly separate the data domain with respect to the XOR task. Therefore, there are functional relations which are not expressible by a simple perceptron. We can however solve this problem by switching to multilayer perceptrons (MLPs). By adding an intermediate layer

of neurons which correspond to a hidden representation between the input and output, it is possible to obtain the non-linear representation required to solve the XOR task, shown in the bottom left of Fig. 2.2. The shown decision boundaries are only an example solution, other boundaries are possible.

It again has to be stated, that the multi-layer setup alone is not sufficient and that the non-linear activation of the hidden representation is essential for the concept to function [87]. If not performed, the full network representation collapses to the representation of a simple perceptron. This can be easily shown in Eq. 2.2 for the network given in Fig. 2.1 right:

$$y = \mathbf{W}_2 \cdot (\mathbf{W}_1 \vec{x} + \vec{b}_1) + \vec{b}_2 = (\mathbf{W}_2 \mathbf{W}_1)\vec{x} + (\mathbf{W}_2 \vec{b}_1 + \vec{b}_2) = \mathbf{W}' \vec{x} + \vec{b}' \tag{2.2}$$

where $\mathbf{W}' = \mathbf{W}_2 \mathbf{W}_1$ and $\vec{b}' = \mathbf{W}_2 \vec{b}_1 + \vec{b}_2$. The relation between y and $\vec{x}$ is again given by a linear transformation, prohibiting any non-linearly separable tasks. The non-linear activation function prevents this simplification of the representation, since a direct combination of the weights and biases of the different layers is not possible anymore.

Although already requiring an MLP setup, the XOR task is still a very simple task to solve for a neural network. Today, different forms of neural networks like convolutional neural networks, recurrent neural networks and graph neural networks are used for the effective solving of several different and complex tasks. The concepts are going to be explained in more detail in the next part of the thesis. The tasks can be presented e.g. in form of classification, regression, segmentation or registration problems or a combination of these challenges. The neural network is presented with an input in form of e.g. a feature vector, an image or a video and its weights have been optimized towards providing the correct solution for the given input, e.g. the correct classification of an image content. A famous example is the classification of different handwritten digits for the MNIST dataset [63]. For the XOR example, we have seen that the weights need to be chosen with a certain configuration in order to solve the task. For the small amounts of weights, this process might still be manually feasible, for larger networks and more complex tasks, this is however not possible. Additionally, the whole purpose of a machine learning task is to learn the correct behavior from the data automatically, optimizing its weights towards the best solution for the given data domain. This optimization enables a processing of the data domain which can go beyond manually designed filters and is performed during the neural network training process.

2.1.2 Neural network training process

In the previous section, we have seen that a neural network is essentially a collection of neurons which are connected with other neurons via weighted connections to forward information from one neuron to the next. The performance and success of a neural network prediction is crucially dependent on the numerical values of these weights, which basically determine the representation that the network is encoding. In order to obtain the correct values to solve a given task, a training process is performed. This training process aims to optimize the network with respect to a so-called loss function which mathematically represents the task which should be solved. The loss function corresponds to a penalty function for the network, which returns high values if wrong predictions have been performed. The training process therefore aims to minimize the value that the loss function returns when it is applied onto the

networks predictions [87]. For a standard supervised training process, the training is realized with a collection of data samples for which the true solution is known, the so-called ground truth value. The prediction is then compared against this ground truth value within the loss function. A standard loss function is e.g. the Mean Squared Error (MSE) loss [87]. This loss returns the mean of the squared distances of all predicted values and their corresponding ground truth values as shown in Eq. 2.3.

$$\text{MSE}(\vec{y}, \vec{y}_{\text{GT}}) = \frac{(\vec{y} - \vec{y}_{\text{GT}})^2}{N} = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_{i,GT})^2 \tag{2.3}$$

where $\vec{y}$ corresponds to the network predictions y_i stacked to a vector, $\vec{y}_{\text{GT}}$ corresponds to the corresponding ground truth values $y_{i,GT}$ and N is the number of vector entries / predictions. It directly becomes apparent that the value of the MSE loss will be minimized when the predictions $\vec{y}$ are equal to the ground truth values $\vec{y}_{\text{GT}}$. Now, the question is how to get the network weights to a configuration where the output of the network for a given input will approximate the correct ground truth value. For the largest parts of performed neural network trainings, this will be based on a gradient-based approach called gradient descent. The term gradient here refers to the gradient of the loss function, essentially its slope in the hyperdimensional input space. In order to properly define this gradient, the loss function needs to be mathematically differentiable. This automatically requires that the neural network setup is differentiable as well, since its prediction will be part of the loss function. An activation function like the Heaviside step function is therefore not useful within a training process, since its gradient is zero within both the positive and negative value area. Activation functions like the sigmoid and ReLU function are common choices for the representation of non-linearity within neural networks. It is then possible to calculate the gradient for every weight of the neural network with respect to the loss function for a given prediction to determine the required change direction for reducing the loss function value. The weight values are modified using this gradient direction and a new and more accurate prediction is performed to receive the next update. This process is repeated until no significant improvement can be achieved anymore.

To mathematically describe the gradient descent process, we look at a very simple example of a neural network. Our network has only one layer, two weights w_1 and w_2 and one bias b. Following the above definition, for a set of N inputs with two values $x_{i,1}$ and $x_{i,2}$ our neural network performs the prediction $y = \sigma(w_1 x_1 + w_2 x_2 + b)$, where here we define σ as the ReLU activation function. The network therefore represents a function with a multilinear regression in the positive outcome area and the constant value zero in the negative. Note that for e.g. a regression task, the last outcome of the network would usually not be activated, here we chose the activation for the purpose of demonstrating the gradient descent process. Using $\text{f}(x) = \text{MSE}(x)$ defined in Eq. 2.3, we receive the loss function:

$$f(w_1, w_2, b) = \frac{1}{N}\sum_{i=1}^{N}(\sigma(w_1 x_{i,1} + w_2 x_{i,2} + b) - y_{i,GT})^2 \tag{2.4}$$

In order to optimize the neural neural network for the prediction task, w_1, w_2 and b have to be changed towards the direction that minimizes the loss function in Eq. 2.4. This direction can be obtained by the gradient of the function in multidimensional space. For a simpler description, we define $g = w_1 x_{i,1} + w_2 x_{i,2} + b$ and $h = \sigma(g)$. To receive the gradient ∇f of

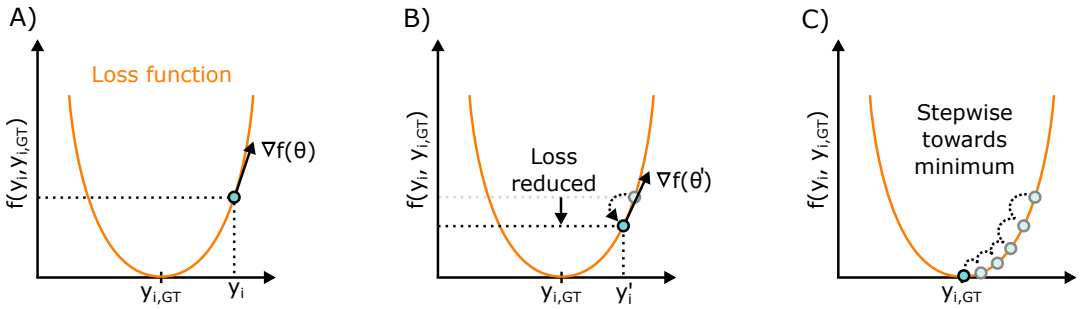


Fig. 2.3. Schematic description of the gradient descent process (for simplicity only one data point is assumed here). A) For given values of network weights θ (representative for weights and biases) the outcome y_i of the network for a given input x_i is calculated. The output is used within loss function f and the gradient $\nabla f(\theta)$ with respect to the weights θ is calculated. B) The weights θ are updated to θ' using one step of the gradient descent algorithm and a new output value y_i' is calculated. C) The process is repeated until no more improvement can be achieved and the network performs nicely on the given input data (prediction y_i very close to $y_{i,GT}$).

f, we will calculate the partial derivatives of Eq. 2.4 with respect to w_1, w_2 and b. They are given by:

$$\nabla f(w_1, w_2, b) = \begin{pmatrix} df/dw_1 \\ df/dw_2 \\ df/db \end{pmatrix} = \begin{pmatrix} df/dh \cdot dh/dg \cdot dg/dw_1 \\ df/dh \cdot dh/dg \cdot dg/dw_2 \\ df/dh \cdot dh/dg \cdot dg/db \end{pmatrix} \tag{2.5}$$

The representation of the partial derivatives on the right side of Eq. 2.5 is obtained using the chain rule. The partial derivative dh/dg is dependent on the ReLU activation function and yields 1 if $g \geq 0$ and 0 otherwise. We can calculate this approach for the loss function defined in Eq. 2.4 and receive the following partial derivatives:

$$\nabla f(w_1, w_2, b) = \begin{pmatrix} \frac{1}{N}\sum_{i=1}^{N} 2(\sigma(w_1 x_{i,1} + w_2 x_{i,2} + b) - y_{i,GT}) \cdot dh/dg \cdot x_{i,1} \\ \frac{1}{N}\sum_{i=1}^{N} 2(\sigma(w_1 x_{i,1} + w_2 x_{i,2} + b) - y_{i,GT}) \cdot dh/dg \cdot x_{i,2} \\ \frac{1}{N}\sum_{i=1}^{N} 2(\sigma(w_1 x_{i,1} + w_2 x_{i,2} + b) - y_{i,GT}) \cdot dh/dg \cdot 1 \end{pmatrix} \tag{2.6}$$

Using Eq. 2.6, we have now received the direction of optimization for the current weight and bias values. Since the gradient points towards the direction of steepest ascend, we have to move into the opposite direction for the steepest descend. We however only know this direction for the current point. Within the hyperplane of the loss function, gradients can strongly change depending on the current position, also local minima are possible. Additionally, we do not know how far we have to move in the direction of the gradient. Therefore, we have to rely on an iterative process where we take a small step in the opposite direction of the gradient and then reevaluate our position by calculating the partial derivatives for the new weight and bias values before moving in the corresponding direction (Fig. 2.3). This step-wise optimization process is called gradient descent [87]. The size of the performed step is given by the so-called learning rate l of the training process. An update step of w_1, w_2 and b is therefore given by:

$$\begin{pmatrix} w_1' \\ w_2' \\ b' \end{pmatrix} = \begin{pmatrix} w_1 - l \cdot df/dw_1 \\ w_2 - l \cdot df/dw_2 \\ b - l \cdot df/db \end{pmatrix} \tag{2.7}$$

The new values w_1', w_2' and b' should result in a better prediction of the neural network on the given training set than the previous values. One update step of the neural network performed

on the full training set is called an epoch. Repeating this process for several epochs moves the values closer and closer to the ideal setup of the neural network for the given training set. Here, the learning rate plays a crucial factor, a very large learning rate can e.g. result in a overshooting of the minimum, potentially even worsening the performance. When the learning rate is chosen correctly, after a sufficient number of epochs, the performance does not significantly improve anymore and the training is stopped. Ideally, the neural network has automatically learned to perform a strong prediction on the dataset without the necessity of a human defining the weights explicitly.

If the given dataset was representative of the data domain, the learned network setup is generalizable to new unseen data points from the same data domain. To supervise and evaluate this, the used data is normally split into training, validation and testing data. The training data is used for the calculation of the gradients and updating of the network weights. The validation data is used to receive an evaluation of performance and generalizability of the current state of the network on an independent dataset during training, normally after every epoch. It can also be used to perform a hyperparameter tuning (e.g. adaption of learning rate) and an early stopping of the training for the best performance on the validation data. Finally, the testing set is used to receive an independent evaluation of the neural network prediction performance on new data, after the training has finished. We will briefly give an overview about important evaluation metrics which are used in this thesis in the next section.

2.1.3 Important evaluation metrics

The evaluation of the model performance on an independent and previously unseen test dataset is an important step to evaluate how well the representation learned by the neural network has generalized to the data domain. The following metrics are used within this thesis:

Regression metrics: The provided information is the ground truth values $y_{\mathrm{GT},i}$ of all N samples of the test set and the corresponding prediction values y_i of all neural network predictions on these test cases.

Root Mean Squared Error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{\sum_i (y_i - y_{\mathrm{GT},i})^2}{N}} \tag{2.8}$$

Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{\sum_i |y_i - y_{\mathrm{GT},i}|}{N} \tag{2.9}$$

Multiclass classification metrics: The provided information is the ground truth class label $y_{\mathrm{GT},i}$ of all N samples of the test set and the corresponding predictions y_i of the neural network on these test cases. For the metric calculation, each class c of the number of classes C is treated as a binary classification problem. Correspondingly, for every class we define a number

of true positives (TP_c), for all cases where the class was correctly predicted, a number of false positives (FP_c), for all cases where the class was wrongly predicted, although another ground truth class was given, and a number of false negatives (FN_c), for all cases where another class than the ground truth class was predicted.

Accuracy (Acc):

$$\text{Acc} = \frac{\text{TP}}{N} \tag{2.10}$$

where TP is simply the sum of all correct predictions.

Precision for class c (P_c):

$$\text{P}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \tag{2.11}$$

Recall for class c (R_c):

$$\text{R}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \tag{2.12}$$

Micro precision (P_m):

$$\text{P}_m = \frac{\sum_c \text{TP}_c}{\sum_c \text{TP}_c + \text{FP}_c} \tag{2.13}$$

Micro recall (R_m):

$$\text{R}_m = \frac{\sum_c \text{TP}_c}{\sum_c \text{TP}_c + \text{FN}_c} \tag{2.14}$$

F1 micro score (F1 micro):

$$\text{F1 micro} = 2\frac{\text{P}_m \cdot \text{R}_m}{\text{P}_m + \text{R}_m} \tag{2.15}$$

F1 macro score (F1 macro):

$$\text{F1 macro} = \frac{2}{C}\sum_c \frac{\text{P}_c \cdot \text{R}_c}{\text{P}_c + \text{R}_c} \tag{2.16}$$

The metrics given in bold will be used throughout the thesis for the evaluation of our developed methodologies. Both RMSE and MAE reflect on the performance of the regression task, while the RMSE gives a larger weighting to strong outliers. While the accuracy estimates the performance over all classes, the F1 score metric involves more information about the performance on individual classes. For classification tasks, where every case has a single label, the accuracy and F1 micro score are however identical.

2.1.4 Data and performance dependency

The description of the neural network training process has highlighted a major strong point of the concept of machine learning. The setup of the optimization process allows a non-linear functional approximation of in principal arbitrary complex learning tasks. Using

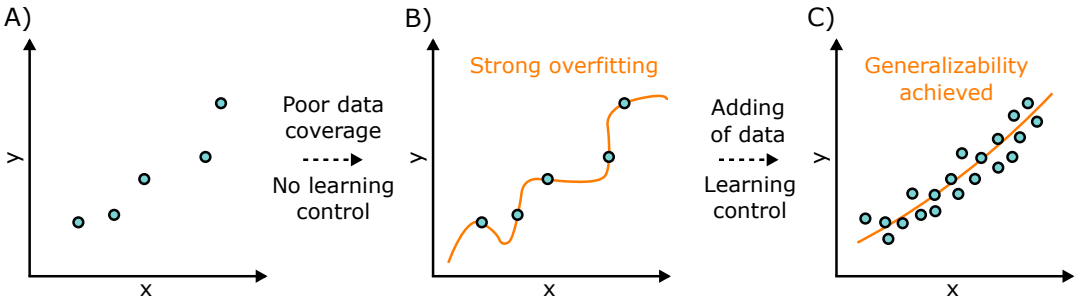


Fig. 2.4. Schematic description of overfitting process. A) The provided data domain for the learning task. B) Since the data domain is very sparse and at the same time no proper learning control has been performed, the powerful neural network trained on the data can learn a representation which results in almost perfected predictions on the training set. Every training data point is essentially learned by heart and the correct value is given. From the slope of the learned functional relation it however becomes obvious that the resulting representation is not generalizable and will lead to poor performance on new data points of the domain. C) Adding more data points to the training and increasing the quality of the learning control (e.g. reducing network size), results in properly learned relationship that has the potentially to correctly predict new data points.

learning concepts like gradient descent, the approach does not require the manual crafting of weight constellations or parameter dependencies but instead automatically learns the optimal representation of the data that it is provided with. However, within this major strong point of machine learning lies also its major weakness. Since the approach fully relies on the information extracted from the provided data, it is highly dependent on the quality and quantity of the data provided. A bad quality of the data will automatically result in poor prediction performance, since wrong relationships have been extracted from the provided dataset. Additionally, the non-linear and therefore very powerful representation capabilities of machine learning approaches like neural networks have a tendency towards overfitting on small amounts of data. Overfitting can be interpreted as a form of learning the provided data by heart without actually extracting a more meaningful and generalizable representation for the prediction on new unseen cases (see Fig. 2.4). The approach is therefore also relying on a sufficient amount of data provided [87].

There are several ways for a data scientist to improve the performance of a machine learning approach like a neural network. Among them are [87]:

- Hyperparameter tuning: Network and learning specific settings like the network size, choice of activation function and learning rate can have a major effect on the training process and network performance. Typically, a validation set is used to optimize these settings before evaluating the performance of the network on an independent test set.

- Regularization: Within e.g. an addition to the loss term the network is penalized for learning unnecessary complex representations of a given dataset. This increases the chance to receive a learned representation that generalizes well on new given data points of the same data domain.

- Augmentation: Another approach to improve the generalizability of the learned representation. Given a set of data points, new data points are generated by randomly modifying the input using a couple of transformations. For e.g. an image classification

task, the image could be rotated, mirrored or distorted. The goal is to create a multitude of new images from a real image which also could be part of the data domain. The chosen transformations therefore have to result in realistic transformations. It is e.g. perfectly realistic to mirror an image of a cat, effectively changing left and right, less realistic to turn it upside down.

- Ensemble learning: Instead of relying on one model for the prediction task, several different models are trained for the same task and their predictions are combined. This reduces the effect of individual weaknesses of certain approaches and stabilizes the overall outcome of the prediction task.

- Dropout: During the training process, random units of the network during a training step are set to zero. The network is therefore forced to learn its prediction in a more robust way to disturbance. The concept essentially mimics the ensemble behavior, since effectively different networks need to be trained for the prediction.

The above described techniques can be very effective and result in major improvements of the neural network prediction performance. At the same time, they are not introducing any new additional information of the data domain whose representation should be learned. Also techniques like augmentation which are generating new data points are in principal only modifying already existing information. It can however be strongly beneficial to introduce further prior knowledge about the data domain into the learning process. Domain prior knowledge can highly stabilize the prediction process and compensate for both quality- and quantity-related data insufficiencies. Within the next section, we will look further into this topic.

2.2 Domain prior knowledge

2.2.1 Forms of domain prior knowledge

As described previously, the incorporation of domain prior knowledge can assist machine learning concepts like neural networks to achieve better outcomes and higher stability of their prediction results. The term domain knowledge in general refers to any information that is available about the data domain which the system should learn to represent. As described in the introduction, within this thesis we are going to focus on three main areas of domain prior knowledge, namely relational knowledge, logical knowledge and scientific knowledge [201]. These three groups are categorized as follows:

- **Relational knowledge**: This type of domain prior knowledge refers to information which correlates data samples or features from a domain with each other. The value of this form of domain prior lies in the structural context that it can add to the prediction task. Instead of considering the information of a sample isolated, it is possible to link further information from the related instance to the sample representation. This form of knowledge therefore has a stabilizing and regularizing effect. Additionally, in a multi-modal data context, it might be beneficial to not treat every provided information in an

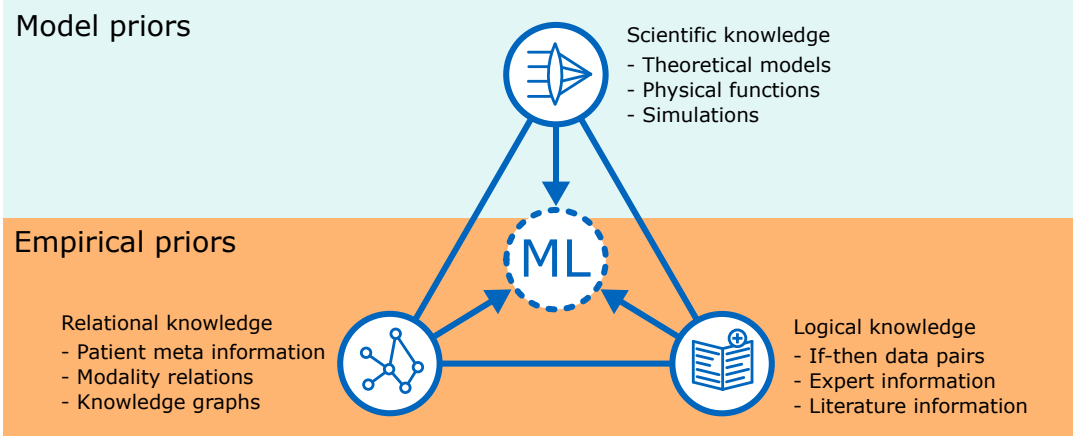


Fig. 2.5. Forms of domain prior knowledge discussed in this thesis. Broadly, we separate into empirical data-based priors as well as model priors relying on theoretical functions and physical formulas, which can be integrated into the machine learning (ML) training process. The three corresponding subgroups are given by relational, logical and scientific knowledge. As described in the text, the term "scientific" follows the terminology of [201] and does not refer to the quality of the prior compared to the other forms.

equal manner. Some information might yield a larger value in the form of a contextual instead of a purely sample-based use.

- **Logical knowledge**: Based on e.g. empirical findings, it is possible to derive data domain-related rules for the prediction task at hand. Within the medical field, logical knowledge is a common form of domain priors via diagnostic reasonings based on patient symptoms or other medical findings. The physician can examine the patient for a group of symptoms and derive a rule-based diagnosis based on the findings. Logical knowledge therefore represents decision guidance often motivated from expert knowledge and can be represented in if-then rule data pairs.

- **Scientific knowledge**: The third form of considered prior knowledge corresponds to general theoretical concepts on the data domain. This can be represented by e.g. physical formulas which provide a complete mathematical coverage of the domain and are fully differentiable. Especially functional representations which cover the full data domain are a very powerful domain prior since they essentially contain a large knowledge advantage on the full domain area. The defining property of this form of domain prior is its universal character for the prediction task.

The terminology of these three types of domain prior knowledge is following from [201]. We would like to highlight that the term "scientific" does not reflect on the quality of the obtained domain knowledge, rule-based logical knowledge for instance can also be derived from highly scientifically sound findings. Here, the terminology is chosen to describe theoretical concepts which provide a continuous coverage for the full domain. This stands in contrast to the first and second type, which provide guidance based on either related samples or features from the data cohort or isolated if-then rules for the provided input. We will therefore group the first two forms of domain prior knowledge under the concept of **empirical priors**, the third form under **model priors** (see Fig. 2.5). The thesis is separated into these two parts correspondingly. Again, we would like to highlight that the chosen distinction is not absolute.

Theoretical concepts are e.g. often motivated or corroborated by empirical findings and the defined relation between different samples of the data cohort can be based on theoretical modeling. Still, the main distinguishing factor that we see between the two groups is the fact that scientific knowledge as defined above aims to concisely model a full data domain in a differentiable way, while relational and logical knowledge operate more on the individual empirical sample level.

There is a vast body of work on domain prior information and its incorporation for prediction tasks. Within this thesis, we will focus on the incorporation of domain prior knowledge for the training of neural networks and corresponding machine and deep learning applications. For the empirical priors, the focus will be on the knowledge incorporation via graph convolutional neural networks (GCNNs). This rather new type of neural network approach is capable of leveraging non-Euclidean structural information within the data domain. It is therefore very nicely suited for the incorporation of empirical priors. For model priors, we will focus on the incorporation of especially physical theories into the neural network structure and training process. For both directions of prior knowledge incorporation, we will provide a detailed review of the historical development of the approaches from the theoretical as well as application perspective, and evaluate the current state of the art and its medical implications. Additionally, for the chosen medical application fields in which we will demonstrate the effectiveness of our newly developed approaches, we will provide a corresponding review as well.

2.2.2 Challenges of domain prior incorporation

As described above, domain prior information can appear in various forms. Several challenges need to be addressed with respect to the incorporation of domain prior information into medical decision support:

- Methodological challenges: A basic question which needs to be answered by the developed approaches is how to learn from the provided different domain prior knowledge forms most effectively. Since today's learning procedures under incorporation of neural networks require gradient based training, it is essential to represent domain prior knowledge in a differentiable way. Only like this, the provided prior is processable within the concept of loss function minimization. A prior in the form of e.g. logical knowledge provides if-then statements which determine hard decision rules. These need to be transferred into mathematical representations usable within the neural network. A physical theory describing certain relationships is by itself not directly useful as long as it cannot be transferred into the data-based learning process. These challenges will need to be tackled individually for every type of domain knowledge, additionally potential combinations have to be evaluated.

- Medical challenges: Especially for the inclusion of medical expert information, it needs to be considered that the description of experiences of medical doctors are subjective and its documentation can vary strongly from expert to expert. Additionally, the experience itself can vary substantially leading to different conclusions within the provided expert information. These factors result in sometimes very unstructured behavior for the

provided knowledge, a challenge which will become strongly apparent later in Chap. 5. It is therefore essential to first determine the actual expert information. This is described e.g. in [250] in terms of diagnostic patterns in which a medical expert performs his or her diagnosis, potential expressible in terms of logical knowledge. The developed methodologies require the capability to deal with these uncertainties, additionally the knowledge itself needs to be curated correspondingly.

- Data challenges: Especially for empirical priors and relational knowledge, it is required to determine which informative priors are actually present within the provided data. Informativeness of the chosen data sources is an essential factor to ensure performance for the developed methods. This will be clearly demonstrated within Chap. 4 and 5.

- Resource challenges: As mentioned before, for the incorporation of empirical priors, the thesis will explore the very promising field of graph neural networks. This type of network can be very powerful, at the same time relational information processed within a graph structure always requires the presence of such a context-providing graph. This will directly result in methodological challenges how to prevent an excessive memory usage, especially for the image-based approaches in Chap. 4.

Within the next chapter, we will start the process to tackle these challenges by a detailed introduction into the field of graph neural networks, their developmental journey as well as important related works in the field.

Part II

Empirical Priors: Graph Convolutional Networks

3 Theory of neural network-based graph processing

3.1 The graph as a powerful empirical prior

3.1.1 The value of structural information

Medical decision support based on machine and especially deep learning requires a sufficient amount of medical data to train on. Here, data can come in various forms like images or volume scans of the human body by e.g. magnetic resonance imaging (MRI), X-Ray scans or optical coherence tomography (OCT), numerical or categorical features (blood values, medical pre-conditions, ...) and additional information like patient age, clinical sex or profession. It is far from trivial to process these various forms of data effectively and in consequence often not all available information is used for a task of medical decision support or clinical inference.

Nonetheless, human medical experts intuitively consider additional information when performing a diagnosis on a patient. When e.g. evaluating an X-Ray imaging scan under suspicion of containing tumorous tissue, the physician does not only see the image but implicitly considers all the additional information about the patient in his or her examination as well. This incorporation of additional information is based on the medical expert's experience, which helps the expert to place the image into context with other medical cases examined before. For very young patients, the physician might expect a different visual appearance for the imaging scan than for older patients, allowing to analyze the data more effectively within this context. A structure is created into which the case at hand is integrated based on the relations to other cases. This structure provides a background which makes a diagnosis on the data more robust and understandable. It is intuitive to assume that computer-based decision support can profit from such a consideration of additional modalities as well.

When processing data using computer-based methodology, the standard approach to consider information sources from different modalities is the concatenation of the features in order to process them collectively. This approach can however lead to a suboptimal consideration of the different sources. When combining e.g. hundreds of features coming from an imaging modality with one numerical meta feature of the patient, e.g. the age, the overwhelming amount of imaging-based features can easily undermine the single numerical information. Additionally, such a processing approach does not reflect the medical expert's intuition, since especially the patient meta information such as age and clinical sex does not unfold its full potential within an individual consideration for a single case but more in the context that it provides with respect to other patients. It forms a topology within the data domain and allows the useful consideration of other cases for the prediction task at hand. Separating this

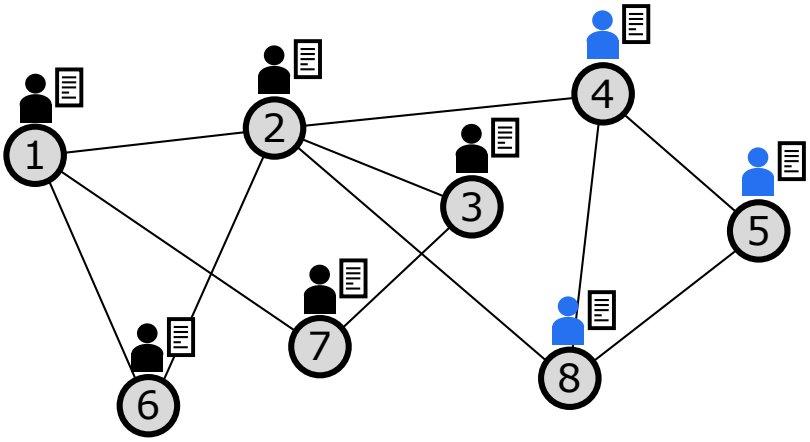


Fig. 3.1. Visualization of a patient graph containing eight patients which are connected based on similarity of their meta information. Additionally, every patient still has an individual collection of features. The colors correspond to a corresponding prediction task, in this case the detection of a disease. Here, patients in black are healthy while patients colored in blue suffer from the disease. The connections between the patients help the medical expert to distinguish the two groups from each other since the graph provides a structure within the patient cohort. A single case is therefore predicted under consideration of its neighborhood.

information from the processed features and using it to shape a structure within the data can highly enhance the value for a decision support task. One example is given in Fig. 3.1. The medical expert has to determine if a patient suffers from a certain disease. The cases in black are healthy while the patients in blue are ill. The connections between the patients reveal a structure within the patient cohort, which the medical expert can use to improve the prediction. Considering neighboring cases of a patient provides an additional data source besides the patient's medical record alone.

One very effective way to express such a structure in mathematical terms is the construction of a graph. A graph consists of multiple instances called vertices which are connected via edges. The edges allow an information flow between the different patient representations and can therefore provide structural context for an individual decision on a case by changing its representation within the graph. Especially for complex graph data structures, the application of machine learning approaches like neural networks is a very promising expedition. They can provide an optimized way to incorporate the empirical priors of the data structure into the medical decision process. The plethora of approaches following this idea is collected within the methodological field of graph neural networks (GNNs). Graph neural networks enable an end-to-end learning procedure on the graph structure itself and allow the consideration of the complex topology and underlying empirical priors within the network. This chapter will focus on providing a detailed theoretical background and motivation of GNNs before focusing on the development and evaluation of new methodologies in the next two chapters.

3.1.2 Overview on graph neural networks

The term graph neural network (GNN) encapsulates all neural network based approaches which are operating on graph-based data structures [248]. We are first going to provide the mathematical terminology which will be used throughout this thesis to describe graphs,

following the common terminology within the literature [221]. We define a graph G as a data structure which encodes the relation between different data points in the form of connections within the graph. Every data point is represented by a vertex $v_i \in \mathbf{V}$ with feature representation $x_i \in \mathbf{X}$ within the graph structure, a connection in the form of a graph edge $e_{ij} \in \mathbf{E}$ between two vertices v_i and v_j within the structure corresponds to a relationship and similarity between the two data points. The structure of the graph can be encoded within a so-called adjacency matrix $\mathbf{A}$, where $\mathbf{A}_{ij} = 1$ if $e_{ij} \in \mathbf{E}$ and $\mathbf{A}_{ij} = 0$ otherwise. Using a weighting mechanism on the graph's edges with weights w_{ij} on edge e_{ij} can further improve the graph's meaningfulness. For the resulting weighted adjacency matrix $\mathbf{W}$, we correspondingly define $\mathbf{W}_{ij} = w_{ij}$. Graphs can be distinguished into directed and undirected graphs [221]. An undirected graph is a special case of a directed graph with $w_{ij} = w_{ji} \; \forall \, i, j$. The collective $\mathbf{X}$ of all feature representations x_i represents the signal on the graph $G(\mathbf{V}, \mathbf{E}, \mathbf{W})$.

Depending on the learning task, we can distinguish between three major graph-based applications with respect to medical decision support [248]:

- Vertex-based: Vertex-based prediction approaches focus on the prediction on vertex level. Considering e.g. the patient graph given in Fig. 3.1 the vertex based objective would be the classification of the health status of every individual vertex, corresponding to the patient. Especially for patient-level diagnosis, vertex-based approaches are therefore very important. Also the developed methodologies in this thesis are patient- and therefore vertex-based. The training input to a vertex-based training task can therefore constitute of multiple but also only a single graph.

- Edge-based: Interpretable as a counterpart to vertex-based approaches, edge-based prediction tasks perform a link prediction between different vertices. Given e.g. the patient medical records of multiple patients in Fig. 3.1, an edge prediction would perform a prediction if two patients should be linked within the graph. Again, the training input can consist of one or multiple graphs.

- Graph-based: Finally, graph-based predictions focus on the prediction of a graph as a whole. Considering the case of e.g. patient brain connectomes from MRI scans expressed in form of a graph, a graph-based prediction task could focus on the classification of these scans. As a result, graph-based approaches are normally trained using multiple graphs.

Following a nice review and proposed taxonomy for GNNs in [248], GNNs can in general be categorized into four groups, namely Recurrent GNNs (RGNNs), Convolutional GNNs (CGNNs), Spatio-temporal GNNS (STGNNs) and Graph Autoencoders (GAEs). In the following, we will provide a brief overview about the underlying ideas of each group followed by an introduction into the underlying mathematical concepts in Sec. 3.1.3 and 3.1.4. From Sec. 3.2.1 onwards, we will then provide a detailed explanation of the different groups and corresponding important methodologies developed within recent years. Here, we will especially focus on Convolutional GNNs, since our developed methodologies in Chap. 4 and 5 are based on CGNNs.

Recurrent graph neural networks (RGNNs): As already expectable from the name, RGNNs are motivated from recurrent neural networks (RNNs) or more broadly recursive neural network approaches, e.g. [89, 90, 204, 219]. They constitute the group of first works performed in the direction of GNNs [248]. RNNs are designed for the processing of time series data [200]. As will be motivated in Sec. 3.1.3, since the general idea of graph signal processing and information propagation can be deduced from digital signal processing [178] (which is focusing on time series data), it was therefore intuitive that the first works in the field of GNNs would be based on RNN concepts. RGNNs rely on an information propagation approach, where a graph feature representation $\mathbf{X}$ is iteratively processed via a recurring transition function. During each recurring application of the transition function, information is exchanged between the vertices v_i until a stable equilibrium is reached. The resulting feature representation of the graph is then processable via an output function. Details on RGNNs will be provided in Sec. 3.2.1.

Convolutional graph neural networks (CGNNs): One of the most impactful developments in the GNN field is the development of CGNNs. The underlying idea of CGNNs is to extend the convolution operation defined for convolutional neural networks on regular grids to non-Euclidean domains and irregular structures [29]. The information of a vertex v_i and its neighboring vertices v_j with $e_{ij} \in \mathbf{E}$ is convoluted within the new representation for the vertex. In general, these networks can be divided into spectral (e.g. [28, 29, 58, 128]) and spatial (e.g.[32, 34, 94, 231]) approaches. While spectral approaches follow the ideas of Laplacian-based graph signal processing (explained in Sec. 3.1.3) to filter a graph feature representation in its spectral domain, spatial approaches perform a direct information propagation (explained in Sec. 3.1.4) in graph domain. Spatial approaches therefore show a similarity to RGNNs. CGNNs however differ strongly in the fact that CGNNs apply individual graph convolutional layers to the feature representation which can be stacked within a network to extract high-level vertex representations [248]. This advantage justifies the huge impact of CGNN models. Since as explained above the developed graph based methodologies in this part of the thesis for the processing of empirical priors in Chap. 4 and 5 are based on CGNNs, we will perform an extensive review of CGNNs in Sec. 3.3 and 3.4.

Spatio-temporal graph neural networks (STGNNs): While RGNNs and CGNNs focus mainly on the processing of the spatial information given within the graph, STGNNs transfer this concept to the processing of spatial and temporal information at the same time (e.g. [115, 143, 209, 258]). Additional to the spatial dimension the graph feature representations therefore also have a time dimension which can be processed together with the topological structure. The work within this part of the thesis does not address spatio-temporal data, we will still address this group of GNNs and provide and overview within Sec. 3.5.1.

Graph Autoencoders (GAEs): The last group of GNNs can be summarized within GAE approaches. Similar to standard autoencoders, GAEs work on unsupervised approaches to encode a given graph representation in a latent representation which can be used to reconstruct the graph. The applications of GAEs lie therefore in the field of graph encoding into a graph embedding (e.g [40, 129, 232, 260]) as well as graph generation (e.g. [24, 57, 86, 136]) [248]. As stated already for STGNNs, we will focus on CGNNs in this part of the thesis, but still provide an overview of GAEs within Sec. 3.5.2.

As discussed for RGNNs and CGNNs, the underlying concepts of GNN approaches are motivated from graph signal processing and the resulting information propagation. Before diving deeper into GNN theory, we will therefore provide an introduction into these concepts in the following two sections. Sec. 3.1.3 will motivate the spectral filtering concepts which are also applied in spectral CGNN approaches, from these Sec. 3.1.4 will deduce the direct spatial information propagation approaches used in both RGNNs as well as spatial CGNNs.

3.1.3 Graph signal processing

The concept of Graph Signal Processing (GSP) focuses on the filtering of a graph signal $\mathbf{X}$ given on a graph structure G [221]. GSP can be seen as an extension of Digital Signal Processing (DSP) which processes time signals [178]. As described in [178], DSP focuses on signals and their representations, filters which process these signals, signal transformations and sampling of signals. To understand the idea of DSP, we are assuming a time signal $\mathbf{X}$ represented as a discrete number of N samples x_t with $t = 0, 1, ...N-1$. The samples are ordered chronologically, meaning that x_t at time t precedes x_{t+1} and succeeds x_{t-1}. Often, the processing of the signal is performed in its spectral domain. In order to transfer the signal into its spectral representation, we now need to apply a corresponding transformation. Since we are dealing with a discrete time signal, the transformation used within DSP is the so-called z-transform $\mathbf{X}(z)$, which transfers the discrete time signal into a complex frequency domain. The z-transform constitutes the discrete-time equivalent of the Laplace transform. The variable $z = ae^{i\omega}$ is a complex variable with amplitude a and phase ω. The representation of the z-transform is given by the polynomial equation with variable z^{-1} in Eq. 3.1 [178]:

$$\mathbf{X}(z) = \sum_{t=0}^{N-1} x_t z^{-t} = \sum_{t=0}^{N-1} x_t a e^{-i\omega} \tag{3.1}$$

To provide an intuition of the effect of the z-transform, one can examine the special case of $a = 1$. In this case, Eq. 3.1 corresponds to the discrete-time Fourier transform, which converts the time signal into its corresponding frequency signal. In order to process the time signal in DSP, we now want to apply a filter $\mathbf{H}$ with components h_t. Similarly, we can define a transformed signal filter $\mathbf{H}(z)$ as described by Eq. 3.2 [178]:

$$\mathbf{H}(z) = \sum_{t=0}^{N-1} h_t z^{-t} \tag{3.2}$$

The convolution of time signal $\mathbf{X}$ with filter $\mathbf{H}$ simplifies to a multiplication in frequency domain [221]. We can therefore receive a filtered signal $\mathbf{X}_{out}$ by applying the filter onto the input signal as shown in Eq. 3.3:

$$\mathbf{X}_{\text{out}}(z) = \mathbf{H}(z) \cdot \mathbf{X}_{\text{in}}(z) \tag{3.3}$$

Within Graph Signal Processing (GSP), these concepts applied to time signals are now generalized on the above mentioned graph structures. We rely on the definition of a graph $G(\mathbf{V}, \mathbf{E}, \mathbf{W})$ with feature representation $\mathbf{X}$ and adjacency matrix $\mathbf{W}$ as described in Sec. 3.1.2. The feature representation $\mathbf{X}$ can be related to the time signals x_t, but instead of being represented within a sequence, it is given by a vector or one one-column matrix (in case that $x_i \in \mathbf{X}$ are

scalars) or a matrix (in case that $\vec{x}_i \in \mathbf{X}$ are vectors). For the scalar case, we therefore receive $\mathbf{X} = [x_0 x_1 ... x_{N-1}]^T \in \mathbb{R}^N$ (for simplicity we only consider real-valued inputs). The filter $\mathbf{H}$ correspondingly is now given by a matrix $\mathbf{H}$ [178]. To stay within the analogy of DSP, this filter needs to be applied within the spectral domain of the graph. Such a spectral representation of the graph can be received through the so-called graph Laplacian. As given in [19], the Laplacian operator provides a harmonic analysis on the graph. Since the Laplacian is a very important concept in GNN theory [135], we will give a brief overview on the concept.

The graph Laplacian is the discrete version of the continuous Laplacian or Laplace operator [180]. The Laplace operator $\Delta = \nabla \cdot \nabla$ is defined as the divergence of the gradient of a scalar function in Euclidean space. It is therefore given by the sum of second partial derivatives for each independent variable in Cartesian coordinates [180]. This definition provides an intuition about the meaning of the Laplacian applied onto a functional relation. For a function $f(x)$ evaluated at point x_0, the Laplacian $\Delta f(x)$ represents the size of the average deviation of $f(x)$ from $f(x_0)$ in the infinitesimal space around $f(x_0)$. The same holds true for the discrete version of the Laplace operator, the graph Laplacian $\mathbf{L}$, with respect to the neighboring vertices of the graph. It provides a natural link between the discrete representation of the graph and the continuous representation of its signal on the manifold [109]. In unnormalized form, it is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ corresponds to the degree matrix of $\mathbf{A}$ defined as $\mathbf{D}_{ii} = \sum_{j=0}^{N-1} \mathbf{A}_{ij}$ and $\mathbf{D}_{ij} = 0 \; \forall \; i \neq j$ [221]. The graph Laplacian encodes the information propagation process within a graph structure. This can be nicely demonstrated with a simple example, namely the heat diffusion process as e.g. addressed in [1]. We assume a graph with vertices $\mathbf{V}$ and edges $\mathbf{E}$ with individual heat values $x_i \in \mathbf{X}$. Newton's law of cooling states that the heat transfer from location A to B, in this case from vertex v_i to v_j, is proportional to the temperature difference $x_i - x_j$. For vertex v_i and the given adjacency matrix $\mathbf{A}$, the temperature change dx_i/dt with thermal conductivity k is defined by:

$$\frac{dx_i}{dt} = -k \sum_j \mathbf{A}_{ij}(x_i - x_j) = -kx_i \left(\sum_j \mathbf{A}_{ij} - \sum_j \mathbf{A}_{ij} x_j \right) = \\ - kx_i \left(\mathbf{D}_{ii} - \sum_j \mathbf{A}_{ij} x_j \right) = -k \sum_j (\delta_{ij} \mathbf{D}_{ii} - \mathbf{A}_{ij}) x_j = -k \sum_j (\mathbf{L}_{ij}) x_j \quad (3.4)$$

This formulation can be transferred into the matrix formulation $d\mathbf{X}/dt = -k\mathbf{L}\mathbf{X}$, clearly showing the direct relation between the heat diffusion process and the graph Laplacian.

After this short excursion, we will introduce the role of the Laplacian within graph signal processing. Here, the normalized graph Laplacian $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is used as a mathematical representation of the graph as e.g. introduced in [29]. Both the unnormalized and normalized graph Laplacian are constraint to undirected graphs, but have the nice property of being real symmetric positive semidefinite. For these types of matrices, it is possible to define an eigenspace $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U} = [u_0, u_1, ..., u_{N-1}]$ is the matrix of orthonormal eigenvectors ($\mathbf{U}^T\mathbf{U} = \mathbf{I}_N$) and $\mathbf{\Lambda} = \text{diag}(\vec{\lambda})$ is the diagonal matrix of eigenvalues with $\mathbf{\Lambda}_{ii} = \lambda_i$. We can therefore define a graph Fourier transformation F and inverse graph Fourier transformation F^{-1} on the graph signal $\mathbf{X}$ in Eq. 3.5 [221]:

$$F(\mathbf{X}) = \hat{\mathbf{X}} = \mathbf{U}^T\mathbf{X}\,, \qquad F^{-1}(\hat{\mathbf{X}}) = \mathbf{X} = \mathbf{U}\hat{\mathbf{X}} \quad (3.5)$$

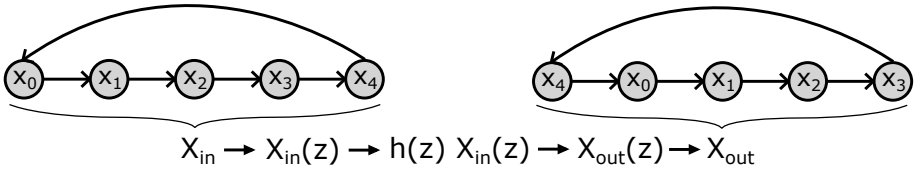


Fig. 3.2. Visualization of the z-transform and resulting shift of a time signal $\mathbf{X}_{\text{in}}$ with five samples x_t using the filter $h(z) = z^{-1}$ as described in Eq. 3.7. Due to the periodic extension of the time signal, the value of the fifth sample transfers to the first sample.

In this context, the term graph Fourier transformation is chosen as a transfer of the concept of Fourier transformation into the graph signal processing domain. It is used for the transformation of the graph signal into the spectral domain of the graph [214]. Throughout the thesis, the term Fourier transformation is used correspondingly. Following the analogy of Eq. 3.3, we can define a filter $\mathbf{H} \in \mathbb{R}^N$ and apply it to the graph via multiplication in the spectral domain:

$$\mathbf{X} * \mathbf{h} = F^{-1}(F(\mathbf{X}) \odot F(\mathbf{H})) = \mathbf{U}(\mathbf{U}^T\mathbf{X} \odot \mathbf{U}^T\mathbf{H}) \tag{3.6}$$

This concept provides the basic building block of spectral CGNNs to encode a graph signal in a latent representation which is then processable for a prediction task. The Fourier and inverse Fourier transformations in the operation in Eq. 3.6 are however computationally very expensive ($\mathcal{O}(N^3)$), additionally the approach is unlocalized and bound to a given graph structure due to the fixed eigenspace representation [58]. A detailed explanation of these limitations as well as of the several approaches developed to overcome these limitations with respect to spectral graph convolutional networks will continue in Sec. 3.3 and Sec. 3.4.

3.1.4 Information progression in spatial graph domain

After giving a first introduction to the ideas of spectral based graph processing, we will now motivate the concept of spatial processing, the direct information progression in the graph domain. Eq. 3.2 provides us with the underlying idea of the DSP filtering process. Since this filtering process can result in a polynomial of a degree larger than $N-1$ (via the multiplication of z^{-t} within $\mathbf{H}(z)$), certain boundary conditions have to be applied like e.g. a periodic extension of the signal, resulting in $x_t = x_{t \bmod N}$. As a result, the filter $h(z) = z^{-t'}$ produces a simple shift in the time signal in the size of t'. The shift filter $h_{\text{shift}} = z^{-1}$ itself constitutes the basic component of DSP, since every filter $\mathbf{H}$ is build from a combination of these shifts [178]. Applying the shift onto the input signal yields Eq. 3.7, also visualized in Fig. 3.2:

$$\mathbf{X}_{\text{out}}(z) = h_{\text{shift}} \cdot \mathbf{X}_{\text{in}}(z) = \sum_{t=0}^{N-1} x_t z^{-t-1} \rightarrow \mathbf{X}_{\text{out}} = (x_{N-1}, x_0, x_1, ..., x_{N-2}) \tag{3.7}$$

We can therefore receive a shift in the time domain by applying a filter $h(z) = z^{-t'}$ to the time signal in spectral domain. More complex filtering operations on the signal are based on combinations of these basic shift operations.

We now again rely on the definition of a graph $G(\mathbf{V}, \mathbf{E}, \mathbf{W})$ with feature representation $\mathbf{X}$ and adjacency matrix $\mathbf{A}$ given in Sec. 3.1.3. As before, every sample of the time signal now

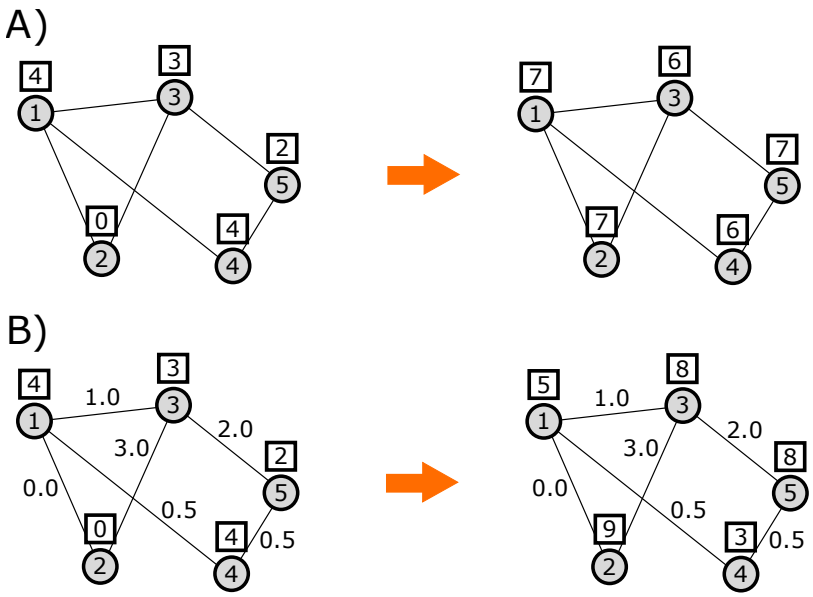


Fig. 3.3. Visualization of a simple graph structure G with five vertices. Every vertex contains a scalar feature $x_i \in \mathbf{X}$ (values in squares) and is connected to other vertices via an edge $e_{ij} \in \mathbf{E}$. **A)** For binary edges, the adjacency is described within the adjacency matrix **A** in Eq. 3.10. The edges encode the information how the graph signal will progress through the graph. One progression step is applied by multiplying the graph signal **X** with the adjacency matrix **A**. The scalar features (the signal) progress to neighboring vertices and accumulate, forming the new graph signal on the right. For e.g. vertex 4, the scalar features of vertices 1 and 5 with value 4 and 2 form the new vertex feature value 6. **B)** The same process is repeated for a weighted graph and weighted adjacency matrix **W** given in Eq. 3.10.

corresponds to a vertex in the graph. The chronological ordering of the samples is transferred into an ordering label for each vertex. As shown in Fig. 3.2, the shift performed in the spectral domain of the graph is directly interpretable in the spatial domain. Every vertex value is transferred to the next vertex. Since for DSP, the basic building block is defined by a shift, we can use this for the definition of a filter based on the adjacency matrix **A** [178]. We receive Eq. 3.8 for the matrix multiplication:

$$\mathbf{X}_{\text{out}} = \mathbf{A} \cdot \mathbf{X}_{\text{in}} \tag{3.8}$$

As an example we consider a signal with $N = 5$. In matrix form, the shift z^{-1} can be therefore represented as an adjacency matrix $\mathbf{A}_{\text{shift}}$:

$$\mathbf{A}_{\text{shift}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{3.9}$$

The matrix $\mathbf{A}_{\text{shift}}$ encodes the cyclic transfer of one vertex signal to the next vertex. It is intuitive to deduce that this representation allows also more complex signal progression within the graph structure.

To understand this idea, we consider a simple example of a graph structure in Fig. 3.3 similarly described in [221]. Our graph consists of five individual cases v_i, each with a representation x_i and edges e_{ij} between the vertices v_i and v_j. For simplicity, the feature representation on

every vertex contains only one scalar value (in the squares). The connections between the vertices can be encoded within the adjacency matrix **A**. As explained in Sec. 3.1.3, within this matrix every column / row corresponds to one vertex v. If a connection between two vertices v_i and v_j exists, the edge e_{ij} is entered in $\mathbf{A}_{ij}$. For the simple example given in Fig. 3.3, we consider undirected edges, so a connection from v_i to v_j automatically exists vice versa, resulting in a symmetric matrix. The corresponding binary edges (either 0 or 1) for the graph shown in Fig. 3.3A are given within adjacency matrix **A** in Eq. 3.10. In case the edges are also weighted with a weight w_{ij} to distinguish the impact that different vertices have on the update of a certain vertex (see Fig. 3.3B), the weights are stored within a weighted adjacency matrix **W**:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \quad \mathbf{W} = \begin{pmatrix} 0.0 & 0.0 & 1.0 & 0.5 & 0.0 \\ 0.0 & 0.0 & 3.0 & 0.0 & 0.0 \\ 1.0 & 3.0 & 0.0 & 0.0 & 2.0 \\ 0.5 & 0.0 & 0.0 & 0.0 & 0.5 \\ 0.0 & 0.0 & 2.0 & 0.5 & 0.0 \end{pmatrix} \tag{3.10}$$

The adjacency matrices allow us an easy mathematical description of the transition process of the signal within the graph. A signal shift based on the adjacency matrices **A** or **W** can be defined as Eq. 3.11 [221]:

$$\mathbf{X}' = \mathbf{A} \cdot \mathbf{X} \quad \text{or} \quad \mathbf{X}' = \mathbf{W} \cdot \mathbf{X} \tag{3.11}$$

Matrix $\mathbf{X}'$ represents the graph signal after one transition through the graph. For **A**, every connected vertex is considered equally for this transition step, since the adjacency matrix only consists of binary connections between each vertex. A weighted adjacency matrix **W** will additionally give a different impact to different vertices of the update step. The resulting graph signal can be seen in Fig. 3.3B. Every weight provides an information of the importance of the corresponding neighboring vertex to the representation update of a connected vertex. The information progression within the graph can therefore be shaped for flexibly. The update of e.g vertex 3 is more dependent on vertex 5 than it is on vertex 1 due to the different weighting of the corresponding edges.

From the above relation, it becomes clear that the application of the adjacency matrix **A** corresponds to the shift in time-based DSG. It is important to notice that here the adjacency matrix is applied directly within the spatial domain of the graph and is valid for directed and undirected graphs in comparison to the Laplacian, which operates in the spectral domain of undirected graphs only. Both approaches have their advantages and disadvantages, which will be discussed in the following sections.

The shown processing based on the graph adjacency matrix or the graph Laplacian as described in Sec. 3.1.3 relies on a given graph structure and predefined weights to determine the graph signal update process. Especially for larger and more complicated graph structures, it is therefore intuitive that a learning process to automatically optimize the feature processing can be of high advantage. The introduction of machine learning concepts, specifically neural networks, was therefore a logical next step. As described in Sec. 3.1.2, this step was performed via recurrent graph neural networks, a form of neural network designed for the processing of temporal and sequential data, which we will explain in the following section.

3.2 Beginning of GNNs: Recurrent GNNs

3.2.1 From RNNs to RGNNs

Since Graph Signal Processing constitutes a generalization of Digital Signal Processing which is used on time series data, it is intuitive that the first attempts of a neural network processing arbitrary graph structures performed by Sperduti et al. [219] and later Gori et al. [90] was inspired by the concept of recurrent neural networks (RNNs) or more generally recursive neural networks. A recurrent neural network contains a feedback loop to process time series data in a way that considers the information of older states for the prediction of newer ones and was first proposed in [200]. We consider a time signal $\mathbf{X} = (x_0, x_1, ..., X_{N-1})$ with scalar values x_t. Note that $\mathbf{X}$ here represents one time sample. For one feedback loop, we receive the input x_t which is the signal at time step t. Additionally, we receive the hidden representation h_t of time step t, which encodes the previously processed information of the signals x_0 to x_{t-1}. The hidden representation for the time step $t+1$ is defined using this hidden representation as demonstrated within Eq. 3.12:

$$h_{t+1} = \sigma(w_x x_t + w_{h,1} h_t + b_n) \tag{3.12}$$

where w_x, w_h and b_n are learnable parameters and σ is a non-linear activation function. h_0 is initialized with the value zero. h_{t+1} provides the new hidden representation of time step $t+1$. Finally, the prediction result for a time step t is given by Eq. 3.13:

$$y_t = w_y h_t + b_y \tag{3.13}$$

This processing is visualized in Fig. 3.4. For a time series of five data points, every point x_t and the previous hidden representation h_t is used for the calculation of the hidden representation h_{t+1} within an RNN layer. As discussed before, through the visualization of the input time sequence in Fig. 3.4 it becomes clear that it is possible to interpret the time sequence as a directed graph where every data point x_t is connected only to its directly neighboring predecessor x_{t-1} and successor x_{t+1} (similar to the shift operator in Eq. 3.9 introduced in Sec. 3.1.4 without the boundary condition). h_2 for example can be seen as the updated representation of x_1 after an interaction with the updated representation h_1 of x_0. Neglecting the output calculation of y_t for the moment, the RNN processing of the time sequence data points therefore can be interpreted as the calculation of a state update of a graph [89]. Additionally to the graph signal $\mathbf{X}$ every vertex is provided with a vertex label $l \in \mathbf{L}_X$. This label does not correspond to a ground truth value for a learning task, but instead can carry additional information in e.g. a positional graph. The equation shown by [89] describing the update is given by Eq. 3.14, a two-layer MLP is used for the loop network within the RNN:

$$x_{t+1} = w_{x,2} \cdot \sigma(w_{x,1} \cdot x_t + w_l \cdot l_{t+1} + b_{x,1}) + b_{x,2} \tag{3.14}$$

The update process performed in Eq. 3.14 resembles the signal progression described in Eq. 3.11 in Sec. 3.1.4 (within A, every vertex would only have an adjacency to its preceding vertex), but is now trainable for a target y_t via learnable weights (Eq. 3.15).

$$y_t = w_{y,2} \cdot \sigma(w_{y,1} \cdot x_t + b_{y,1}) + b_{y,1} \tag{3.15}$$

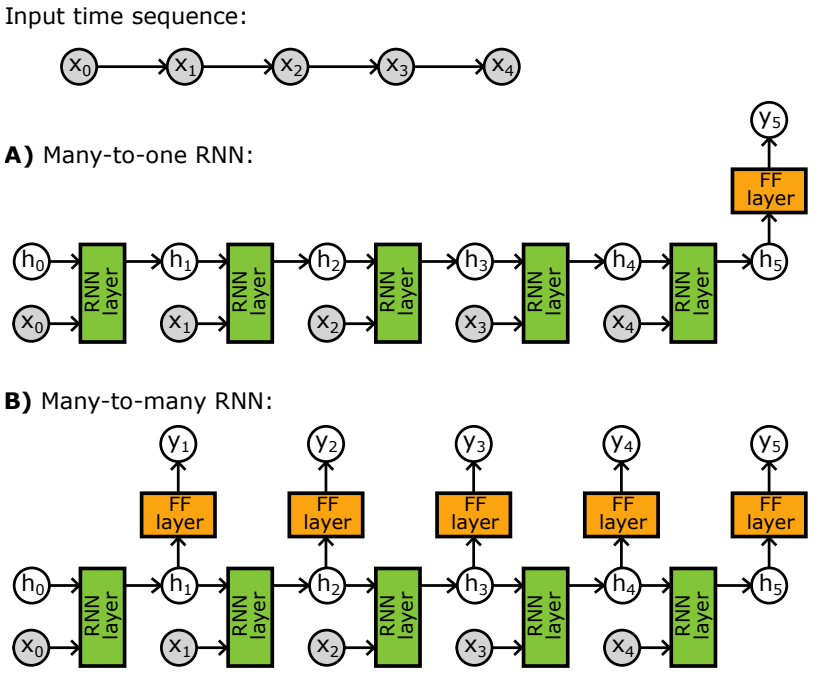


Fig. 3.4. Schematic description of a recurrent neural network (RNN), two examples for the processing of a time-series consisting of five data points are given: **Many-to-one**: The full time series results in one output of the network. **Many-to-many**: The hidden representation for every time step results in an output, providing one output per data point in the time series.

This target can be either representative for the whole graph (Fig. 3.4 many-to-one, output of one vertex represents the graph) or for each vertex (Fig. 3.4 many-to-many). Correspondingly the learning task focuses on a graph as a whole or on each vertex individually.

[89] showed that it is possible to broaden the applicability of the above approach using recursive neural networks. A recursive neural network is a generalization of a RNN which can be applied to more complex data structures like certain graphs types. The recursive network can be unfolded on the graph, resulting in an encoding network which provides the state transition functions for the data points within the graph. The state of a vertex is updated based on its dependent vertices. For the graph given in Fig. 3.5A, vertex 1 is dependent on vertices 2, 3 and 5, vertex 3 is dependent on vertices 4 and 5. Vertices 2, 4 and 5 have no dependencies. The graph is therefore interpretable as a tree with roots and children. Since the recursive neural network needs to receive the same input dimensions, the largest number of children vertices is chosen, in this case 3 (vertex 1). Then, for every other vertex with missing children, a so-called frontier state x_0 is assumed. Following these definitions, Eq. 3.14 can be modified to Eq. 3.16 [89]. Since the graph is not interpretable as a linear time series anymore, we replace x_t with x_i, where i indexes every data point within the graph structure:

$$x_i = w_{x,2} \cdot \sigma(\sum_{k=1}^{p} w_{x,1,k} x_{\mathrm{ch}_k[i]} + w_l \cdot l_i + b_{x,1}) + b_{x,2} \tag{3.16}$$

where $p = \max(\mathrm{od}[v])$, $\mathrm{od}[v]$ is the number of childs present for a vertex v and $x_{\mathrm{ch}_k[v]}$ is the kth child of vertex v. We set $x_{\mathrm{ch}_k[v]} = x_0$ if the kth child is missing for vertex v. In the example in Fig. 3.5A, $p = 3$. This processing allows a graph signal progression within the provided graph, which is learnable via the given weights w and b. However, the structure of the update process

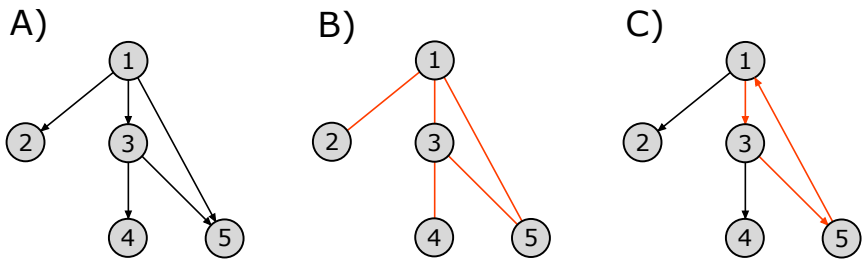


Fig. 3.5. **A)** Simple graph structure, which can be processed by the recursive neural network defined in Eq. 3.16. **B)** and **C)** Even slight modifications to the graph in form of undirected edges or cycles within the graph prevent the applicability of Eq. 3.16, corresponding edges are shown in red. Therefore, the recursive neural network approach is not sustainable for the general processing of graph information.

directly reveals that the approach is only usable for directed and acyclic graphs. Otherwise, the children dependency is not clearly defined. After slight modifications, the undirected graph in Fig. 3.5B and cyclic graph in Fig. 3.5C for example are not processable via Eq. 3.16.

3.2.2 Transition to arbitrary graph structures

To overcome the previously described limitations and create a neural network which is capable to process arbitrary graph structures, [90] therefore modified this approach. Following the definition of a graph G consistent of vertices $v_i \in \mathbf{V}$ and edges $\mathbf{E}$, every vertex obtains a vector $x_i \in \mathbb{R}^s$, called state, which is connected to other vertices via the edges $\mathbf{E}$, representing their relationship. For every vertex v_i, a neighborhood $N(i)$ is defined. Now, the state x_i is defined by a parametric transition function, which expresses the dependence of the vertices on their neighborhood $N(i)$. Following the intuition of Eq. 3.16, we can treat every neighboring vertex v_j with $e_{ij} \in \mathbf{E}$ as an individually contributing data source. The contributions of every vertex are added up to receive the new representation x_i for vertex v_i. The update of x_i is therefore performed via Eq. 3.17 [90]:

$$x_i = f_w(l_i, x_{N(i)}, l_{N(i)}) = \sum_{j \in N(i)} h_w(l_i, x_j, l_j) \tag{3.17}$$

The function h_w can be represented via a standard neural network, where $l_i \in \mathbf{L}_X$ is the label of node v_n and $l_{N(i)}$ are the labels of the neighboring nodes respectively. The neighborhood itself can be encoded within an adjacency matrix introduced in e.g. Eq. 3.10 in Sec. 3.1.3. For the graph given in Fig. 3.5B, we e.g. can define the following adjacency matrix A and resulting neighborhood constellations $N(i)$:

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \qquad \begin{aligned} N(1) &= (2,3,5) \\ N(2) &= (1) \\ N(3) &= (1,4,5) \\ N(4) &= (3) \\ N(5) &= (1,3) \end{aligned} \tag{3.18}$$

We can receive all states simultaneously with the composition of f_w, namely $\mathbf{X} = F_w(\mathbf{X}, \mathbf{L}_X)$. Since the neighborhood is defined individually for each vertex, undirected and cyclic graphs

do not provide an issue to the update step. Modifying the recursive neural network approach, [90] therefore allowed the processing of arbitrary graph structures. Using $F_w(\mathbf{X}, \mathbf{L}_X)$, the update is performed simultaneously for all states, resulting in new representations for each state. This process can now be iteratively repeated until a convergence is reached. Then, for every state an output vector $\vec{y}_i$ defined by the output function g_w can be calculated as shown in 3.19, which incorporates the respective state x_i as well as vertex labels l_i.

$$\vec{y}_i = g_w(x_i, l_i) \tag{3.19}$$

The composition of g_w is given by $\mathbf{Y} = G_w(\mathbf{X}, \mathbf{L}_X)$. Both equation F_w and G_w provide a parametric function $\phi_w(G, v_i) = \vec{y}_i$ which constitutes their graph neural network approach. For the learning set $(G_k, v_k, \vec{y}_k)|1 \leq k \leq p$, [90] gets triples (G_k, v_k, y_k) with graph G_k, vertices v_k and targets y_k. Examples are chemical compound detection where vertices represent the atoms and edges the chemical bonds (graph-focused) or web page classification where every vertex corresponds to a page connected via hyperlinks representing the edges (vertex-focused) [204]. The classification of a patients health status as shown in Fig. 3.1 can be considered as a vertex-focused problem as well. The quadratic loss for training is:

$$e_w = \sum_{k=1}^{p} (y_k - \phi_w(G_k, v_k))^2 \tag{3.20}$$

Using their network, [90] performed evaluations on multiple toy examples. Within an experiment which focus on the localization of all vertices belonging to a sub-graph, they were able to show that their network considers also the topology of the system for the classification of the vertices and not only the label information, outperforming corresponding standard neural networks.

Within a work published by Scarselli et al. [204], the approach by [90] was again elaborated in more detail. It especially focused on a more detailed description of necessary mathematical constraints. In order to e.g. assure a unique solution for the convergence problem of X, the functional relation F_w has to be a contraction mapping according to the Banach theorem [204], which shrinks the distance between two vertices after their processing. In order to achieve this, [204] showed mathematically that the loss term in Eq. 3.20 needs to be modified to the following term:

$$e_w = \sum_{i=1}^{p} (y_i - \phi_w(G_i, v_i))^2 + \beta L(\left|\frac{\partial F_w}{\partial x}\right|), \quad L(y) = \begin{cases} (y-\mu)^2 & \text{if } y > \mu,\ \mu \in (0,1) \\ 0 & \text{otherwise} \end{cases} \tag{3.21}$$

This modification assures the convergence of the training process. The graph neural network as defined by [90] and [204] falls into the class of recurrence-based graph neural networks (RCNNs) [248]. This group of approaches defines the developmental starting phase of GNNs. Since the approach requires an iterative convergence of the states within F_w before every calculation of the outcome in Eq. 3.19, the defined algorithm is computationally very expensive. Several works in the past years have tried to overcome these problems as we will continue to discuss in the coming sections. A few other works on RGNNs have however been performed, as e.g. in [53, 78, 145].

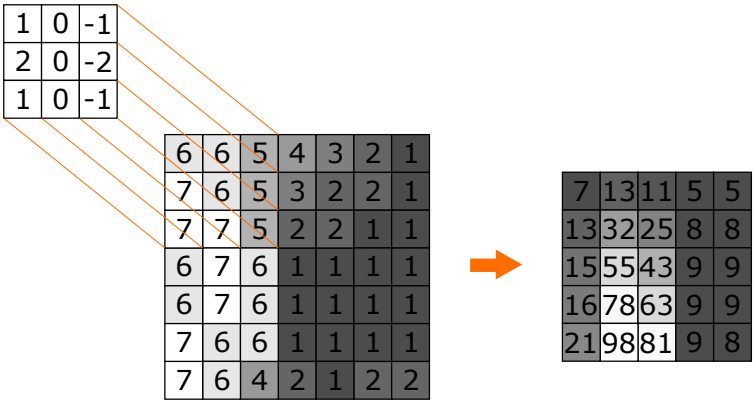


Fig. 3.6. Example of Sobel filter [68] for vertical edge detection: The filter is designed to produce large absolute output values, when a strong intensity difference (an edge) occurs within the input from left to right. The result of the stepwise application of the filter onto the image is shown on the right. The largest values are created within the area where the strongest intensity gradient occurs within the input.

3.3 Spectral convolution on graphs

3.3.1 Convolutional neural networks

The essence of every machine learning based model approach lies in a processing of the provided data for discovering relations within the given input domain with respect to the learning task. While for numerical features, a processing can be straightforward (e.g. with a simple regression approach) more complex data like images, volumes or even video data can be significantly more challenging due to the normally substantially larger amount of information as well as the geometrical relation within the data input itself. If the task of the machine learning algorithm is e.g. to distinguish between cats and dogs within an image, a simple consideration of every individual pixel intensity value independently from the surrounding values will not lead to any reliable classification, since the animal within the image only manifests itself as a relational constellation between the pixels. To effectively perform the classification task, it is therefore crucial to extract these relations from the imaging data. Considering the structural nature of the discussed graph data, it is therefore intuitive that approaches used for imaging data could prove effective on graph data as well.

One of the most effective ways to achieve the feature extraction is by using filters designed to recognize certain patterns within the image [87]. A filter is a tensor which is applied to a given section of the imaging data via element-wise multiplication with the given pixel intensity values and subsequent addition of the resulting outcomes, providing a filter value for the analyzed image part. Fig. 3.6 shows an example of this extraction process. The shown Sobel filter [68] is designed to detect a vertical edge within the image part by providing a large outcome value when a strong intensity shift occurs in horizontal direction. The resulting value of the filter is entered into a so called feature map, which preserves the geometrical relation between the extracted values. It is now possible to again apply a filter on the given feature maps to extract more high-level features like a corner (two edges connected with an angle to each other). This process can be repeated to detect complex relationships within the imaging data (like a dog). A proper design of these filters is however very challenging

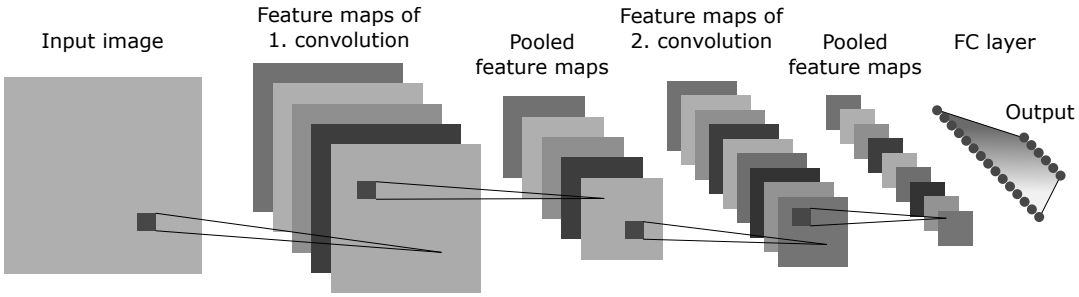


Fig. 3.7. Schematic description of a convolutional neural network [138]. An input image is processed by multiple learned filters within the first convolutional layer. The resulting feature maps are pooled to half their size in each dimension to condense the information within the feature maps. This process is repeated in a second convolutional and pooling layer. Finally, the resulting output is converted into a feature vector and processed by a fully connected layer (FC layer). The output of the fully connected layer is e.g. trained for a classification task.

when performed manually. Especially for the extraction of more high-level features, design choices become very complicated. With the rise of machine and deep learning in recent year's, the usage of trainable neural networks for the analysis and extraction of information using automatically learned filters therefore has become substantially more feasible and attractive. The ability of especially convolutional neural networks [138] to learn highly non-linear filters on complex data structures is favorable and very successfully used in various applications inside and outside the medical domain to process the complex relationships between individual representations by automatically learning and optimizing the used filters. Within Fig. 3.7, a typical setup for a convolutional neural network is shown. One layer of the network consists of a number of learnable filters with a given size, which are randomly initialized. These filters are applied to the input in a specified pattern and result in a feature map for every filter. For e.g. a 3×3 filter **F** and $M \times N$ input **X**, every entry of the resulting feature map **X'** is defined by [87]:

$$\mathbf{X'}_{m,n} = \sum_{u=m,v=n}^{m+2,n+2} \mathbf{X}_{u,v} * \mathbf{F}_{u-m+1,v-n+1} \tag{3.22}$$

with $m \in [1, M-2]$ and $n \in [1, N-2]$. This process has been performed for a small example image of size 4×4 and example filter in Fig. 3.8A. We can see directly, that the feature map shrinks in size compared to the original input since the filter cannot be applied onto the edges of the input. We can however apply measures like padding, where the edges of the image are artificially extended with e.g. zeros to allow for processing of the edge values. In general, when no padding or special striding procedure is used, the size of an image with dimensions $M \times N$ and filter with dimensions $U \times V$, the resulting feature map has the dimensions $(M-(U-1)) \times (N-(V-1))$.

Repeating the described process layer after layer results in a condensed version of the input which is processable via e.g. a fully connected layer to achieve the final prediction. Using the previously described concept of backpropagation from the used loss function, the filters **F** automatically learn to extract the most important features for the prediction task.

We will now show a direct relation of this convolution to the processing of a given graph structure within a data domain. Instead of describing the image in Fig. 3.8A via the pixel position with two indices (m, n), it is possible to encode it in form of a graph, where every pixel

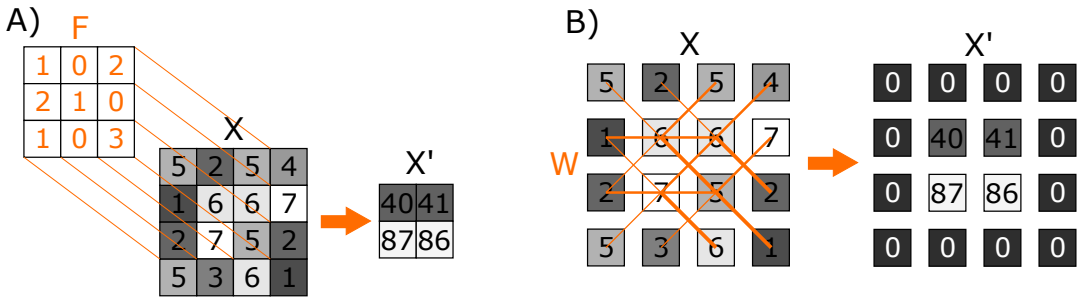


Fig. 3.8. Comparison of convolutional process on example image: A) A standard convolutional filter is applied to the image. Since no padding is used the convolution results in a feature map of reduced size. For every entry in the feature map, the filter is element-wise multiplied with the corresponding pixel intensity values of the image, the resulting values are summed up. B) A weighted graph structure using each pixel as a vertex is applied onto the image. Every vertex has the pixel's intensity value as its scalar graph signal. The directed weight graph itself represents the weight of a pixel for another pixel within the convolution (line thickness corresponds to strongest connection in either direction). Multiplying the graph signal matrix X with the graph weight matrix W therefore directly performs the full convolution.

is described as a node v [28, 248]. Following the definition in Sec. 3.1.3, we can therefore define the image as a feature matrix $\mathbf{X}$, where each row refers to the feature representation x_i on node v_i. Since for the image, every node only contains an intensity value, every x_i is given by a scalar. Following a row-wise pattern, we can therefore describe the 4×4 example image with the following 16 features:

$$\mathbf{X}^T = \begin{pmatrix} 5 & 2 & 5 & 4 & 1 & 6 & 6 & 7 & 2 & 7 & 5 & 2 & 5 & 3 & 6 & 1 \end{pmatrix} \tag{3.23}$$

The graph consists of 16 nodes correspondingly. Now, we define that an edge $e_{ij} \in \mathbf{E}$ from node v_i to v_j represents the usage of the pixel value on node v_j for the convolution process of v_i, where the weight of edge e_{ij} corresponds to the importance of v_j for the convolution. As described in Sec. 3.1.3, we can encode this process in an adjacency matrix $\mathbf{A}$ and a weight matrix $\mathbf{W}$ for the graph. Following the 3×3 convolutional filter, a pixel is considered adjacent, when it is connect vertically, horizontally or diagonally to another pixel. Since this results in an undirected graph, the adjacency matrix $\mathbf{A}$ is symmetric. The weigh matrix $\mathbf{W}$ encodes the weight which a pixel has in the convoluted signal for another pixel in the processed graph signal $\mathbf{X}'$. For the 12 edge pixels no convolution will take place, resulting in rows filled with zeros for these pixels. We therefore define matrices $\mathbf{A}$ and $\mathbf{W}$ to perform the convolution step as given in Eq. 3.24:

$$\mathbf{A} = \begin{pmatrix}
1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\
1&1&1&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&1&1&1&0&1&1&1&0&0&0&0&0&0&0&0\\
0&0&1&1&0&0&1&1&0&0&0&0&0&0&0&0\\
1&1&0&0&1&1&0&0&1&1&0&0&0&0&0&0\\
1&1&1&0&1&1&1&0&1&1&1&0&0&0&0&0\\
0&1&1&1&0&1&1&1&0&1&1&1&0&0&0&0\\
0&0&1&1&0&0&1&1&0&0&1&1&0&0&0&0\\
0&0&0&0&1&1&0&0&1&1&0&0&1&1&0&0\\
0&0&0&0&1&1&1&0&1&1&1&0&1&1&1&0\\
0&0&0&0&0&1&1&1&0&1&1&1&0&1&1&1\\
0&0&0&0&0&0&1&1&0&0&1&1&0&0&1&1\\
0&0&0&0&0&0&0&0&1&1&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&1&1&1&0&1&1&1&0\\
0&0&0&0&0&0&0&0&0&1&1&1&0&1&1&1\\
0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1
\end{pmatrix} \quad \mathbf{W} = \begin{pmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&2&0&2&1&0&0&1&0&3&0&0&0&0&0\\
0&1&0&2&0&2&1&0&0&1&0&3&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&2&0&2&1&0&0&1&0&3&0\\
0&0&0&0&0&1&0&2&0&2&1&0&0&1&0&3\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{pmatrix} \tag{3.24}$$

The convolution described as in Eq. 3.22, can therefore equally be performed by Eq. 3.25. Here, the adjacency matrix **A** is not considered, since **W** already contains the required information of **A**, also visible in Eq. 3.24.

$$\mathbf{X}' = \mathbf{W} \cdot \mathbf{X} \tag{3.25}$$

The resulting graph signal $\mathbf{X}'$ represents the progression of the graph signal **X** as defined by **A** and **W**, see also Fig. 3.8B. The new graph signal shows the successful convolution when compared to Fig. 3.8A:

$$\mathbf{X}'^{T} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 40 & 41 & 0 & 0 & 87 & 86 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.26}$$

Only four nodes have a remaining signal larger zero, directly corresponding to the 2×2 feature map in Fig. 3.8. We have therefore successfully encoded the convolutional process within the graph.

Although the shown process is in principal functional, the standard convolution is the preferred choice on image data since the matrices **W** and **A** show the overhead resulting from the necessity of graph construction. This argumentation however only holds as long as we operate in an Euclidean domain like the image data above. As soon as we enter a non-Euclidean domain like a social graph or a patient cohort graph with arbitrary connections operating on unstructered connections, the standard convolution of e.g. Eq. 3.22 is not defined anymore [28]. The process of convolution can become arbitrary complex using a varying number of vertices for the update of the graph signal of a specific vertex. In this situation, while the image convolution is not applicable anymore the graph convolution is still fully functioning. This analogy shows the high value and versatility of the graph convolutional process. So far, we have however only shown the processing of a graph signal using a fixed graph structure. The high advantage of deep convolutional networks lies in their ability to automatically learn filters for the extraction of the most important features. It is therefore necessary to find a formulation for a graph convolution learning process. Within the following sections, beginning with Sec. 3.3.2, we will provide a detailed overview about the history of graph convolution development. Here, we will start with the evolution of spectral approaches, before moving over to spatial approaches in Sec. 3.4.

3.3.2 From eigenvectors to Chebyshev networks

As described before, the idea of a graph convolutional network is to learn a representation of a data distribution not only by its features, but also by an underlying structure within the data. This underlying structure is defined by the graph, which consists of edges and vertices, where the vertices are connected by the edges. This concept allows a variety of applications, ranging from the analysis of social networks and population cohorts over non-Euclidean surface mapping to physical and chemical interaction analysis [28]. Within the medical field, CGNNs have been used e.g. for the analysis of brain connectivity and surfaces (e.g. [122, 132, 133]), a classification of neurological diseases (e.g. [126, 133, 156, 185]), mammogram analysis for the detection of abnormal lesions (e.g. [67]) and drug-drug interactions (e.g. [8]) (within Sec. 3.6 a detailed overview is provided).

Within Sec. 3.1.3, we have given a first introduction of graph signal processing and convolutional filtering in spectral domain on a graph structure using the graph Laplacian. After providing more detail on the general convolution process, we will now continue in this direction. As introduced before, for our graph setup we are assuming a number N of vertices $\mathbf{V}$ with feature representations $\mathbf{X} \in \mathbb{R}^{N \times M}$, where M is the length of the feature representations, and corresponding edges $\mathbf{E}$ with weights $\mathbf{W}$. Every vertex $v_i \in \mathbf{V}$ holds its corresponding feature representation $x_i \in \mathbf{X}$. The vertices are connected by a graph $G(\mathbf{V}, \mathbf{E}, \mathbf{W})$, where an edge $e_{ij} \in \mathbf{E}$ with weight $w_{ij} \in \mathbf{W}$ corresponds to a connection between vertices v_i and v_j. The existing edges $\mathbf{E}$ are therefore represented in the weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$. To take the data structure provided by the graph into account for the processing of signal $\mathbf{X}$ that is placed on the graph, a convolution can be applied on the graph structure. As was demonstrated in Sec. 3.3.1, this operation is relatable to the convolution on an image. For images, every pixel corresponds to a vertex with a scalar as a signal ($M = 1$). As shown in Fig. 3.8, a 3×3 convolutional filter could be represented by an adjacency matrix that only connects directly touching pixels. The signals of neighboring vertices are now aggregated to update the representation of the central vertex, similar to the convolutional filters of a CNN.

A convolution operation on a graph is more complex due to its non-Euclidean structure. It can be performed either by using a spatial approach, working on the vertex signal directly or a spectral approach in the frequency domain of the signal. By definition, the spatial approach will be localized (similar to a filter kernel of a CNN), an important property for an effective convolutional operation, but can have problems of matching local neighborhoods. We will address the spatial convolution operation later in Sec. 3.4. Since the convolution operation is defined as a linear operator that can be diagonalized in the eigenbasis of the Fourier domain, the spectral approach is provided by transferring the graph signal into Fourier space [221]. As described before, here the eigenbasis is represented by the eigenvectors of the graph Laplacian $\mathbf{L}$. To recap, the graph Laplacian can be seen as a representation of the information propagation process in the graph and is defined as the matrix subtraction of the (weighted) adjacency matrix $\mathbf{W}$ from the degree matrix $\mathbf{D}$ of the graph. The degree matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ is defined as $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$, which provides us with the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W} \in \mathbb{R}^{N \times N}$. Normalizing $\mathbf{L}$ results in $\hat{\mathbf{L}} = \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ with $\mathbf{I}_n$ as the identity matrix. The graph Laplacian enables a mathematical description of the graph structure (for details on the Laplacian we refer the reader again to Sec. 3.1.3). For the convolution process, we now want to transfer our graph signal into frequency space. To receive the underlying signal frequencies, we have to calculate its Fourier transformation. Since $\hat{\mathbf{L}}$ is a real symmetric positive semidefinite matrix, it provides a complete set of eigenvectors $\{u_l\}_{l=0}^{N-1}$ and corresponding eigenvalues $\{\lambda_l\}_{l=0}^{N-1}$. It therefore follows that $\hat{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ with $\mathbf{U} = [u_0, ..., u_{N-1}] \in \mathbb{R}^{N \times N}$ and $\mathbf{\Lambda} = \text{diag}([\lambda_0, ..., \lambda_{N-1}]) \in \mathbb{R}^{N \times N}$. We can therefore define the Fourier transform of the graph signal $\mathbf{X} \in \mathbb{R}^{N \times N}$ as $\hat{\mathbf{X}} = \mathbf{U}^T X \in \mathbb{R}^{N \times N}$ as well as the reverse transform $\mathbf{X} = \mathbf{U}\hat{\mathbf{X}}$ [221].

As stated above, the convolution process corresponds to a linear transformation in the frequency domain of the signal. Although this theoretical assumption and the non-Euclidean nature of the graph prohibits the direct application of a translation operator in the spatial domain, the operator can therefore be defined in the Fourier space. [29] proposed this idea as one of the first works to create a CNN-type architecture on graphs. They define this translation

operator as a filter g_θ. The convolution operation in Fourier space can therefore be described as described in Eq. 3.27 (see also Eq. 3.6):

$$g_\theta * \mathbf{X} = \mathbf{U} g_\theta \mathbf{U}^T \mathbf{X} \tag{3.27}$$

With respect to Eq. 3.6, g_θ can be seen as a spatial filter converted into Fourier domain. This concept would allow learning the translation operator g_θ, applying it to the signal $\mathbf{X}$ in Fourier space and transferring the result back to the vertex domain. From $\hat{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ and 3.27, it further becomes clear that g_θ can be treated as a function of the Laplacian eigenvalues lambda $g_\theta(\mathbf{\Lambda}) = \text{diag}(\vec{\theta})$. This corresponds to a non-parametric filter, where $\vec{\theta} \in \mathbb{R}^N$ represents the Fourier coefficients. Since the process requires the multiplication with the eigenvector matrix $\mathbf{U}$, it however has an unfavorable complexity of $\mathcal{O}(N^2)$. Additionally, a non parametric filter $g_\theta(\mathbf{\Lambda})$ has a learning complexity of $\mathcal{O}(N)$ and also does not provide localization [58].

To tackle these issues, [96] therefore proposed an effective polynomial approximation of the filtering process. A filter designed as an approximation in the form of Eq. 3.28 solves both problems of a non-parametric filter.

$$g_\theta(\mathbf{\Lambda}) = \sum_{k=0}^{K-1} \theta_k \mathbf{\Lambda}^k \quad \text{with} \quad \vec{\theta} \in \mathbb{R}^K \tag{3.28}$$

First, the complexity of the parameter learning is reduced to $\mathcal{O}(K)$ with a constant K. Second, the approach has to lead to a filter localization. To analyze this prerequisite, it is necessary to understand which contribution every vertex in the graph provides to a vertex of interest v_i, when a convolution using g_θ is performed. Since $\hat{\mathbf{L}}^k = (\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T)^k = \mathbf{U}\mathbf{\Lambda}^k\mathbf{U}^T$, $g_\theta(\mathbf{\Lambda})$ can be written as a function of $g_\theta(\hat{\mathbf{L}})$. We can determine the contribution of a vertex v_j within the filter $g_\theta(\mathbf{\Lambda})$ at vertex v_i by performing a convolution with the Kronecker delta function $\delta_i \in \mathbb{R}^N$ [58].

$$(g_\theta(\hat{\mathbf{L}})\delta_i)_j = (g_\theta(\hat{\mathbf{L}}))_{ij} = \sum_{k=0}^{K-1} \theta_k (\hat{\mathbf{L}}^k)_{ij} \tag{3.29}$$

Since it was shown in [96] that $(\hat{\mathbf{L}}^k)_{ij} = 0$ if the path length d_G within the graph is longer than k, Eq. 3.29 automatically results in a K-localization for a filter updating vertex v_i. We can specify:

$$(\hat{\mathbf{L}}^k)_{ij} = 0, \quad \text{if } d_G(i,j) > k \tag{3.30}$$

Therefore, a polynomial of degree $K-1$ containing $(\hat{\mathbf{L}}^{K-1})_{ij}$ updates vertex v_i based on a $K-1$-hop neighborhood and is localized within this neighborhood.

The provided solution solves the mentioned problems of the filter, but still requires a Fourier transformation shown in Eq. 3.27 of complexity $\mathcal{O}(N^2)$. To remove this problem, [96] proposes to parameterize $g_\theta(\hat{\mathbf{L}})$ as a polynomial recursively computable from $\hat{\mathbf{L}}$ directly. The concept of Chebyshev polynomials allows for this description, demonstrated in [58]. They are defined recursively as:

$$T_k(x) = \begin{cases} 1, & \text{if } k = 0 \\ x, & \text{if } k = 1 \\ 2xT_{k-1}(x) - T_{k-2}(x), & \text{if } k > 1 \end{cases} \tag{3.31}$$

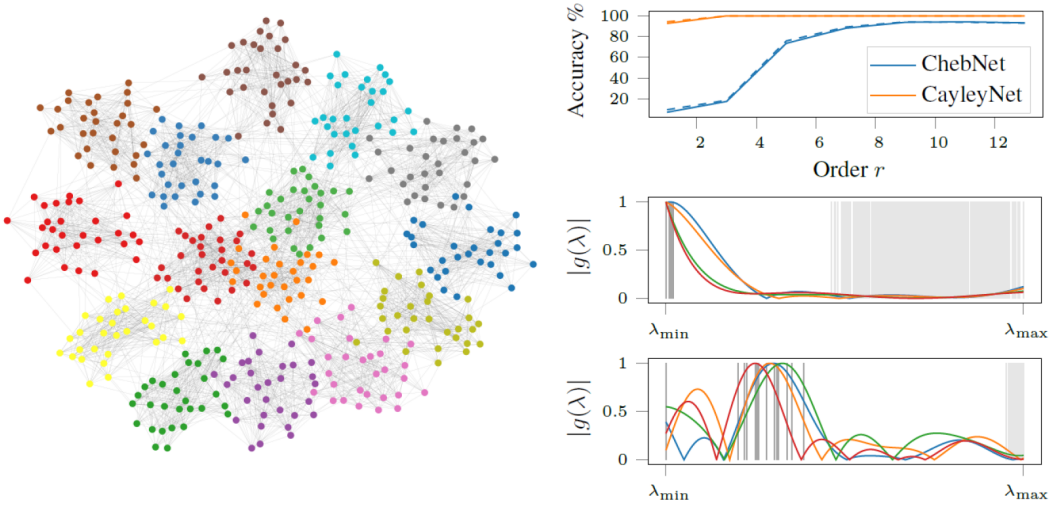


Fig. 3.9. Left: Synthetic 15 community graph used in [139]. Right top: Community detection accuracy for increasing polynomial order for ChebNet and CayleyNet as demonstrated in [139], Right middle and bottom: normalized responses of four filters for ChebNet and CayleyNet respectively. It becomes visible that the filters of the CayleyNet approach of [139] can focus on the low frequency band with most information present, requiring a smaller polynomial order than ChebNet for a strong performance. *Note:* Image from [139]

We can now describe the filter using a Chebyshev polynomial as described in Eq. 3.31 up to the K^{th} order:

$$g_{\theta'}(\mathbf{\Lambda}) \approx \sum_{k=0}^{K} \theta'_k T_k(\bar{\mathbf{\Lambda}}) \tag{3.32}$$

where $\bar{\mathbf{\Lambda}} = \frac{2}{\lambda_{max}}\mathbf{\Lambda} - \mathbf{I}_N$ describes a scaling of the eigenvalues into the area of $[-1, 1]$, λ_{max} denotes the largest eigenvalue of $\hat{\mathbf{L}}$ and $\vec{\theta'} \in \mathbb{R}^K$ is a vector of Chebyshev coefficients. We can therefore define the convolution of the graph signal $\mathbf{X}$ with filter g'_θ as:

$$g_{\theta'} * \mathbf{X} \approx \sum_{k=0}^{K} \theta'_k T_k(\bar{\mathbf{L}})\mathbf{X}\,, \tag{3.33}$$

where $\bar{\mathbf{L}} = \frac{2}{\lambda_{max}}\hat{\mathbf{L}} - \mathbf{I}_N$. The transfer of $\mathbf{\Lambda}$ to $\hat{\mathbf{L}}$ within Eq. 3.33 follows from the above mentioned relation $\hat{\mathbf{L}}^k = (\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T)^k = \mathbf{U}\mathbf{\Lambda}^k\mathbf{U}^T$. For a sparse Laplacian this operation has a complexity of $\mathcal{O}(K|E|) \ll \mathcal{O}(N^2)$. We can define a graph convolutional layer using Eq. 3.33 as shown in Eq. 3.34:

$$\mathbf{X}' = \sigma\left(\sum_{k=0}^{K} \theta'_k T_k(\bar{\mathbf{L}})\mathbf{X}\right)\,, \tag{3.34}$$

where σ is a non-linear activation function and $\mathbf{X}'$ is the new feature representation after the filter has been applied. In their work, [58] used a simple scalar per vertex as graph signal, resulting in $\mathbf{X} \in \mathbb{R}^N$. When using a feature vector for every vertex as described above, it is possible to transfer the learned parameters θ_k into learnable parameter matrices $\mathbf{\Theta}_k \in \mathbb{R}^{M\times F}$, where M corresponds to the input and F to the output dimension of the feature representation processed by Eq. 3.34.

As nice extension to the work performed by [58] was performed by [139] in 2019 with the introduction of CayleyNet, which we would like to briefly mention. Following a similar

idea than ChebNet, this concept relies on Cayley polynomials. The group highlighted that despite all the advantages of Chebyshev filters the usage of polynomials makes it hard to create narrow-band filters due to the required high order of the polynomial and the resulting non-locality. This is addressed via the usage of the Cayley transformation for the defintion of filters. The Cayley transformation $C(x)$ is given by Eq. 3.35:

$$C(x) = \frac{x-i}{x+i} \tag{3.35}$$

where i denotes the imaginary unit. This transformation is applied onto the Laplacian as $C(h\mathbf{L})$. The parameter h plays a crucial role, since it effectively allows the zooming of the function into different parts of the spectrum, which is mapped onto the complex half-circle via the transformation (see Fig. 3.9). The computation of a Cayley filter $\mathbf{G}$ on a signal $\mathbf{X}$ is correspondingly defined as:

$$\mathbf{G} * \mathbf{X} = c_0\mathbf{X} + \sum_j (c_j C^j(h\mathbf{L}) + \bar{c_j} C^{-j}(h\mathbf{L}))\mathbf{X} \tag{3.36}$$

using the fact that for the real value $\mathrm{Re}(z)$ of a complex value z we can calculate $2\mathrm{Re}(z) = z + \bar{z}$, where $\bar{z}$ is the conjugate of z. The calculation of the individual Cayley coefficients does not need to be performed precisely, instead any incorrections of the used Jacobi approximation method are corrected via the learned parameters [139].

3.3.3 Linear approximation for layer-wise graph convolution

The approach by [58] shown in Sec. 3.3.2 allows an effective formulation of a graph convolutional network. Especially for wide vertex degree distributions present in e.g. social networks and knowledge graphs, these networks are however prone to overfitting on local neighborhood structures. To solve this problem, [128] proposed a simplification of Eq. 3.33 by using only the first two terms of the Chebyshev polynomial corresponding to $K = 1$, effectively receiving a representation linear in $\hat{\mathbf{L}}$. From Eq. 3.31, it follows that $T_0(\bar{\mathbf{L}}) = 1$ and $T_0(\bar{\mathbf{L}}) = \hat{\mathbf{L}}$. Inserting this into Eq. 3.33 leads to $g_{\theta'} * \mathbf{X} \approx \theta'_0\mathbf{X} + \theta'_1\bar{\mathbf{L}}\mathbf{X}$. By fixing $\lambda_{max} = 2$ under the assumption that the network parameters will learn the rescaling, this further simplifies to:

$$g_{\theta'} * \mathbf{X} \approx \theta'_0\mathbf{X} + \theta'_1(\hat{\mathbf{L}} - I_N)\mathbf{X} = \theta'_0\mathbf{X} - \theta'_1\mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}\mathbf{X} \tag{3.37}$$

Every time, this operation is applied to the graph signal $\mathbf{X}$, the 1-hop neighborhood of every vertex is considered for the convolution. Stacking K layers and applying them consecutively therefore effectively results in a convolution under the consideration of the K-hop neighborhood, since e.g. the second layer receives a signal that already has incorporated the first neighborhood of every vertex into its representation. Applying the layer to the vertex v_i in the second layer under consideration of a neighboring vertex v_j therefore contains the 1-hop neighborhood of v_j, which is the 2-hop neighborhood of v_i. To further minimize the risk of overfitting, [128] defines only one parameter $\theta = \theta'_0 = -\theta'_1$. Further a renormalization is applied to prevent exploding/vanishing gradients (since the eigenvalues of $\hat{\mathbf{L}}$ are in the range $[0, 2]$) by transferring $\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ to $\bar{\mathbf{D}}^{-\frac{1}{2}}\bar{\mathbf{W}}\bar{\mathbf{D}}^{-\frac{1}{2}}$, where $\bar{\mathbf{W}} = \mathbf{W} + \mathbf{I}_N$ and $\bar{\mathbf{D}}_{ii} = \sum_j \bar{\mathbf{W}}_{ij}$. As explained in Sec. 3.3.2, from the beginning we assumed $\mathbf{X} \in \mathbb{R}^{N \times M}$ to contain a feature vector for every vertex instead of a scalar. We can therefore introduce a linear transforma-

tion $\mathbf{\Theta} \in \mathbb{R}^{M \times F}$ as learnable parameter matrix instead of a single parameter θ. Again, F corresponds to the new dimension of the filtered feature representation for every vertex. Our convolutional layer is therefore approximated as:

$$\mathbf{X}' = \sigma(\bar{\mathbf{D}}^{-\frac{1}{2}}\bar{\mathbf{W}}\bar{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\mathbf{\Theta}) \,, \tag{3.38}$$

where σ is a non-linear activation function and $\mathbf{X}'$ is the new graph signal representation after the filter. A consecutive application of the layer defined in Eq. 3.38 allows the definition of a deep graph convolutional network.

3.3.4 Further spectral approaches

Especially the approach of [58] to use Chebychev polynomials in the constructions of the graph convolutional filters was widely recognized and reused in several works. Next to the demonstrated improvements further spectral methods were developed in the field we want to briefly reflect on.

Graph usage enhancement

As described for the development of CayleyNet, the balance between the incorporation of local and global information on the graph structure is important for a proper learning behavior. This was addressed as well in a work by [268], which proposed a dual graph convolutional network (DGCN) consisting of two network branches for the simultaneous consideration of local and global information. They achieve this by adding a second graph concept in the form of a positive pointwise mutual information (PPMI) matrix based on a random walk next to the standard adjacency matrix. The outputs of these two network parts are compared against each other within an unsupervised regularization function which aims to minimize the difference between the two representations. It is interesting to highlight that the weights of both networks were shared which led to a major performance improvement compared to a separated approach. The group of [268] interpreted this as an indication that without weight sharing both networks learn representations too different from each other.

A major drawback of spectral methods leveraging the analogy of graph Laplacian eigenvectors and the Fourier domain is their limitation to undirected graph structures. The group of [167] introduced MotifNet, a spectral GCNN applicable to directed graph structures. This is achieved via small subgraph structures, called motifs, which are introduced into the prediction process and were previously proposed by [20]. Via a reformulation of the graph weights, a motif Laplacian is designed which is symmetric and whose underlying concept acts anisotropically with preferred directions along the motif. This approach is introduced into a polynomial approximation similar to that of [58] with certain approximations to prevent unfeasible computation time.

Another very important aspect of graph neural networks is the impact of their graph structure on performance, as demonstrated in [184]. Within [269], the group demonstrated the effectiveness of small perturbations on the graph structure, which led to a significant reduction in prediction performance. A year later, the group therefore introduced a method to verify certifiable robustness of the graph model to perturbations [23].

Within [18], the group designed a graph-based learning procedure to perform a reasoning about physical relations of objects. The state of each object is encoded within its vector representation, the corresponding relation is given via a graph structure. A work Finally, we would like to mention a recent work focusing on multi-label one shot learning [179]. Next to a nice overview of related work, a label correlation graph is designed to achieve improved predictions for unseen target groups.

Graph learning

The previously discussed approaches work under the assumption that the underlying graph structure is known. A work by [99] lies the ground work for the creation of graphs when no underlying graph structure is given. They propose two approaches, the first one being an unsupervised approach based on data similarity with respect to the Euclidean distance or the Z-score of representations within the initial feature space. Similarity is therefore determined based on the input space. The second proposal follows from a supervised strategy, where a standard feed-forward neural network is trained for a prediction task, after which first level features of the network are extracted under the assumption that similarity within these features corresponds to a representation relevant for the prediction task. The method therefore operates in the same direction as later proposed by [185] to include feature information into the graph creation process.

In 2018, [141] proposed Adaptive Graph Convolutional Neural Networks which allow an adaptive learning of the underlying graph structure. Different from approaches like [29] or [99], they highlighted the importance of flexible layer size to allow the undisturbed processing of the data. They argue that a modification of the graph structure to meet dimensionality requirements is not always legit. For datasets consisting of e.g. molecules, the coarsening of the graph structure is chemically not justifiable. They overcome this by designing an individual graph Laplacian for each sample in the dataset. Additionally, a residual graph Laplacian focusing on unused parts of the graph is learned and added to the initial graph representation.

The group of [242] proposed an interesting work towards the generation of more meaningful graphs with respect to the task at hand. They introduced a CGNN for the classification and segmentation of point clouds. Here, after every layer the graph connections are newly determined based on a k nearest neighbors approach. Like this, non-local information can be processed for the representation and proximity can be reinterpreted. The group shows that for later layers of the network, the proximity of two points in the graph shifts from similarity of geometrical position to similarity in semantic meaning.

Further, we want to highlight two more recent interesting works proposed by [51] and [123]. Within [51] an end-to-end approach for graph learning from the embedding space of input features is proposed. [123] on the other hand proposes Differentiable Graph Module (DGM) for an ideal prediction of edge probabilities for the downstream task. Both methods are inductive and work for new unseen cases as well. In [43], the group proposes the methodology for multi-label recognition within images and the learning of a correlation matrix to capture inter-class relationships. Three other interesting works for an improved understanding of graph creation as well as graph influence were performed by [71, 155, 206] using differentiable edge masking, causal inference and disentanglement.

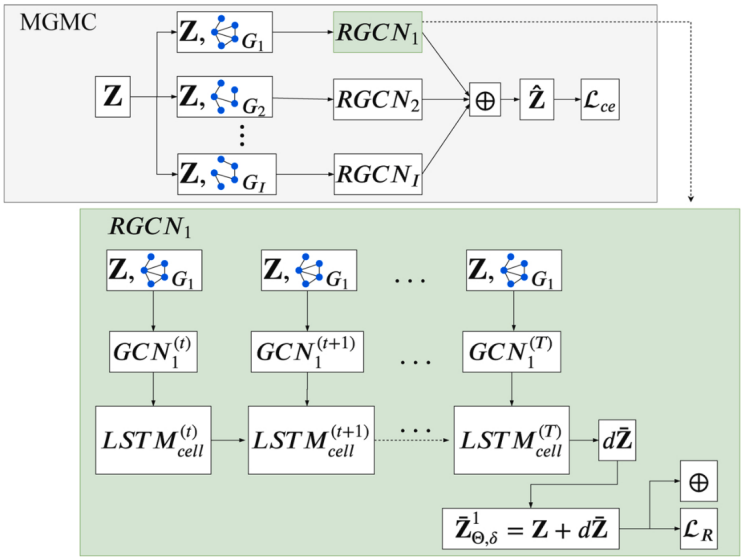


Fig. 3.10. Working principle of MGMC network [234]. Representations learned from individual graphs are combined within a self-attention approach. *Note:* Image from [234]

Geometric matrix completion

Another important area we would like to briefly highlight is the field of geometric matrix completion. The problem of missing information in provided data is a ubiquitous problem in the medical field. Effective approaches for the imputation of these missing values can potentially highly increase the usability of given datasets. Within [119], the problem of matrix imputation is tackled via a graph based approach. A matrix $\mathbf{M}$ with missing information contains observations $\mathbf{M}_{ij}$, $A_\Omega(\mathbf{M})$ defines the observed elements of $\mathbf{M}$ (Ω effectively masks all present elements). We are now predicting a matrix $\mathbf{X}$ and correspondingly try to achieve a maximum match between the features already known in $\mathbf{M}$, so namely $A_\Omega(\mathbf{X}) = A_\Omega(\mathbf{M})$, while at the same time the rank of the matrix $\mathbf{X}$ should be minimized (a low rank solution is targeted, since only a few factors are assumed to be influencing). [119] follow the argumentation of [39] that under certain conditions this minimization problem can be reformulated to a minimization of the trace norm $||\mathbf{X}||_*$ and the Frobenius norm defined as $||A_\Omega \odot (\mathbf{X} - \mathbf{M})||_F^2$. In the next step, they are interpreting the matrix rows and columns as undirected row and column graphs G_r and G_c, effectively describing every row and column as a vertex on a graph. Assuming a smoothness of the graph signal, closely connected vertices should obtain similar features. The idea is to use this behavior as a regularization to the minimization problem. Smoothness implies that for $e_{ij} \in \mathbf{E}_c$, the corresponding representations x_i and x_j should not deviate to strongly. In general form, this can written as $\sum_{i,j} w_{i,j}^c ||x_i - x_j||_2^2$. Interestingly, this can be expressed in terms of the graph Laplacian as $\text{tr}(\mathbf{X}\mathbf{L}\mathbf{X}^T) = ||\mathbf{X}||_D^2$, which is the definition of the graph Dirichlet semi-norm. This smoothness assumption can be used as a regularization term on the learning process. The overall minimization problem to impute the missing values of $\mathbf{X}$ therefore becomes:

$$\min \gamma||\mathbf{X}||_* + ||A_\Omega \odot (\mathbf{X} - \mathbf{M})||_F^2 + \frac{\gamma_r}{2}||\mathbf{X}||_{D,r}^2 + \frac{\gamma_c}{2}||\mathbf{X}||_{D,c}^2 \tag{3.39}$$

A very similar approach proposing the same regularization term is presented a year later by [191]. Based on these initial works, in 2017 [166] proposes the extension of this idea via the usage of graph convolutions within the learning process. To achieve this, they presented

RMGCNN. To reduce the degrees of freedom of the matrix completion problem, they propose a factorization of the matrix $\mathbf{X} = \mathbf{W}\mathbf{H}^T$. Given $\mathbf{X}$ with dimensions $m \times n$, the factorized version is given via $\mathbf{W} \in \mathbb{R}^{m \times r}$ and $\mathbf{H} \in \mathbb{R}^{n \times r}$ with $r << \min(m, n)$. This concept is used within a multi-graph convolutional approach resulting in an entry for each matrix element. Following this convolution process, an RNN in form of an LSTM [104] is introduced to perform the diffusion process of the information. [166] argue that this approach is nicely suitable since for a sparse matrix a convolutional approach would need a large amount of layers to perform the diffusion. The designed concept results in the following optimization problem:

$$l(\Theta, \sigma) = ||\mathbf{W}^{(T)}_{\theta_r,\sigma}||^2_D + ||\mathbf{H}^{(T)}_{\theta_c,\sigma}||^2_D + \frac{\mu}{2}||A_\Omega \odot (\mathbf{W}^{(T)}_{\theta_r,\sigma}(\mathbf{H}^{(T)}_{\theta_c,\sigma})^T - \mathbf{Y})||^2_F \tag{3.40}$$

where θ_r and θ_c are the learnable parameters of the CGNN and σ of the LSTM. A year later, [255] introduced an approach CGMC which is fully based on CGNNs as well as a fully connected layer. The group of [21] addressed the same problem via a modified version of the Variational Graph Autoencoder (VGAE) introduced by [129], which will be explained in more detail in Sec. 3.5.2.

Finally, we would like to highlight an approach proposed by our group called Multigraph Geometric Matrix Completion (MGMC) [234]. This work expanded on [166] with the incentive to address the problem of graph influence. The underlying graph structure has a strong impact on the performance of graph-based approaches [184], which was also nicely shown in [269]. Inspired by [185], for medical tasks similarity between meta data from patients can be used for the graph creation. Instead of integrating the information of all meta features into one graph structure, within [234] we create a graph for every meta feature and perform a separate feature prediction based on the methodology proposed by [166]. Then a self attention mechanism is used to combine the representations of all features into one final representation (see Fig. 3.10). For the training process, we additionally combine the feature and corresponding label information into one matrix which allows us to add a label-based cross entropy loss to the overall loss function and perform an imputation and classification at the same time, effectively treating the classification as an imputation problem.

3.3.5 Requirements of spectral graph convolution

As explained in Sec. 3.3.2, the above defined spectral graph convolution as performed in [29] is given in the graph Fourier space of the graph signal. Since this frequency spectrum is depending on the full graph signal and sensitively depending on it, all data that should be processed by the learned filters has to be present during training time and already incorporated by the graph structure. Adding additional samples into the graph would change the eigenbasis of the graph Laplacian [231]. Since the learning process is indirectly depended on this representation, an inference on this data using the previously trained network would not be valid according to the underlying theory. The shown approach is therefore a transductive learning procedure that is only valid on the data it has trained on. This finding has strong implications:

As an advantage, the overall concept is nicely usable to perform a semi-supervised training for e.g. vertex classification. Here, every vertex corresponds to a data point with a representation x_i and an individual label y, but only a small amount of the vertices has underlying ground

truth information y_{GT}. In a standard supervised training procedure, where every data point is processed individually the additional training data without ground truth information would be unusable. Within the graph it is however possible that for a vertex v_i with representation x_i having ground truth information the feature representations from neighboring vertices v_j are aggregated within the resulting representation x_i. Since it is assumed that a meaningful graph is building clusters between the vertices that correspond to similar data points with respect to e.g. its label, this transferred information can be very helpful to stabilize the prediction especially for situations where the individual representation x_i of a vertex v_i is less representative for its class. Since the graph contains the test data that should be inferred later already in its structure and incorporates it into the eigenbasis this signal propagation is learned very effectively.

But the necessity for all data to be present in the graph during training time also results in several methodological drawbacks. Especially in the medical field, where it is required for a system to provide a diagnosis or treatment planning for new individual patients, the necessity of a retraining of the full network for new data points is time-consuming and infeasible if e.g. the calculation power is not provided or the decisions are time-critical.

Even if all data is present during training time and the application does not require a frequent inference on new data points, depending on the amount of considered vertices and the size of individual representations x_i, the approaches do not scale well and can run into memory problems quickly [94]. Since the network needs to train on the full amount of data, a batchwise training approach is not applicable which makes the processing of especially imaging or even volumetric data points nearly impossible for larger datasets.

One possibility to circumvent the transformation into the eigenbasis for spectral approaches are the above described approximations performed in e.g. [58]. Another way to tackle these drawbacks very effectively is the usage of spatial graph convolutional approaches that work on the graph signal directly. We will discuss the underlying idea of these approaches in detail in the next section, since they also form the basis of the work presented.

3.4 Spatial convolution on graphs

3.4.1 Inductive vertex embedding

To overcome the previously described limitations, we will now focus on inductive graph learning approaches. Inductive refers to the property of the trained network to be valid and applicable to new unseen data points, therefore enabling a normal inference process and batchwise training. Different from methodologies like [29] described in Sec. 3.3.2, the here discussed methodology will not dependent on the graph Laplacian and instead work on the feature representation directly. The focus of these approaches lies on a smart and effective aggregation of local neighborhood that is transferable for the full graph and even other graph structures following the same setup rules. It therefore strongly resembles the information aggregation process of a standard convolution on image data as described in Sec. 3.3.1. Here,

information from neighboring pixels is aggregated to receive a new representation for the image within a feature map.

One of the first works performed is this direction was published already in 2009 by [164] as also stated in [248]. Within their work, they introduced a neural network for graphs (NN4G) which allowed the consecutive application of layers for the processing of the graph structure, a major difference to the previously discussed RGNNs. In simple terms, NN4G sums up a node's neighboring information to receive the new representation of each vertex. Here, every vertex contains a label vector which essentially can be interpreted as the feature vector $\vec{x}_i$, containing attributes of the vertex. The first layer is distinguished from the following layers, since no neighborhood representation is given for the first layer as input.

$$\vec{h}_i^{(1)} = \sigma\left(\mathbf{W}^{(0)}\vec{x}_i\right) \quad , \quad \vec{h}_i^{(k)} = \sigma\left(\mathbf{W}^{(k)}\vec{x}_i + \sum_u^{k-1}\sum_{j\in N(i)} w_{ij}^{(k)}\vec{h}_j^{(k-1)}\right) \tag{3.41}$$

where $\vec{h}_i^{(k)}$ is the representation of vertex v_i after application of the k-th layer, $\mathbf{W}^{(k)}$ is the linear transformation applied to the feature representation and $w_{ij}^{(k)}$ is the weight between vertices v_i and v_j. [13] has expanded on this work with Contextual Graph Markow Models (CGMM). Within their work, they highlight the basis of their method in NN4G, but use a hidden Markov model for neighborhood incorporation. Different from standard hidden Markov models, a vertex state is determined by the frozen states of neighboring vertices from previous layers.

Different from these approaches, in 2016 two works were presented, namely Diffusion CNN (DCNN) [11] and PATCHY-SAN [172], also highlighted in [248]. As already demonstrated for heat diffusion in Eq. 3.4, DCNN interprets the graph processing as a diffusion process, where information is gradually transferred between nodes. To achieve this behavior, a transition probability matrix $\mathbf{P}$ is used. The representation after layer k is correspondingly achieved by an application of $\mathbf{P}^k$ within layer k. The effectiveness of diffusion approaches was also demonstrated in [81]. A work we will address later for STGNNs went into a similar direction [143]. The method PATCHY-SAN [172] developed in the same year as [11] takes a different approach and performs a neighborhood ordering based on the concept of Select-Assemble-Normalize (SAN): a fixed-length sequence of nodes from the graph is selected, a fixed-size neighborhood for each node is assembled and the extracted neighborhood is normalized. Then a neighborhood representation is learned from the resulting sequence of patches.

Another interesting approach for graph-based feature aggregation was proposed by [94]. Its underlying concept is relevant also for the understanding of its evolution provided by [231], a very important work for this thesis. Their network GraphSAGE focuses on the direct usage of the weighted adjaceny matrix $\mathbf{W}$. We again consider a number of vertices $\mathbf{V}$ with overlying graph signal $\mathbf{X}$ and connecting edges $\mathbf{E}$ with weights $\mathbf{W}$ within a graph $G(\mathbf{V}, \mathbf{E}, \mathbf{W})$. For simplicity, we assume that the edge weights are binary only allowing values 0 and 1. $\mathbf{W}$ therefore becomes an unweighted adjacency matrix, where entries $w_{ij} = 1$ corresponds to the existence of an edge e_{ij} between vertices v_i and v_j. First, the neighborhood of every data point is aggregated using an aggregator function Ω, which takes into consideration the neighboring vertices $N(i)$ of vertex v_i:

$$\vec{x}_{N(i)}^{(k)} = \Omega(\{\vec{x}_u^{(k-1)}, \forall u \in N(v)\}) \tag{3.42}$$

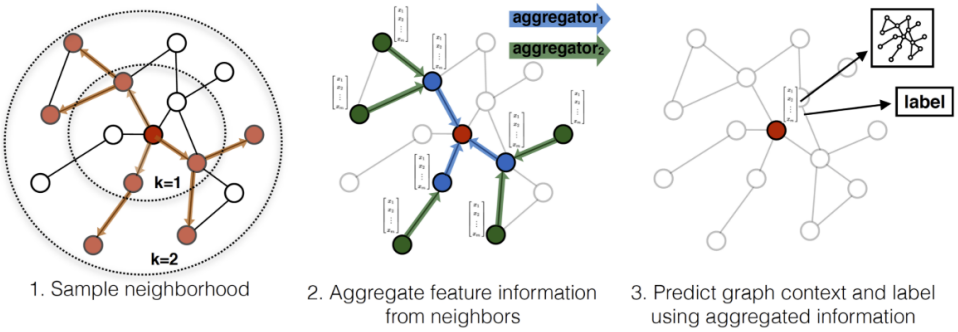


Fig. 3.11. Working principle of GraphSAGE network [94]. A neighborhood of fixed size is sampled for the processing within an aggregation function to perform the feature update. *Note:* Image from [94]

The aggregation function Ω is defined by [94] in three ways: as a simple mean aggregation, as a pooling operation and as a trainable LSTM. Depending on the size of the graph and the sparsity of the adjacency matrix, it might not be feasible to aggregate the complete neighborhood $N(i)$. Therefore, a fixed amount Z of neighboring vertices is aggregated for every data point. If the amount of vertices $v_j \in N(i)$ is larger than Z, a random subset is chosen, if it is smaller than Z, samples from $N(i)$ are randomly repeated until the necessary amount is reached. The resulting feature vector corresponds to an aggregated representation of the full considered neighborhood (see Fig. 3.11). It now can be incorporated into the representation of $\vec{x}_i^{(k-1)}$ using a learnable linear transformation $\mathbf{\Theta}$:

$$x_i^{(k)} = \sigma(\mathbf{\Theta}^{(k)} \cdot [x_i^{(k-1)} || x_{N(i)}^{(k)}]) \,, \tag{3.43}$$

where $[||]$ corresponds to the concatenation of the two feature vectors. The feature representation of vertex v_i does therefore not anymore correspond to the individual representation $\vec{x}_i$, but has incorporated information of the whole graph cohort (see Fig. 3.11). A prediction on $\vec{x}_i$ is therefore stabilized by the cohort it is present in. Apart from this work, the group also published an interesting review paper [95].

The group of [82] developed a more general framework for the description of CGNNs, namely message-passing neural networks (MPNN). They defined messages $m_i^{(k)}$ for the k$^{\text{th}}$ update step of representation $\vec{x}_i$ as:

$$m_i^{(k+1)} = \sum_{j \in N(i)} M_k(\vec{x}_i^{(k)}, \vec{x}_j^{(k)}, e_{ij}) \quad , \quad \vec{x}_i^{(k+1)} = U_k(\vec{x}_i^{(k)}, m_i^{(k+1)}) \tag{3.44}$$

where M_k and U_k are message passing and update functions respectively. Via a readout function R, an output from the last representation $\vec{x}_i^K$ can be achieved. They were able to show that among others the works [18, 29, 58, 69, 129, 145] are reproducible with their framework.

One of the major drawbacks of the described approaches is their focus on the immediate neighborhood, the transition process makes it unlikely that the vertices positioned further away are considered for the update process [248]. Within [228], the network PGC-DGCNN was developed which introduced adjacency matrices $\mathbf{S}^{(k)}$ encoding, if the shortest path between vertices v_i and v_j is of length k. Then, these matrices are applied as adjacency matrices within

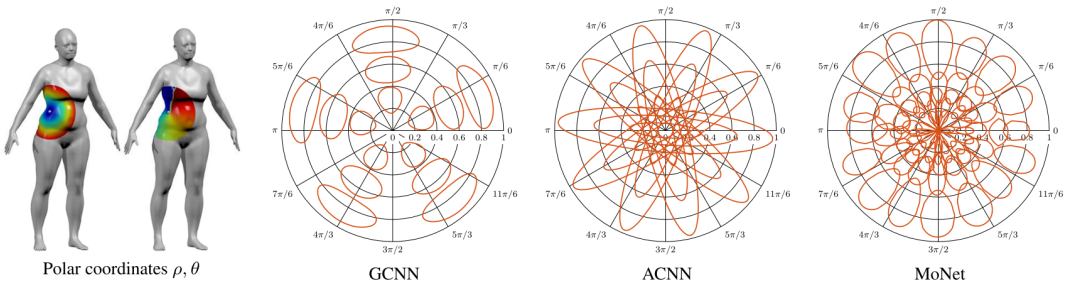


Fig. 3.12. Concept of pseudo-coordinates in MoNet [165]. Several other CGNN approaches can be realized within the developed framework. *Note:* Image from [165]

each layer k to gradually increase the considered neighborhood for the update step, while the representation after each layer is concatenated.

The group of [165] introduced another more general framework which also incorporates multiple other approaches like [25, 72, 128, 160], the network approach is called MoNet. They were tackling a similar issue as [228] a year later, namely the proper incorporation of the near and far graph neighborhood. This was achieved via the introduction of a pseudo-coordinates vector $\vec{u}(v_i, v_j)$ for every neighboring vertex v_j of v_i. On these, a weighting function $w_\theta(\vec{u}) = (w_1(\vec{u}), ..., w_J(\vec{u}))$ is applied, where θ contains learnable parameters (see Fig. 3.12). Via a corresponding definition of $\vec{u}(v_i, v_j)$ and $w_\theta(\vec{u})$ different methodologies could be reproduced. The filtering operation on the graph signal is then given by:

$$(f * g)(v_i) = \sum_{k=1}^{K} g_k \sum_{j \in N(i)} w_k(\vec{u}(v_i, v_j)) f(j) \tag{3.45}$$

where K is the dimension of the extracted patch. Here, for the weighting function, a Gaussian kernel is suggested. The work by [72] is building on top of this work. Here, a network SplineCNN is introduced which relies on the same concept of pseudo-coordinates.

Within [251], the group criticized that most CGNN approaches fall short for the distinction between certain different graph structures and propose Graph Isomorphism Network (GIN) to tackle this issue. The developed network achieves the same expressiveness as the Weisfeiler-Lehman graph isomorphism test for graph structure distinction (the test is demonstrated in e.g. [211]). Within their network, the update process of a vertex representation $\vec{x}_i$ is given by:

$$\vec{x}_i^{(k)} = \text{MLP}^{(k)}((1 + \epsilon^{(k)}) * \vec{x}_i^{(k-1)} + \sum_{j \in N(i)} \vec{x}_j^{(k-1)}) \tag{3.46}$$

where $\epsilon^{(k)}$ are learnable parameters. Also addressed in [248], [79] proposed a new approach of neighborhood selection. Their learnable graph convolutional layer (LGCL) is designed to rank the vertices according to their informativeness and chooses a fixed number correspondingly. Within another work presented by [265], a geometric encoding of the graph spatial information was used generated via a spatial path neural network (SPNN).

The shown concepts allow a purely spatial aggregation of neighborhood not based on a spectral eigenbasis and therefore not depended on a complete graph structure. It is therefore possible to train the model on a graph system and then transfer it for inference on new data

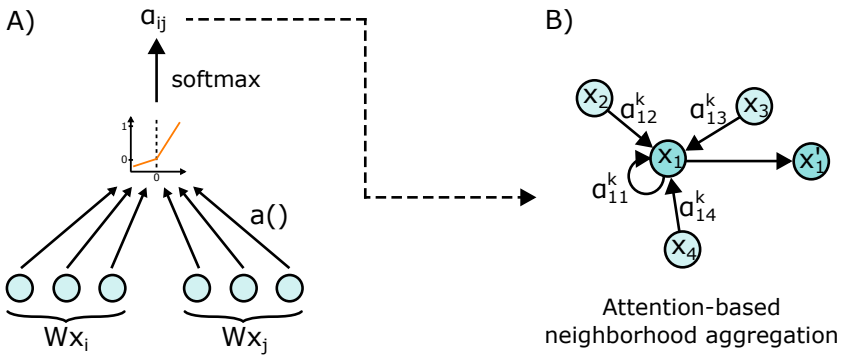


Fig. 3.13. Working principle of Graph Attention (GAT) network. A) Using the transformed feature representations of two neighboring vertices v_i and v_j, a shared linear attention function a() is applied on the concatenated representations. The softmax of the activated representation yields the attention coefficient α_{ij}. B) The representations of multiple neighboring vertices are aggregated based on their calculated attention coefficient with respect to the center vertex v_i.

that either gets included in the graph structure or even is part of a new graph. Within works like [94], the network treats every aggregated neighbor equally independent of the increased importance that certain neighbors might have for the update over others. As described, this issue was addressed by other works in the field. One effective way, which is also an important concept for our developed methodologies, is an attention approach proposed by [231].

3.4.2 Attention-based neighborhood aggregation

The graph structure itself may not provide direct information about the real importance of neighboring vertex representation $\vec{x}_j$ for a vertex v_i. In order to determine, how important a neighboring feature is for the update of $\vec{x}_i$ for vertex v_i, [231] proposed the usage of an attention mechanism. Instead of aggregating all neighboring vertices with equal importance, an attention weight is learned to incorporate every vertex based on this importance measure. To calculate this value, a shared linear transformation $\mathbf{W} \in \mathbb{R}^{M \times F}$, where F is the dimension of the transformed representation, is applied within an attention function a: $a(\mathbf{W}\vec{x}_i, \mathbf{W}\vec{x}_j)$. For simplicity, a is chosen to be a linear transformation $\vec{a} \in \mathbb{R}^{2F}$ applied to the concatenated representations:

$$e_{ij} = \vec{a}^T \cdot [\mathbf{W}\vec{x}_i || \mathbf{W}\vec{x}_j] \tag{3.47}$$

For every neighboring vertex v_j of v_i, the attention is calculated, the resulting coefficient can however vary largely in size depending on the individual representation. To perform a normalization over all calculations, a softmax is applied resulting in the normalized attention coefficients α_{ij}:

$$\alpha_{ij} = \frac{\exp(\sigma(\vec{a}^T \cdot [\mathbf{W}\vec{x}_i || \mathbf{W}\vec{x}_j]))}{\sum_{k \in N(i)} \exp(\sigma(\vec{a}^T \cdot [\mathbf{W}\vec{x}_i || \mathbf{W}\vec{x}_k]))} \tag{3.48}$$

These coefficients can now be used to create a weighted linear combination of all neighboring representations $\vec{x_j} \in N(i)$ for the update of $\vec{x_i}$ (see Fig. 3.13). To further stabilize this, [231] suggests the usage of multiple independent attention-based linear combinations, so-called attention heads.

$$\vec{x_i}' = ||_{h=1}^{H} \sigma \left(\sum_{j \in N(i)} \alpha_{ij}^{(h)} \mathbf{W}^{(h)} \vec{x_j} \right) , \tag{3.49}$$

where H denotes the number of heads, h is the index of the corresponding head. The resulting representations of each head are concatenated, indicated by $||$.

Next to this approach, there are a few more approaches using attention mechanisms [93, 151, 262]. GeniePath [151] introduces an adaptive path layer, which aims to learn both the breadth (which direct neighbors should be considered) and depth (which neighbors at further hops are of relevance) of the informative graph area around a vertex. This is achieved via two adaptive functions which incorporate an attention mechanism and learn an importance measure for both direct and distant neighbors. Here, the adaptive depth function is based on an LSTM. [93, 262] fall more into the category of Spatio-Temporal GNNs addressed in the next section. The approach Gated Attention Networks (GaAN) [262] adds a soft gate to the attention mechanism to filter out the contribution from an attention head capturing a useless representation. [93] introduces ASTGCN which extends the concept of spatial attention with a temporal attention function.

3.5 Further GNN methodologies

3.5.1 Spatio-temporal GNNs

The previously discussed approaches mostly assume that the input given by the feature vectors of the graph are static and do not contain relation information in itself. This is however not always correct. For e.g. a system of sensors of a water pipeline, not only the values of closely located sensors are relevant for each other (represented by the graph) but also the dynamic behavior and temporal development of the sensor outputs are important. To address this information dimension given by temporal data on the graph, Spatio-temporal GNNs have been developed [248]. This type of networks and underlying data structure is not addressed within this thesis, we still would like to give a short overview about the general underlying concepts.

STGNNs can be divided into RNN- and CNN-based approaches [248]. As we introduced previously in Sec. 3.2.1, RNNs are based on recurrent units which are applied onto the input data while taking previous states into account. Within [248], a nice analogy for a simple RNN approach is given. As defined in Eq. 3.12, we can perform the update step of a representation x_t at time step t with:

$$\mathbf{H}_{t+1} = \sigma(\mathbf{W}_x\mathbf{X}_t + \mathbf{W}_h\mathbf{H}_t + \vec{b}_n) \tag{3.50}$$

where $\mathbf{X}_t$ represents the feature representations, $\mathbf{W}_x$ corresponds to the linear transformation applied onto $\mathbf{X}_t$ and $\mathbf{W}_h$ to the transformation applied onto the hidden state $\mathbf{H}_t$ at time step t. These feature representations are now converted into a graph signal $\mathbf{X}_t$ on graph G. We can correspondingly define a convolution operation on the graph based on one of the above described approaches and redefine a spatio-temporal network:

$$\mathbf{H}_{t+1} = \sigma(\text{CGNN}(\mathbf{X}_t, \mathbf{A}; \mathbf{W}_x) + \text{CGNN}(\mathbf{H}_t, \mathbf{A}; \mathbf{W}_h) + \vec{b}_n) \tag{3.51}$$

where CGNN() corresponds to a graph convolutional layer and $\mathbf{A}$ is the underlying adjacency matrix of the graph structure.

Approaches based on this concept are given by e.g. [115, 143, 209, 262]. One of the first works was performed by [115]. Here, RNNs are applied to both vertices and edges. Since this would be computationally prohibiting if performed for all vertices and edges individually, the vertices and edges are split into semantic groups with the same vertex and edge factor respectively. The RNNs are processing the temporal information of the vertex and edge representations, additionally the vertex RNNs receive the output of the edge RNNs as input to incorporate the spatial information of the graph. A simpler approach for the incorporation of temporal information was performed by [209]. They relied on the concept of the previously introduced ChebNet [58] and essentially integrated the convolutional concept into an LSTM structure, therefore extending the representations processed within the LSTM to graph level. In the same year, [143] introduced a similar concept by integrating a diffusion convolution into gated recurrent units, a RNN concept introduced by [46].

A second group of STGNNs tackles the issues of the previously described approaches (time-consuming iterative propagation, exploding gradients) via the introduction of CNNs instead of RNNs for the processing of the temporal data dimension [248]. An approach by [258] called CGCN e.g. creates a spatio temporal block by stacking 1-D convolutional layers with ChebNet layers. [253] follows the same idea by stacking 1-D convolutions with a partition graph convolution (PGC) layers. Both approaches assume a given graph structure [248]. Graph WaveNet introduced by [249] in 2019 moves away from this assumption and integrates and adaptive adjacency matrix which is learned during the training process.

3.5.2 Graph Autoencoders

The concept of Graph Autoencoders (GAEs) follows the same idea than normal autoencoders, namely the encoding of its input into a latent representation and its corresponding decoding. Following a differentiation proposed by [248], GAEs are mainly used for the generation of new graph structures [24, 54, 57, 86, 136, 146, 157, 216, 257] and network embedding [40, 129, 182, 229, 232, 260]. We will not go into details for all these works, since the concept of GAEs is not addressed within the developed approaches of this thesis. Still, we will provide a brief overview about the general concepts of GAE approaches.

The first group of GAEs focuses on the generation of graph structures from learned latent representations. Within [248], GAE approaches for graph creation are grouped into sequential approaches where the graph is build up iteratively or approaches which offer a global setup. Deep generative model of graphs (DeepGMG) [146] for instance defines a joint distribution $p(G, \pi)$ over the graphs G with an edge ordering π. At the same time, $q(\pi|G)$ corresponds to any proposal distribution over permutations. The probability of a graph is then given via the expectation value E of $p(G, \pi)/q(\pi|G)$ over all these distributions [146]:

$$p(G) = E_{q(\pi|G)}\left[\frac{p(G, \pi)}{q(\pi|G)}\right] \tag{3.52}$$

DeepGMG now creates the new graph setup via consecutive decisions whether to add a node or an edge. On the other end, approaches like Graph variational autoencoder (GraphVAE) [216] create the graph as one output directly. A CGNN is used as encoder and an MLP as a decoder for the generation of a graph with adjacency matrix as well as vertex and edge

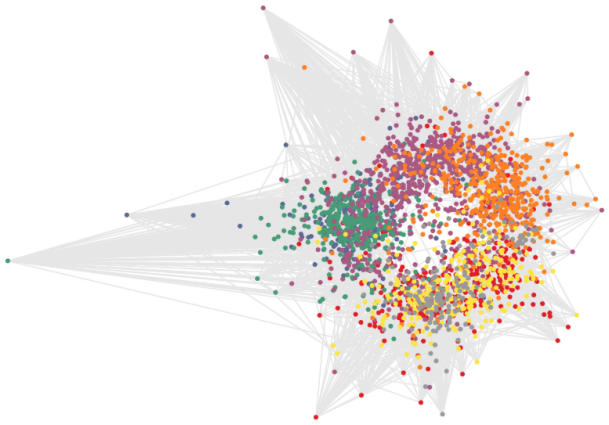


Fig. 3.14. Visualization of graph latent representation learned with an unsupervised VGAE [128] trained on the CORA dataset [161]. The colors represent the document class *Note:* Image from [128]

features. Further approaches incorporate validity constraints [157], reinforcement learning [57] and Wasserstein GANs [24] ([248]).

The second group of GAEs focuses on network embeddings. In general, the learning of a network embedding refers to the learning of a graph representation that keeps a low-dimensional vector representation of each vertex while preserving graph structural information [40, 248]. Within e.g. [237], a two loss approach is proposed for the training of structural deep network embedding (SDNE). The first loss function enforces that the network embedding of a vertex is similar to a network embedding of a neighboring vertex. The second loss function minimizes the difference between vertex input and its reconstructed input. This effectively preserves the first- and second-order proximity of the vertex. Both [237] and [40] use MLPs for the encoding process and only rely one the graph structural information that is encoding connectivity between a vertex v_i and v_j, not on the feature representations on the vertices, potentially losing important information [129, 248]. Graph Auto-Encoder (GAE*) and Graph Variational Auto-Encoder (VGAE) [129] change this via the incorporation of graph convolutional layers. The encoder is based on two consecutive GCN layers (given in Eq. 3.38), creating a latent representation (see Fig. 3.14). The decoder corresponds to an inner product decoder. They evaluate both methods on three different datasets for link prediction and demonstrate that the incorporation of feature information can strongly improve performance. Further works in the field are performed by [182, 229, 232, 260] ([248]). Interestingly, also GraphSAGE from [94] previously shown in Sec. 3.4.1 can be used for network embedding.

3.6 GNNs in medical

As already motivated in the introduction, the incorporation of additional relational knowledge can have a potential strong benefit for medical prediction tasks. For various methodological approaches, graph neural networks are very nicely suited for the processing of such relational information. Within this section, we will give an overview about different application areas in which relational knowledge incorporation via GCNNs was used to improve performance in the

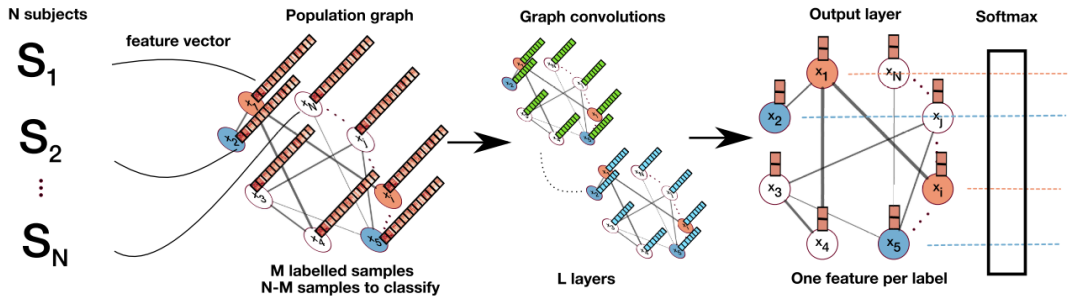


Fig. 3.15. Working principle of GCN network used within [185]. A population graph based on patient meta information is used for an improved feature processing. *Note:* Image from [185]

given prediction task. Here, the area of computer-aided disease diagnosis is the most relevant for our developed approaches, we are going to briefly highlight other areas as well.

Computer-aided disease diagnosis (CAD)

Computer-aided disease diagnosis (CAD) is an application field where the creation of meaningful patient inter-relation graphs can have a very beneficial effect. Information within the patient cohort can be processed jointly instead of relying on individual feature representations. Although graph-based neural network approaches were already developed for some years, one of the first applications of GNNs in this field and in the medical domain in general was performed by [185] in 2017 in the field of Autism and Alzheimer prediction. The basic idea of the group was to connect patients into cohorts based on the similarity of their meta information, in this case age and clinical sex, which was normally not directly involved in the prediction task. Within their work, the group used phenotypic patient information provided in the Autism Brain Imaging Data Exchange (ABIDE) [66] and Alzheimer's Disease Neuroimaging Initiative (ADNI) [113] dataset to form a graph structure between the patients with respect to similar values of this form of meta data. The feature vectors of each patient vertex of the graph are based on the extracted imaging information of functional MRI scans. Processing this graph setup with a standard GCN [128], resulted in an improved prediction performance (see Fig. 3.15). The intuition of the group was that information like age and clinical sex can be linked to disease specific aspects or the patient condition in general. These forms of relation have also been shown in other non-graph-based works as e.g. in [168, 187] for the prediction of age and sex from eye fundus and OCT imaging. In 2018, the group further expanded on [185] with a more detailed analysis of the impact of the graph structure on the performance [184]. They demonstrated that changes in the setup of the graph lead to strong variations in prediction performance, showing the importance of a proper choice of the graph structure [269]. This strong dependency on the graph setup was addressed by [5] in 2019, who proposed a bootstrapping method for multiple weakly trained GCNNs to improve the stability of the prediction on the ABIDE dataset. Since every GCNN was trained on a different randomized population graph, uninformative graph setups could be compensated to a certain extend. Another work was proposed by [88] for brain surface analysis using adaptive pooling. Further [67] proposed an interesting approach of mammogram analysis. A graph setup is used to mimic the radiologist's behavior to zoom into the most relevant areas. Another interesting work on the ADNI dataset was proposed by [156] for longitudinal Alzheimer's disease analysis. A year later, [245] proposed the usage of GNNs for the early detection of liver fibrosis.

Within works like [184, 185], the different meta information was combined in a criterion to provide one graph structure. In 2018, [124] used a GCN setup for an optimized incorporation of different modalities from the ADNI-based TADPOLE dataset [159]. Individual GCN layers were applied to two patient-graphs, one dependent on age similarity, the other on clinical sex similarity. Both graphs were processed in separate GCNs before combining their representations in a self-attention layer. A similar approach was proposed by us in [125], where each non-imaging feature was used for the creation of an individual graph structure. The learned representations were combined using an LSTM network. Another approach was proposed by us in [233], where different imaging-modalities were treated individually in the input step, additionally the presence of missing features was considered. A year earlier, [235] targeted the problem of missing features within the TADPOLE dataset via geometric matrix completion. The idea of this concept is to use the graph structure of the data domain to find the best imputation for each missing parameter value. Here, information from neighboring nodes leads to an improvement of this imputation and correspondingly to a better prediction of the transition from Mild Cognitive Impairment (MCI) to Alzheimer's disease. Within [126], another interesting methodological approach was evaluated by our group. Considering the previously discussed fact that different graph cohorts may require different neighborhood sizes and corresponding receptive fields for a proper processing, the network InceptionGCN was developed, which combines different neighborhood sizes within one processing unit. This allows for a individual adaption of the network to the graph structure requirements.

Functional brain network processing

While the previous approaches consider brain scans and corresponding extracted feature information as representation for one vertex within a patient graph, it is also possible to consider the functional brain network itself as a graph. For example [122], [132], [133] and [7] worked in this direction. The estimation of brain development can be a very important prediction task for e.g. infants born preterm. [122] proposed BrainNetCNN which leverages the topological locality of a structural brain network to predict the neurodevelopmental outcome. In general, it is an important but non-trivial task to perform a comparison between different brain connectivity networks in order to reveal patterns related to brain development and disease. Within [132], the group tackled this problem on the ABIDE dataset via a siamese graph convolutional network using a pairwise similarity global loss function. The general idea of the setup is to process graphs from two different brain scans within two GCNNs with shared weights and a subsequent metric learning part consisting of a dot product layer and fully connected layer to maximize the similarity estimate for two brain graphs from the same class / minimize for different classes. The two classes correspond to people with Autism spectrum disorder (ASD) and a healthy control group. The work was extended within [133] by applying it onto 2500 subjects of the UK biobank [36]. In general, it is of interest to detect patterns within functional brain networks that can be linked to certain phenotypic or disease-related differences. Using a ChebNet [58] approach as well as class activation mapping (CAM) [266], [7] showed that they could obtain an identification of regions of interest (ROIs) for a certain classification task. As an application example, they performed a prediction of clinical sex for the given brain connectivity networks. Within [142], an interpretable Brain Graph Neural Network (BrainGNN) was introduced for fMRI analysis. In the same year, a GNN called multi-scale triplet graph convolutional network (MMTGCN) was proposed for the processing of templates with different levels of definition applied on brain fMRI and dMRI scans for brain disorder detection [254]. For the modeling of of spatio-temporal patterns of brain functional

networks, [252] introduced Multi-Head Guided Attention Graph Neural Network (Multi-Head GAGNN).

Drug-drug interaction

The concept of connectivity does not only work for brain networks but can also be applied to other network like structures. [8] used a GCNN approach for the prediction of drug-drug interaction. The text embedding linking two drugs was processed within a CNN approach, while the molecular structure of the drugs is encoded using a NFP [69] and Gated Graph Neural Network [145]. They were able to show an enhancement of text-based drug-drug interaction prediction via the incorporation of the molecular information, showing nicely how the inclusion of this form of relational information can assist the learning task.

Image and volume segmentation / registration

As described in the theoretical motivation of CGNNs, the convolution operation on the graph structure can be interpreted in a similar way as the convolution operation on an image. This idea has been leveraged in a few works for segmentation of medical imaging data [162, 217, 226, 246]. These works do not technically add relational information in form of additional domain parameters into the prediction process, but they still use relational information via the incorporation of geometrical information of the structure to be segmented. [217] applied CGNNs for the improvement of the label prediction for voxels with uncertain predictions in pancreas and spleen segmentation. The uncertainty for every voxel is calculated via Monte Carlo Dropout to estimate the model expectation for a given voxel, then the entropy is used as uncertainty measure while the calculated expectation is inserted as probability of the voxel to belong to a given class within the binary segmentation task. A graph is created on the segmentation volume, where every voxel is connected to its six direct neighbors as well as 16 random vertices within the graph. The edges are weighted based on intensity and location similarity as well as diversity between the nodes. For the semi-supervised classification task, every voxel considered as uncertain is given without a label. Like this, during the training process, the CGNN can improve the predictions on the given voxels via neighboring nodes, which showed a performance improvement on both segmentation tasks (see Fig. 3.16). In the same year, [246] applied the inductive graph approach GraphSAGE [94] for the automatic extraction of a coronary artery centerline from coronary CT angiography images (CCTA). To do so, every segment of the artery was separated into different angle sectors and the corresponding image information in that sector was present as a feature representation for the vertex in the graph. For every vertex, the corresponding radius to the center line of the artery needed to be predicted. They could show that the usage of the CGNN approach led to a smoother artery surface prediction than classical approaches.

In 2020, [226] allowed for the inclusion of an additional information component via interactive feedback from the user. The method is applied onto the segmentation of prostates within MRI scans. The feature representation of a CNN encoder enriched with information from the contour location is provided to the GCNN. Every point of the contour is represented by a vertex in the graph. The network is then trained to perform a refined prediction of neighboring vertices of the contour when one vertex is corrected by the user. In the same year, [162] created a CNN encoder / GCNN decoder approach with integrated Attention refinement to improve the stability of learned segmentation contours. They applied their methodology on

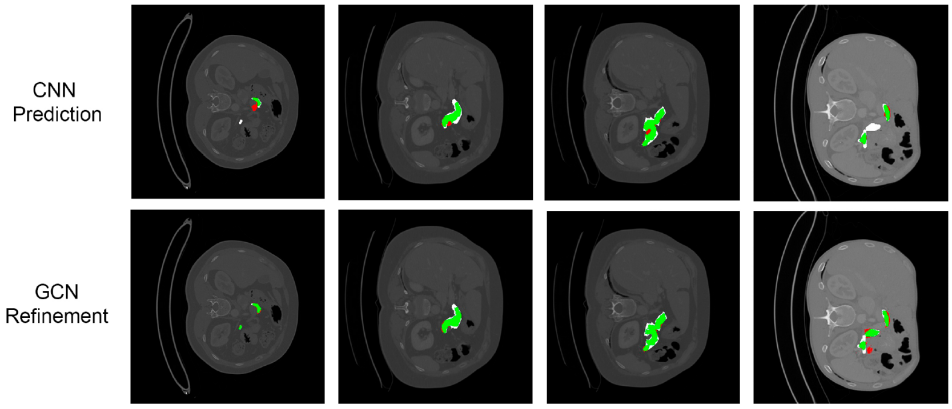


Fig. 3.16. Comparison of CNN prediction and GCN refinement for pancreas segmentation with method proposed in [217]. Green color indicate true positive, red color false positive, white color false negative regions. *Note:* Image from [217]

the segmentation of the fetal head in ultrasound images as well as the optic disc and optic cup in color fundus images. Two more interesting works were performed by [77, 224] in 2021, focusing on the semantic segmentation of airways [224] and landmark-based anatomical segmentation [77].

In 2021, GraphRegNet was introduced by [97] for improved registration of 3D lung CTs, introducing improved displacement regularization using a combination of convolutional and graph neural networks. In the same year, two more interesting papers were published [45, 170]. [170] introduced a method for shape reconstruction from a single CT scan using statistically reconstructed mean shapes and patient-specific models processed within a sequential GCN. In [45] a similar concept is used for the creation of the 3D human pose from a 2D human pose via human mesh vertices processed in a GNN.

CAD for COVID 19

Given by recent events, we would also like to highlight the usage of relational knowledge via graph-based approaches for the fight against COVID 19. Motivated by the outbreak of COVID 19 in the beginning of 2020, research tried to assist medical experts in a fast diagnosis of COVID 19 cases from CT scans as well as treatment recommendations. Within this time, also multiple methods based on graph neural networks were introduced for this topic [65, 110, 147, 218, 239, 259]. For an improved distinction between COVID-19 and community acquired pneumonia, [65] proposed Uncertainty Vertex-weighted Hypergraph Learning (UVHL). Again, every patient is a vertex within the designed graph connected via the determination of k nearest neighbors for both the regional and radiomics features. An uncertainty measure is calculated, uncertainty is either given by erroneous measurements or cases which are close to the decision boundary. Using the weighting concept, the system can learn to reduce the impact of uncertain cases on the training process, making the system more robust to noisy data. ResGNet-C, introduced by [259] can essentially be seen as an application of graph convolutional networks on extracted imaging features, similar to the approach of [185]. The pre-trained weights of a ResNet101 are frozen and the last layers are replaced to match the classification task. Then, the extracted features are combined using the normalized adjacency

matrix and the GCNN is trained based on the given feature representations. The graph itself is build from the k nearest neighbors in feature space.

[218] performed a similar approach, however in their work focused on the inclusion of training samples from multiple medical sides. A 3D-CNN [116] is used for feature extraction of the CT scans. Then, a graph structure is created which connects samples whose feature representations are similar enough and at the same time have the same treatment center, clinical sex and equipment type used for the scan. Additionally, for the training set, only cases with the same label are connected. This basic graph setup is now complemented with an augmented graph setup, in which the similarity metric is replaced with the similarity of the GCNN output for two cases, after it has been pre-trained on the training data. The input for the final GCNN training is then defined by three identical sets of the training dataset concatenated together as well as three adjacency matrices: the identity matrix, the adjacency matrix and the augmented adjacency matrix. This concept allows the simultaneous incorporation of the original extracted features (via the identity matrix), the standard graph as well as the augmented graph containing information of training behavior. Finally, only the q top edges are retained to receive a sparse graph. The group could show that using this graph setup a performance improvement against competitor methods could be achieved. [147] focused on a similar task, namely an accurate detection of COVID-19 cases for patients scanned with different CT machines. An interesting approach was used to prepare the network for the prediction task, closely related to the inclusion of further prior domain knowledge. Before focusing on the COVID-19 prediction task, the 3D-CNN was pre-trained to distinguish scans from the different machines. The trained weights are now transferred to the actual prediction task. The trained network is then used for the extraction of corresponding imaging features. Finally, within the graph structure edges are established between patients which were examined with the same type of machine. Within these separated clusters, a further weighting is applied based on feature similarity and similarity of the prediction outcome of the 3D-CNN. The trained GCN, which is again based on [128], resulted in a strong prediction performance.

The approach for COVID-19 detection proposed by [239] follows the same concept of patient clustering based on feature similarity, but performs a late feature fusion of the graph features and the initial features extracted by the CNN. Especially [239] and [218] leverage this concept of the combination of CNN and relational GCNN features for an improved prediction performance. This concept was introduced by us already in 2019 [32], where we combined it with an end-to-end training on imaging data and its methodological advantages will be discussed in detail within Chapter 4.

4 Relational knowledge through patient meta information

4.1 Introduction

In the previous chapter, we have provided a detailed overview about the incorporation of relational information in form of graph structures into the graph neural network learning process. The described approaches constitute an effective way to leverage the structural information as a topological input of the data domain instead of another individual feature. Here, within works from e.g. [126, 184, 185, 233] it was shown that for a multimodal dataset with imaging and non-imaging features, the incorporation of non-imaging features can be effectively performed as a metric used for affinity graph creation. Based on the similarity of the provided meta information for two patient cases, it is decided if a corresponding connection should be defined in the graph. As described already in Sec. 1.2, from a medical perspective a physician will always implicitly consider patient meta information during diagnosis via his or her medical experience. A clustering process of patients based on this information therefore nicely follows this intuition. For most of the previously described approaches which were operating on imaging data, the processed imaging information is however considered as already extracted, leaving only the condensed feature information as input to the graph neural network. We hypothesize that valuable information relevant for the graph processing is missed out during this process, since the feature extraction and corresponding convolutional filter learning was not optimized towards the graph processing. Within the following sections, we will motivate this reasoning, develop an effective methodology for the inclusion of relational knowledge into the feature extraction process via an end-to-end graph based image processing pipeline and evaluate it on a designed MNIST toy dataset [63] as well as a medical chest X-ray dataset, namely the NIH ChestX-ray14 dataset [240]. To the best of our knowledge, by the time of publication our work was one of the first graph based approaches that tackled end-to-end image processing.

Our methodological approaches will be evaluated on a disease classification problem on chest X-ray images. Although the chest X-ray modality is only used as a medical example and the developed methodology is generally applicable to other imaging datasets (providing corresponding meta information) without any modifications, we still would like to briefly introduce into the topic. First, we will give a short overview about the X-ray modality, its underlying measurement concepts and its applications in the medical field. Then, we will provide an overview about related work performed in the field in disease classification on Chest X-ray images.

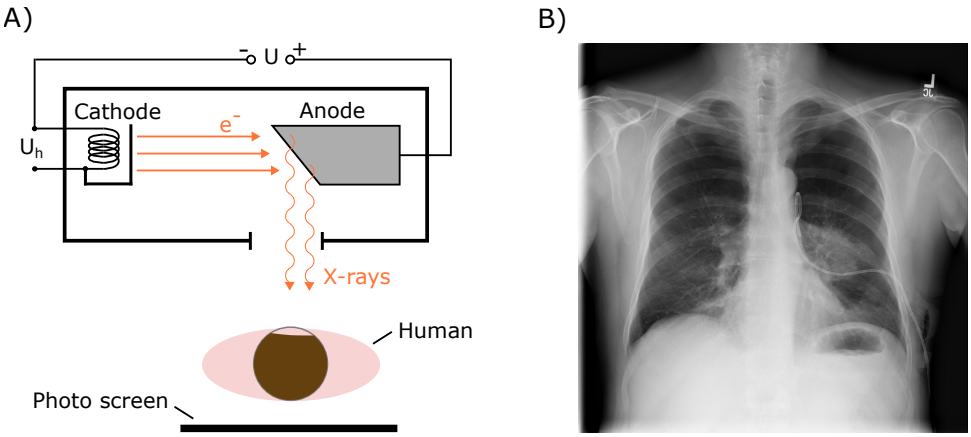


Fig. 4.1. Schematic explanation of X-ray measurement concept. A) X-ray scanning process. Electrons are emitted from a heated cathode and accelerated onto an anode via a voltage U. Within the anode material, the electrons' energy is partially converted into X-ray radiation (consistent of bremsstrahlung and characteristic X-ray wavelengths.) These rays are directed onto the human body, where depending on the transitioned material different ratios of the rays are absorbed. A photo screen behind the patient collects the transmitted X-ray radiation and makes it visible B) Example of a chest X-ray image (darker areas were exposed to more radiation) *Note:* Image from NIH ChestX-ray14 dataset [240].

4.2 Computer aided diagnosis on X-ray scans

4.2.1 The X-ray modality

The usage of X-ray scanners is an important imaging modality in medical context. Given e.g. the medical condition of pneumonia, a serious disease of the lung for which over 1 million adults are hospitalized every year and over 50,000 die from the disease in e.g. the US alone [271, 190] and approximately over 4 million worldwide [202], Chest X-ray scans are currently the best available method for detection [202, 247]. Also other modalities like CT (which is based on X-ray technology) and MRI exist, but the low cost and high availability make the X-ray imaging process very attractive.

The underlying concept of X-ray imaging was discovered by Wilhelm Conrad Röntgen already in 1895, who experienced an electromagnetic radiation capable of transitioning human tissue when experimenting with gas discharge tubes [62]. This radiation is resulting from the partial energy conversion of accelerated electrons. Based on this finding, the concept of an X-ray scanner consists of a heated cathode from which electrons are emitted, which are then accelerated towards and anode target (Fig. 4.1). When hitting the anode under high speed, a part of their energy is converted into X-ray radiation, consisting of a continuous bremsstrahlung spectrum and characteristic X-ray wavelengths depending on the anode material. While traveling through a medium of thickness d, the rays are absorbed by the medium, leading to a reduction in power of $P(x) = P_0 \cdot \exp -\mu x$ with absorption coefficient μ. Depending on the material which the X-rays are emitted on, the absorption of the radiation differs due to a strong dependency on the atomic number Z. The absorption coefficient μ is dependent on the absorption cross section $\sigma_a \propto Z^4\lambda^3$, which scales for the atomic number with the power of 4 (λ is the X-ray wavelength). Therefore, material like lead with an atomic number of $Z = 82$ absorbs the radiation substantially better than e.g. aluminum ($Z = 13$)

for the same material thickness [62]. The same goes for calcium ($Z = 20$) mostly present in bones compared to e.g carbon ($Z = 6$) in human tissue. The human body can therefore be transitioned by the radiation as well, while the different structures and elements within the body provide a different absorption and scattering rate. Collecting the transitioned radiation on a photo detector can therefore provide an imaging scan of the human's internals given in intensity values of each pixel [275] (see also Fig. 4.1).

Today, the X-ray modality is widely used in the medical field. Applications lie in the detection of bone fractures, certain tumors and other abnormal masses, pneumonia, certain injuries, calcification, foreign objects or dental problems [171]. Further, the modality is used in different technical realizations. Among them are Mammography (cancer detection of the breast), Computed Tomography (CT, processing of series of cross-sectional X-ray scans to obtain 3D volume scans) Fluoroscopy (real-time imaging of movement within the human body) and therapeutic applications (destruction of cancerous cells with high-energy radiation) [171].

4.2.2 Disease classification for chest X-ray imaging

Today, X-ray imaging scans are broadly used for various medical applications as already described above. Within the deep learning community, for applications like e.g. pneumonia detection a vast body of work exists. The recent COVID-19 pandemic and corresponding diagnosis requirements for the disease from X-ray as well as CT scans have boosted this development even further with the appearance of extensive amounts of papers in the area. A few of them, which incorporated graph based methods, have been addressed by us already in Sec. 3.6. It would exceed the scope of this thesis to provide a full overview on the area, since the focus of this thesis also does not lie on this topic. We still want to shortly highlight a few important aspects with a focus on available datasets and point to interesting review papers in the field.

Until the year 2017, the largest publicly available dataset was Open-i [60, 273], which at the time had 7,470 chest X-ray images. The group of [212] did an interesting work on this dataset using both the image and its report for disease localization, severity and affected organs. In [112], disease classification was realized on Open-i and two chest X-ray datasets using standard deep convolutional networks, localizing activations with occlusion mapping. The two other datasets (JTRS dataset [84, 213] and Shenzhen dataset [114]) have only 247 and 662 chest x-ray cases respectively. Publicly available data was limited. This changed with the introduction of "ChestX-ray8" [240], which contains 108,948 frontal view X-ray images of 32,717 unique patients for eight different diseases, constituting a jump in dataset size by an order of magnitude. The group performed multi-label disease classification and disease localization using class activation mapping (CAM) [266] on this dataset. Within [190], a DenseNet of 121-layers called ChexNet was used for classification of all 14 diseases available for the whole dataset, with an emphasis on pneumonia. Apart from these approaches further related work was performed in recent years. A nice review on pneumonia detection was e.g. performed by [144]. From the 276 papers they initially found, 59 papers remained after an deeper examination of the scope of each work. Further, after an evaluation on the used data and study design, only 15 papers remained eligible for their comparison. Two other datasets

from the field of chest or lung scans worth mentioning are the LIDC-IDRI (CT, X-ray) [6] dataset and LUNA16 (CT) [210] dataset with 1018 patients and 888 images respectively.

All the works discussed so far relied on standard CNN approaches and did not leverage any non-imaging information. In this direction, [250] performed a comprehensive survey of domain knowledge incorporation in deep learning for medical image analysis. Their review focuses on medical image analysis in general and not specifically on the X-ray modality. The shown comparison of integrated knowledge sources, types and incorporation methods is however very interesting. Methods including graph-based approaches are e.g. [42, 150, 264]. [150] uses a bipartite graph (graph, where a group of vertices is exclusively linked to another group of vertices without any in-group links) to effectively combine the processing of different views of mammography scans for a better prediction. [264] focuses on the problem of automatic radiology report generation and proposes the inclusion of a knowledge graph. [42] proposes an interesting approach to represent connected lymphatic systems of lymph node gross tumor volume with a GNN approach, essential incorporating the pathological behavior into the network structure. Another extensive review on deep learning for chest X-ray analysis was performed in [37], which grouped and analyzed 296 publications. Finally, another very interesting review work, we would like to highlight was performed in [110]. They focus on the topic of feature fusion and the importance of medical context for image-based disease diagnosis within 17 found studies. Although not relating to graph learning aspects, the paper still nicely highlights the importance of the inclusion of non-imaging information into the prediction process. Imaging data with similar appearance can still lead to a different diagnostic result when the medical / demographic situation differs. The methodologies developed within the following sections are motivated from this thinking.

4.3 Adaptive image feature learning

4.3.1 Motivation of end-to-end graph approach

As discussed in the previous section, graph neural networks have shown very promising results for several tasks in both the medical and non-medical domain. One field, where they have been used rarely is the field of image-based processing. Papers like [185] and [184] have shown that improved results can be achieve on pre-extracted imaging features by the inclusion of non-imaging data as relational information. The feature extraction is however not performed within the methodology itself, but instead done in a pre-processing step. This procedure does not only require a potential effort-intensive additional step before the training procedure, but more importantly also might result in a not optimized feature representation for the processing within the graph structure. Certain features within the imaging data might be more beneficial for the classification task, when processed within the graph. At the same time, they are not properly considered when the feature extraction is performed without the underlying cohort and relational information. First, the connection of images from the same cohort can assist in stabilizing the learned feature representation, since related information from the connected images can compensate for individual images of poor quality or informativeness. Additionally, it might be possible to learn and extract features which would have not been considered without the relational graph context. To visualize this idea within a schematic example,

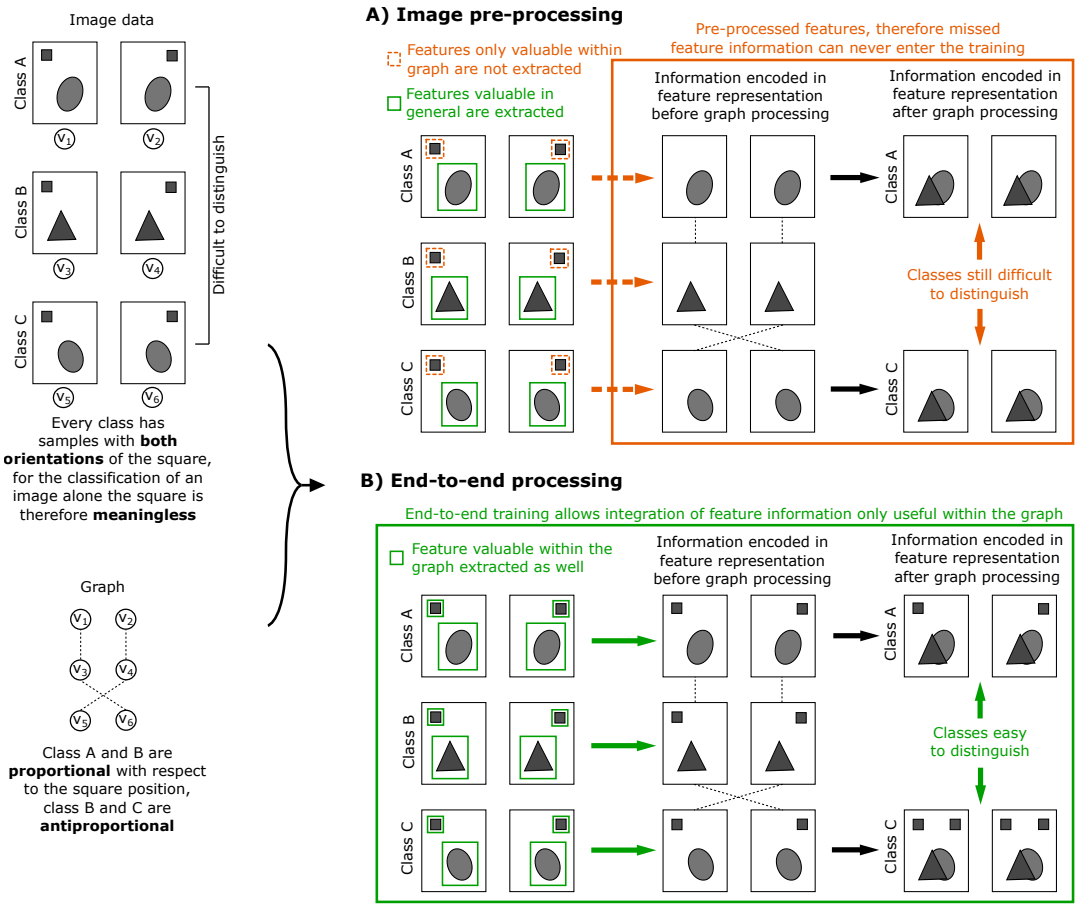


Fig. 4.2. Schematic example to visualize the advantage of graph-based end-to-end image feature extraction. A classification task for three classes A, B and C should be performed, class B being easily distinguishable, while the features in class A and C are very similar, posing a prediction challenge. Additionally, the small square feature at the top exists for all classes and is oriented either to the left or the right, reoccurring for all classes in the same way. The graph in the left bottom however shows that connected samples from class A and B have a proportional relation to the position of the square, while class B and C have an antiproportional behavior. This information is very valuable, in a pre-processing approach it is however not detectable from the image data alone and therefore not meaningful for the classification task. Hence, it will likely be neglected within a feature extraction pre-processing. This is shown on the upper right, the final representations do not distinguish classes A and C. Only an end-to-end backpropagation of the graph's gradients shown in the lower right leads to distinguishable features for classes A and C.

consider the explanation in Fig. 4.2. Three different classes A, B and C are given, which should be distinguished from each other in a classification task. While class B provides features which are very distinguishable from the other two classes (triangle shape), the features for classes A and C are very similar (circular shape). Additionally, a square shaped feature is present within the images which might be right or left-oriented at the top of the image. This feature occurs for every class and is equally distributed to the right as it is to the left. When only the imaging information is considered, it therefore provides no value to the classification task and is therefore likely to be neglected within a feature extraction process focusing on the classification task. In the graph on the bottom left side of Fig. 4.2, we can however see that samples from class A and B as well as samples from class B and C are equally connected. For class A and B, the orientation of the square feature to the left or right has a proportional relation. When the image sample of class A contains the square feature on the left so does class B and vice versa. At the same time, for class B and C the relation is antiproportional.

This valuable information is not considered in an only image-based pre-processing step. As a result, we can see that within the processing in Fig. 4.2A the resulting representation after the processing and feature accumulation (here, we consider equal contribution from every neighbor as in e.g. GraphSage [94]) within the graph network is not nicely distinguishable for class A and C, leading to a probably poor performance on the classification task. The missing square feature not extracted in the pre-processing step cannot be regained in a later step. At the same time, for the pipeline given in Fig. 4.2B, the end-to-end gradient backpropagation from the graph to the feature extraction process allows the detection of the underlying relation between the classes and the square feature. The resulting representation is therefore easily distinguishable and should lead to a better performance outcome.

This simple example motivates the advantage that an end-to-end processing of image-feature extraction can provide within a graph based learning approach. It enables the system to detect feature information and feature relations within the imaging data which provides its value only in the graph structure but not in a pre-processing step alone. It is therefore also only detectable and processable if an end-to-end learning is enabled. While the relation in the example is fairly simply and could be solved with a rule-based approach, relations within the data structure can be more complex and hidden. An approach to find these relations and adapt the feature extraction correspondingly can therefore be of high advantage.

4.3.2 End-to-end graph-based image processing

To incorporate the previously motivated idea of adaptive image feature learning into a graph-based neural network approach, within a work presented at the MICCAI 2019 conference we have developed CNNGAT [32], an end-to-end graph convolutional neural network (CGNN) for image data processing. The general idea of the network CNNGAT is the classification of imaging data in form of 2D data represented via intensity arrays $\mathbf{X}$ using non-imaging meta data $\mathbf{Q}$ in an inductive end-to-end approach. The network architecture is given in Fig. 4.3. The objective function that needs to be optimized by the training process is therefore given by $f(\mathbf{X}, G(\mathbf{V}, \mathbf{E})) : \mathbf{X} \rightarrow \mathbf{Y}$. $G(\mathbf{V}, \mathbf{E})$ represents the graph with vertices $\mathbf{V}$ containing a feature representation $\vec{r}_i$ of every image x_i and binary edges $\mathbf{E}$ as connections between the vertices. The feature representation $\text{CNN}(x_i) = \vec{r}_i$ is extracted from the corresponding image x_i, the extraction process itself is updated during training time, since the extraction-performing convolutional neural network (CNN) introduced in Sec. 3.3.1 is trained end-to-end with the graph network. $\mathbf{Y}$ corresponds to the set of classes which needs to be learned. The whole set of feature representations $\vec{r}_i$ of the graph for every image x_i is given by: $\mathbf{R} = \{\vec{r}_1, \vec{r}_2, ..., \vec{r}_N\}, \vec{r}_i \in \mathbb{R}^F$, where F is the dimension of the feature representation.

Since we want to process imaging data, we need to consider the increased memory consumption of this data type. Within Sec. 3.3.5, the requirements of spectral and spatial CGNN approaches were discussed. Spectral approaches are motivated by a conversion of the graph signal X into spectral domain defined by the eigenbasis of the graph Laplacian $\mathbf{L}$ [29]. Since the eigenbasis is strongly dependent on the graph setup itself the resulting network representation is therefore bound to the graph G on which it was trained. New data not part of the previous training therefore requires a retraining of the approach. This automatically excludes the batchwise processing of the graph by using subgraphs in the training, since every subgraph

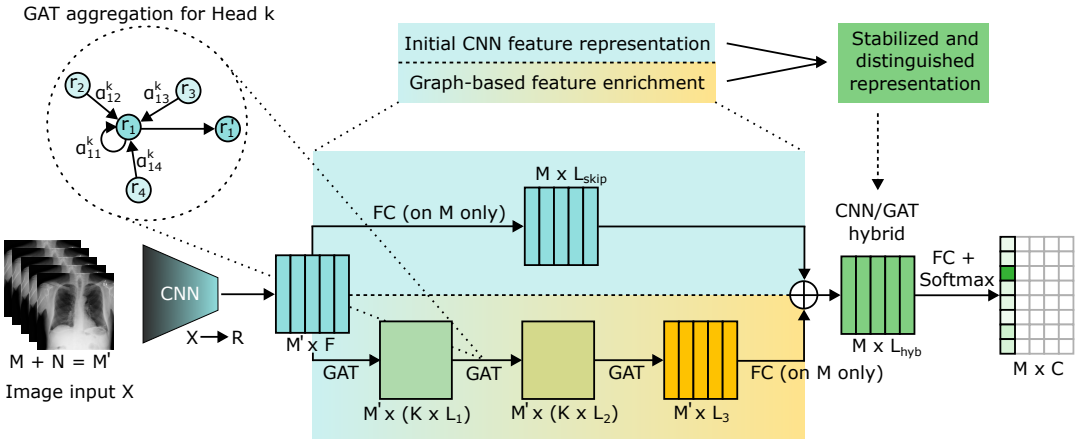


Fig. 4.3. Schematic description of CNNGAT classification process. An image batch of M images and N neighboring images is processed by the CNN, the extracted features are distributed to their corresponding vertices within the graph and passed through the GAT layers. Only the M main batch representations of the GAT and CNN pipeline are concatenated and used for the classification and loss backpropagation.

would result in an individual eigenbasis destroying the consistency of the training. Since the processing of imaging data is not feasible without batchwise processing (due to GPU memory limitations), we therefore decided to rely on spatial approaches.

For the feature representation update of a vertex, it is required that a sufficient amount of neighboring information is available. At the same time, the memory constraints of an image-based approach have to be considered. The neighborhood $N(i)$ of vertex $\vec{v}_i$ is based on the similarity of the sample's meta information $\vec{q}_i \in \mathbf{Q}$. For a given vertex $\vec{v}_i$ another vertex $\vec{v}_j$ is considered to be connected to the node if their similarity is above a certain threshold. We can therefore define the edge criterion in Eq. 4.1:

$$e_{ij} \in \mathbf{E} \; \forall \; \frac{1}{N} |\vec{q}_i - \vec{q}_j| < \theta \tag{4.1}$$

where θ is a given threshold and N is the dimension of the meta data vector. Since the amount of neighbors can strongly vary from vertex to vertex, it is not possible to simply consider every connection of a vertex. Hence, we will follow the approach of [94] and only consider a fixed number n of neighbors $\vec{v}_j$ with $e_{ij} \in \mathbf{E}$ for each vertex $\vec{v}_i$. This sub-sample represents the neighborhood $N(i)$. If for a given vertex $\vec{v}_i$, the number of neighbors $\vec{v}_j$ with $e_{ij} \in \mathbf{E}$ is smaller than n, the neighbors are resampled until the corresponding size is reached. The computational effort of the whole graph is therefore set to a fixed limit, allowing us to control the memory-consumption. The corresponding resulting directed adjacency matrix $\mathbf{A}$ is given by Eq. 4.2:

$$\mathbf{A}_{ij} = \begin{cases} 1, & \text{if } e_{ij} \in E \wedge j \in N(i) \\ 0, & \text{if otherwise} \end{cases} \tag{4.2}$$

We will now forward the extracted feature information from the CNN via to separate channels. Since we want to leverage image interdependencies, the first channel is given by a CGNN. As has been shown in [127, 184], the performance of CGNNs highly depends on the quality of the graph and knowledge gain via relational knowledge, it provides to the learning task. Since high expressiveness of the graph connections cannot be guaranteed, the feature aggregation process should be able to optimize itself for the prediction task. Therefore, it is necessary to

influence and optimize the impact of the individual connections. For our network, we therefore chose the conceptual idea of graph attention layers proposed by [231]. As explained in Sec. 3.4.2, graph attention (GAT) layers learn an importance measure in form of an attention coefficient for every neighbor for the update of another vertex feature representation. Every vertex $\vec{v}_i$ has its own neighborhood $N(i)$ with vertices $\vec{v}_j$. In a first step, within the GAT layer a higher representation of the extracted feature representation needs to be achieved. This representation is obtained by applying a shared learnable linear transformation $\mathbf{W} \in \mathbb{R}^{F' \times F}$ to the representation vector $\vec{r}_i$ and its considered neighbors $\vec{r}_j \in N_i$.

$$\vec{r'}_i = \mathbf{W}\vec{r}_i \tag{4.3}$$

Onto these new representations $\vec{r'}_i$ a shared attention mechanism a can be applied to determine an attention coefficient α representing the importance of a representation for the update of another representation. The coefficient is computed as:

$$a(\mathbf{W}\vec{r}_i, \mathbf{W}\vec{r}_j) = \vec{a}^T[\mathbf{W}\vec{r}_i || \mathbf{W}\vec{r}_j] \tag{4.4}$$

where [||] symbolizes the concatenation of $\mathbf{W}\vec{r}_i$ and $\mathbf{W}\vec{r}_j$ and $\vec{a} \in \mathbb{R}^{2F'}$. This coefficient is processed via an activation function σ = LeakyReLU and then forwarded into a softmax function to obtain normalized attention coefficients α for the weighting of each image feature contribution, the softmax therefore ensures controlled value ranges.

$$\alpha_{ij} = \frac{\exp(\sigma(\vec{a}^T([\mathbf{W}\vec{r}_i || \mathbf{W}\vec{r}_j])))}{\sum_{p \in N_i} \exp(\sigma(\vec{a}^T[\mathbf{W}\vec{r}_i || \mathbf{W}\vec{r}_p]))} \tag{4.5}$$

As explained in Sec. 3.4.2, we can now receive the GAT-layer processed new image feature representation $\vec{r'}_j$ for a given vertex by multiplying the new feature representation of every vertex $\vec{v}_j$ given by $\mathbf{W}\vec{r}_j$ with its corresponding attention coefficient α_{ij} and adding up all contributions. This is performed for multiple attention heads containing individual $\mathbf{W}_k$, which perform the same prediction for $\vec{r}_j$. For every intermediate GAT layer, these representations are $\vec{r'}_j$ are concatenated. In the final attention layer, concatenation is not anymore sensitive and is therefore replaced by averaging, with the activation being performed as the final step. Both functional relations are given in Eq. 4.6:

$$\vec{r'}_i = \|_{k=1}^{K} \sigma\left(\sum_{j \in N_i} \alpha_{ij}^k \mathbf{W}^k \vec{r}_j\right) \quad , \quad \vec{r}_{\text{GAT},i} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in N_i} \alpha_{ij}^k \mathbf{W}^k \vec{r}_j\right) \tag{4.6}$$

where α_{ij}^k constitutes the attention coefficient of head k for the vertex representations $\vec{r}_i$ and $\vec{r}_j$, K is the total number of heads and $\vec{r}_{\text{GAT},i}$ is the final representation of of image $\vec{x}_i$ received from the GAT pipeline.

In parallel to this graph based processing, the original image feature representation $\vec{r}_i$ extracted by the CNN from image x_i is forwarded via a skip connection consistent of a simple single-layer feed-forward network. The idea of this skip connection is to forward the original image feature information without interference with other samples in the graph, an important aspect, which will be explained in detail in the next section. We receive the representation $\vec{r}_{\text{skip},i}$ of image x_i after the skip connection via the simple relation in Eq. 4.7:

$$\vec{r}_{\text{skip},i} = \sigma(\mathbf{W}_{skip} \cdot \vec{r}_i + \vec{b}_{skip}) \tag{4.7}$$

where $\mathbf{W}_{skip}$ and $\vec{b}_{skip}$ are the weights and bias of the single-layer feed-forward network and σ is the ReLU activation function. The outcome of the GAT network and the skip connection are then concatenated to receive the hybrid representation $\vec{r}_{\text{hyb},i} = [\vec{r}_{\text{GAT},i}||\vec{r}_{\text{skip},i}]$ containing both the original extraction information as well as the graph-modified representation after neighborhood interaction. The representation is again activated and forwarded through a last single-layer feed-forward network to receive the mapping onto the output classes of the classification task, before applying a softmax function and evaluating the predictions in the cross entropy loss function.

The overall end-to-end processing of this complete pipeline allows the backpropagation of the gradients to the graph as well as to the convolution neural network, allowing a potentially optimized feature representation learning for the graph structure.

4.3.3 Reasoning of CNNGAT pipeline setup

As explained in the previous Sec. 4.3.2, our developed methodology CNNGAT consists of two separate end-to-end processing pipelines of the image features extracted by the CNN. While the GAT-based pipeline performs a graph-based processing resulting in a mixture representation of neighboring vertices, the second pipeline forwards the individual representations without interference from neighboring data points. The underlying idea behind this concept manifests itself in three conceptual ideas:

Performance preservation: As stated in the beginning, we cannot assume that for every image-based problem a graph can be created which can provide additional relational knowledge to the problem. It is e.g. possible that the meta information $\mathbf{Q}$ used for the setup of the graph is not meaningful with respect to the classification task but instead creates a lot of problematic connections. As has been shown in e.g. [184] and [269], these wrong connections within the graph can even harm the performance of the model by polluting the feature representations. The parallel processing of the original feature representation in a neighborhood-interacting network branch and a preserving network branch can allow the model to rely its predictions mostly on the skip connection branch in case the graph model does not provide assistance in the prediction task. As a result, the performance of our network should never drop significantly below the performance of a standard CNN approach, even when the graph information provides no meaningful addition to the task.

Feature identification: As explained in Sec. 4.3.1, the graph network within CNNGAT can benefit the prediction tasks in two ways. The first more obvious one is the connection of different vertices belonging to the same class and therefore stabilizing the prediction task. The second way is the detection of hidden relations within the feature occurrences of different classes. In order to leverage these two concepts, the corresponding involved feature representations need to be aggregated into a new representation for the prediction case. For the effect of connecting different samples belonging to the same class, it is expectable that the resulting representation should be very representative for the class so that within the final fully connected layer an easy classification of the representation can be performed. In the second case also explained in detail in Fig. 4.2, where relations from different classes are aggregated together to receive a stronger distinction for previously difficult to distinguish classes, the

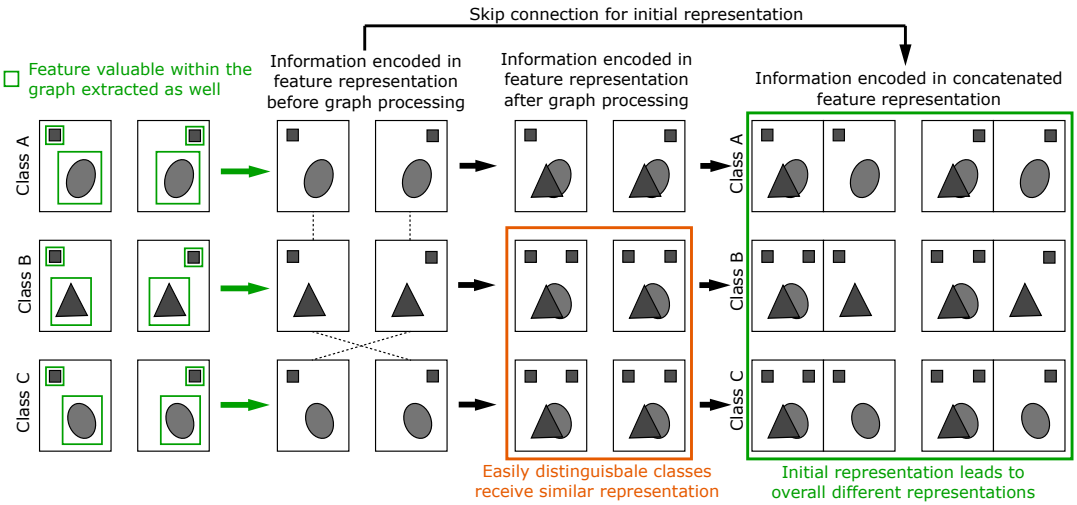


Fig. 4.4. Visualization of the effect of the CNNGAT skip connection on the example provided in Fig. 4.2. After the processing via the graph network, the representations of samples of class B and C might end up similar (under the assumption of a GNN with equal neighborhood consideration like [94], for GAT with its attention concept the problem might not occur as explained in the main text). The concatenation of features from the initial representations to the graph processed representations via the skip connection however ends up with overall nicely distinguishable samples on the right side.

representations can however be more confusing. Following the example of Fig. 4.2, let us consider the resulting representation of class B and C given in Fig. 4.4. Although the initial representations $\vec{r}_i$ of class B and C are very easily distinguishable, the final representations $\vec{r}_{\text{GAT},i}$ after the graph processing are potentially more alike due to the graph interaction. Here, we have to highlight that we are showing the result for an undirected graph and a CGNN which considers every neighborhood contribution equally like e.g. GraphSAGE [94]. Since within CNNGAT a graph attention approach is chosen which is able to weight neighborhood contributions based on their importance, the problem shown in Fig. 4.4 might not occur. The network could realize that samples with a representation representative for class B should not strongly interfere with their neighborhood, since they are already easy to classify without. Using the skip connection, we are still providing an additional safety measure. Even if the representations $\vec{r}_{\text{GAT},i}$ for class B and C would end up to be similar, the skip connection does forward their original representations to the final output layer and concatenates them. The final representation does therefore again contain the easily distinguishable initial representation, resulting now in three easily separable classes (Fig. 4.4 right).

Representation separation: The third motivation of the skip connection and the resulting hybrid representation focuses again on classes difficult to distinguish and provides the opposite side of feature identification. Next to the concept described in Sec. 4.3.1 for the extraction of features only valuable within the graph, which is the main motivation for the end-to-end processing of image data, the graph network inside CNNGAT can serve the final classification task in a second way. This concept is visualized in Fig. 4.5. Different from the previously described scenario for feature identification, we now consider samples from two classes for which the extracted feature representation is very similar. A direct classification based on this extracted information would therefore likely lead to a poor prediction performance. Additionally, within the image data no hidden feature relation as shown in Fig. 4.2 might

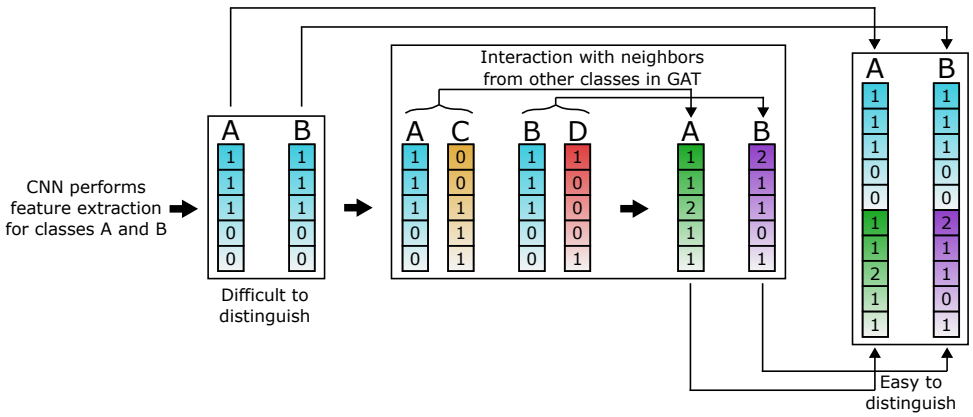


Fig. 4.5. Visualization of the effect of distinct inter-class connections on the prediction performance. In the shown example, for images A and B very similar feature representations are extracted. A direct distinction of the two samples might therefore be difficult. Within the graph, it is however possible that samples from class A (besides being connected to other samples from class A) are more likely to be connected to samples from other classes (in this case C) than samples from class B (in this case D). The resulting aggregated feature representations for class A and B are therefore clearly distinguishable. In the final outcome they are again concatenated with the initial representations to assure that a clear difference between all classes exists (otherwise, the inter-class connections between A and C as well as B and D might lead to similar representations for these classes). We find that inter-class connections can be very helpful in the chosen network architecture, an interesting finding since the general idea of a meaningful graph and relation knowledge focuses on connections between samples from the same class.

be present, so that no feature adaption can assist the training process. Still, the graph can offer another possibility for feature distinction. As shown in Fig. 4.5, the samples from class A and B are provided with very similar feature representations. Within the graph, due to a certain meta data similarity, samples from class A might be more likely to be connected to samples from class C (besides being connected to other samples from class A) while samples from class B are more frequently connected to samples from class D. Therefore, within the graph the feature aggregation results in distinct feature representations for samples from class A and B through the individual fingerprint of their inter-class connections. As soon as such a class-specific pattern of inter-class connections exists, the graph can therefore improve feature distinction. To still allow a clear distinction of the final representation, the skip connection concatenates the initial feature representation. Similar to the process in Fig. 4.4, this concatenation guarantees a clear difference between the final representation of samples from class A and C as well as B and D. Therefore, we receive the interesting and unintuitive observation that inter-class connections could be beneficiary in the used methodology.

4.4 Experimental setup

4.4.1 Datasets

In the following, we will evaluate the performance of CNNGAT on real datasets. Here, we will separate the evaluation of our proposed method into two separate parts, the evaluation on an Modified MNIST toy dataset as well the NIH ChestX-ray14 dataset. The focus of this evaluation will lie on the graph impact on image processing and feature extraction.

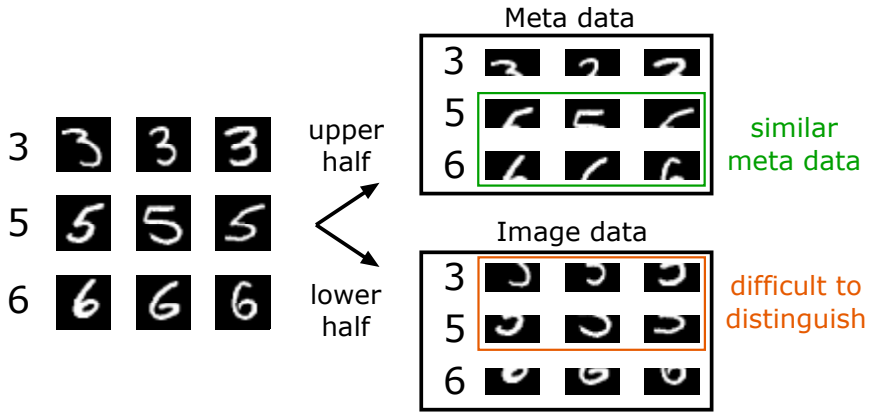


Fig. 4.6. Generation of modified MNIST dataset. The lower halves of the chosen digits are used to create the image data, the upper halves are used for the creation of a meta data vector by flattening the 2D image tensor. As can be seen in the data, while the lower parts of 3 and 5 look very similar, the digit 6 is clearly distinct. In the same way, for the upper parts, 5 and 6 have a stronger similarity compared to 3 and 6. This behavior enables us to create a graph structure with helpful inter-class connections to facilitate the distinction between the digits 3 and 5.

Modified MNIST dataset

In the first part, we will run an evaluation on a modified dataset designed from the well-known MNIST dataset to evaluate our previously explained methodological assumptions. The MNIST dataset consists of 70,000 equally balanced handwritten digits from the numbers 0 to 9 [63]. We will use only the lower half of the images as image input x_i and the upper half of the image as meta data $\vec{q}_i$ for that image. To do so, we will flatten the upper image half into a vector of intensity values. Our goal is to show the improvement of the prediction performance on samples normally difficult to distinguish. We will therefore create our dataset based on samples from the digits 3, 5 and 6. The reason for this is nicely visible in Fig. 4.6. If we only consider the lower half of the images, the numbers 3 and 5 are very difficult to distinguish at least for some of the cases, since the lower part of the number (the half circle) looks identical for both. The upper half of 3 and 5 however looks distinct, therefore providing different meta information. We now require a digit, which provides different image features then these two numbers and at the same time connects only to one of the two samples via its meta information. This is nicely performed by the digit 6. This digit does show different features in the lower part of the image (full circle), while being similar to 5 in the upper half of the image, but not 3. By setting the correct threshold θ in Eq. 4.1, it is possible to connect a substantially larger amount of samples from 6 to 5 than to 3. We can therefore recreate a situation described in Sec. 4.3.3 where the graph can be of clear advantage to the prediction task. We therefore design a training subset of 6000 images equally distributed between the numbers 3, 5 and 6 and provide the same balance for 2,860 images for testing.

NIH ChestX-ray 14 dataset

After evaluating the performance of our designed network CNNGAT on the modified MNIST dataset, in the next step we will transfer the approach onto a real medical dataset. As already stated in the introduction of the thesis, larger image-based medical datasets are rare, especially if a larger amount of meta information for the images is required. One dataset providing a larger amount of imaging data is the NIH ChestX-ray14 dataset [240]. It consists of 112,120 labeled X-ray images, containing one or more of 14 different diseases, unfortunately with

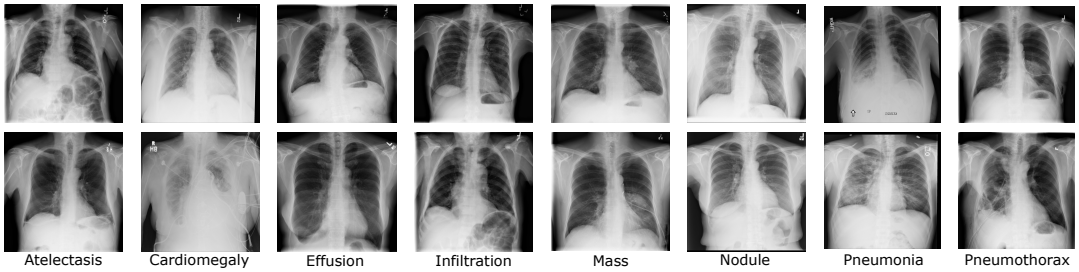


Fig. 4.7. Samples from the NIH ChestX-ray 14 dataset for all eight diseases (selection based on [240]) considered within the classification task. For every disease two samples are shown. For every image, an image-level label is provided for the present disease, no details about the location within the images is given. Additionally, for every image the patient id, patient age and clinical sex are provided.

substantial imbalance for some classes. We will therefore focus on the eight diseases suggested by [240] (namely Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pheumonia and Pneumothorax) and create a dataset based on the images which provide only a single label (only one disease present). The labels are given on image level, no detailed information about the location within the image is given. Within Tab. 4.1 a statistical overview about the setup of the dataset is given. We want to create a dataset in the overall size of 16,000 images. Since not all of the eight disease classes provide a sufficient amount of images to obtain a fully balanced dataset (2000 cases per class), we design the dataset based on the following process: For every class, we add a random image from all images of that class into the dataset, therefore increasing the dataset size in steps of 8. As soon as one class does not provide anymore images (e.g. Pneumonia after 322 iterations, see Tab. 4.1) the process is repeated for the remaining classes. The process stops when overall 16.000 images are in the dataset. This process provides a good trade-off between balancing the dataset and receiving a dataset of sufficient size for proper training. We receive the following distribution of samples: Atelectasis (2563 images), Cardiomegaly (1093 images), Effusion (2563 images), Infiltration (2563 images), Mass (2139 images), Nodule (2563 images), Pheumonia (322 images), Pneumothorax (2194 images).

The dataset additionally provides the patient id, age and clinical sex of every examined patient, so we will use this information to build up a meta data similarity graph similar to the work of [185]. Whenever two patients have a high enough similarity in their meta information, a corresponding connection $e_{ij} \in \mathbf{E}$ is established.

4.4.2 Affinity graph creation

For processing of the previously described datasets, it is required to setup a meaningful graph structure to incorporate additional relational knowledge priors into the otherwise fully image based prediction process. This additional knowledge should result out of the provided meta information.

For the modified MNIST dataset, the construction of the adjacency matrix and the corresponding affinity graph relies on the upper half of every handwritten digit image. By design of the dataset, we know as discussed in Sec. 4.4.1 that the affinity graph should provide valuable information to the classification problem. As a metric for the setup of the adjacency matrix, we

chose the averaged $l1$-distance of the flattened image intensity vectors, which constitutes the meta information **Q**. The intensity values of every pixel are in the range $[0, 1]$. For every two vertices v_i with meta information $\vec{q}_i \in \mathbf{Q}$, we calculate the averaged $l1$-distance to all other $\vec{q}_j$ and receive an edge $e_{ij} \in \mathbf{E}$, when Eq. 4.1 is fulfilled. After an analysis of the resulting affinity graphs, we found that within the training set the best trade-off between connectivity between digits 5 and 6 and non-connectivity between 3 and 6 was given for $\theta = 0.1$ for the average pixel intensity difference.

In a second step, we want to use the modified MNIST dataset for an explicit analysis of the assumption of inter-class connections. As explained in Sec. 4.3.1, the main contribution of the graph to classification performance should result out of connections between samples from the same class, since it provides stability to the prediction task within the same data cohort. At the same time, within Sec. 4.3.3 we have discussed that our developed approach can leverage certain types of inter-class connections as well. To evaluate this assumption explicitly, we design three additional graphs. The first is set up as a complete random graph, therefore providing no additional information as a baseline graph. Then, we create a label graph, which connects all samples with the same labels with each other. Finally, we create a label graph, which also includes all inter-class connections between the digits 5 and 6. These graphs are for evaluation purposes only and are obviously not usable for an actual performance comparison, since label information was introduced into the graph creation.

For the NIH ChestX-ray14 dataset, we can use the provided meta information of patient id, age and clinical sex that was given for every patient. Here, we are following the approach of [185]. They defined a similarity criterion based on the available meta information for the ADNI [113] and ABIDE [66] dataset. We have performed a statistical analysis of the available meta information (see Tab. 4.1). Within this analysis on the training dataset, it was visible that no strong distinction or inter-correlation was possible via the provided meta information, only minor effects were visible. We could e.g. determine that woman were more likely to experience Cardiomegaly and Pneumothorax than man, although less woman are present

Tab. 4.1. Statistical overview of NIH ChestX-ray 14 dataset for the eight classes proposed by [240]. Only single label cases are considered (only one label present). The abbreviations are defined as follows: M = Male, F = Female, A = Average age, AM = Average age male, AF = Average age female, AR = Age range, ARM = Age range male, ARF = Age range female.

Disease	Cases	M	F	A	AM	AF	AR	ARM	ARF
Atelectasis	4215	2603	1612	50.6	51.1	49.8	0-91	0-91	2-87
Cardiomegaly	1093	508	585	46.4	44.6	48.0	3-91	3-89	5-91
Effusion	3955	2158	1797	50.6	51.0	50.1	5-90	6-88	5-90
Infiltration	9547	5383	4164	44.8	44.7	45.0	0-90	1-90	0-90
Mass	2139	1301	838	48.0	47.9	48.1	1-88	1-88	2-83
Nodule	2705	1524	1181	50.0	50.5	49.3	2-93	2-93	3-91
Pneumonia	322	194	128	41.1	42.3	39.2	3-87	3-87	5-72
Pneumothorax	2194	1001	1193	45.5	42.9	47.7	5-85	5-83	7-85

in the overall dataset. Following a similar approach as [185], we therefore provided the following connectivity criteria for two vertices v_1 and v_2:
1) patient id$_{v_1}$ = patient id$_{v_2}$
2) clinical sex$_{v_1}$ = clinical sex$_{v_2}$ and |age$_{v_1}$ - age$_{v_2}$| $< 1y$
3) clinical sex$_{v_1}$ $\neq$ clinical sex$_{v_2}$ and age$_{v_1}$ = age$_{v_2}$
The logic behind this setup is given as follows: 1) If multiple scans were taken for the same patient, these should be considered as related to each other. Within the setup of the dataset, it was assured that multiple samples for the same patient id were all placed in either the training or testing set. 2) If two patients have the same clinical sex and therefore a higher similarity, their age is allowed to differ by one year. 3) If two patients have a different clinical sex, their age has to match exactly. This setup was designed to receive a sparse graph with a high distinction between clinical sex and a few linking edges.

4.4.3 Network setup

The idea of our evaluation process is focusing on the evaluation of the proposed approach and methodological assumptions we have made to create the network setup. Our evaluation therefore needs to focus on the individual aspects of the methodology. This is best achieved in form of an ablation study. We will therefore create three baseline and one alternative method to evaluate these methodological implications. In a first step, we want to demonstrate the general superiority of CNNGAT compared to a standard CNN approach not relying on a relational prior in form of a graph structure. Then we will compare against a standard GAT network [231] trained on pre-extracted features to demonstrate the advantage of end-to-end image processing. The same will be done with a modification to the GAT network by adding a skip connection identical to CNNGAT but still training on pre-extracted features, we will call this network "SkipGAT". Finally, to evaluate the impact of the skip connection within CNNGAT, we compare against an end-to-end trained GAT network, called "EndGAT". We would like to highlight that the goal of this evaluation is not a performance-driven comparison against other methodologies for disease prediction on chest X-ray images, but instead focused on the evaluation of the graph impact on image processing.

For all networks the same CNN setup was chosen. For the modified MNIST dataset, a simple CNN based on two convolutional layers and one pooling layer was set up (32 and 64 channels, ReLU activation, no dropout), the NIH ChestX-ray 14 dataset was processed via AlexNet [131]. This simple setup allowed an easy prevention of overfitting and a focus on the impact of the different aspects of the methodology.

In general, the following parameters were chosen for the training: The modified MNIST dataset was processed by a GAT network with 60 features as input, 2 GAT layers with 30 and 10 units correspondingly and 5 heads each. For training, a dropout rate of 0.3, weight decay of 5e-3, a learning rate (lr) of 0.02 and a lr decay of lr $\times$ 0.3 every 20th epoch was chosen, for both the first and second neighborhood (corresponding to the two GAT layers) 4 neighbors were sampled. For the NIH dataset, most settings were kept, but the GAT units were set to 30 and 32 for the two layers, the lr was set to 0.01 and a lr decay of lr $\times$ 0.3 every 30th epoch was chosen.

4.4.4 Training and experiment setup

The training for the performance comparison evaluation will be based on a split of the datasets into training and unseen test data. As described above, for the MNIST dataset this split consists of 6000 images for training and 2860 images for testing, both subsets equally balanced between the classes. For the NIH ChestX-ray 14 dataset, the training consists of 16,000 images in total, split up into 12,800 images for training and 3200 images for testing. The training is automatically stopped after 200 epochs while performing an ongoing learning rate reduction as described previously to obtain better convergence behavior. The resulting network models are then evaluated on the corresponding unseen test set.

In a second step, we will perform an occlusion analysis to evaluate if additional information from the graph assists in the classification of images where important parts are occluded. For the two datasets, we sample 1000 random images from the test sets and perform a prediction on the images using the previously trained models while sliding an occlusion window over the images. Like this, we evaluate if the feature representation is maintained in a more robust way for the CNNGAT approach. For every prediction performed on an occluded image, the probability of the correct class given by the softmax function is recorded for every position of the occluding window.

4.5 Results and Discussion

4.5.1 Methodological evaluation on modified MNIST dataset

The performance results of the evaluation on the modified MNIST dataset are shown in Tab. 4.2. For all networks the overall accuracy as well as the lowest accuracy obtained for a class were noted. As expected the prediction task on the lower parts of the images is challenging, resulting in a higher confusion between the digits 3 and 5. The obtained results overall confirm the assumptions that were previously defined. The CNN provides an acceptable base performance. The GAT performing on pre-extracted features slightly reduces in performance, since neither an end-to-end processing of the data is possible nor a skip connection is established. Adding the Skip connection in SkipGAT again improves the performance. SkipGAT essentially corresponds to the exact same network setup as CNNGAT, but does not perform an end-to-end processing of the images but only processes the pre-extracted features. Adding the end-to-end processing in EndGAT boosts the performance of the approach, showing the value that the end-to-end processing can bring to the approach. Finally, adding the skip connection in CNNGAT pushes the network to its highest performance, since now the aspects discussed in Sec. 4.3.3 can be leveraged as well. As a result, CNNGAT is significantly outperforming all other approaches (Wilcoxon signed-rank test, significance level $p < 0.001$). The accuracy on the lowest performing class is highly improved, showing the clear benefit for the difficult classes.

It needs to be kept in mind that the processed graph structure is based on the MNIST meta information of the upper image half, therefore the graph is not perfect but contains a not insubstantial amount of wrong edges not helpful for the prediction task (connections

Tab. 4.2. Performance comparison within an ablation study on the modified MNIST dataset. For the affinity graph creation, the criterion θ in form of the l1-distance was used. The p-value is reported against CNNGAT. While the individual aspects of CNNGAT already show improvements within the ablation study, our full CNNGAT approach significantly outperforms all compared methods (p-value: ≤ 0.05 $*$, ≤ 0.01 $**$, ≤ 0.001 $***$).

Network	Affinity	Accuracy	Lowest class acc	p-val
CNN	/	0.822 ± 0.005	0.690 ± 0.049	10e-5
GAT	θ	0.794 ± 0.003	0.450 ± 0.008	10e-5
SkipGAT	θ	0.829 ± 0.014	0.538 ± 0.066	10e-5
EndGAT	θ	0.881 ± 0.016	0.780 ± 0.087	10e-4
CNNGAT	θ	$\mathbf{0.911 \pm 0.005}$	$\mathbf{0.810 \pm 0.035}$	/

between 3 and 5 as well as 3 and 6). Therefore, the performance results need to be seen under consideration of these disturbing data aspects. To achieve a pure evaluation of the methodological assumptions, we therefore evaluate CNNGAT on an artificial scenario, in which the graph is based on label information. If the assumptions from Sec. 4.3.3 are correct, the integration of inter-class connections into a label graph should outperform a pure label graph only connecting samples from the same classes. Within Tab. 4.3 we can see that this assumption is indeed justified. As expected, the random graph performs on a similar level as the CNN, since the graph can provide no valuable information. At the same time, the skip connection assures that the performance cannot drop substantially below the CNN performance. The label graph results in a almost perfect prediction result. Here, it is important to note that the label graph, although providing essential information to the classification task, is not identical with actually providing the label as an input parameter. The label graph does only group every sample with the same class together into one cohort, strongly unifying their feature representation via the graph aggregation process. However, if the feature representations of two cohorts are very similar, it still might be difficult for some cases to perform the right predictions. For these cases, an inter-class connection distinguishing the two representations from each other can therefore lead to another performance boost. Within Tab. 4.3, it is clearly visible that this is the case. While the label graph results in an already very

Tab. 4.3. Performance comparison on modified MNIST dataset using the described label graph setting. The following affinity mechanisms were used: random = random connections within the graph , L = samples with same label are connected, CL = samples with same label are connected as well as samples 5 and 6. The results emphasize that the inter-class connections are indeed beneficial for the prediction task and provide complementary information the pure label information.

Network	Affinity	Accuracy	Lowest class acc	p-val
CNNGAT	random	0.816 ± 0.004	0.613 ± 0.052	/
CNNGAT	L	0.980 ± 0.036	0.956 ± 0.079	/
CNNGAT	**CL**	$\mathbf{0.992 \pm 0.001}$	$\mathbf{0.989 \pm 0.003}$	/

Tab. 4.4. Performance comparison on the described NIH ChestX-ray 14 dataset for 16,000 images of eight disease categories. Although a minor improvement is visible, significance against the CNN approach without graph information was not achieved (Wilcoxon signed-rank test). This outcome reflects the insufficient information that is provided by the available patient meta data.

Network	Affinity	Accuracy	Lowest class acc	p-val
CNN	/	0.429 ± 0.015	0.144 ± 0.102	0.109
CNNGAT	meta	$\mathbf{0.437 \pm 0.014}$	0.086 ± 0.070	/

strong performance, the label graph polluted with inter-class connections between 5 and 6 boost the performance even higher, resulting in an almost perfect classification.

From the obtained results on the modified MNIST dataset, we can therefore deduce that our general assumptions about the advantages of graph-based image processing via CNNGAT were justified.

4.5.2 Disease classification of chest-Xray images

In the second step, we will perform an evaluation on a medical dataset, namely the NIH ChestX-ray 14 dataset. We again would like to highlight that the evaluation is focused on the graph impact on image processing and not the performance itself compared to other methods. Using the graph setup described in Sec. 4.4.2, we create the corresponding graph structure for the training process. As we have shown in Tab. 4.1, the underlying meta information does not show a clear relation to the classification task, therefore it can be assumed that the graph can only provide limited assistance to the task compared to the setup of the modified MNIST graph.

Within Tab. 4.4, we see the corresponding result of the evaluation on the NIH ChestX-ray 14 dataset. As a baseline, we apply a standard CNN (AlexNet) to the classification task. The application of our full approach CNNGAT shows only a slightly better performance than the used CNN, but the improvement is not significant (Wilcoxon signed-rank test). As expected, the graph does not provide sufficient relational information to substantially exceed the purely image-based classification. However, we can again see that the performance is protected via the skip connection. Our assumption with respect to the overall setup of the learning procedure was justified. When the graph structure does not assist in the learning task, CNNGAT falls back to the CNN default performance. At the same time, the dependency of the approach on a meaningful graph structure becomes painfully apparent.

4.5.3 Performance under occlusion

As described in Sec. 4.4.4, to further evaluate the impact the graph setup has on the classification performance, we will perform an occlusion analysis for both datasets. The idea of the occlusion analysis is to cover parts of an image when providing it as input to the network. Like

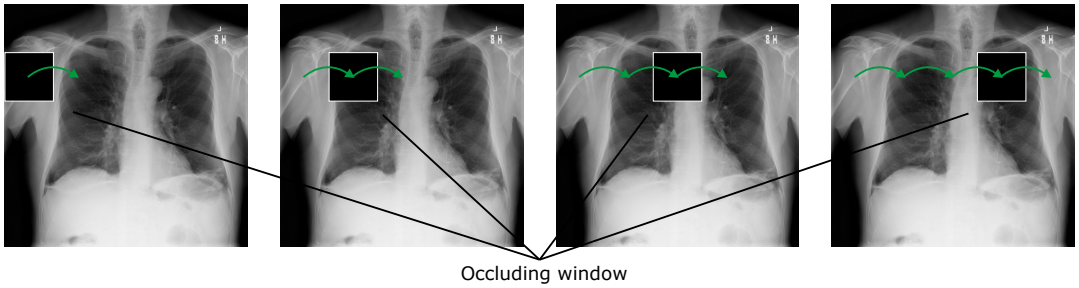


Fig. 4.8. Schematic description of occlusion shift experiment. A window of given size (7×7 for modified MNIST dataset, 50×50 for NIH ChestX-ray dataset) is moved across the image to occlude certain areas of the imaging information. The trained network models are then applied onto the imaging information, the resulting probability of the correct class is stored for the corresponding window position. Repeating this process for all window positions on the image results in a probability map for the correct class prediction.

this, it is possible to evaluate the robustness of the classification method to this disturbance. For every dataset, 1000 random images from the test set are chosen for this evaluation. Since the image dimensions of the two datasets differ, the occlusion process is slightly adapted for the analysis on each set. For the modified MNIST dataset, a 7×7 window is placed on every image and shifted across with a stride length of 1, resulting in $22 \times 8 = 176$ individual evaluations on every image. For the NIH ChestX-ray 14 dataset, a 50×50 window and a stride length of 50 are used, resulting in 25 evaluations for one image. For every evaluation, the probability for the correct class within the softmax function is collected, in the end the average over all evaluations is reported. The results of the occlusion experiment are shown in Tab. 4.5. For the 1,000 images from the modified MNIST dataset, a clear improvement in prediction accuracy can be seen. For the NIH ChestX-ray14 dataset, the improvement is less pronounced but still reaches statistical significance (Wilcoxon signed-rank test). As expected, the probability to predict the correct class under occlusion is stabilized by the

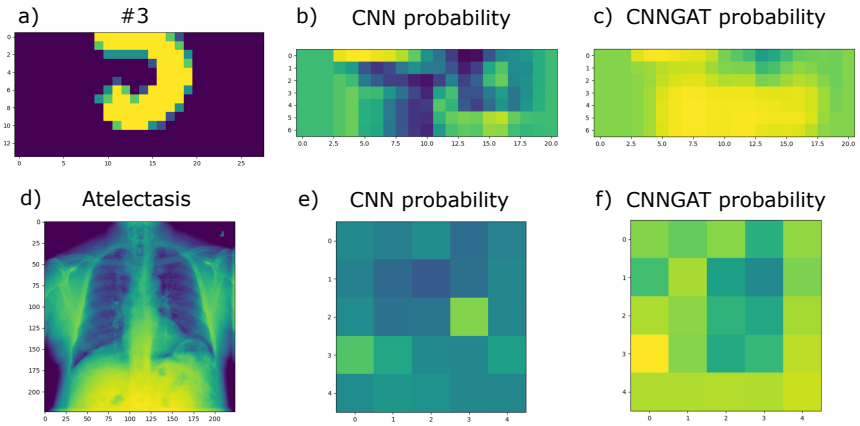


Fig. 4.9. **Top**: (a) Example image of modified MNIST dataset, the lower part of a number 3 is depicted. (b and c) Occlusion shift on (a) with a 7×7 window and a stride length of 1. The coloring corresponds to the probability in the softmax function for the correct class when the occlusion is performed in that image region, the brighter the region the higher the probability (bright yellow: highest achieved probability for compared occlusions, for b) and c) as well as e) and f) the same coloring scale is chosen for fair comparison). **Bottom**: (d) Example image of NIH ChestX-ray14 dataset, an image for Atelectasis is depicted. (e and f) Occlusion shift on (d) with 50×50 window and a stride length of 50. The color coding follows the same logic as above.

Tab. 4.5. Performance under occlusion of CNN and CNNGAT on the modified MNIST and NIH ChestX-ray 14 dataset for 1,000 images respectively. For both datasets, the occlusion and resulting image data information loss can be compensated partly by the relational knowledge provided by the graph structure.

Network	Dataset	Affinity	Accuracy	P-val
CNN	Mod. MNIST	/	0.69	0.0
CNNGAT	Mod. MNIST	meta	**0.81**	/
CNN	NIH ChestX-ray 14	/	0.27	2.6e-5
CNNGAT	NIH ChestX-ray 14	meta	**0.30**	/

graph structure. Its relational information can partly compensate for the missing imaging information. Within Fig. 4.9, two exemplary probability maps for the occlusion of an image from the modified MNIST dataset and the NIH ChestX-ray14 dataset are shown respectively. Bright areas correspond to regions with a high probability for the correct class while dark areas refer to regions with low probability. Especially for the MNIST example, it can be seen that the occlusion of the center area of the image leads to strong performance drops. The performance in this area is highly stabilized by the graph approach within CNNGAT, leading to an overall strong prediction result.

Before concluding on this evaluation, in the next section we briefly would like to point out to an extension of this work, which we have performed motivated by the previous findings.

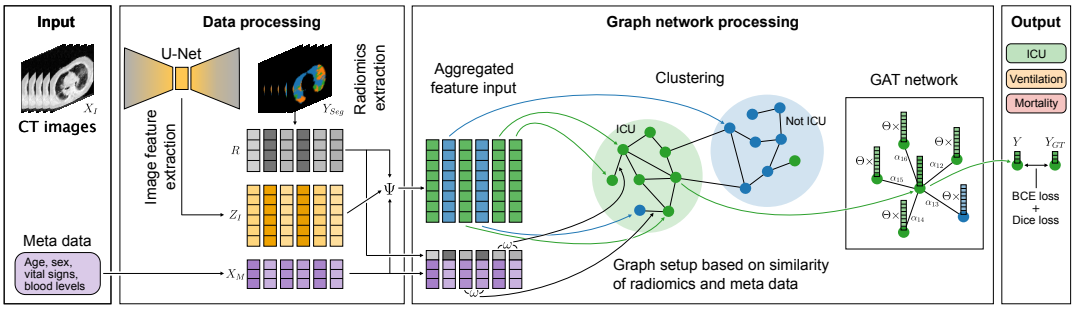
4.6 Extension: U-GAT for outcome prediction

Within the previously described work, we have demonstrated the potential of graph-based adaptive image feature learning, but also highlighted its challenges. We briefly would like to raise the attention to another of our works (currently in revision process for the journal Medical Image Analysis, pre-print available in [127]) which builds up on the previous findings. The work was started during the COVID-19 pandemic and aims to extend the concept of graph-based end-to-end image processing for improved early patient treatment advice and outcome prediction, which was especially challenging in the early months of the new unknown disease. Motivated from the previous results, we wanted to further investigate the limiting factors of the CNNGAT approach. An emphasis of this work was on a thorough investigation of the graph creation. As described in Sec. 4.5.2, the performance of graph-based approaches is strongly dependent on its corresponding graph structure. Within the developed methodology U-GAT, the latent representation of a U-Net is connected to a graph attention network (GAT) in an end-to-end pipeline for the processing of CT scans. The purpose of this structure is the usage of the segmentation as an auxiliary task, while the features of the latent space are extracted for processing within the CGNN. Since the U-Net encoder is therefore used as a feature extractor, this setup allows for a backpropagation of graph information into the extraction process as introduced for CNNGAT. Below, a conceptual figure of the U-GAT setup as well as the abstract of the paper is provided for the interested reader, the work can be found in [127]. The paper is also again listed at the end of the thesis for undiscussed works.

U-GAT: Multimodal Graph Attention Network for COVID-19 Outcome Prediction

Matthias Keicher*, Hendrik Burwinkel*, David Bani-Harouni*, Magdalini Paschali, Tobias Czempiel, Egon Burian, Marcus R. Makowski, Rickmer Braren, Nassir Navab, Thomas Wendler

* Authors contributed equally to this work



U-GAT is an end-to-end trained model that combines image and radiomic features (ZI and R respectively) with clinical metadata XM like age, sex, vital signs and blood levels to predict the patient outcome of a disease. The segmentation YS eg of disease-affected areas in a CT image XI is used as an auxiliary task to extract the radiomic features R from the segmentation output and to regularize the extraction of the image feature ZI from the bottleneck of the U-Net [127].

During the first wave of COVID-19, hospitals were overwhelmed with the high number of admitted patients. An accurate prediction of the most likely individual disease progression can improve the planning of limited resources and finding the optimal treatment for patients. However, when dealing with a newly emerging disease such as COVID-19, the impact of patient- and disease-specific factors (e.g. body weight or known co-morbidities) on the immediate course of disease is by and large unknown. In the case of COVID-19, the need for intensive care unit (ICU) admission of pneumonia patients is often determined only by acute indicators such as vital signs (e.g. breathing rate, blood oxygen levels), whereas statistical analysis and decision support systems that integrate all of the available data could enable an earlier prognosis. To this end, we propose a holistic graph-based approach combining both imaging and non-imaging information. Specifically, we introduce a multimodal similarity metric to build a population graph for clustering patients and an image-based end-to-end Graph Attention Network to process this graph and predict the COVID-19 patient outcomes: admission to ICU, need for ventilation and mortality. Additionally, the network segments chest CT images as an auxiliary task and extracts image features and radiomics for feature fusion with the available metadata. Results on a dataset collected in Klinikum rechts der Isar in Munich, Germany show that our approach outperforms single modality and non-graph baselines. Moreover, our clustering and graph attention allow for increased understanding of the patient relationships within the population graph and provide insight into the network's decision-making process.

In revision for Medical Image Analysis (2021), pre-print in arXiv:2108.00860

4.7 Conclusion

Within the proposed network structure CNNGAT, a methodology is presented to leverage graph-structured relational information for a better image processing. The methodology focuses on an improvement of the learned features as well as an improved separability, targeting specifically the better distinction of classes with otherwise similar feature representations. This is achieved via an end-to-end learning process of the features using relational graph information as well as a combination of graph-processed and pure features into a hybrid representation. The relational knowledge is manifested in additional meta information provided together with the imaging data, which allows the connection of samples based on meta data similarity within the graph. The motivation behind the methodological design choices was explained in detail within Sec. 4.3.1 and Sec. 4.3.3. The evaluation of the proposed methodology on two datasets, namely a modified MNIST dataset and the NIH ChestX-ray14 dataset, showed that the expected performance improvements can be achieved, while at the same time clearly the dependencies of the methodology's success were highlighted. These dependencies are mainly related to the quality and information-richness of the relational information from which the graph structure is created. For the MNIST dataset, this inclusion led to an almost 10% improvement in classification accuracy compared to the CNN baseline. For the NIH ChestX-ray14 dataset, the inclusion of the available meta information could achieve a mathematically significant performance improvement.

The value of the graph structure information can be seen in two terms. The first and common one is its ability to connect samples from the same class within the same graph cohort. This behavior leads to the creation of a more unified and stable feature representation for this class and the distribution of this representation among the connected samples, resulting in improved prediction results. The second aspect which was motivated in this work is related to unique inter-class connections of samples from one class to samples of other classes. In general, this concept is at first counterintuitive since the pollution of a feature representation with representations from other classes should harm the prediction performance. In case of random or meaningless inter-class connections, this statement is true as becomes visible for the evaluation of the NIH ChestX-ray14 dataset, where no performance improvement could be achieved. At the same time, inter-class connections which are distinct for different classes can help to provide unique feature representations to two classes previously difficult to distinguish (as hypothesized in Sec. 4.3.3). This concept can be further leveraged via the end-to-end processing of the image data as described schematically in Fig. 4.2 in Sec. 4.3.1. The achievable advantages were clearly shown on the modified MNIST dataset and could be shown for both datasets within an occlusion shift experiment.

In conclusion, it can be stated that the impact of the introduced methodology has a strong dependency on the provided relational knowledge and corresponding graph structure. For its successful application, a focus should be set on the collection of meaningful relational meta information. At the same time, the design of the method assures that also meaningless graph information should not harm the network's performance below a purely CNN-based alternative. Since the methodology is designed in a way which makes it generally applicable to image processing networks, it can provide a valuable contribution whenever expressive empirical priors in form of meta information can be collected. As discussed in Sec. 4.6, we

have performed further research in this direction within [127], focusing on image based outcome prediction for COVID-19 cases. Within this work, it was possible to demonstrate the added value that the graph provided to the feature extraction and prediction task.

Within the next chapter, we are going to further enhance the concept of relational knowledge inclusion by integrating a second form of empirical priors which can assist in a network learning process: Logical knowledge provided by expert information.

5 Expert knowledge through literature information

5.1 Introduction

Within the previous chapter, we have introduced a methodology to leverage empirical priors in the form of relational information to improve decision support outcomes on image processing. The relational information is characterized as sample-specific meta information for e.g. a patient which can be set into context with the meta data from other patients. By analyzing the relational information, it is possible to perform a cohort clustering within a graph structure which provides complementary information to the pure imaging data. A sufficient quality and informativeness of this meta information is however essential for the creation of a meaningful underlying graph structure.

Another form of empirical prior can be provided by logical or expert knowledge. Different from relational knowledge, expert knowledge is not defined from pure sample-to-sample perspective but in our definition instead provides expert-derived rule sets often deduced from empirical experience relevant for the decision support task at hand. In the medical realm, a very common form of expert knowledge is the medical description of disease characteristics and symptoms in if-then data pairs. A medical doctor is trained to perform a diagnosis for a patient based on the characteristics of his or her medical condition and the symptoms which are present. Within the medical literature, these witnessed conditions and symptoms are linked to likely conditions and allow the diagnosis of the disease or medical situation of the patient. It is therefore intuitive to assume that besides the patient relational information also this expert knowledge could assist a machine learning based approach in its prediction task. The posed challenge is how to effectively incorporate this expert knowledge into a network learning process. Additionally, the combination with existing relational knowledge should be optimized as well.

To address these questions, we would like to develop and evaluate approaches in a medical field with a high emphasis on fast literature symptom to medical condition matching. One nicely suiting medical area, for which fast rule-based decision support is an essential requirement, is the area of toxicology and literature-informed toxin classification, as it is performed in poison control centers. Within the next section, we will provide an overview of the medical background of poison control and related work performed in the area of literature-informed toxin classification. Then we will introduce a new concept called ToxNet developed for the optimized leveraging of relational and expert information for the purpose of fast poison classification.

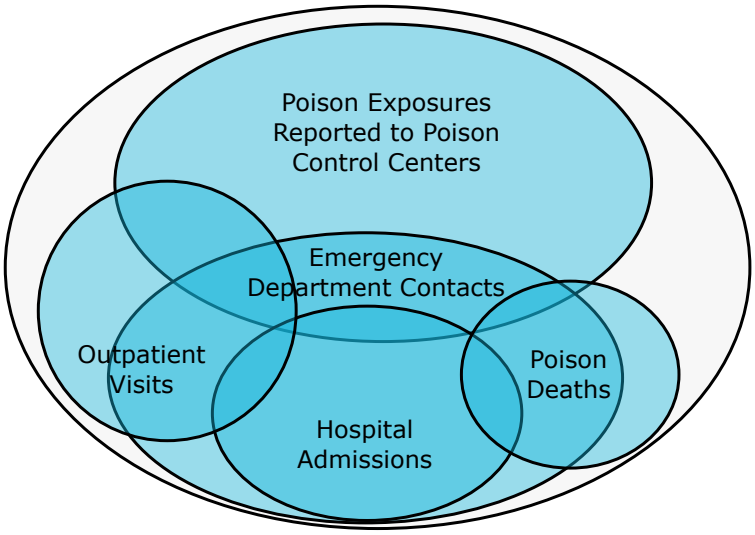


Fig. 5.1. Poison exposures in the United States [49]. The corresponding circles provide a relation between the occurrences and relationships of poison exposures reported to poison control centers, emergency department contacts, outpatient visits, hospital admissions and poison deaths (not drawn to scale).

5.2 Literature-informed toxin classification

5.2.1 History and medical background of poison control

Impact of intoxication and drug abuse

Intoxication and drug abuse are unfortunately still a major impacting factor on global health and life expectancy. From a clinical perspective, human poisoning subsumes any toxin-related injury [49]. The term poisoning or intoxication can therefore refer to the toxic effects of classic toxins, but also an overdose of a prescription medication or other substances like herbal medicine and dietary supplements. Correspondingly, the source of the toxin can be chemical, naturally occurring in plants or animals or being part of a mineral substance.

The general impact of any form of disease or health-related disability like intoxication is collected within the quantity "disability-adjusted life-years" (DALY). The quantity DALY accumulates all health related issues as well as life-years lost into one quantity to allow a comparison between different global regions. Stated in [59], in 2016 alcohol abuse alone lead to 2.8 million deaths worldwide and resulted in 99.2 million DALYs, which corresponds to 4.2% of all DALYs. Including other drugs into the statistic, additional 451,800 deaths and 31.8 million DALYs are added to these numbers. Within case documentation, different terminologies might be used for documentation (e.g. "envenomation" in case of insect stings) which complicates a structured statistical evaluation. Within Fig. 5.1, a rough overview about poison exposures and corresponding outcomes in the US in 2004 is given.

The history of poison control centers

Drug abuse and other forms of intoxication are an important aspect of the medical care sector. Nevertheless, at the beginning of the 20th century the modern medical field of toxicology

was not established yet in Europe or the US [49, 92]. There was little to no oversight of pharmaceutical companies, their manufacturing or drug labeling and studies about the treatment of poisoning and intoxication cases were still rare and not advanced. Especially for children, this situation led to frequent incidents with poisonous compounds. A survey on this matter resulted in the problematic finding that a large percentage of childhood injuries and deaths was caused by accidental poisoning [205]. The resulting increasing concern about food and drug safety in the US led to the Pure Food and Drug Act in 1906, the "Wiley Act" [140]. As a result of this act, the Food, Drug and Insecticide Administration was founded, which since 1930 is famously known under the name Food and Drug Administration (FDA). Over the years, more legislation followed to increase the protection of the population like the "Food, Drug and Cosmetic Act" in 1938 and the Child Protection Act in 1966. Pushed by the efforts of people like Jay Arena, M.D. and Louis Gdalman, R.Ph. a first toxicological information system was established, which classified over 9,000 consumer products [49]. Especially Gdalman paved the way for the founding of modern poison control by privately offering a 24 hours phone service to provide advice for intoxication cases. In consequence, he and Edward Press, M.D. developed the first official poison control center including a phone service and a data collection form in 1953 [26]. To account for the urgency of intoxication cases, poison control centers (PCCs) provide a direct assistance to the caller. Independent from the level of expertise, a person can contact the PCC and request assistance in the handling and treatment of an intoxication case at hand. This service enabled the possibility of early patient treatment and at the same time prevented unnecessary medical visits.

Within Europe, already in 1949 after the second world war, there exist reports about a 100 bed hospital ward dedicated to the clinical treatment of intoxication patients in Budapest, at the same time a poison information service was established in the General Infirmary Hospital in Leeds [92]. This was followed up with Gaultier's emergency service in Paris in 1959 and the founding of the National Poison Information Service in Great Britain in 1963. One year later, the European Association for Poison Control Centers was created. In 1965, a conference in Istanbul was used to note done the concepts most relevant for the creation of further PCCs. From two of the main points determined within this conference the necessity and general purpose of PCCs can be nicely deduced [92]:

- "Considering the constantly increasing number of chemicals and, consequently, of drugs, household products, pesticides, cosmetics, and industrial products that can induce acute intoxication under certain circumstances, it is quite impossible for physicians to insure proper care of the victims of such intoxication without full information on the product involved."

- "Poison information centers are the best way to provide this information. They can offer round-the-clock information service, sometimes to the public as well as to physicians."

Following their recommendation, a poison control center should ideally consist of a treatment center, a laboratory for toxicological analysis, a research department and an information office. In the US, already eleven centers were established in Chicago alone by 1954, a number which nationally quickly increased to 265 in 1958 and over 600 in 1970 [205]. To standardize the process the National Clearinghouse for Poison Control Centers (NCHPCC) and American Association of Poison Control Centers (AAPCC) was founded in 1958 [49]. The developed

certification system of the centers and the correspondingly required verification lead to a rapid drop in the number of PCCs, leaving 64 remaining centers in 2002. At the same time, with the rising interest in the field of poison control, the medical field of toxicology became more and more established, with PCCs beginning to offer fellowship programs for physicians. In 1994, an official qualification in the sub-specialty of medical toxicology became available, a medical field responsible for the research, diagnosis and treatment of poisoning.

Challenges of diagnosis for poison control

Modern poison control centers are an essential factor of the medical care sector. Especially severe intoxication cases require immediate attention by medical personal. A fast diagnosis and corresponding treatment are crucial to prevent permanent organ damage or even death [134]. The consequent establishment of PCCs has substantially reduced poison-related mortality by accounting for the urgency of intoxication cases. In the US, over 2 million people contact PCCs annually to receive advice on an intoxication case, showing the high need for this service [243]. For many of these cases, the substance which is causing the intoxication is known so that the request is focusing on treatment advice. For some cases, the responsible substance is however unknown, requiring an immediate diagnosis and poison classification by the medical expert based on the described symptoms alone. The term symptom refers to any physical or mental condition that is induced by the poison and shows itself as an effect of this poison on the patient. This time-sensitive diagnosis poses a challenging task due to several reasons, especially for still inexperienced doctors.

The first reason is the not preventable mismatch between the symptom descriptions by the caller and the symptoms described within the literature, the expert knowledge. Symptoms can be experienced subjectively and wrongly described due to the stressful situation of the caller. This can potentially highly increase the difficulty of an accurate symptom matching. Regional and institutional differences in symptom description add to this difficulty. Secondly, every intoxication has an individual character and the patient might react different to the toxin than expected. Further medical conditions might result in symptoms which are not related to the intoxication. Thirdly, the medical situation of a patient is influenced by many more factors than the intoxication. Factors like his age or clinical sex can have an influence on experienced symptoms, additionally external factors like the weekday of the intoxication can have a strong statistical information. This relational information which complements the expert information is however not taken into account in a systematic and quantitative way during the diagnosis, since it does not directly correspond to medical information and might be regionally different.

As a result, the medical experts in PCCs find themselves repeatedly confronted with time-sensitive and at the same time very challenging diagnosis and treatment advice requests. Over the years, there have been multiple attempts to assist the medical experts in this challenging task. The following section will provide an overview about this area.

5.2.2 Development of poison prediction

To structure the provided treatment advice of the PCCs, early centers relied on so-called drug information and poisoning management cards. The cards would contain treatment advice

under the assumption that the corresponding poison is known. These cards were later replaced by microfiche as a reaction to the increasing amount of data required. Databases like Toxifile and management cards published by the NCHPCC provided collections of poison reports to physicians, but were only temporarily updated. In 1973, with POISINDEX a first larger commercial product was available which provided a treatment algorithm, the "management", for every stored entry of consumer product [199]. In 1985, it was restructured for usage on PC. In 1983, the AAPCC began a structured collection of intoxication data by using its developed Toxic Exposure Surveillance System (TESS) [243]. It was connected to the product codes of POISINDEX and allowed the digital data collection for PCC calls. For drug identification, the database Drugdex was connected.

All these approaches were designed to assist the physician in the provision of treatment advice, but as described were operated under the assumption that the corresponding source of poisoning is known. Additionally, it did not consider intoxication details like the dose of poison, its aetiology or the potentially more complicated treatment necessary for substances with multiple toxins present. To assist the medical expert with the diagnosis and treatment advice, computer-aided decision support systems were developed. One of the first works addressing the issue of multiple toxins present was performed by [197] in 1985. They developed a rule-based algorithm following simple if-then statements to provide detailed information to the user about the toxicity of a product in the data-base to account for the problems of the POISINDEX database that required a longer search process as well as the consideration of potential interactions between toxic ingredients. As output, the algorithm would provide advice how to act against the poisonous substance. This work was continued by [55] within the expert system SETH, which is based on a level graph approach. Each level represents a step of the reasoning, starting with initial conclusions on delay, dose and sign, resulting in a final conclusion which defines the required monitoring and treatment advice for the patient, the last layer of the algorithm. As input, the system requires patient data as well detailed information about the drug including dose information. The system performs a rule-based analysis of these inputs to determine a corresponding required treatment.

Both approaches [197] and [55] did however not address the main challenge of poison prediction in the case of an unknown substance as described in Sec. 5.2.1. This changed with the work of [4], who started to work on a prediction approach called Induction and Reasoning from Cases (Inreca), focusing on medical decision support based on case based reasoning (CBR). The work was performed within a European project and a first application was targeted on diagnosing poison cases caused by psychotropes. The concept of CBR solves a problem based on the solution for similar past cases. Every case consist of a problem (symptoms) and the corresponding solution (diagnosis and treatment). For the project, eight poison classes were used: Ethanol, Barbiturates, Methanol, Amynotryptelene, Malathion, Acetic acid, Parathion and Dichloroethane. Additionally, 86 symptom classes for patients were considered as potential input parameters. Interestingly, patient meta information like age and clinical sex were treated as the same kind of input category as actual poisoning symptoms like sweating. So, although already considering meta information, this data is probably not used in the most effective way. Given a set of poisoning cases, the algorithm, which is based on a decision tree, requests the status of different patient attributes to structure the cases within different buckets. Therefore, after this preparation step, the dataset has been distributed into different subgroups at the leaf of each tree branch. For a new case, it is now determined into which

bucket the new case should be placed based on the decision rules at each node of the tree. Then, this process is recursively repeated for the parent nodes to determine which additional child nodes should be visited and which not using a search algorithm within the tree based on Ball-Overlap-Bounds and Ball-Within-Bounds. A similarity metric for all cases within the visited buckets is kept in order to determine the case which is most similar to the evaluated patient case. For an increasing amount of data, [4] then further proposed the gradual shift of the CBR approach to the training of a decision tree with the decision classes instead of the cases in the buckets. This requires only one call of the tree, essentially corresponding to an inductive machine learning approach. From the description, it however becomes clear that the approach is not incorporating explicit literature information and expert knowledge but instead relies on the patient cases and their case reportings.

This changed with the ESTHER algorithm developed by [137], which was designed for the classification and treatment advice on poisonings resulting from medical drugs. Within the algorithm, the available medicines are allocated into 19 groups. Overall, 63 diagnostic signs are determined and an individual value scale is defined for every value. For e.g. the diagnostic sign pulse rate, the options tachycardia, normal pulse or bradycardia are possible. Within the algorithm, a decision base is designed resulting from the expert feedback of physicians to receive a rule set for which sign values correspond to which medicine class. Since these definitions can overlap additionally a level of typicality for the specific intoxication is given to each sign value. The system then allows for the insertion of data received from a new patient, provides the corresponding poison prediction and provides treatment advice.

The previously discussed approaches mainly focused on the classification of medical drugs. In 2010, [17] introduced their Expert System for Poisoning (ESP), which additionally targeted other chemical compounds. With the system, a medical attendant enters poisoning related symptoms into the data base. These symptoms are converted into decision rules and submitted to the inference engine. The engine will then provide a decision based on these rules and other rules in the knowledge base using CLIPS [272]. It will then return a list of potential poisonings, additionally it will provide further treatment advice. The knowledge base encodes connections between symptoms and poisons. The inference engine determines a hit ratio which measures the intersection of the set of input symptoms with the symptoms given by the literature for the poison. This hit ratio can be described in simple terms as the ratio n/N of the number of symptoms n present for the current patient to the number of all symptoms N. At the same time, with the knowledge acquisition tool it is possible for the expert to enter and update poison specifications.

In 2017, [153] developed the Antidote Application (AA), a methodology containing 29 antidote "algorithms" to provide the correct treatment and dose advice for a given poisoning and the corresponding antidote. 19 different poison classes are considered for this purpose, consisting of 31 specific toxins. The corresponding poison needs to be provided as an input to the system, it is therefore not usable for the purpose of poison prediction. Additionally, the patient age, weight and height are required. It is interesting that, like in [4], patient meta information is considered, but only used as a standard input value. Here, is it used to determine the required dosages of the antidote within the treatment algorithm. Since this output has crucial implications on the treatment procedure, a safety measure is implemented to detect inconsistencies within the input values.

Tab. 5.1. Comparison on decision support systems for poison control. The abbreviations correspond to the following aspects: Treat. P. = Treatment prediction is performed, Poison P. = Poison prediction is performed, EK used = Expert knowledge used, RK used = Relational knowledge used, ML based = Machine learning based. Many of the previously developed approaches are focusing on intoxication treatment device, only some include a prediction of the underlying poison. Machine learning is only involved in one of the approaches.

Method	Treat. P.	Poison P.	EK used	RK used	ML based
POISINDEX [199]	✓	×	✓	×	×
[197]	✓	×	✓	×	×
SETH [55]	✓	×	✓	×	×
Inreca [4]	✓	✓	×	✓	✓
ESTHER [137]	✓	✓	✓	×	×
ESP [17]	✓	✓	✓	×	×
AA [153]	✓	×	✓	✓	×

We can summarize the previously described decision support systems in Tab. 5.1 to receive an overview about their advantages and drawbacks. It becomes immediately clear that previously developed decision support systems for toxicology are focusing on the provision of treatment advice to the expert. Here, they mostly rely on the usage of expert knowledge given direct by the medical physician or collected from different literature and online sources. Only for two approaches, patient information beyond the symptoms are considered, they are however only used as further direct input into the system [4, 153]. Only in one of the approaches, a machine learning methodology in form of a decision tree was introduced [4]. Additionally, as described above, we want to focus on the challenging task of poison prediction, since reliable intoxication classification constitutes a very important aspect of a successful treatment advice and can be challenging even for experienced doctors, as we are going to show later. However, only three of the previously developed approaches perform a poison classification next to the provided treatment advice, although physicians could highly benefit from decision support in this area for challenging cases.

Considering the approaches, which we have previously addressed, it becomes clear that so far no effective combination of relational and expert knowledge has taken place for the field of decision support, especially in the area of poison prediction. Within the following sections, we will therefore develop and evaluate a new methodology called ToxNet to address these issue. Considering the promising results when dealing with relational information from the previous chapter and other works in the field [58, 128, 185], we will again focus on the usage of Graph Convolutional Networks (CGNNs) especially for the effective incorporation of patient relational knowledge. This concept allows us to treat patient information and circumstantial information (like the weekday of intoxication) as a connecting factor between cases by treating the cases and their symptom features as nodes in the graph and the similarity between their meta information as edges.

5.3 ToxNet - Poison classification under expert knowledge inclusion

5.3.1 General motivation of developed approach

As was described in detail in Sec. 3.6, CGNNs have been successfully applied in the medical realm for multiple approaches. The two main requirements to our methodology are time-efficiency (due to the urgency of poison diagnosis) and performance stability (due to the consequential treatment resulting from the prediction). For time-efficiency, we will rely on spatial graph approaches as described in Sec. 3.1.4, since they allow an easy inductive prediction process compared to spectral graph methods like [29] as described within their comparison in Sec. 3.3.5. Additionally, local graph inaccuracies can potentially harm the performance for a set of patients, so that compensating measures are required. It was shown that attention mechanisms [44, 148] improve filtering by weighting similarity scores between nodes based on node features. This setup helps to compensate for locally inaccurate graph structures and stabilizes the prediction outcome and correspondingly performance. The CGNN layers of our developed methodology will therefore again be based on the spatial graph approach Graph Attention Networks [231] which is (beginning of 2020) one of the strongest spatial graph networks using attention. Within the following sections, we will motivate the design choices of our new approach, which leverages structured incorporation of patient meta information and expert literature information to significantly boost performance as our experiments will demonstrate.

The main challenge for our newly developed methodology is the consistent incorporation of both relational and literature information within one network structure. In general, we can separate the data into three separate aspects: the concrete symptoms experienced by the patient, which we encode in a symptom vector **P**, the literature symptoms which are related to a toxin according to the expert literature, encoded in literature symptom vector **H** and the additional patient meta information **Q**. As described before, a symptom constitutes any physical or mental condition change induce by the poison to the patient like e.g. sweating or somnolence. The first challenge derives from the fact that the two vector groups **P** and **H** are not necessarily of the same dimension. There might be symptoms reported for a patient which are not described for any toxin in the literature. At the same time, a symptom might be described in a different way and therefore is not directly matchable to the literature description. The extend of this mismatch can potentially be large. Our developed methodology therefore needs to provide a way to perform a consistent matching to the literature provided symptoms while at the same time still allow the consideration of the remaining symptom descriptions not part of the literature.

The second challenge stems from the effective incorporation of the additional relational information. This can come in the form of patient-specific information like age and clinical sex as well as situation-specific information like aetiologie and the weekday of the intoxication. This information provides a second impacting factor that requires its own form of processing. As described previously we want to use this information to cluster the patient cases within a graph structure for the processing within a CGNN.

Therefore, while the focus for the expert knowledge evaluation needs to lie on a condensing of the information onto literature-reported symptoms to allow for a comparison, the relational knowledge should focus on the holistic reporting of the case to find underlying statistical relations in the data which are complementary to the literature information. From this description of the required processing pipelines, it is intuitive to process the patient information in two parallel pipelines which should later combine their findings. We therefore created a methodology called ToxNet based on two parallel network branches, which we presented at the MICCAI 2020 conference [33]. We are going to describe the methodology and its evaluation in detail in the following sections.

5.3.2 Feature setup and graph processing branch

As described in the previous section, the data is distributed into three separate groups: patient symptom vectors **P**, literature symptom vectors **H** and patient meta data vectors **Q**. The symptom vectors follow a binary encoding logic, therefore we provide an entry for every symptom, 1 if the symptom is present, 0 if it is absent. Every patient receives a symptom vector $\vec{p}_i$ encoding the symptoms occurring for this patient and every poison a symptom vector $\vec{h}_i$ according to the literature symptom description. This leads to the overall symptom vector sets $\mathbf{P} = \{\vec{p}_1, \vec{p}_2, ..., \vec{p}_M\}, \vec{p}_i \in \{0,1\}^{F_P}$ and $\mathbf{H} = \{\vec{h}_1, \vec{h}_2, ..., \vec{h}_C\}, \vec{h}_i \in \{0,1\}^{F_H}$, where M is the number of patients, C is the number of poison classes and F_P and F_H are the dimensions of the patient and literature symptom vectors, respectively. For real patient cases, we will also receive symptom descriptions which are not present in the literature. The vector dimensions F_P and F_H therefore refer to the total number of individual symptoms S_P and S_H that are given within all patient cases and poison descriptions respectively. For the patient symptom vectors, all symptoms S_H mentioned in the literature are automatically considered for **P** and a binary entry is reserved in the first F_H entries of every vector $\vec{p}_i \in \mathbf{P}$, we call this sub-vector $\vec{p}_{i,\text{lit}}$ (see Fig. 5.2). Additionally, **P** contains $F_P - F_H$ binary entries for all additional symptoms only mentioned for the patient cases, giving overall symptoms S_P. Therefore, we can define that $F_H < F_P$ and $S_H \subseteq S_P$.

The patient meta information is provided in the vector set $\mathbf{Q} = \{\vec{q}_1, \vec{q}_2, ..., \vec{q}_M\}$, where every vector $\vec{q}_i$ contains the patient's meta data. Additionally, for every patient case i, a poison label $y_i \in \mathbf{Y}$ of the C poison classes is provided. We want to achieve a functional relation $f(\mathbf{P}, G(\mathbf{P}, \mathbf{Q}, \mathbf{E}), \mathbf{H}) : \mathbf{P} \rightarrow \mathbf{Y}$, where $G(\mathbf{P}, \mathbf{Q}, \mathbf{E})$ corresponds to the graph with edges **E** created from meta data similarity. Every vertex in the graph has a representation x_i concatenated from $p_i \in \mathbf{P}$ and $q_i \in \mathbf{Q}$ with dimension F. For the graph processing, the meta information is therefore used both for the graph creation and the feature input.

In order to generate the graph, we define a corresponding similarity function $\text{Sim}(\vec{q}_i, \vec{q}_j)$ to determine if the edge $e_{ij} \in \mathbf{E}$ exists. This similarity function consists of a threshold test for a given threshold vector $\vec{\theta}$. The set of edges and the similarity function are given by Eq. 5.1:

$$e_{ij} \in E \;\forall\; \text{Sim}(\vec{q}_i, \vec{q}_j) = 1 \qquad \text{Sim}(\vec{q}_i, \vec{q}_j) = \begin{cases} 1, & \max(|\vec{q}_i - \vec{q}_j| - \vec{\theta}) \leq 0 \\ 0, & \text{if otherwise} \end{cases} \tag{5.1}$$

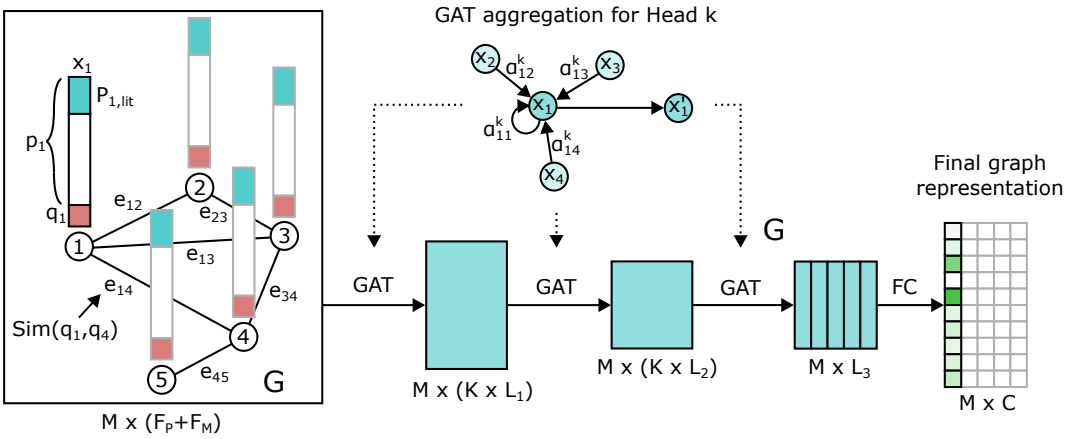


Fig. 5.2. Schematic architecture of ToxNet graph branch. The symptom vectors $\vec{p}_i$ are concatenated with the meta data vectors $\vec{q}_i$ and then processed in graph-based GAT layers. The graph is resulting directly from the similarity function results when comparing meta data for different patients. Here, an example for five vertices is given, the update process for vertex one and head k of the GAT layer is depicted. The final graph representation has already class level dimension due to a proper contribution balancing with the literature branch described in the next section.

An edge is therefore only created, if every entry of the meta data vectors $\vec{q}_i$ and $\vec{q}_j$ is below its corresponding threshold in the threshold vector $\vec{\theta}$. The neighborhood $N(i)$ of patient vector $\vec{x}_i$ is therefore given by $\vec{x}_j \in N(i) \; \forall \; e_{ij} \in \mathbf{E}$.

The designed input vectors are now processed within a CGNN. As described in Sec. 4.3.2, the usage of a CGNN in the form of a GAT allows the inclusion of neighboring information into the prediction process of a patient case. For completion, it is again briefly stated how the GAT layer operates. As shown in Fig. 5.2, we are applying three consecutive GAT layers onto our representation, each resulting in a hidden representation of size L_1, L_2 and L_3 respectively. Within the GAT, a shared learnable linear transformation $\mathbf{W} \in \mathbb{R}^{F' \times F}$ is applied to all patient vectors $\vec{x}_i \in \mathbf{X}$, resulting in a new representation with dimension F'. For every neighboring representation $\vec{x}_j \in N_i$, we are calculating an attention coefficient α using a shared attention mechanism a. Here, we are using the two updated representations $\mathbf{W}\vec{x}_i$ and $\mathbf{W}\vec{x}_j$ to obtain $a(\mathbf{W}\vec{x}_i, \mathbf{W}\vec{x}_j) = \vec{a}^T[\mathbf{W}\vec{x}_i || \mathbf{W}\vec{x}_j]$, where $[\,||\,]$ is the concatenation operation of $\mathbf{W}\vec{x}_i$ and $\mathbf{W}\vec{x}_j$, and $\vec{a} \in \mathbb{R}^{2F'}$ is a single feed-forward layer. To enable non-linearity and achieve a normalization of all calculated coefficients, after applying a leakyReLU activation σ, the softmax function is calculated for all $\vec{x}_j \in N_i$. This provides the attention coefficients α_{ij} given in Eq. 5.2:

$$\alpha_{ij} = \frac{\exp(\sigma(\vec{a}^T([\mathbf{W}\vec{x}_i || \mathbf{W}\vec{x}_j])))}{\sum_{r \in N_i} \exp(\sigma(\vec{a}^T[\mathbf{W}\vec{x}_i || \mathbf{W}\vec{x}_r]))} \tag{5.2}$$

These coefficients can now be used for an attention-weighted aggregation process of the symptom vectors $\vec{x}_j \in N_i$ for the update of $\vec{x}_i$. The concept is also shown in Fig. 5.2. The feature representations $\mathbf{W}\vec{x}_j$ are weighted with the corresponding attention coefficients α_{ij} and added to receive the representation $\vec{x'}_i$. This process can be performed multiple times in parallel, using different learned linear transformations $\mathbf{W}^k$, the overall K heads of the GAT layer. This further assists in the stabilization of the prediction process by mimicking an ensemble of networks. Correspondingly, K different attention coefficients α_{ij}^k are created for every symptom vector $\vec{x}_i$. Since these representations need to be used collectively in the

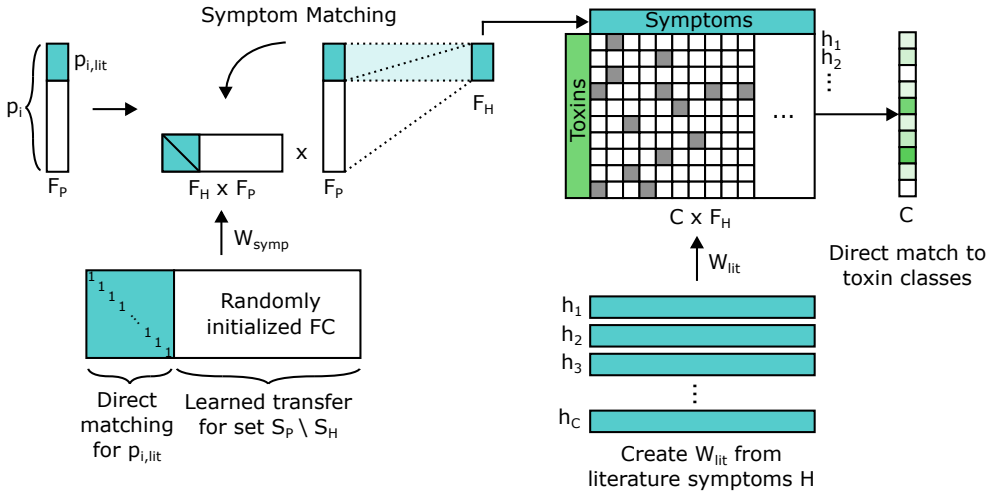


Fig. 5.3. Schematic architecture of ToxNet literature branch. Via a learnable linear transformation $\mathbf{W}_{\text{symp}}$, the symptom vector $\vec{p}_i$ is mapped into literature symptom dimension F_P, essentially encoding all symptoms into literature dimension. The mapping from literature symptom to literature symptom is preserved. Then, the new representation is processed via a hardcoded transformation $\mathbf{W}_{\text{lit}}$ which contains the literature symptom to poison matching.

next layer, the outcome of every head k is concatenated to receive the final presentation (represented as $||$).

$$\vec{x'}_i = \|_{k=1}^{K} \sigma \left(\sum_{j \in N_i} \alpha_{ij}^k \mathbf{W}^k \vec{x}_j \right) \tag{5.3}$$

where K is the number of used heads and α_{ij}^k is the attention coefficient of head k for the feature update of $\vec{x}_i$ using $\vec{x}_j$.

After the GAT layers have been applied to the symptom vectors, a fully connected layer maps the outcome vector onto its final representation $\vec{y}_{i,GAT}$. The representation size is chosen to match the number of toxin classes C in dimension. This will allow an equal contribution to the final prediction compared to the developed second network branch which we will introduce in the next section.

5.3.3 Literature matching branch

While the graph branch needs to perform an effective inclusion of the relational knowledge, the literature branch needs to optimize the mapping between the reported patient symptoms and the expert literature information. As described in the previous section, every toxin class c_i of all considered toxins C has a reported number of symptoms S_H, which are encoded within the corresponding binary literature symptom vector $\vec{h}_i \in \mathbf{H}$. The number of symptoms in these vectors is identical to all reported unique symptoms S_H for all considered C poison classes. The first F_H entries of the patient symptom vector $\vec{p}_i \in \mathbf{P}$ correspond to these symptoms. The remaining entries of $\vec{p}_i$ are additional symptoms which have been only reported for the patient cases but are not mentioned in the literature. In order to use the expert literature information, we need to provide a direct mapping to the patient reported symptoms. For the F_H entries directly matching S_H this is intuitive, for the remaining symptoms the process is difficult,

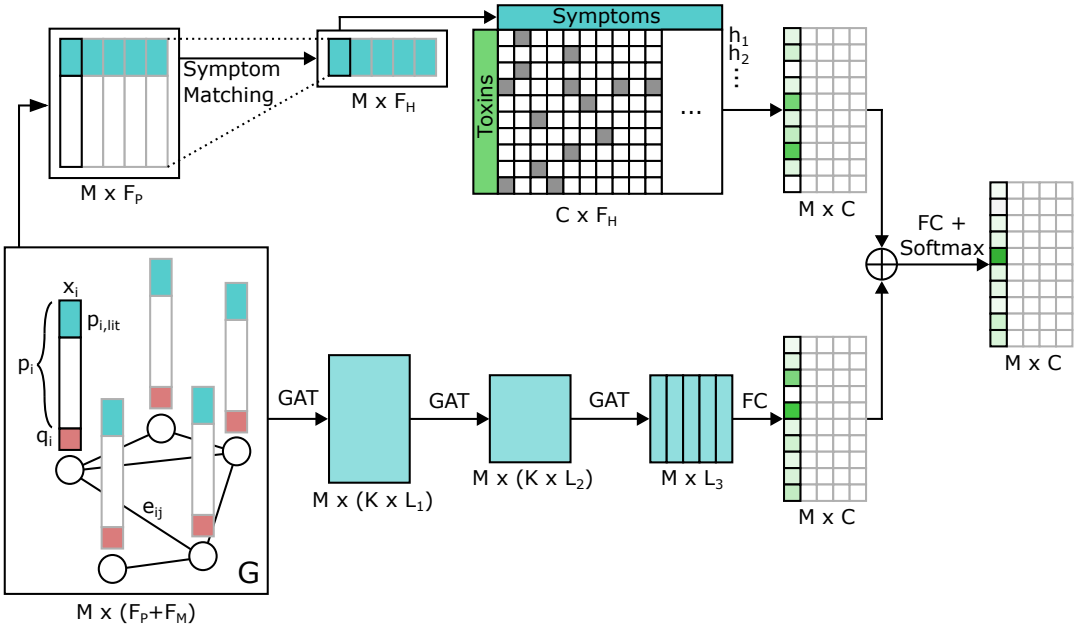


Fig. 5.4. Schematic architecture of ToxNet. The symptom vectors are processed in the graph-based GAT layers and the literature matching network in parallel. Both representations are fused for the final poison prediction after they have interacted with the individual forms of prior knowledge.

since no direct intuition exists. We therefore design a specialized symptom matching layer $\mathbf{W}_{\text{symp}} \in \mathbb{R}^{F_H \times F_P}$ which learns the complete matching of the patient symptom vectors $\vec{p}_i$ to the literature symptoms (the patient meta data is not considered in this branch). Since the first F_H entries of $\vec{p}_i$ correspond to S_H, we can directly map them to the literature symptoms. Within $\mathbf{W}_{\text{symp}}$, the first square of entries with dimension $F_H \times F_H$ is initialized as the unity matrix I_{F_H}. The diagonal remains frozen to ones, since the mapping from a symptom of S_H to itself should remain intact. The rest of the layer matrix is randomly initialized and trainable via gradient backpropagation (see Fig. 5.3). It can therefore learn a contribution of the symptoms in S_P not reported in the literature to S_H and also an interdependence of symptoms in S_H. These can be very helpful for e.g. the case of a strong coexistence of two symptoms. If only one of them is reported, the other one is triggered as well. The transformation therefore provides an interpretable mapping of the patient symptom to the literature symptom space while conserving the direct literature-based information. The resulting vector now has the dimension F_H of the literature symptoms.

Within a second network layer, we now encode a frozen representation of the expert knowledge. This is achieved by creating a matrix $\mathbf{W}_{\text{lit}} \in \mathbb{R}^{C \times F_H}$ from the combined literature symptom vectors $\vec{h}_i \in \mathbf{H}$, where the ith row of the matrix is initialized with the corresponding $\vec{h}_i$. The whole matrix remains constant during the training process and therefore provides a hard-coded symptom to poison matching based on the expert literature information. The resulting outcome vector has the dimension C of considered poison classes and its entries' value size directly corresponds to the level of similarity between incoming symptom vector and literature symptom vector. The overall transformation $\vec{y}_{i,lit} = \mathbf{W}_{\text{lit}} \cdot \sigma(\mathbf{W}_{\text{symp}}\, \vec{p}_i)$ therefore performs a mapping of patient symptoms onto the poison classes under explicit usage of literature information.

5.3.4 Representation fusion

Using the previously described branches, we have received two patient feature representations, one motivated by the relations between the patient cases and a corresponding collective processing within a GCN, the other one by expert knowledge provided from the literature. Both branches therefore have incorporated individual and potentially complementary information into the feature processing. The dimension of both representations is chosen identical for two reasons. First, it allows an equal contribution of both branches to the final representation. Second, the dimension was chosen to be of size C, so that each branch alone is already forced to provide an outcome representative of the final prediction task. For the final outcome, the GAT and literature representations $\vec{y}_{i,GAT}$ and $\vec{y}_{i,lit}$ are concatenated, activated and processed through a last fully connected learnable layer and a softmax function onto the class output $\vec{y}_i$. The full network setup is shown in Fig. 5.4. During training, this prediction outcome for a patient is now compared against the patient's class label within a cross-entropy loss function. The corresponding loss gradients are backpropagated through the network to both the graph and literature branch.

Within the following sections, we will analyze our developed methodology in detail based on real poisoning cases from the poison control center (PCC) in Munich, one of the largest PCCs in Europe. First, we will perform an ablation study to evaluate the individual aspects of our methodology, then we will compare our method against the poison classification performance of actual medical experts.

5.4 Experimental setup

5.4.1 Dataset

In order to evaluate our methodology, we rely on real patient cases which have been processed within the PCC in Munich. The valuable dataset was created by our medical experts from the PCC focusing on the most important intoxications. Overall 8995 patient from the years 2001-2019 were extracted from the database. All cases were mono-intoxications (only one poison present) and the toxin has been determined, so our data is fully labeled. The ten toxins given in Tab. 5.2 are considered for our analysis. These toxin groups were selected based on one or multiple of the following criteria: 1) The intoxications occur very frequently and are therefore important for a classification methodology. 2) The intoxications have a distinct treatment and intervention necessity. 3) They result in a severe intoxication. 4) They require a very specific treatment and are therefore important to detect. As a result, the corresponding dataset shows an imbalance between the different classes. Poisonings involving alcohol (ethanol) are e.g. very common while other intoxications occur more rarely. Within the dataset, $F_P = 607$ different unique symptoms S_P were reported within all 8995 patient cases (including all literature symptoms). At the same time, only $F_H = 140$ unique symptoms S_H are reported for the ten chosen poison classes in the literature. The mismatch between the reported patient symptoms and the literature is therefore substantial, showing again the mentioned requirement for a strong matching algorithm.

Tab. 5.2. Type and number of poisons present in the dataset. The poisons were chosen based on their frequency of occurrence and the their difference in treatment necessities.

Poison	Cases	Comment
ACE inhibitors	119	/
Acetaminophen	1376	/
Antidepressants	1073	selective serotonin re-uptake-inhibitors
Benzodiazepines	577	/
Beta blockers	288	/
Calcium channel antagonists	75	/
Cocaine	256	/
Ethanol	2890	/
NSAIDs	1462	excluding acetaminophen
Opiates	879	/

Additionally, it needs to be considered that the symptom descriptions especially within the literature are not initially designed to be used in a computational way, but instead should provide guidance to a medical expert who wants to receive information on specific toxin effects. Within the reporting, there are therefore simplifications and generalizations which are clear to the expert but pose a challenge to a computational system. Considering for example the effect of a poison on the state of consciousness, within our literature information for the poison "Ethanol" only the strongest impact is reported, which is "coma". It is however obvious that also intermediate states like somnolence are possible. For a computational system, this black and white reporting however poses a larger challenge, which we aim to address with our literature matching branch. Another problem stems from the way that data is recorded within a PCC. The treating doctor will note down the occurring case and its symptoms, but for the moment focus on the treatment advice to the caller for obvious reasons. The data is then later transferred into the database. From this manual process, three major sources of noise are introduced into the data:

- Spelling mistakes: These come in the form of both accidental typing errors and missing letters as well as misspellings of difficult scientific terms.
- Spelling variants: Some medical terms actually have multiple ways of spelling, resulting in different terms meaning the same symptom.
- Term variants: For some symptoms, the corresponding condition is written down in different languages, additionally some experts note down the layman term, others the scientific term for the same symptom.

All this impacting ambiguities needed to be corrected manually for the dataset, often under consideration of expert feedback, posing a substantial amount of work. After this cleaning process, 6444 patients had symptoms reported for which at least one matched a literature

Tab. 5.3. Available meta information for toxicology dataset.

Meta data	Examples	Meta data	Examples
Year	2018, 2019	Age group	teenager, adult
Month	January, March	clinical sex	male, female
Day	17., 31.	Weight group	overweight, normal
Weekday	Monday, Tuesday	Latency unit	days, hours
Hour	16, 21	Latency value	3, 20
Country	Germany, Austria	Severity	critical, medium
ZIP code	80, 83 (1st 2 digits)	Point of entry	oral, trans-mucous
Caller	layman, medical doctor	Aetiology	suidice attempt

symptom. Still, 2551 patients remained for which no reported symptom matched any symptom reported in the literature. Additionally, for the 6444 patients 3097 had symptoms reported for which no symptom matched the symptoms reported in the literature for the corresponding poison. This mismatch can be linked to two mechanisms. First, as described before, the literature symptoms do not consider all levels of a condition and therefore e.g. somnolence does not occur for an ethanol poisoning. Secondly, not every reported symptom is resulting from the poisoning. This idea can be quite impressively shown within the statistics of the cases with ACE inhibitor poisoning. ACE inhibitors are used for the treatment of blood hypertension and therefore can result in hypotension. Correspondingly, hypotension is noted as a literature symptom, hypertension is not, since ACE inhibitors will not cause hypertension. The leading symptom among all reported cases is however hypertension. This is not due to the intake of the drug but due to the simple fact that most people using ACE inhibitors are suffering from hypertension. Therefore, the reported and often occurring symptom hypertension does not match the literature (correctly so). Here, intelligent algorithms can assist in interpreting these underlying and sometimes at first not intuitive relations.

Complementary to the symptom information, every intoxication case also provides a collection of meta information. Within Tab. 5.3, an overview about all available meta information is given. The data was not complete, for various cases certain fields were left empty. For our methodology, we are relying on the patient age group (child, adult, elder), clinical sex, aetiology, point of entry and week day and year of intoxication. These parameters were determined within a hyperparameter analysis and resulted in best model performance. They will be used in the graph branch to set up the graph structure.

5.4.2 Affinity graph creation

The graph used within the graph processing branch of ToxNet is based on the described meta information for every patient. The meta data for the patient data vector $\vec{x}_i$ and $\vec{x}_j$ has to fulfill the similarity criterion given in Eq. 5.1 with respect to the defined threshold vector. Following a hyper parameter analysis, we have determined that the strongest results were achieved for a very strict thresholding vector, namely the zero vector. Using this vector within

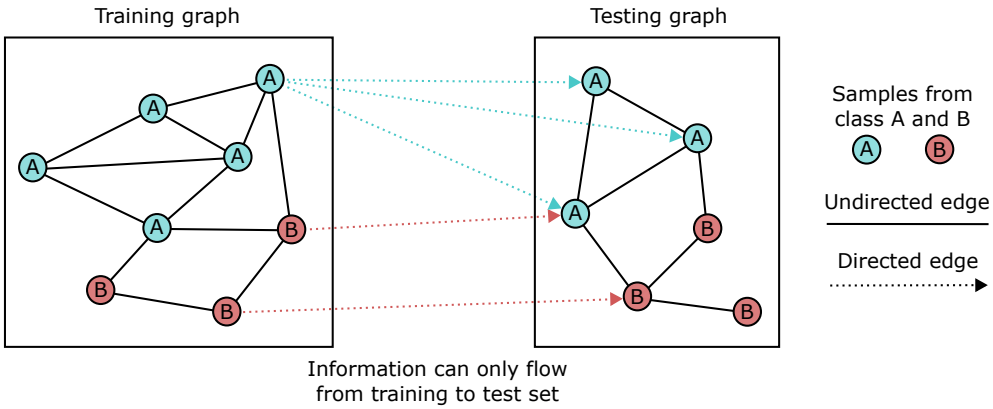


Fig. 5.5. Schematic description of affinity graph creation. Within the individual sets, undirected connections are used (if $e_{ij} \in \mathbf{E}$ then $e_{ji} \in \mathbf{E}$). In between sets, only directed connections from the training to the test set and training to the validation set exist. Like this, a graph approach can be used even if only one test sample is present due to the training graph background. At the same time, any information leakage from the test set to the training set is prevented (if $e_{ij} \in \mathbf{E}$ then $e_{ji} \notin \mathbf{E}$).

Eq. 5.1 results in isolated fully connected graph clusters, the graph is therefore very sparse, the corresponding connections are however created between very similar patients leading to over 50% of edges between cases with the same poisoning. This seems to increase the value of the graph structure so that it is actually useful for the prediction task as will be shown in the results.

One very important aspect was considered for the connection of training, validation and testing samples. Our developed methodology should be able to perform a graph-informed prediction for an individual new patient case. Every new case effectively corresponds to a new testing set. If we would only connect training with training and testing with testing patients, this would work for a larger testing set. For an individual case however, no graph would be left, since the testing set consists of one case. Additionally, the network has trained on the training data and could potentially benefit, if it could take parts of this information into an inference process. Therefore, we establish two types of connections within our graph. Within the respective sets (training, validation and testing), we have undirected edges in the graph. This means that for the corresponding edge sets $\mathbf{E}$ we can define that if $e_{ij} \in \mathbf{E}$ then $e_{ji} \in \mathbf{E}$ as well. At the same time, we connect the validation and testing samples to the training set via directed edges. If there is a connected training vector $\vec{x}_i$ and testing vector $\vec{x}_j$, then $e_{ji} \in \mathbf{E}$ but $e_{ij} \notin \mathbf{E}$. Therefore, information can travel from training to testing samples, but no knowledge can travel from testing to training samples, effectively preventing any label leakage. Using this setup, the training set provides the graph background during inference, even if only one new case is present.

5.4.3 Network setup

For the two branches of ToxNet (graph-based branch and literature-based branch), we chose the following settings:

For the graph processing branch, every GAT layer consists of 5 heads. The first GAT layer provides $L_1 = 128$ features per head, resulting in a $128 \cdot 5 = 640$ hidden representation via concatenation. The second layer decreases this to a $L_2 = 64$ and correspondingly $64 \cdot 5 = 320$ feature representation. The last layer repeats this process with $L_3 = 64$, but averages the outcome of the head to a 64 feature representation. After activation with an ELU function, the last fully connected layer with dimension 10×64 brings the dimension down to 10.

For the literature branch, the network dimensions are already given by the available data. For $\mathbf{W}_{\text{symp}}$ a matrix with dimensions 140×607 is chosen to map all reported unique symptoms to all unique literature symptoms. After activation, the matrix $\mathbf{W}_{\text{lit}}$ with dimensions 10×140 maps this representation down to toxin class level. Since $\mathbf{W}_{\text{lit}}$ consists of the binary vectors $\mathbf{H}$, this multiplication sums up the contributions of all symptoms which are reported in the literature and also present in the incoming vector. Both graph and literature representation are then concatenated and processed in a final fully connected layer with dimension 10×20.

For the training process, an Adam optimizer with the cross-entropy loss function and a learning rate of 0.001 was used. The weight decay was set to 5e-4. As an activation function between layers, the ELU function was used, no dropout has been applied.

5.4.4 Training and experiment setup

While within the second part of the evaluation a performance comparison against medical doctors is conducted, the first part focuses on the detailed evaluation of the proposed methodology ToxNet itself. Here, we will perform an ablation study to analyze the impact of different parts of the method. Within this evaluation, we refer with "GAT" to the pure graph processing branch and with "LitMatch" to the pure literature matching branch, each used on their own. Additionally, we will compare against alternative prediction approaches:

- "Naive Matching": This method refers to the simplest possible approach. Here, the number of matching symptoms to the literature for every of the ten poisons is counted and the poison with the most hits is chosen. If there is a tie between multiple poisons, the poison with less reported symptoms is chosen. For cases without any matching symptom no evaluation is performed.

- "Decision Tree": For this approach, a decision tree is trained on an artificial dataset of patients we created. The artificial dataset contains patients which show a random subgroup of symptoms for a poison reported in the literature and obtain the corresponding poison as label. The decision tree should therefore learn from the artificial data to predict patients in the same way as the literature does. The reason, why we created this artificial dataset is that a training on the ten sets of reported symptoms for the considered toxins (essential a training on one sample per class) did not seem feasible.

- "MLPLit": Here, we train a standard multilayer perceptron on the patient symptom dataset $\mathbf{P}$. The number and dimension of the layers is chosen to correspond to the number of learnable parameters of the literature matching branch for a fair comparison.

The two layer sizes are correspondingly chosen as $140 \times F_P$ and 10×140. All training hyperparameters were kept identical to the ones of the ToxNet training.

- "RandomMatch": Again, this setup corresponds to the literature matching branch, but instead of matrix $\mathbf{W}_{\text{lit}}$ being initialized from the literature symptom vectors $\mathbf{H}$, the initialization is performed randomly. Only the amount of symptoms per toxin class is kept identical to the literature.

- "DataMatch": In the previous data analysis in Sec. 5.4.1, we have seen that the informativeness of the literature knowledge with respect to the prediction task seems to be rather limited. We therefore create an artificial literature which is fully based on the patient data. To achieve this, we again use the LitMatch setup, but the matrix $\mathbf{W}_{\text{lit}}$ is initialized as follows: For every poison row and symptom column, the corresponding entry is set to the number of patients with the poisoning who experienced this symptom. Then a softmax function is applied over the complete row. The resulting matrix is therefore optimized towards the prediction on the data.

- "MLP with meta": This approach has the same setting as "MLP", but the meta data of every patient is concatenated to the symptom vector. Therefore the full vector $\vec{x}_i$ is used as input for a patient.

- "ToxNet (S)": This approach provides a modification of the ToxNet network structure to allow for a successive instead of a parallel processing of the inputs using multiple loss functions. The literature and non-literature symptoms are split into two separate input vectors. The literature vector is processed via a 140×140 matrix corresponding to the unity matrix part of $\mathbf{W}_{symp}$. The outcome is used as input for $\mathbf{W}_{lit}$ to receive the literature informed outcome $\vec{y}_{lit}$. At the same time, the non-literature symptoms are processed via a 64×467 fully connected layer. This is processed within another fully connected layer of size 10×64 to receive the symptom outcome $\vec{y}_{symp}$. Both first representations of size 140 and 64 are additionally concatenated together with the meta information vector and processed via the GAT pipeline to receive the final prediction $\vec{y}$. To all three outcomes $\vec{y}_{lit}$, $\vec{y}_{symp}$ and $\vec{y}$ the cross-entropy loss is applied using the provided ground truth label. The loss contributions are weighted with 0.1, 0.1 and 0.8 respectively.

For all experiments, a 10-fold cross validation was used. For every of the 10 runs, 10% of the data is used as test data, of the remaining 90 % of data 80 % is used for training and 20 % is used for validation. Like this, the whole dataset can be used for an evaluation process on unseen data, enhancing the statistical meaningfulness of our findings. After performing the described ablation experiments and comparisons against other methods, we will perform an evaluation against 10 medical doctors from the poison control center in Munich. They will classify the same unseen subset of data as ToxNet. The subset of the data used for this comparison is divided into two parts. First, every MD receives 25 individual cases for classification, leading to 250 cases in total. Additionally, 25 cases were provided to all doctors to also analyze the variability of doctors for the same prediction task. The subset therefore consists of overall $250 + 25 = 275$ individual poisoning cases. The poisons provided within each questionnaire to the MDs were randomly sampled with respect to the occurrence ratio in

Tab. 5.4. Occurrence ratio of poison within dataset and number of cases in one questionnaire. Overall, 11 questionnaires are designed, 10 are given to the 10 MDs, one questionnaire each, to receive a larger test set of 250 cases. The 11th questionnaire is provided to all MDs to allow for an inter-variability analysis.

Poison	Ratio	# in one questionnaire
ACE inhibitors	1.3 %	1
Acetaminophen	15.3 %	4
Antidepressants	11.9 %	3
Benzodiazepines	6.4 %	1
Beta blockers	3.2 %	1
Calcium channel antagonists	0.8 %	1
Cocaine	2.8 %	1
Ethanol	32.1 %	7
NSAIDs	16.3 %	4
Opiates	9.8 %	2

the overall dataset, increasing the realism of the experiment (see Tab. 5.4). If for a poison a ratio below $1/25 = 4\%$ compared to the overall dataset occurred, still one case was given to the dataset. The MDs are informed about the ten possible poisons. The distribution of cases and the fact that all ten poisons are present in the questionnaire was however not disclosed to the MDs to allow for a fair comparison (ToxNet does not have this information as well). The setup will allow us to evaluate ToxNet against the MDs on a larger test set, while at the same time also receive the inter-variability between MDs to differentiate between easy and difficult prediction cases.

5.5 Results and Discussion

5.5.1 Ablative evaluation of network performance

In a first step, we are going to analyze the performance of our method against different benchmark approaches and the ablation models discussed in Sec. 5.4.4. For all comparisons, the described 10-fold cross validation was performed. As metrics, we will look at the F1 micro and F1 macro score (defined in Eq. 2.15 and Eq. 2.16). The results are shown in Tab. 5.5. The first clear finding we can obtain is the status of the expert literature knowledge. The "Naive matching" approach, where simply the toxin with the most matching symptoms is chosen for every test set, performs very poorly on the dataset. This is a consequence of the already within Sec. 5.4.1 described challenges of the provided expert information. $2551 + 3097 = 5648$ patients had symptoms reported, which either did not even exist in the literature or did not match the symptoms described in the literature for that specific poisoning case. Additionally, the way that symptoms are described within the expert literature is not optimized for the usage within a computational approach. Within Sec. 5.5.2, we are going to further investigate the impact of the literature information and its design.

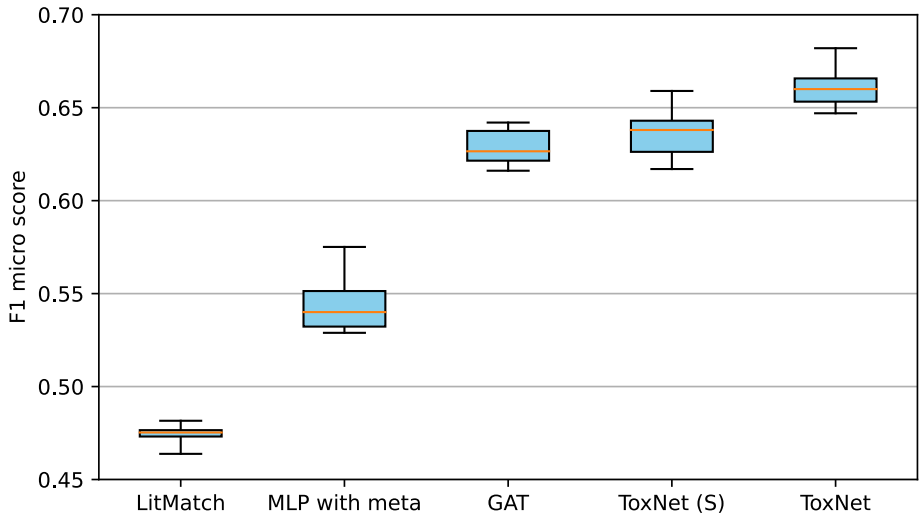


Fig. 5.6. Boxplot comparison of ToxNet prediction performance for 10-fold cross-validation. The box shows the area where the center 50% of results are contained, the whiskers stretch to the minimum and maximum values. ToxNet is compared against its individual branches LitMatch and GAT, additionally the MLP approach using concatenated meta data is depicted. Finally, its sequential setting is compared. ToxNet significantly outperforms all approaches (see. Tab. 5.5).

The decision tree, which was trained on a simulated dataset which completely behaves according to the literature, improves this outcome. It is interesting to see that the performance increases compared to the "Naive Matching" approach, although no additional information was added to the prediction method (so the method still fully relies on the literature information). However, the decision tree has learned to treat the symptoms reported in the literature with a different level of importance to the prediction task (e.g. symptoms almost always occurring do not substantially assist the task). This can explain the improvement in prediction accuracy. Incorporating the literature information into a network branch which at the same time can still learn from the data clearly boost the outcome compared to these baseline approaches, as can be seen for "LitMatch". This setup directly corresponds to the literature matching branch

Tab. 5.5. Performance comparison of ToxNet against different methods for poison prediction. Methods are described in detail in Sec. 5.4.4. We can clearly determine the value that is added to the prediction by the graph processing branch of ToxNet, since the MLP with the same input information is clearly outperformed. Adding the literature matching branch to the method further boosts the performance, ToxNet is significantly outperforming all other approaches (p-value: <0.01 $*$, <0.005 $**$).

Method	F1 Sc. micro	F1 Sc. macro	p-val micro	p-val macro
Naive Matching	0.201 ± 0.012	0.127 ± 0.007	$**$	$**$
Decision Tree	0.246 ± 0.016	0.227 ± 0.016	$**$	$**$
LitMatch	0.474 ± 0.005	0.342 ± 0.023	$**$	$**$
MLP with meta	0.544 ± 0.015	0.429 ± 0.019	$**$	$**$
GAT	0.629 ± 0.010	0.458 ± 0.021	$**$	$**$
ToxNet(S)	0.637 ± 0.013	0.478 ± 0.023	$**$	$**$
ToxNet	$\mathbf{0.661} \pm 0.010$	$\mathbf{0.529} \pm 0.036$	/	/

of ToxNet. Providing a learnability of the features (given the suboptimal condition of literature information) is therefore crucial to receive a stronger performance. As said before, we will analyze the literature impact in more detail in the following section.

Next to the literature data, we want to evaluate the importance that the relational information has on the performance of ToxNet. To achieve this, we will use the concatenated symptom and meta data vectors $\vec{x}_i$ as input for a standard MLP which has the same amount of learnable parameters as the ToxNet graph processing branch (as described in Sec. 5.4.4 for "MLP with meta"). This is compared against the actual graph processing branch of ToxNet using multiple GAT layers as well as the created affinity graph ("GAT"). From the results in Tab. 5.5, we can clearly see the added value that the graph neural network provides to the prediction task. The GAT network results in a substantially stronger performance compared to the MLP. This clearly shows the value that the graph-based processing of the relational information adds to the method and at the same time again highlights the importance that a proper processing of meta data can have on performance.

Finally, we evaluate the outcome of the combination of both branches, the literature matching and graph processing branch to receive the fully functioning ToxNet. Tab. 5.5 shows that our proposed approach again boosts the performance even further, resulting in the strongest performing method that significantly outperforms all other approaches (p-value < 0.005 for Wilcoxon signed-rank test). This is also clearly visible in the boxplot evaluation in Fig. 5.6. ToxNet shows a superior behavior and good prediction stability on all folds (the whiskers of the boxplot refer to the complete value area of the ten folds). This improvement was achieved although we previously showed that the literature knowledge was not very informative on the task. It can therefore be assumed that a stronger contribution of the literature information could boost the performance even further. The alternative successive application of the literature and graph branch performed for "ToxNet (S)" also shows a strong performance, can however not outperform the parallel setup of ToxNet. The reasoning of parallel contributions, we provided with respect to the method design in Sec. 5.3.1, is therefore corroborated by the results.

5.5.2 Performance of literature matching

The poor performance of the purely literature based methods "Naive Matching" and "Decision tree" have shown that the literature data is not very informative with respect to the prediction task. We therefore want to evaluate to which extend our literature matching branch can capture value from the expert knowledge. Before diving deeper into this evaluation, we want to highlight that within the full ToxNet setup the "LitMatch" branch shows some synergy effects and adds value to the prediction, since the already strong performance of the GAT can be further boosted (see Tab. 5.5). This is again confirmed in the successive approach ToxNet (S). Since the graph processing branch can perform the more complex data analysis, the "LitMatch" branch can focus on providing at least some literature-related complementary information. Still, we want to evaluate the literature matching in more detail. To do so, we will first compare its performance with the performance of a standard MLP of the same dimensions trained with the same patient symptom vectors ("LitMLP"), additionally we will compare with different setups of the literature matrix $\mathbf{W}_{\text{lit}}$ ("RandomMatch", "DataMatch").

Tab. 5.6. Performance comparison of the literature branch against other processing approaches. Methods are described in detail in Sec. 5.4.4. It is visible that the literature information provide to the branch is not very informative. This becomes obvious especially from the performance of "RandomMatch" and "DataMatch" (p-value: <0.01 $*$, <0.005 $**$).

Method	F1 Sc. micro	F1 Sc. macro	p-val micro	p-val macro
LitMatch	0.474 ± 0.005	0.342 ± 0.023	$**$	$**$
LitMLP	0.481 ± 0.009	0.370 ± 0.022	$**$	$**$
RandomMatch	0.474 ± 0.011	0.349 ± 0.022	$**$	$**$
DataMatch	0.498 ± 0.015	0.345 ± 0.027	$**$	$**$
ToxNet	$\mathbf{0.661} \pm 0.010$	$\mathbf{0.529} \pm 0.036$	/	/

This performance comparison is interesting since "LitMLP" has a fully learnable $\mathbf{W}_{\text{lit}}$ matrix. It will be therefore very interesting to see, if a certain setting of the $\mathbf{W}_{\text{lit}}$ can outperform "LitMLP", showing the value of the hard-coded literature matrix. The setup of these methodologies was already introduced in Sec. 5.4.4.

From Tab. 5.6 we can see the result of these comparisons. The "LitMLP" approach performs even slightly better than "LitMatch". This again demonstrates the limited quality of the literature information introduced into the training process of "LitMatch". Also the performance of "RandomMatch" is on the same level as "LitMatch", again highlighting the poor informativeness of the provided literature for the task. "RandomMatch" contains a randomly generated literature matrix, but the performance is not largely effected. At the same time, a very interesting finding is that "DataMatch" which is based on a literature matrix $\mathbf{W}_{\text{lit}}$ optimized on the data is showing the best performance of all approaches. It is able to also outperform the learned "LitMLP" matching. This highlights that with a stronger literature matrix potentially better outcomes could be achievable. Additionally, as already discussed above, we have to differentiate between the usage of the LitMatch branch for the full prediction task (as it is done here) and the usage for the provision of complementary literature-based information only (as it is the case for ToxNet). Within ToxNet, the addition of the literature branch adds value since the load of the data analysis is already taken over by the graph network. Here, it has to be performed completely by the literature branch itself. We can deduce from this evaluation (especially via the comparison against the random literature and data-based literature) that more informative expert knowledge (combined with a better symptom description in the data) could be beneficial for the prediction task.

5.5.3 Performance comparison against medical experts

After performing a detailed analysis of the implications of our design for ToxNet in the previous two sections, we now want to evaluate the potential clinical impact. For this evaluation, we designed 10 questionnaires with 50 patient cases each, which contained poisonings within a similar ratio as also occurring in the full dataset. As described in detail in Sec. 5.4.1, 25 cases of every questionnaire were individual different cases, while 25 cases were reoccurring for all questionnaires to receive insights on the inter-variability of the predictions. The questionnaires

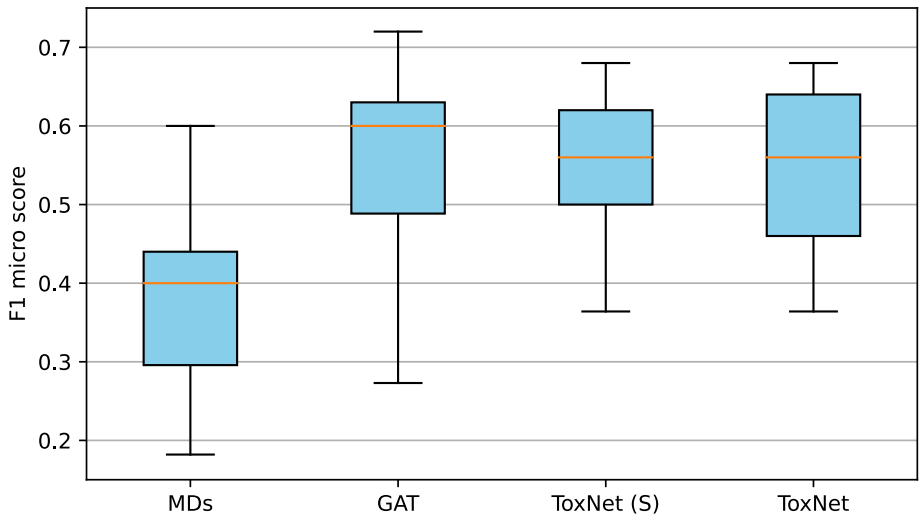


Fig. 5.7. Boxplot comparison of ToxNet prediction performance against medical doctors (MDs). The box shows the area where the center 50% of results are contained, the whiskers stretch to the minimum and maximum values. ToxNet as well as its graph branch (GAT) and its sequential version (ToxNet (S)) clearly outperform the MDs in both prediction accuracy and stability. On the smaller data subset used for the experiment, ToxNet is on a similar level than the GAT setup, but has smaller variation.

were handed out to 10 medical experts from the PCC in Munich. Four of them are specialists, while the other six are on the level of assistant doctors. Each doctor was asked to predict the poisonings for the given cases based on the provided symptoms and meta information. ToxNet was given the exact same exercise. The results for the individual and shared cases are given in Tab. 5.7, Tab. 5.8 as well as Fig. 5.7. It is clearly visible that ToxNet outperforms the medical doctors on the prediction task. In Fig. 5.7, we see that ToxNet reaches a substantially stronger F1 score compared to the doctors. The performance improvement against the "GAT" setup is not as severe on the small subset of data (250 cases) as on the full dataset, but the performance stability is improved (whiskers which indicate the full value range on the 10

Tab. 5.7. Performance comparison of ToxNet and medical doctors on 10 toxin classes for the 250 test cases.

Poison	Cases	F1 score MDs	F1 score ToxNet
ACE inhibitors	10	0.000	0.000
Acetaminophen	40	0.393	**0.425**
Antidepressants	30	0.276	**0.316**
Benzodiazepines	10	0.195	**0.421**
Beta blockers	10	0.300	**0.353**
Calcium channel antagonists	10	0.167	0.167
Cocaine	10	0.303	0.286
Ethanol	70	0.585	**0.839**
NSAIDs	40	0.222	**0.426**
Opiates	20	0.700	**0.857**

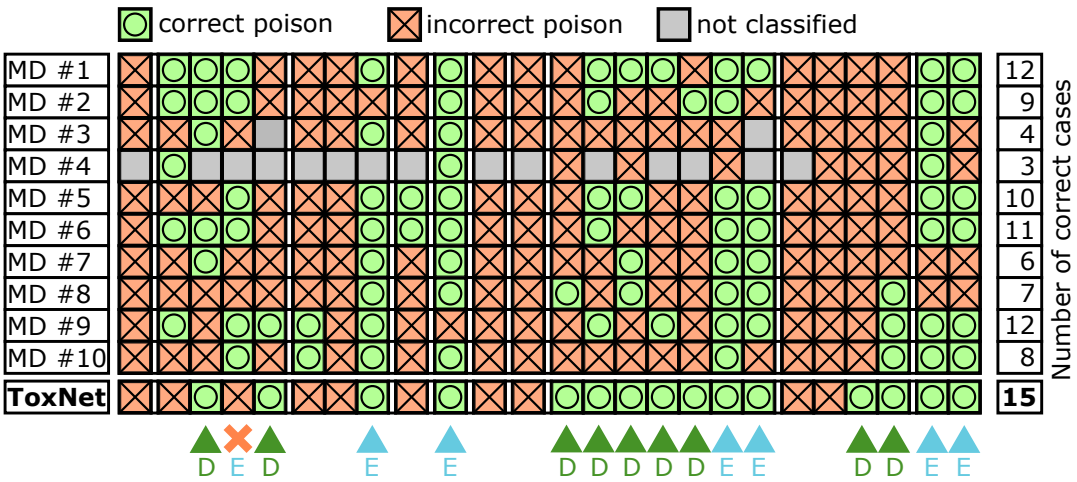


Fig. 5.8. Evaluation of clinician inter-variability and comparison with ToxNet. The poison classes are ordered alphabetically from left to right (see Tab. 5.2 for name details), each group separated with a white spacing. ToxNet provides the largest amount of accurately classified posions. Here, it is very stable on easy cases (marked with an "E") where most MDs correctly classified the correct poison and superior on difficult cases (correctly identified cases marked with "D") where half or more MDs failed [33].

questionnaires provided to the doctors are smaller). These results are in agreement with the expectations of our medical experts. As discussed in the introduction, for some cases only very generic symptoms are available which poses a major challenge to the prediction task. At the same time, as a MD it is difficult to consider the patient meta information in a statistically effective way. Here, a machine learning approach like ToxNet can strongly improve prediction outcome by analyzing and using these learned relationships.

Within Tab. 5.7, the performance of the MDs and ToxNet is reported on the individual poison classes within the 250 case test set provided to the ten doctors in form of individual questionnaires containing 25 cases. The distribution of the cases within a 25 case set follows the distribution of the overall dataset to receive a realistic situation potentially occurring within a real PCC. Shown in Tab. 5.4, when for a class the ratio to the overall amount of data is below $1/25 = 4\%$, we still provide one case of that poison to the set. As described, like for ToxNet the MDs are informed about the 10 possible poison classes, the distribution of cases and the fact that all 10 poisons are present in the questionnaire was however not disclosed to the MDs to allow for a fair comparison (ToxNet does not have this information as well). We see that ToxNet shows a stronger performance on seven poison classes, an equal performance on two and a worse performance only on one. This result nicely confirms that

Tab. 5.8. Performance comparison of ToxNet against medical doctors. Both the outcomes for the individual dataset (250 cases) and the shared dataset (25 cases) are shown. It is clearly visible that our developed ToxNet outperforms the MDs on the intoxication prediction task.

	F1 Sc. Individual		F1 Sc. Shared	
Method	micro	macro	micro	macro
Medical doctors	0.393	0.314	0.340	0.264
ToxNet	**0.538**	**0.409**	**0.600**	**0.417**

ToxNet is indeed capable to perform strong predictions on the full data domain. The overall results on the 250 individual cases are given in Tab. 5.8 left.

In Fig. 5.8, additionally the analysis of the 25 shared patient cases is given in detail. Again, ToxNet produces the largest amount of correct predictions with 60% correct answers for the ten toxin classes. More importantly, we see that with one exception for every patient case where the majority of MDs provided the correct prediction, ToxNet predicts the toxin correct as well. At the same time, for nine cases where only half of the MDs or less performed a correct prediction, ToxNet still provided the correct classification of the intoxication. This indicates that ToxNet shows a stable and reliable expert-level prediction performance for easy cases, while it at the same time provides a stronger or more successful prediction behavior for challenging intoxication cases, also visible within the results in Tab. 5.8 right. This is exactly the behavior which we were hoping for in the design of this method. The two strongest doctors correctly predicted 12 cases, ToxNet overall performed 15 correct poison predictions. We want to further highlight six cases which were wrongly classified by all doctors and our method. These are exemplary data samples which show an insufficient documentation quality (e.g. only a single generic symptom present), again underlining the challenges which are present in medical data in the wild.

5.6 Conclusion

Within this chapter, we have continued the evaluation of the inclusion of empirical priors into the learning process and have developed a methodology which not only considers relational knowledge but also allows the incorporation of rule-based literature knowledge into the training and prediction process. The developed methodological concept was evaluated on the task of intoxication prediction as performed in poison control centers and correspondingly named ToxNet. This specific task was chosen since within the medical field of toxicology a large literature data base of direct symptom to poison matching exists which is also used for predicting intoxications in the field. At the same time, patient-related and circumstantial information is available, making poison prediction a very interesting and at the same time challenging task to evaluate our methodology of empirical prior incorporation. Data vectors correspondingly contain individually experienced symptoms which are processed in a literature-informed network branch and at the same time are combined in a graph based prediction approach relying on meta data similarity.

Within the evaluation of ToxNet a few interesting findings could be made. One of the first major outcomes of our experiments is the superiority of our approach compared to the other evaluated methodologies and the superiority against the medical experts. The feedback we received from the medical experts was that the weaker results that were achieved by the MDs on the prediction task were not surprising to them. Poison prediction suffers from several previously described challenges, the largest ones being the existence of very generic symptom descriptions for patients, the stronger mismatch between the literature-provided rule sets and the real patient data and the unstructured incorporation of circumstantial information. ToxNet was designed to tackle these major issues. Via the combination of multiple cases within the graph a generic symptom representation can be enriched with neighboring closely related representations. The mismatch between literature symptoms and real data symptoms

is automatically learned and corrected by the literature matching branch. And the graph attention network can learn to use the patient meta information in a quantitatively optimal way. The clustering process occurring during this learning phase helps to further stabilize the ToxNet predictions. The results especially against the MDs show that this process was successful. Both on the overall performance as well as on the individual class level, our method showed a significant performance improvement. Our developed approach allowed the successful incorporation of relational and logical knowledge as domain priors into the learning process.

We did however also find clear limiting factors especially with respect to literature data quality. For the full ToxNet pipeline, the inclusion of the literature branch did improve the performance both in the parallel and the sequential setting, showing that to some extend complementary information is provided by the literature. However, within a direct evaluation of the "Naive Matching" and "Decision tree" approaches and the different ablative evaluations of the literature matching branch, we clearly found that the available literature in general provided only very limited information with respect to the prediction task. From the data analysis performed before, this was already expectable. At the same time, we could show that literature information directly derived from the data could boost the performance. We can therefore assume that a more expressive literature and in general a better match between reported symptoms and literature symptoms would increase the performance of the method further.

Within the next part of the thesis, we are going to focus on the second big area of available domain priors, namely scientific priors. These priors are not given in the form of empirical meta data or empirically derived logical hard-coded if-then rules, but are instead represented by a domain-covering theoretical description in the form of e.g. physical laws. They therefore provide a mathematical context which might be motivated and validated by empirical findings but constitutes in itself an underlying theoretical concept that governs the data domain as a whole. If available for the task at hand, providing such a rich domain prior to the training can therefore be off substantial value. Within the next sections, we will design and evaluate a new methodology for the effective incorporation of physical domain knowledge into the challenging medical task of intraocular lens prediction for cataract surgery. Here, a strong domain prior exists in the form physical optics of the human eye.

Part III

Model Priors: Physics-based Deep Learning

6 Theory of Physics-informed deep learning

6.1 Underlying theory

6.1.1 Types of physically informed learning

The term "physically informed learning" or "physics-based learning" refers to any kind of incorporation of physics-based knowledge into the learning process in the form of differentiable equations. Here, the used physical priors are relevant to the learning task by e.g. relating the used network input to the prediction target from a physical perspective. The intuition behind explicitly incorporating scientific model knowledge into the training process is motivated by the limitations of classical purely data-based training. We assume we know that a learning task at hand can be approximated by a physical model. The task therefore at least roughly follows the rules provided by this model, potentially more or less strongly disturbed by measurement errors and model insufficiencies. Within a standard machine learning approach, the network which in its initialized state is completely oblivious to these underlying concepts, has to learn all relations from the data and tries to deduce the underlying theory within the data domain based on the examples provided. Depending on the complexity of the task, this process can be very data-demanding and still is not able to properly cover the full domain. If instead an approach is provided that allows the system to learn the underlying scientific rules of the task, this huge knowledge gain is already incorporated into the model, highly simplifying the learning task. Here lies the great strength of the concept. Instead of relying on data samples which can only cover the full data domain in certain points, are expensive to collect in great numbers and are in general error-prone, the approach uses a concept that covers the whole data domain completely. Additionally, the differentiability of physics-based priors conceptually fits nicely to gradient-based training concepts of modern neural networks.

There are different approaches to realize this incorporation of physical knowledge, which can be roughly divided into four different categories [12]: physical fusion, residual physics, physical regularization and embedded physics (Fig. 6.1). These concepts can occur individually or overlap within one method.

- **Physical fusion**: Physical fusion describes the process of incorporating the solution of a physical model as an input to a neural network. This physical "truth" is processed by the network together with additional relevant input (e.g. the initial input of the physical model) to perform a prediction task. The network therefore always receives an assumption about the correct solution and can learn a correction based on the actual ground truth.

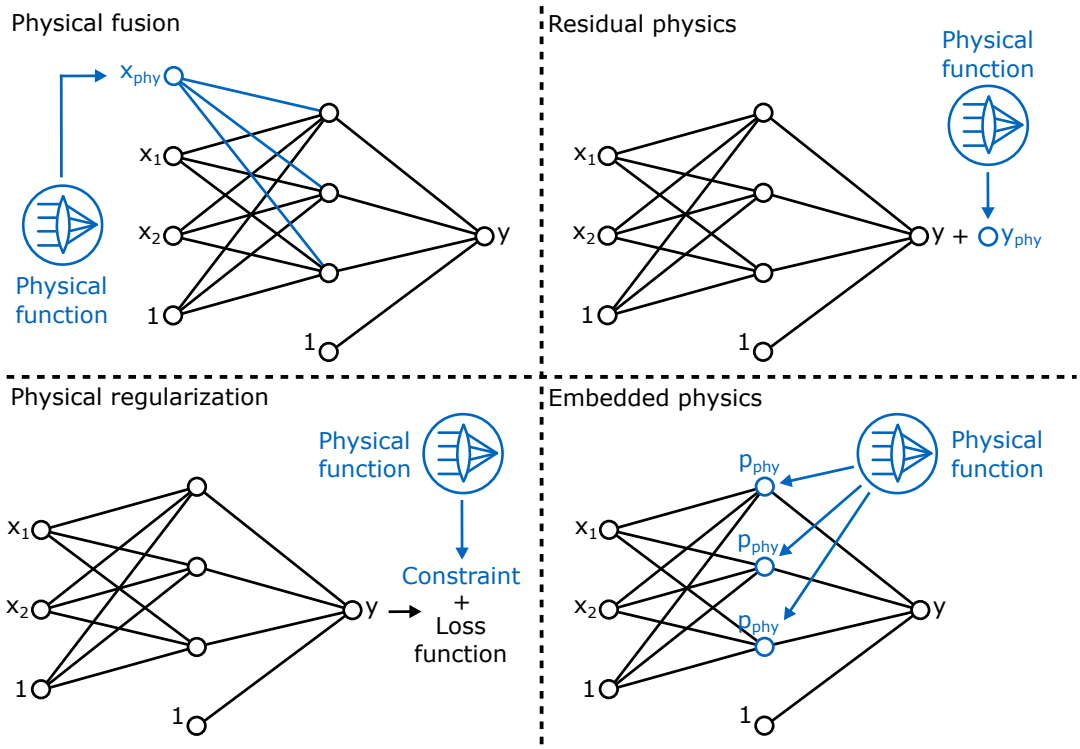


Fig. 6.1. Overview of physical knowledge integration concepts. For physical fusion, outputs of the physical model or motivated by the physical model are introduced as input into the network. The concept of residual physics uses the physical prediction as an anchor and predicts only the residual from the physical concept via the neural network. For physical regularization, a constraint motivated by physical concepts is introduced into the loss function. Finally, for embedded physics, the network is designed to resemble the physical model and its parameters correspond to learned units within the network.

- **Residual physics**: The approach of residual physics follows a similar idea and also relies on an explicit usage of the solution provided by a physical model for a given input. This solution is however not incorporated as an additional input to the network, but instead added to the output of the network. When compared against the ground truth the task of the network is therefore not to learn the complete solution but only the deviation from the physical prediction. This approach can increase the stability of the prediction and improve the interpretability of the results.

- **Physical regularization**: Physical regularization goes a step further. Instead of only incorporating the solution of a physical formula, the formula is explicitly connected with the learning process of the network by encoding it within a physical constraint and incorporating it into the loss function. The constraint penalizes the network, when its solution for a given task deviates to strongly from the physical assumptions and therefore operates as a regularization term that enforces physical consistency. Since the mathematics of the formula and therefore its gradients are directly connected with the update process of the network, this approach nicely highlights the effectiveness of physics-based priors. The used constraint automatically and elegantly enforces itself on every input provided to the network and therefore realizes agreement with the scientific assumption without the necessity of additional measures.

- **Embedded physics**: Finally, embedded physics, the most direct incorporation of scientific knowledge, uses the neural network as a shell for the physical model, the learnable parameters within the network therefore become directly correspondent to a parameter within the physical description. Using constraints within the loss function, similar to the approach described for physical regularization, it can be enforced that these parameters are representing their physical counterparts within the network system. The usability for this approach however requires that the scientific parameters and the corresponding calculation within the equation are expressible in terms of individual parameters within the network.

Within the next sections, we will introduce a specific form of physical knowledge integration via the creation of a domain specific loss, a concept which can be located in the field of physical regularization. "Domain specific" here refers to the fact that the physical information about the full data domain is differentiably encoded within the loss function. We will motivate how this concept can introduce detailed physical knowledge into a neural network in an unsupervised manner. After this, we will provide an overview about the historical development of the field of physical knowledge incorporation and related work in the field, focusing on the area of physical regularization. Then, within the next chapter, we will develop and evaluate a methodology based on these concepts for the challenging field of intraocular lens (IOL) calculation for cataract surgery. Within this field, detailed physical knowledge in the form of human eye optics is available. We will provide a detailed overview about the development of IOL calculation as well as human eye physics and demonstrate the superiority of our designed approach.

6.1.2 Domain-specific loss design

As described for the concepts of physical regularization and embedded physics, the encoding of domain-covering scientific knowledge within a differentiable constraint can be a very efficient and elegant way to incorporate scientific prior knowledge into the network training process. As described in the previous section, our goal is to incorporate the underlying scientific prior into our model, so that the system does not have to learn this prior from the data, which we can expect to be limited especially for medical tasks. Here, the usage of a constraint motivated from an underlying scientific equation as described for the process of physical regularization describes a very elegant and versatile concept to achieve this goal. Within our developed approach, we want to take this idea a step further. Instead of only incorporating the knowledge within the training process on real medical data with a randomly initialized model, we want to initialize the model as a mirror of the physical model before even considering real data samples. Like this, the training on real data samples simplifies to a fine-tuning process that has the purpose of correcting model incapabilities. Since a huge step towards understanding the data domain has been already performed by mirroring the behavior of the physical model, this process should require a substantially smaller amount of data. Essentially, a transfer learning step of the domain knowledge itself has been performed. The concept of transfer learning is known to increase prediction stability and reduce data requirements and has been successfully applied for various tasks [183, 244], one example being image classification problems [177].

The prerequisite for the described process is to design a specific physics-based constraint description, a domain-specific loss function, that represents the learning process for the medical task. This function can be used for the unsupervised training on artificial data. We will design such a loss function as well as a mechanism for creating an arbitrary amount of artificial data for the field of IOL calculation in cataract surgery. By applying the designed loss function to a network and using it for training on the simulated input parameters, the physical model is enforced to the network in an unsupervised fashion and will automatically result in a function represented by the learned network parameters that has understood the physical priors of the data domain. We will describe the detailed process in the following.

We consider the learning process for a network $\mathbf{\Phi}$ of a task with target $\mathbf{Y}$ and medical input parameters $\mathbf{X}$, therefore we are trying to optimize the objective function $\mathbf{\Phi}(\mathbf{X}) : \mathbf{X} \rightarrow \mathbf{Y}$. Further, we assume that we can create a physical equation $\mathbf{\Psi}$, which sets the input parameters $\mathbf{X}$ into a mathematical relation with the target values $\mathbf{Y}$. In the easiest case, this equation provides a direct calculation process for $\mathbf{Y}$ in the form of $\mathbf{Y} = \mathbf{\Psi}(\mathbf{X})$. In this case, we can force network $\mathbf{\Phi}$ to mirror the calculation behavior of $\mathbf{\Psi}$ by performing the direct learning task on the ground truth values created by the equation. Using this concept, we can create an arbitrary amount of artificial input samples $\mathbf{X}_{\text{simu}}$ that are representative for the data domain we would like to cover and apply function $\mathbf{\Psi}$ for the calculation of $\mathbf{Y}_{\text{simu}} = \mathbf{\Psi}(\mathbf{X}_{\text{simu}})$. Then, using a standard loss like the MSE loss on the network predictions and the physical calculation, we can train the prediction task (Eq. 6.1):

$$\mathbf{\Phi}(\mathbf{X}_{\text{simu}}) : \mathbf{X}_{\text{simu}} \rightarrow \mathbf{\Psi}(\mathbf{X}_{\text{simu}}) \tag{6.1}$$

After convergence of Eq. 6.1, we have received a network $\mathbf{\Phi}$ that mirrors the behavior of the $\mathbf{\Psi}$ when applied onto the data domain from which $\mathbf{X}_{\text{simu}}$ was sampled. We can however not generally assume this simplified situation of a directly applicable physical model and therefore look at the situation that equation $\mathbf{\Psi}$ itself has a different outcome value $\mathbf{Z}$, and can require additional input variables $\mathbf{\Omega}$. The relation between $\mathbf{X}$ and $\mathbf{Y}$ can be therefore implicitly encoded within 6.2:

$$\mathbf{Z} = \mathbf{\Psi}(\mathbf{X}, \mathbf{Y}, \mathbf{\Omega}) \tag{6.2}$$

Due to the additional input values, the more complex optimization task is therefore defined by $\mathbf{\Phi}(\mathbf{X}, \mathbf{\Omega}, \mathbf{Z}) : \mathbf{X} \rightarrow \mathbf{Y}$. Value $\mathbf{Y}$ is still the search target value for given scientific quantities $\mathbf{X}$, $\mathbf{\Omega}$ and $\mathbf{Z}$, but unless the equation is analytically solvable for $\mathbf{Y}$, it prohibits a direct application of $\mathbf{\Psi}$ onto the learning task for target $\mathbf{Y}$ as it has been done within Eq. 6.1. A straight-forward way to overcome this problem is to perform a numerical solving step for each data input to $\mathbf{\Psi}$. We are assuming a current data point represented by quantities $\mathbf{X}_0$, $\mathbf{\Omega}_0$ and $\mathbf{Z}_0$. By subtracting $\mathbf{Z}_0$ from Eq. 6.2, we can achieve the equation:

$$0 = \mathbf{\Psi}_Z(\mathbf{X}_0, \mathbf{Y}, \mathbf{\Omega}_0, \mathbf{Z}_0) = \mathbf{\Psi}(\mathbf{X}_0, \mathbf{Y}, \mathbf{\Omega}_0) - \mathbf{Z}_0 \quad , \tag{6.3}$$

easily solvable for the root by approaches like the Newton-Raphson method. Since equation $\mathbf{\Psi}_Z$ can have multiple roots, in general a second-degree Taylor expansion $\hat{\mathbf{\Psi}}_Z$ around a reasonable estimate $\mathbf{Y}_0$ of $\mathbf{Y}$ should be previously performed. Within the value range of $\mathbf{Y}_0$ we can approximate:

$$\mathbf{\Psi}_Z(\mathbf{Y}) \approx \hat{\mathbf{\Psi}}_Z(\mathbf{Y}) = \mathbf{\Psi}(\mathbf{Y}_0) - \mathbf{Z}_0 + \mathbf{\Psi}'(\mathbf{Y}_0)(\mathbf{Y} - \mathbf{Y}_0) + \frac{\mathbf{\Psi}''(\mathbf{Y}_0)}{2}(\mathbf{Y} - \mathbf{Y}_0)^2 \quad , \tag{6.4}$$

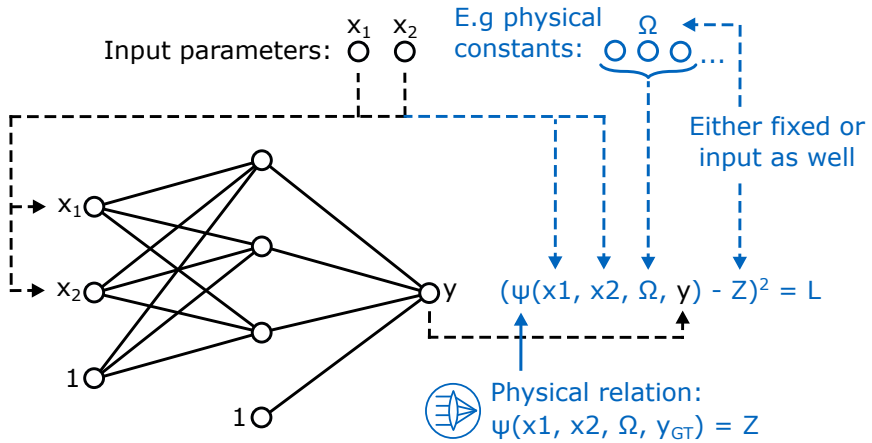


Fig. 6.2. Schematic example of the design of a physically motivated loss L. A monotone physical relation Ψ dependent on three input variables x_1, x_2 and y_{GT} as well as other parameters Ω results in output parameter Z. We consider that y_{GT} is the search parameter which should be predicted by a neural network Φ for a given task which is related to relation Ψ. By reformulating Ψ into loss function L and using the network prediction $\Phi(x_1, x_2) = y$ as input to the function L (Z is fixed in this example), the minimum value $L = 0$ is by definition returned when $y = y_{GT}$.

where $\boldsymbol{\Psi}'$ and $\boldsymbol{\Psi}''$ are the first and second derivation of $\boldsymbol{\Psi}$ respectively and $\mathbf{X}_0$, $\boldsymbol{\Omega}_0$ and $\mathbf{Z}_0$ have been left out for simplicity. Using this approximation, we can solve for the searched solution $\mathbf{Y}$ for given inputs $\mathbf{X}_0$, $\boldsymbol{\Omega}_0$ and $\mathbf{Z}_0$ using iterative steps of the Newton-Raphson method:

$$\mathbf{Y}_{t+1} = \mathbf{Y}_t - \frac{\hat{\boldsymbol{\Psi}}_Z(\mathbf{X}_0, \mathbf{Y}_t, \boldsymbol{\Omega}_0, \mathbf{Z}_0)}{\hat{\boldsymbol{\Psi}}'_Z(\mathbf{X}_0, \mathbf{Y}_t, \boldsymbol{\Omega}_0, \mathbf{Z}_0)} \tag{6.5}$$

Therefore, by simulating an artificial dataset for the input parameters $\mathbf{X}_{\text{simu}}$, $\boldsymbol{\Omega}_{\text{simu}}$ and $\mathbf{Z}_{\text{simu}}$ and individually solving for the corresponding solutions $\mathbf{Y}$, we can achieve the physically determined ground truth values $\mathbf{Y}_{\text{simu}}$ and use them for the training of $\boldsymbol{\Phi}$ within the optimization task given in Eq. 6.6:

$$\boldsymbol{\Phi}(\mathbf{X}_{\text{simu}}, \boldsymbol{\Omega}_{\text{simu}}, \mathbf{Z}_{\text{simu}}) : \mathbf{X}_{\text{simu}} \rightarrow \mathbf{Y}_{\text{simu}} \tag{6.6}$$

This approach has two drawbacks. First, it requires a numerical solving of Eq. 6.2 for the given simulated dataset, which can be quite time consuming depending on the size of the artificial dataset and the complexity of the physical model. Second, the necessity for an explicit solving step prohibits the possibility to learn on the physical model itself. We are therefore following a different approach.

Instead of trying to receive the searched target value $\mathbf{Y}$ as an explicit output of Eq. 6.2, we use the equation to design a physically motivated loss function, that describes the full data domain and can perform an unsupervised training on the data. In order to achieve this, the loss function has to fulfill two properties: it needs to be differentiable and minimizable, while the achieved minimum directly corresponds to the ideal solution of the task. Using Eq. 6.3, here we can distinguish two different scenarios. If Eq. 6.3 is monotone and fully differentiable, then whenever $0 = \boldsymbol{\Psi}_Z(\mathbf{X}_0, \mathbf{Y}, \boldsymbol{\Omega}_0, \mathbf{Z}_0)$ is fulfilled for the given $\mathbf{Y}$, by definition this $\mathbf{Y}$ corresponds to the physically correct solution of the equation. The root of $\boldsymbol{\Psi}_Z$ therefore automatically corresponds to the searched target value $\mathbf{Y}$ for any given input $\mathbf{X}$, $\boldsymbol{\Omega}_0$ and $\mathbf{Z}$. By

using the square of $\mathbf{\Psi}_Z$, we therefore can create a loss function which has the searched $\mathbf{Y}$ as its minimum (similarly described in [121]):

$$\text{loss}_{\text{phy}}(\mathbf{Y}) = \mathbf{\Psi}_Z(\mathbf{X}, \mathbf{Y}, \mathbf{\Omega}, \mathbf{Z})^2 = (\mathbf{\Psi}(\mathbf{X}, \mathbf{Y}, \mathbf{\Omega}) - \mathbf{Z})^2 \tag{6.7}$$

It becomes clear that this function is completely unsupervised and still forces a network to learn and encode the explicit physics of Eq. 6.2 within its parameters, when applied onto an artificial dataset $\mathbf{X}_{\text{simu}}$ (see also Fig. 6.2). The same holds up, if Eq. 6.3 is not monotone. Here, we can use the second degree Taylor expansion of the equation and obtain our approximated physical loss:

$$\text{loss}_{\text{phy, approx}}(\mathbf{Y}) = \hat{\mathbf{\Psi}}_Z(\mathbf{X}, \mathbf{Y}, \mathbf{\Omega}, \mathbf{Z})^2 \tag{6.8}$$

This concept provides a general framework how to use a physical equation, which is describing the given data domain and is setting the different input and target values into a physical context, for the setup of a fully differentiable domain specific loss function that allows a system to understand the physical prior knowledge without the necessity of data-based supervision.

6.2 Related work

Before diving deeper into the description of our designed methodology and the medical task on which we will demonstrate the effectiveness of our approach, we want to provide a detailed overview about the historical evolution and related work in the field of physically informed learning.

First discussions of the inclusion of physical priors into the prediction process of neural networks were already present in the 1990s. In 1997, [74] e.g. introduced the basic idea of residual physics as described in Sec. 6.1.1. Already a year early, ideas were discussed for a similar approach, however a linear instead of a physical model was proposed [152]. Even earlier, e.g. a work by [14] described the idea of physics-based elastic body matching for CT scans of the human brain. Based on this idea, in 2000 a work by [241] used the physical properties of deformable elastic solids as a constraint for non-rigid registration of medical imaging data of the brain and heart. The method was combined with statistical boundary shape information derived from the training set and showed improved performance. In 2003, [47] incorporated explicit information about the physical image generation process with respect to the reflection and scattering of light from surface to assist the detection of melanoma during diagnosis of pigmented skin lesions. Next to the image data, the network receives two additional inputs, the set of parameters characterizing the given tissue (optical properties, quantities, geometry) and a method to determine the spectra emitted from the given parameters. They were able to provide additional information about the lesion from the color image via this spectral analysis and the generation of parametric maps, helping the prediction task and resulting in strong results compared to a clinical assessment of the images and a dermatoscopy (see Fig. 6.3).

In 2016, [227] inverted the problem by using a convolutional neural network for the simulation of Navier-Stokes equations for fluid flow. Their concept moves in a very similar direction as explained in Sec. 6.1.2, they are creating an unsupervised learning task by relying on a

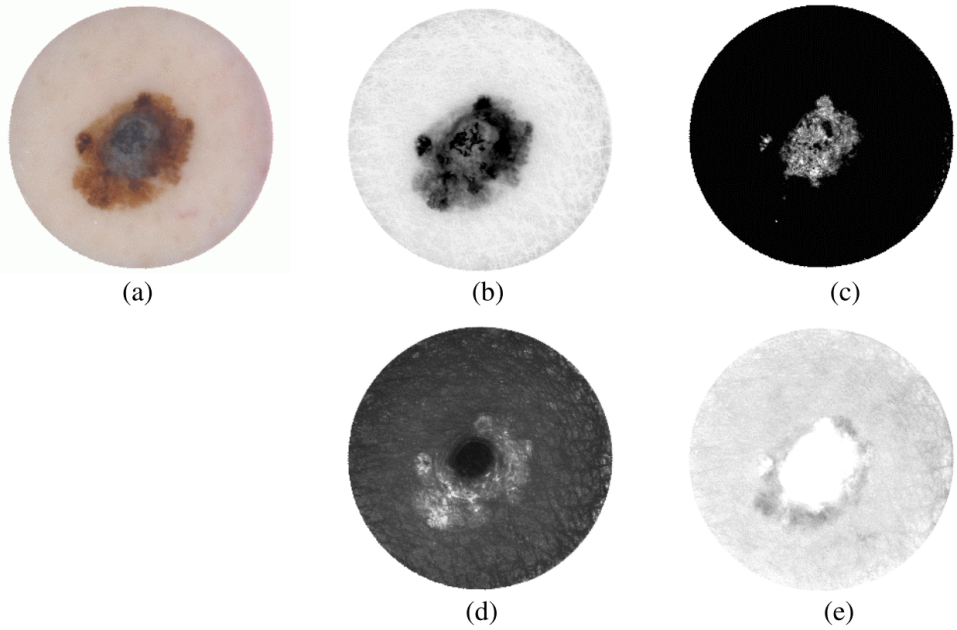


Fig. 6.3. Created parametric maps (b-e) based on a melanoma image (a) within approach of [47]. Information of physical light scattering and reflection behavior based on material is integrated into the learning process. *Note:* Image from [47]

physical formulation which approaches zero in its optimum, namely the calculation of the system divergence. Since for the fluid simulation, the velocity and pressure fields needs to be divergence-free, this formulation can be used for this purpose. In 2017, [189] went into a similar direction by using neural networks for the discovery of non-linear partial differential equations given data following the corresponding relation. Again, the neural network was used as an approximation approach for the equation that achieved high agreement with the underlying functional relation.

In the same year, [222] introduced a nice work about the unsupervised training of neural networks for object location within images only based on the underlying physical assumptions of the situation. The method was used for the tracking of objects in free fall. Given the physical relation of gravitational force, the system was able to predict matching object locations trained purely on the prediction received from the physical assumption, following the logic of Eq. 6.1 (see Fig. 6.4). Another application was the introduction of logical rules into the prediction task, demonstrated on a toy example of game characters present within an image. Another work related to our approach at least from a methodological perspective, [121] showed the improvement of lake-temperature prediction using an auxiliary loss term based on a constraint of temperature and density. Similar to our description above, they define a physical loss function as a continuous and differentiable equation of parameters, which are fulfilling the requirement of being equal or smaller than zero. In their case, this is realized via a parameterization of the density-depth relationship within water for temperature prediction. Their corresponding approach named physics-guided neural network (PGNN) was penalized for learning temperature predictions inconsistent with physical assumptions, showing a nice improvement especially in physical consistency. A very similar approach was used by [111] for assistance in the prediction of remaining useful life (RUL) for lithium-ion batteries. They used a physical constraint motivated from theoretical electrical models to produce label information

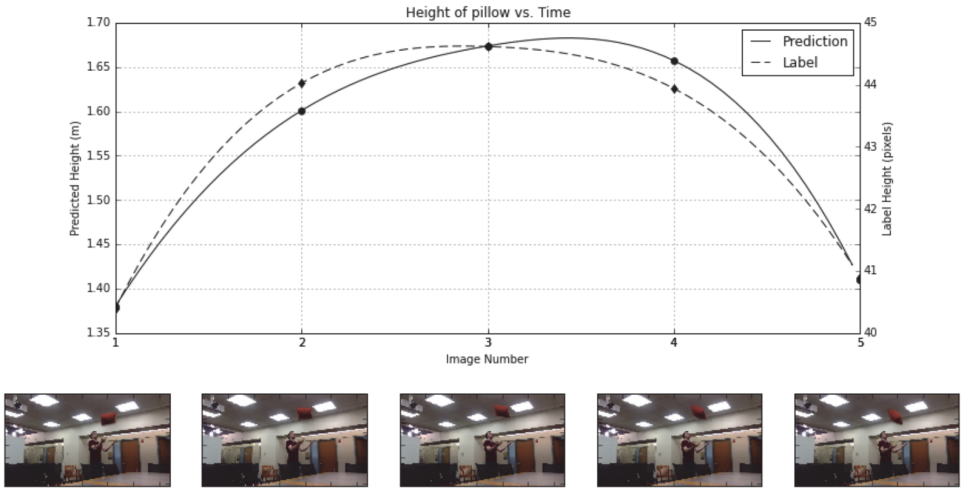


Fig. 6.4. Example application of [222]. The network predicts the parabola of a thrown pillow only trained via the physical relation of gravitation. *Note:* Image from [222]

for a larger amount of unlabeled data. In 2019, [117] followed up on the work of [121] by introducing a recurrent neural network into the methodology to process the temporal development of the water system. Additionally, they used a simple physical model to generate artificial ground truth data to perform a pre-training process for system initialization. Different from out approach, this sampling process was however not performed in an unsupervised manner. A year before, [225] relied on a similar pre-training concept for the prediction of part weights manufactured by injection molding. They used this technique to deal with the very limited amount of real data samples. Interestingly, [173] proposed to use machine libraries like PyTorch [186] for the general solving of physical problems which are expressible in terms of a minimization problem. The strong gradient-based optimization algorithms encoded within this and other libraries are essentially usable for this purpose in the same way as for the optimization of neural networks entrusted with the same learning task.

The addition of a physical constraint term into the training process as discussed by [121] was slightly extended by [169] with an approximation constraint. The approximation constraint essential accounts for noisy data within the training dataset for which reasonable ranges can be defined by expert knowledge. Values outside this range are penalized by the loss term. Their domain adapted neural network (DANN) was trained for representing synthetically generated datasets using this additional constraint. Also in [154], this physical constraint concept is used for the forecasting of photovoltaic power generation by adding physical reasoning both to the loss term and network (e.g. a ReLU activation to avoid negative power generation prediction).

The previously described works are mainly located in the field of physical regularization, which relies on physical constraints to assist in a prediction task. The field is especially interesting for us, since our developed methodologies proposed in this part of the thesis for the inclusion of scientific prior knowledge focus on this area as well. The group of [12] provided a nice review paper, followed up by [118], highlighting some of the above as well as other works from different domain learning fields already introduced in Sec. 6.1.1. Additionally, they introduced neural architecture search (NAS) into the realm of physics-informed learning to

generalize the modeling approach. Similarly, [198] tried to provide a conceptual framework for the interactions between simulation and machine learning approaches.

The previous discussion was more focusing on methodological aspects of physical prior incorporation. In the medical field, interesting work on the incorporation of physical constraints for optimized model training was published as well in recent years. The group of [80] proposed a learning-using-privileged-information paradigm (LUPI) for implicit tissue strain reconstruction for quasi-static ultrasound elastography. The concept is designed to impose a direct influence on the intermediate state of the learning, forcing the neural network to follow the ultrasound elastography process. Within their approach, they additionally introduced a strategy for physics-based data generation to generate triplets of privileged information, training data and labels. These are required for the approximation of the elastography process based on tissue biomechanics and ultrasound physics. In [259], also used the generation of synthetic data to train their neural network approach Model-Informed Machine Learning (MIML) for the purpose of T2 distribution reconstruction from multi-echo T2 magnetic resonance (MR) signals. The creation of the artificial dataset follows biophysical models. In consequence, the network can produce the corresponding T2 distribution for a given MR signal, being constrained with prior knowledge about the in vivo distribution.

In the same year, [31] introduced a method for the simulation of personalized left-ventricular biomechanics using a neural network with a physically constraint restricted output space. The restriction is based on characteristic deformations of the left ventricles described via radial basis functions as well as the application of a loss function based on the energy potential functional tailored for hyperelastic, anisotropic, nearly-incompressible active materials. The group could show that a 30 times faster modeling could be achieved compared to a Finite Element approach. Another interesting work was performed in [223], where physical modeling was leveraged for the the conversion of low-dose PET images into their standard-dose counterparts. The sinogram-based physics of PET imaging systems and their output uncertainty was used to train a deep learning model called suDNN for this purpose. The network produces standard-dose PET images as well the corresponding multi-contrast MRI images and showed an improved robustness, when confronted with out-of-distribution samples.

As discussed before, the general field of domain-knowledge incorporation into deep learning for medical image analysis was covered within a survey paper by [250]. The paper is not focusing on physical domain knowledge, but still provides a nice overview of works in the fields of disease diagnosis, abnormality detection and lesion as well as organ segmentation.

7 Physical knowledge through domain-specific loss design

7.1 Intraocular lens calculation for cataract surgery

7.1.1 Medical background of cataract surgery

To show the effectiveness of the introduced concept of domain-specific loss design within the field of medical machine learning, within this work we will focus on a very important medical procedure, whose success strongly depends on the correct integration of physical aspects, namely the field of cataract surgery. The term "cataract" refers to an ongoing crystallization of the human eye lens as a result of internal chemical processes that leaves the lens increasingly opaque. Before going into detail about this process, we will briefly provide an overview of the optical system of the human eye for perspective. Fig. 7.1 shows a schematic description of the human eye. Light enters the eye through the cornea, a window like tissue in front of the iris, and travels through the eye onto the retina, where the incoming light is focused and processed by light-sensitive cells, which forward their excitation signal through the optic nerve to the brain. Along the optical axis of the human eye, two structures are mainly responsible for the light refraction within the eye, namely the cornea and the human lens, where the cornea typically has the stronger refractive power. The complex human lens is consisting of several nuclei that are incorporated into a capsular bag, which is connected to the zonular muscles via thin zonular fibers. These fibers pull the capsular bag and stretch the lens, making it more flat, when the muscles are relaxed. When the muscles contract and are pulled together, the stress on the lens reduces and it can take a more round shape . This process changes the surface curvature of the lens and as a result its refractive power, effectively allowing the human to focus onto different objects in various distances to the eye [267]. It becomes obvious that the human lens therefore plays a major role in the eye's optical system and its refractive power is an essential aspect of the complex eye optics.

Unlike most other parts of the human body, the human lens continues to grow over a lifetime. During this process, the occurring structural changes within the lens can successively disrupt and rearrange the fiber structure which is essential for the optical clarity of the eye. This process results in an increasing opacification of the lens, the cataract, eventually even leading to complete blindness when left untreated [3]. The development of a cataract is one of the most frequently occurring health related changes of the human eye. There are four main types of human lens cataracts. Age-related cataracts, the most common type, develop especially in the elderly, their occurrence is favored by medical conditions like diabetes or external factors like smoking and drinking. Traumatic cataracts can develop after a serious injury of the eye, sometimes years after the incident. Radiation cataracts can be the result of the eye's exposure

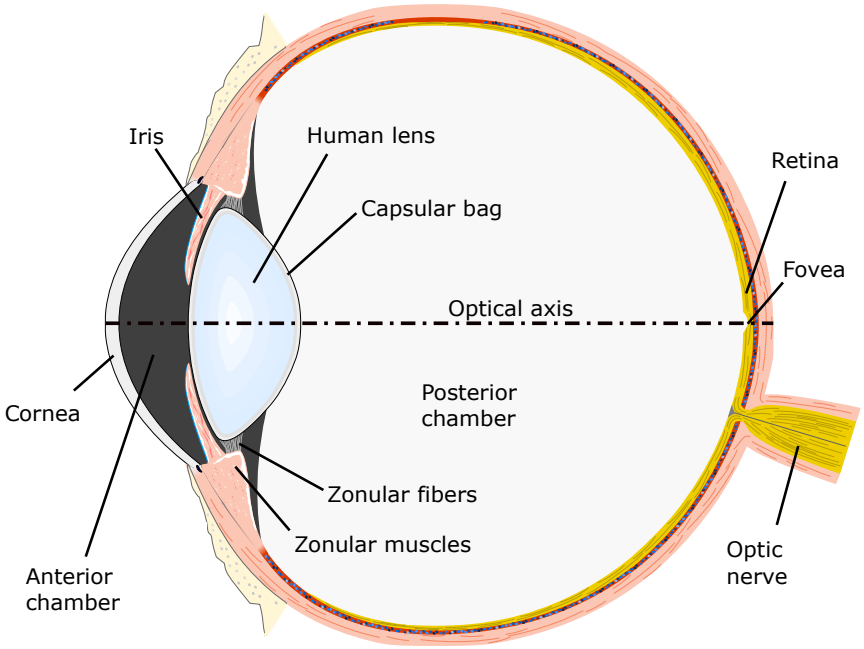


Fig. 7.1. Detailed scheme of the human eye. Light enters the eye through the cornea, is refracted within the complex optical system of the eye and finally focused onto the retina, where light-sensitive cells forward the received signal to the brain via the optic nerve.

to strong electromagnetic radiation. This includes UV light of the sun and radiation treatment of cancer patients. Pediatric cataracts develop for children, are rare, and mostly genetically inherited or due to certain illnesses during childhood [3, 196].

Due to the high impact it has on a patient's life quality, the treatment of cataracts fortunately has become a standard medical procedure today. In fact, cataract surgery constitutes the most frequently performed ophthalmic surgery in the world [3]. In result, cataract surgery is today highly standardized, allowing for fast treatment and at the same time a high demand in efficiency (methodologies developed in this direction in general are e.g. given by [181], [194] or CataNet [158], which is specifically designed for the estimation of remaining cataract surgery duration). During surgery, the surgeon enters the eye through a small incision within the cornea and opens the capsular back within a round shape at its front in a process called capsulorhexis [83]. This hole allows a direct access to the opaque human lens, which is destroyed and aspirated by the surgeon using a small tip vibrating at ultrasound frequency, the so called phacoemulsification. After this process, an empty but intact capsular bag remains in the eye. Within this bag the surgeon now introduces a new artificial intraocular lens (IOL), which substitutes the old natural lens within the optical system of the eye and therefore takes over its refractive role. As described above, the impact of the lens on the optical system is significant, therefore the correct determination of the necessary refractive power of the IOL is crucial for a successful surgery. This process of calculating the correct IOL power is however far from trivial due to several influencing factors and the general complexity of the eye's optical system, sometimes resulting in so-called refractive surprises, a strong deviation from the targeted refractive power of the eye after surgery. Within recent years, a body of work for the calculation of IOL refractive power has been performed with varying success. Within the next sections, we will give an overview about the state of the art in IOL calculation, starting

with measurement concepts and moving over to current calculation techniques. Then we will lay out, how we significantly improve this calculation process by developing a physics-aware machine learning approach called OpticNet. We have presented this work at the MICCAI 2020 conference [34], additionally an extension paper with a deeper analysis of the network as well as methodological improvements was published by us in the journal Medical Image Analysis [35].

7.1.2 Ophthalmic measurement concepts

Objective and subjective refraction measurements

As described above, the calculation of IOL power is essential to receive a good refractive outcome after cataract surgery with little error. The concept of refraction essentially describes how strongly a light ray is refracted (changed in direction) by an optical medium and is measured in the unit diopters (D). The term refractive outcome error represents the deviation of the eye's manifest refraction after surgery from the targeted refraction and corresponds to a spherical approximation of an eye's vision properties. An eye with a perfect vision outcome has e.g. a manifest refraction of 0.0 D (called emmetropic). A detailed description of these concepts and the eye's optical physics in general will be given in Sec. 7.2.1. To obtain the value of the manifest refraction and correspondingly the refraction outcome after surgery, today two types of measurement concepts are used, subjective and objective refraction measurements. The biggest difference between these two concepts is that subjective refraction requires an active feedback from the patient about his or her vision experience while objective refraction automatically provides a measured value of the manifest refraction.

The concept of subjective refractive measurements is already a few hundred years old. Already in 1619, Christoph Scheiner discovered the Scheiner principle, realized via a simple opaque disk with two holes in it which is placed closely in front of the eye. Through the two holes light could pass into the eyeball. When for the patient only one illumination source was visible, the patient could be considered emmetropic, since the rays of the two light sources focus onto the center of the retina. If two light sources were visible, the patient was considered either myopic or hyperopic [56]. It was however not yet possible to distinguish between the two scenarios. The Scheiner slit optometer invented by William Porterfield in 1759 improved on this concept with an adjustable slit light source which was movable with respect to the patient position [236]. In 1801, Thomas Young introduced a measurability to this concept by adding a lens for hyperopic patients and a scale from which the object position could be measured [10]. Later, Jules Badal presented his Badal optometer. An object was placed into the focus point of a lens in front of the eye to create parallel light rays from every point of the object entering the eyeball. In consequence, a non-accommodated eye would see the object in perfect focus [9]. For a myopic eye, the object would need to move closer to the lens to get into focus, for a hyperopic eye it would need to move further away. A scale next to the object would link the object's position to the corresponding manifest refraction. A major problem with these described approaches was a requirement of a considerable amount of cooperation by the patient, starting from the fact that accommodation of the patient eye resulting from the instruments in close proximity can influence the measurements. Therefore, these first approaches are outdated by today's standards and are replaced by modern phoropter devices [38]. A phoropter is essentially a device that allows the seamless placement of optical lenses

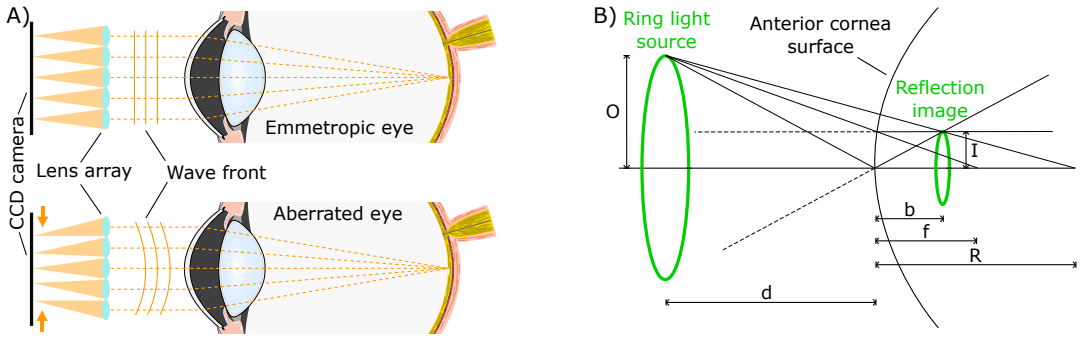


Fig. 7.2. **A)** Schematic description of an aberrometer measurement using Hartmann-Shack wavefront sensor. A plane wave front is emitted into the eye, the reflected signal is measured via the sensor which is placed in front of the eye. For the upper emmetropic eye, the reflected wavefront is again planar. The 2D microlens array focuses the incoming wavefront onto the CCD chip in a correspondingly linear pattern. For the lower aberrated eye, the reflected wavefront is not planar, as a result the focus points on the CCD chip are shifting. This shift within the signal pattern can be directly transferred into a refractive measurement of the eye. **B)** Schematic description of a keratometry measurement used to determine the refractive power of the cornea. The reflection of a ring light source is detected on the cornea surface. The size of the reflected image I can be used to determine the curvature radius of the cornea in the respective area.

of different refractive power in front of the patient eye. The patient views an eye chart (chart with letters and numbers) through these lenses which is placed at optical "infinity" (6 meters away from the patient) and reports his or her visual impression to find the best correcting lens. Like this, the patient eye's refraction is determined. As mentioned above, all the described methods are subjective, since they require subjective feedback from the patient ("I see the object in focus"). Therefore, the individual experience of a patient as well as the surrounding situation of the measurement can have an impact on the refractive outcome. At the same time, the subjective refraction incorporates the individual satisfaction of the patient which ultimately is the highest priority of the surgical intervention, therefore it still is an important measurement concept in today's clinical routine.

In contrast to these approaches, a second group of refractive measurement devices can be considered as objective. They provide a measurement value of the refractive properties of the patient eye without the necessity of their feedback. Objective approaches are autorefractors, aberrometers and retinoscopy. An autorefractor sends a normally infrared light ray into the eye and automatically tunes through a group of refractive lenses [70]. It then measures the reflected signal from the retina on a photodiode. Based on the shape of the measured signal, it can determine the best correcting lens for the patient eye. A slightly different concept is performed by the aberrometer and its wavefront analysis [30]. Here, a wavefront is sent into the patient eye and the returned wavefront is measured via a wavefront sensor. The mostly known sensor is the Hartmann-Shack wavefront sensor [208]. It consists of a lens mask with arrays of small microlenses which focus the incoming wavefront onto a CCD chip. An emmetropic eye without any aberrations would reflect a plane wavefront, resulting in an ordered pattern of focus points on the CCD chip. Any aberrations within the eye disturb the returning wavefront and modify this pattern (Fig. 7.2 A)). From the pattern deviation to the undisturbed case, a refractive map of the eye can be generated. The third approach is the procedure of retinoscopy. Within this measurement technique, the optometrist places correcting lenses in front of the patient eye and uses a retinoscope to shine light into the

eyeball to observe the light spot on the retina. From this, it can be determined which lens should lead to the best refractive outcome [70].

Finally, we want to address another very important measurement concept, which falls under the category of refraction measurements, but focuses on the measurement of the refractive power of the cornea only, namely the keratometry measurements. This measurement concept is based on individual light spots directly reflected from the surface of the cornea [208]. Using the size of the reflection on the cornea I, it is possible to calculate the curvature radius of the anterior cornea surface with the functional relation given in Eq. 7.1 motivated by Fig. 7.2 B).

$$\frac{O}{d} = \frac{I}{b} \quad , \quad b \approx f = \frac{R}{2} \quad \rightarrow \quad R = \frac{2dI}{O} \tag{7.1}$$

where O is the radius of the light source and d is the distance of the source from the cornea. This curvature radius can be directly converted into the refractive power $P_C = (n_C - 1)/R$ of the cornea using an adapted refractive index n_C of the cornea which considers the standard posterior cornea surface within its calculation.

While the refractive measurement of a patient eye is essential to determine the refractive outcome of a cataract surgery, for the calculation of its optical properties and corresponding refractive requirements prior to surgery, another measurement concept is responsible, namely the measurement of the eye's biometry introduced in the next section.

Biometry via Optical Coherence Tomography

The calculation of the required refractive power for an intraocular lens (IOL) and a given eye is mainly based on the eye's biometric measurements. The term biometric measurement refers to measuring defined distances within the eye that enable a theoretical description of the eye's optical system. These measurements were initially performed by applying a thin liquid immersion on the cornea and then using an ultrasound probe placed directly on the cornea to emit and measure and ultrasound signal (Fig. 7.3). The emitted signal is reflected from any surfaces in the eye (transfer boundary from one medium to the next) and the corresponding travel time of the signal is measured. The measured signal spikes directly refer to a positioning information along the central axis. Like this, it is possible to receive an understanding of the measured eye's inner anatomy. The measurement of a single vertical line within the eye is called an A-Scan. Adding multiple neighboring A-scans together to receive one area scan image corresponds to a B-scan.

The first measurement approaches were not very precise, being limited by available US frequencies of $\sim 10\,\mathrm{MHz}$, which corresponds to a resolution of $\sim 0.5\,\mathrm{mm}$ (For completeness, it must be stated that modern US devices like the ArcScan [270] with $50\,\mathrm{MHz}$ can achieve resolutions in the $\mu\mathrm{m}$ range. They are however mostly used in cases where opaqueness of the eye is so advanced that optical approaches are not feasible [207]). Additionally, ultrasound measurements require a direct contact to a patient's eye, which can be experienced as unpleasant. Early approaches did not even use an immersion gel, resulting in deformed corneas and potentially large measurement errors [207]. Therefore, with the approach of optical coherence tomography (OCT), a new generation of contactless and more precise measurement devices was enabled. Already in 1982, Prof. Adolf Fercher patented the concept of optical eye biometry measurements via low-coherence interferometry [207], which is

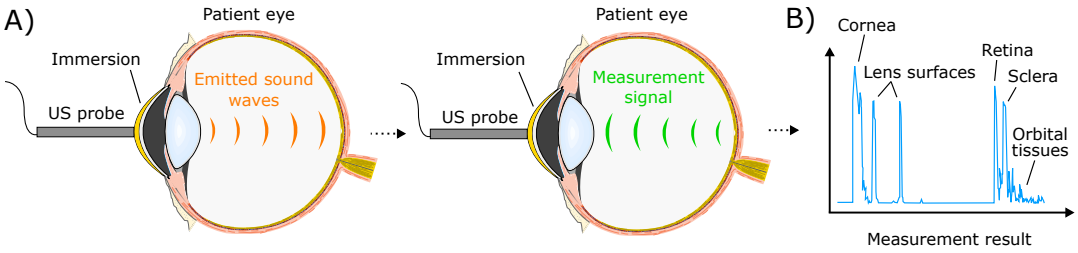


Fig. 7.3. **A)** Schematic description of Ultrasound (US) biometry using an US probe placed on top of the cornea covered under an immersion. The ultrasound signal enters the eye and is reflected from different surfaces within the eyeball. The reflected signals are measured by the system to obtain an estimation of the geometric distances within the eye. **B)** Example results for performed US measurement. The cornea and human lens as well as the retina are detectable from the reflected measurement signal.

the foundational concept of OCT measurements [76]. The patent was bought by the Carl Zeiss Meditec AG and paved the way for modern optical biometry. Instead of a sound wave, within OCT approaches a light wave is send into the eye ball. Since the speed of light (∼300 million meters per second) is around a million times faster than the speed of ultrasound, it is however not feasible to use this measuring modality in the same form as ultrasound (simply measuring its traveling time). Instead, within low-coherence interferometry the physical concept of coherence was used to enable this measuring modality for practical usage. The term coherence and coherence length refers to the difference in travel distance that two light waves emitted from the same source can travel and still perform constructive interference. Constructive interference corresponds to the strengthening of a light signal via the overlap of two electromagnetic waves (the light waves) with the same amplitude an phase (maxima on maxima, minima on minima) [61]. As long as the two waves emitted from the same source travel the same distance, their signals are identical and will perform constructive interference. If however one light wave takes a longer path and later overlaps with the first ray, depending on the difference in traveled distance, dissimilarity will quickly increase. The difference that is still acceptable for constructive interference to occur is called the coherence length of the light source. For an optical coherence tomography device, a source with as short coherence length is used. Therefore, already small differences in travel distance will destroy constructive interference. This concept is now used as follows to achieve biometric measurements of the eye (see Fig. 7.4): A light beam emitted from the device's low-coherence light source is directed onto a beam splitter within the device that splits the beam into two perpendicular directions, effectively resulting in two identical light waves emitted from the source at the same time. One of the light rays is directed onto the eye, where it reflects from the various refractive surfaces (boundaries between different optical media) back into the device. The other light ray is focused onto a mirror on a measuring arm and simply reflects back from there. The reflected ray as well as the reflected signals from the eye ball again meet at the beam splitter. They are both split a second time, where one part of each ray is now directed onto a light-sensitive sensor. Both signals are overlapping when reaching the sensor. Due to the very small coherence length, only signals from the eye that traveled a nearly identical distance to the reference beam reflected from the mirror will result in constructive interference. Therefore, by moving the reference mirror, the received signal spikes at certain positions of the mirror directly refer to the positions of refractive surfaces within the eye. Like this, it is possible to obtain a contactless measurement of the patient eye's biometry. The corresponding measurement technique is called time-domain OCT [76]. The received positions still need

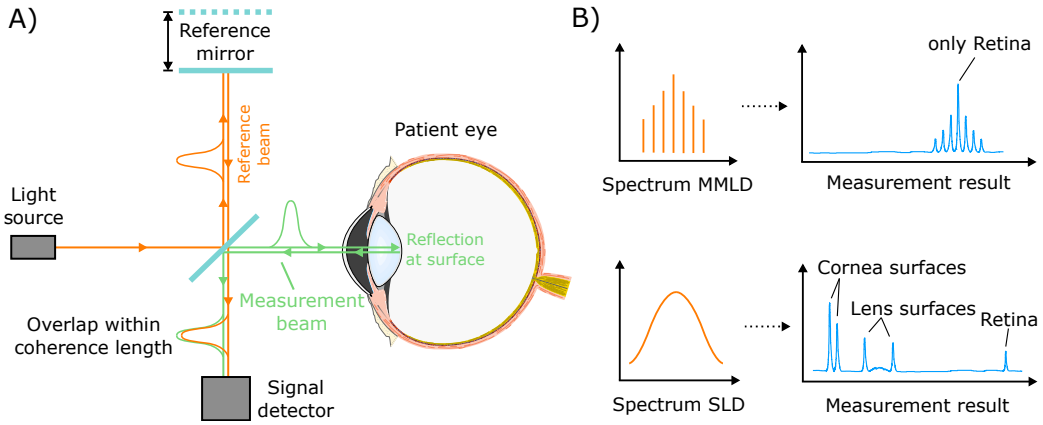


Fig. 7.4. **A)** Schematic description of a time-domain optical coherence tomography device. A light beam with short coherence length is emitted from a light source and split in two beams via a beam splitter. Only if the beam reflected from an eye optical surface and from the reference mirror have traveled nearly the same distance, constructive interference is achieved on the signal detector. **B)** Example spectra for two different light sources. The older multimode laser diode (MMLD) consists of several discrete frequencies, resulting in an inference spectrum with multiple local maxima. As a result, the measurement is less precise and structures have limited resolution. The superluminescent diode (SLD) has a continuous spectrum, resulting in sharp maxima in the detected signal. Resolution is in the area of $10\,\mu\mathrm{m}$ depending on SLD coherence length.

to be converted into actual geometric quantities by converting the optical path lengths into geometric path lengths using the assumed refractive indices for the different eye areas.

Within a first realization of this device type, a multimode laser diode (MMLD) was used which does not provide a continuous frequency spectrum, but multiple fixed emitting frequencies. As a result, the received signal consists of multiple local maxima around the global maximum (interference pattern), which substantially reduces signal quality and measurement accuracy. The signal pattern therefore also prevented any higher resolution of e.g. both cornea surfaces which only have a distance of $\sim 0.5\,\mathrm{mm}$ and was mainly used for the determination of the axial length. The light source was therefore exchanged in a second realization by a superluminescent diode (SLD) which provides a continuous frequency spectrum. This resulted in a substantially better signal resolution, allowing measurements in the range of $10\,\mu\mathrm{m}$ [85].

In order to receive OCT scans of larger areas of the eye, ideally the full volume, the measurement modality did however not only require a high accuracy, but also speed. To improve the devices in that direction instead of a movable reference mirror, a fixed reference arm and a spectrometer was introduced as a receiver of the reflected signals [76]. Within the spectrometer, the frequency spectrum of the signal is decoded, allowing for a Fourier conversion of the signal from frequency into time domain. The frequency scan therefore results in a direct spatial pattern of the eye, yielding the surface positions. Since the need for a mechanical movement of the reference arm was removed, a time gain could be achieved by this approach, called frequency-domain OCT. Since the spectrometer still needs to process the recorded signals from the distributed light beam, it has some time-related requirements. The next leap in processing speed was therefore given by the introduction of the swept source OCT. This concept removes the necessity of a spectrometer altogether by introducing a light source with tuneable light frequency, the swept source laser. Via the tuning of the light source signal over time, the frequency spectrum is encoded in time-space and therefore can be recorded

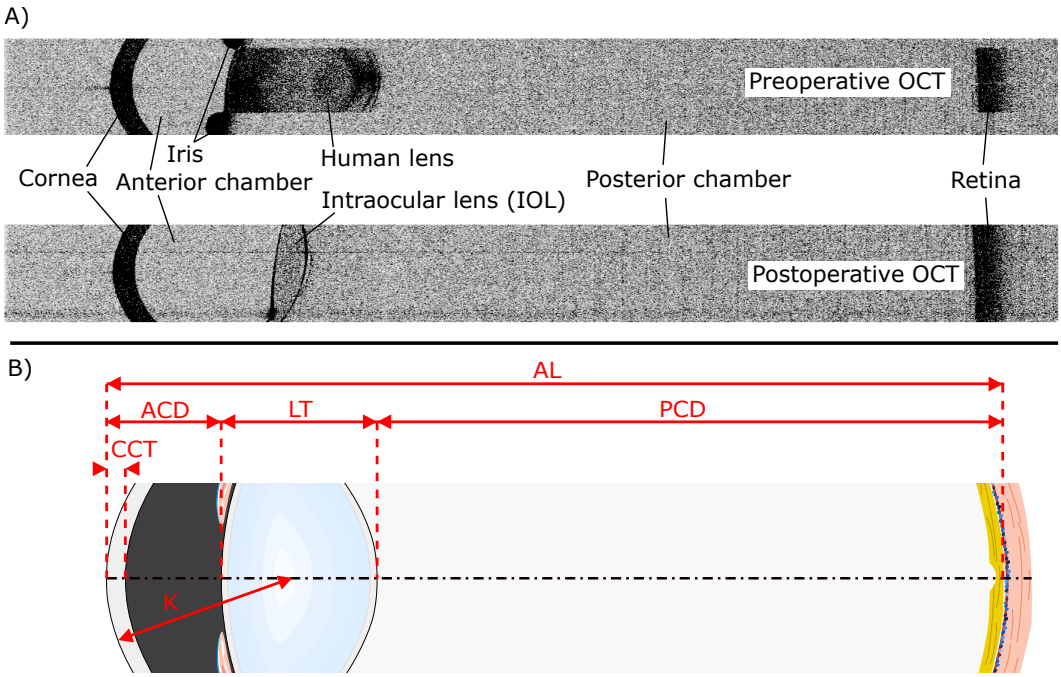


Fig. 7.5. Optical coherence tomography (OCT) imaging scan: **A)** Examples of raw OCT imaging scans for a patient eye taken pre- and postoperatively. It is clearly visble, how the natural lens of the preoperative scan has been replaced with an artificial IOL in the postoperative scan. The attribute "raw" refers to the fact that the distortion due to light refraction of the OCT scanner within the eye is present within the image. **B)** Schematic description of biometric quantities extracted from the OCT scan: axial length (AL), anterior chamber depth (ACD), lens thickness (LT), central cornea thickness (CCT), cornea curvature / keratometry (K), posterior chamber depth (PCD).

by a simple CCD chip. Since the tuning of the swept source laser can be performed faster than the processing time of a spectrometer, another boost in recording speed was possible. Today's state-of-the-art OCT devices like the IOLMaster 700 from ZEISS [41] introduced in 2015 therefore perform fast high resolution B-scans of the eye. The two images shown in Fig. 7.5 A are two examples of images recorded with the measurement device. The upper scan is collected preoperatively, easily visible by the human lens still in place. The lower scan corresponds to a postoperative situation, where the substantially thinner IOL already has been implanted.

7.1.3 Calculation of intraocular lens refractive power

Biometry based on Optical Coherence Tomography

As described in Sec. 7.1.1, a proper prediction of the necessary IOL refractive power fitting to the patient's eye is essential for a successful cataract surgery. The optical system is defined by the patient eye's individual biometry, defined as geometric quantities within the eye like its overall length or the curvature of the cornea. As an example, a longer eye with otherwise identical parameters will need a weaker IOL refractive power to focus incoming light onto the retina compared to a shorter eye. It is therefore intuitive to base the prediction of refractive power on these biometric parameters essentially describing the eye's optical system. As described in Sec. 7.1.2, the standard technology used today to obtain these measured parameters is based on optical coherence tomography (OCT), a fast and non-

invasive measurement technique. It is based on scanning the eye using a light ray with a very narrow coherence length. Using this technique, it is possible to scan and measure the eye's internal surfaces very precisely (details in Sec. 7.1.2). Fig. 7.5A shows an example of real OCT imaging scans taken from a patient before and after surgery. In the postoperative scan, it becomes nicely visible that the human natural lens of the preoperative scan has been replaced by the significantly thinner IOL. The posterior part of the capsular bag has attached itself to the artificial lens during the healing process. Both scans are raw imaging files, therefore still containing the distortions introduced by refraction of the OCT measurement beams within the eye's optical system. These distortions of optical path lengths and geometric angles need to be considered when extracting the biometric measurements from the images, since the description of the optical system requires geometrically correct values. Fig. 7.5B visualizes different biometric parameters extracted from these scans, which are used for the prediction of the necessary postoperative IOL refractive power, the axial length (AL), anterior chamber depth (ACD), lens thickness (LT), central cornea thickness (CCT) and posterior chamber depth (PCD). Additionally, the cornea curvature radius / keratometry (K) is collected via a keratometry measurement. The measurements obtained by today's OCT devices are very precise. Like this, e.g. the measurement error of the axial length, one of the initially most impacting reasons for calculation errors, has been reduced below a 5% impact [103].

Development of IOL calculation approaches

To assure that after a cataract surgery the patient does not require strong visual aids, it is essential that the necessary IOL refractive power is precisely determined prior to surgery. This need for a precise calculation resulted in the development of an increasing amount of IOL calculation methodologies. Based on the used methods, the different IOL calculation approaches that have developed over time can be separated into different generations. The first and second generation were based on either physical modeling or simple empirical approaches. For the physical approaches, all concepts relied on the vergence principle, a simple physical concept that essentially describes the IOL within the optical system as a refractive object that changes the curvature radius of an incoming light wave to focus it onto the retina[22, 75]. The model therefore relies on the description of light as a wave within the wave / particle dualism of light within the physical theory. The vergence $V = n/R$ is described as the refractive index n of a medium which is passed by the wave front divided by the curvature radius R of the wave front and therefore correspondent to a refractive power. The IOL refractive power can therefore be described as the difference in vergence V_1 and V_2 before and after the incoming light wave has passed the IOL, where both the cornea and IOL are approximated as thin lenses, shown in Eq. 7.2 (details on optical physics are provided in Sec. 7.2.1):

$$P_{IOL} = V_2 - V_1, \quad V_2 = \frac{n}{AL - ACD_{IOL}}, \quad V_1 = \frac{n}{\frac{n}{P_C} - ACD_{IOL}} \tag{7.2}$$

Here, n corresponds to the refractive index within the eye media, P_C corresponds to the refractive power of the cornea and n/P_C represents the curvature radius of the wave front at the cornea. The ACD_{IOL} is not the geometrically correct anterior chamber depth after insertion of the IOL, but instead an effective lens position (ELP) corresponding to the thin lens model that is approximated differently depending on the calculation method used (e.g. the cornea dome approximated using the measured keratometry [75]). Inserting the measured

biometric parameters for a patient eye into Eq. 7.2 results in an estimate of the necessary IOL refractive power to achieve emmetropia (zero refractive error) of the eye. The purely empirical approaches on the other hand can be summarized within the SRK I and SRK II formulas [203]. Using a simple linear regression on the two biometric values AL and K for the prediction of P_{IOL}, they obtain a multi-linear model that directly expresses the IOL refractive power for a given patient eye. Within SRK II, the assumption on the intercept value was slightly modified, but the general concept remained the same. By today's standards, all these initial attempts to perform IOL calculation are outdated. The vergence concept is however still the base model for many formulas even used today, although especially the assumptions on the ELP have been substantially optimized. Since the exact position of the IOL after surgery is one of the few parameters that cannot be measured prior to surgery (since the IOL has not been inserted yet), this factor needs to be predicted. The position shows a relation to other measured biometric parameters, but its prediction is far from trivial and subject to ongoing work [102, 174]. At the same time, a wrongly assumed position of the IOL within the system can have a substantial impact on the refractive outcome error. [149] estimated that for a standard human eye the relation between IOL position prediction error ΔACD_{IOL} and eye's refractive outcome error ΔP_{eye} can be estimated as $\Delta P_{eye} = -0.03 + 1.37 * \Delta ACD_{IOL}$, therefore an error of 1mm results in a miscalculation of over 1.3 diopters (D). Besides the limited consideration of additional parameters, this aspect alone explains why the often constant assumed ELP within the first models make them irrelevant by today's standards.

The third generation of IOL calculation formulas, which constitutes of the HofferQ [105], Holladay 1 [108] and SRK/T [193] formula, brought an improvement by calculating an individual ELP for every patient based on their biometry, each relying on different concepts respectively. Additionally, individual lens constants for different lens types were introduced. The formulas however still consider only two biometric measurements, the axial length and the keratometry. This changed with the introduction of formulas like Haigis and Barrett Universal II [15, 16, 230], which marked the beginning of the fourth generation of IOL calculation formulas. With the increasing capabilities of biometry measurement devices, the usage of additional relevant biometric parameters like the anterior chamber depth and lens thickness where introduced, which allowed a more individualized prediction for different patient eyes. Although all these models are still relying on the vergence concept, the model inaccuracies are compensated by the increasing amount of used parameters and additional tuning factors, which were continuously updated and improved over the years. A different approach not relying on the vergence model, but still based on physical assumptions uses the more complex concept of raytracing [101, 106, 176, 188]. Here, the light rays entering the eye are individually modeled on their path through the optical system, allowing a more detailed and distinctive description of the imaging process onto the retina.

Within recent years, there is also an increasing amount of IOL calculation approaches based on machine learning [2, 48, 50, 100, 220, 256]. Unfortunately, for most of the newer concepts, there is no publication about the methodological details available due to proprietary interests. From documentations on e.g. the website of the Hill RBF method the approach is described as a data driven radial basis function network approach, which is trained on more than 30,000 cases for its newest version 3.0. The huge amount of data necessary for training highlights the problem that we are trying to overcome. Instead of trying to incorporate the highly informative modeling of optical physics into the learning process, most of the approaches

rely fully on empirical data only. In general, even when considering physical assumptions, the combination of physical modeling is not ideal. As long as the network is trained on the data provided only, every relation has to be deduced from it, not considering any additional informative priors.

The approach we develop and evaluate within the next sections will overcome this boundary by explicitly introducing physical knowledge into a neural network based on a domain-specific loss design motivated from precise optical physics. The model therefore has already fully incorporated the physical prior into the learning system and only needs to learn the small deviation from reality from the data.

7.2 Modelling of human eye optics

7.2.1 Optical physics of the eye - Geometrical optics

In order to understand the physical modeling approach used as a scientific domain prior within our approach, we have to briefly explain its underlying concepts. Within our description, we are going to use the underlying theory of geometrical optics. A precise description of the propagation behavior of an electromagnetic light wave would normally require the consideration of its wave-like character, which results in a description of concepts like refraction as a result of constructive and destructive interference [61]. Since in an isotropic optical medium (constant refractive index n) the propagation direction corresponds to the normal on the wavefront of the electromagnetic wave, within the approximation of geometrical optics the light propagation can be completely described by this normal, called the light ray. This approximation is valid as long as the diameter of the full light bundle is large compared to the wavelength of the light, since in this case refraction is negligible in the center of the ray. The process of refraction and reflection is then described as a change in the propagation direction of the individual light rays.

Resulting from the assumption of light propagation via rays, we can deduce that the image of an object in front of an optical element is depicted sharply behind it, when all rays propagating from one point of the object through the optical element are focused onto one point behind the element. The resulting image is called the virtual image. This concept of virtual imaging in geometrical optics is essential for the understanding of the optical system within the human eye. The purpose of the complex optical system of cornea and natural lens in the eye is to create a sharp virtual image of surrounding objects on the retina. As described in Fig. 7.6, this image on the retina is upside down and re-inverted during the processing of the incoming signals from the light-sensitive cells of the retina in the brain. Using this knowledge, we can now create a precise description of the optical properties that the eye needs to fulfill in order to map the image of an object of interest properly onto the retina. Within optical vision tests, the target objects are normally in a distance of about 6 m (optical "infinity"), we can therefore approximate the incoming light rays from one point of an object as nearly parallel. In order to map the object sharply onto the retina, these incoming light rays need to be focused on the retina within one point by the optical system of the eye. An eye capable of receiving this focus, is called emmetropic, therefore having a refractive error of zero diopters.

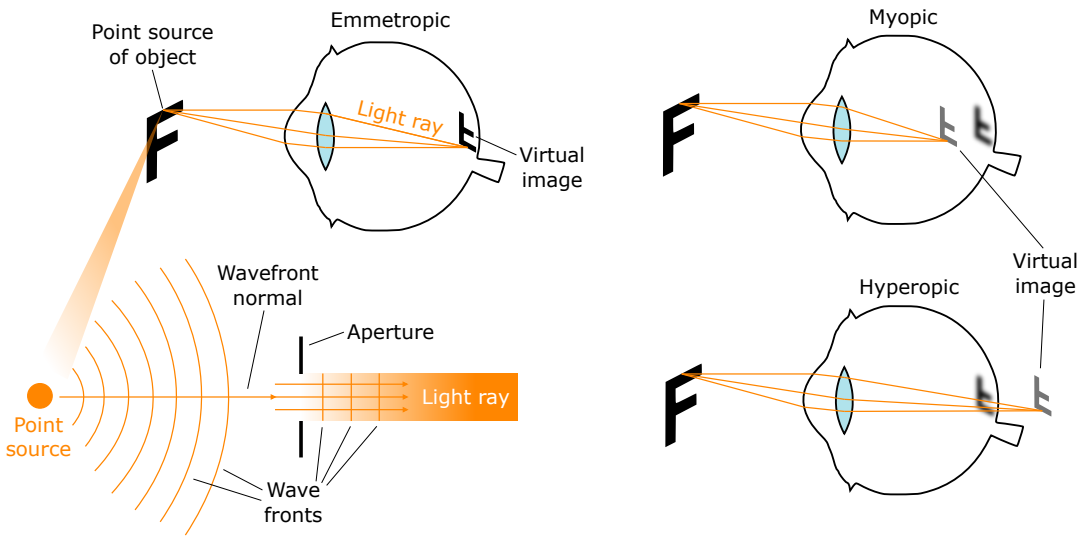


Fig. 7.6. Schematic depiction of the optical imaging process within the human eye. Every point of the object (here the letter "F") corresponds to a point source that emits a light wave, as depicted for an example point at the bottom left. When the wavefronts hit an aperture, which is large in comparison to the light's wavelength, they continue their propagation in the confined direction given by the aperture due to inference processes, describable as a light ray. Such an aperture is e.g. given by the eye's cornea, which is passed by several rays. When all rays emitted from one point source are again focused on the same point on the retina, this object point is seen sharply, since the virtual image forms on the retina. As depicted on the right, this situation changes when the refractive power of the optical system is not properly adjusted. The virtual image forms either in front of or behind the retina, leaving a blurry image on the receptor cells. Depending on the virtual image position, we either talk about myopia or hyperopia.

The diopter $\mathrm{D} = 1/\mathrm{m}$ is the physical unit of the refractive power P of an optical element, being antiproportional to the distance of its focal point f, therefore $\mathrm{P} = 1/\mathrm{f}$ within air [61]. When describing an eye using this terminology, the number of diopters refers to the refraction correction necessary to obtain emmetropia. If the refractive power of an eye is too weak, the correction by e.g. glasses needs to provide additional focusing refractive power to the eye, therefore having a positive diopter value. An eye with this optical condition is called hyperopic (farsighted), since it does not have the refractive power to focus onto near objects, the virtual image is created behind the retina. On the other hand, if the eye's refractive power is too weak, it is described as myopic (nearsighted). It refracts the light too strongly, therefore not allowing to view distant objects properly, the virtual image forms in front of the retina (see also Fig. 7.6). The corresponding correction needs to have a negative refractive power, effectively diverting incoming light rays. Myopic eye's are therefore described with a negative diopter value. The refractive target of a cataract surgery therefore directly refers to the optical correction that would be necessary to achieve emmetropia. In order to express the above explained refractive properties of the eye quantitatively, it is necessary to define a mathematical relation between the eye's properties and its refractive power. As explained above, within an isotropic optical medium with constant refractive index (speed of light is constant in the medium), the normal on a wave front and correspondingly the approximated light ray does not change its propagation direction. Under the constraint that we consider only optical media with different but within one medium constant refractive indices, light refraction and a change in propagation does therefore only occur at the surface boundary between two optical media with different refractive indices n_1 and n_2 (e.g. air and water). This refraction directly follows Snell's law which provides a relation between the angles α_1

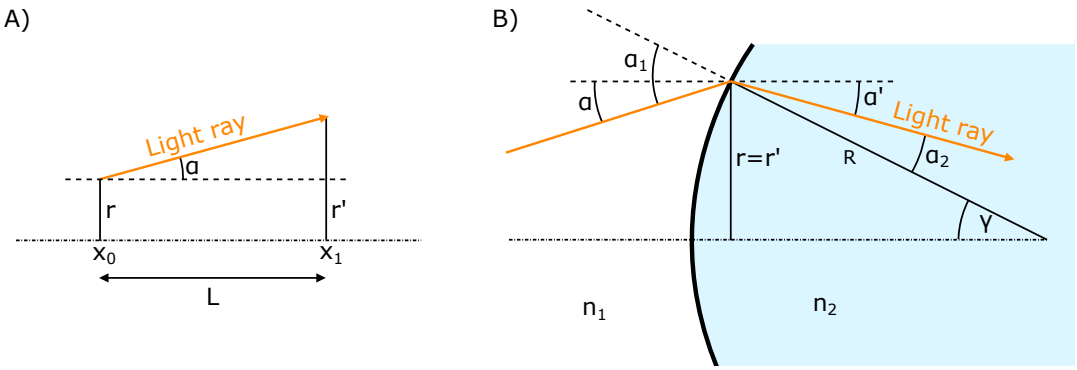


Fig. 7.7. Schematic depiction of the light ray propagation process and transformation of light ray vector $\vec{v}$ with quantities r and α when A) the ray travels through an optical medium with constant refractive index and B) the ray passes a surface boundary between two optical media with respective refractive indices n_1 and n_2 and is refracted at the boundary.

and α_2 of a ray propagation direction with the normal on an optical surface boundary before and after the boundary in Eq. 7.3 [61]. Here, we are further assuming that the light rays are propagating in a similar direction to this normal on the optical boundary and therefore have a rather small angle α. Within this so-called paraxial approximation (the term follows the assumption that the described surface boundaries have normals nearly parallel to the optical axis of the system), we can approximate $\sin(\alpha) \approx \alpha$:

$$\frac{\sin(\alpha_1)}{\sin(\alpha_2)} = \frac{\mathrm{n}_2}{\mathrm{n}_1} \approx \frac{\alpha_1}{\alpha_2} \tag{7.3}$$

The change in angle is therefore directly proportional to the change in refractive index. We can use this relation to derive how a light ray is refracted when passing a surface boundary between to refractive media (e.g. from air into the cornea). Here, we approximate the refractive surfaces within the eye as spherical surfaces with curvature radius R. Within Fig. 7.7, the refraction of the light ray is visualized. We are interested in the focal point f of the optical system, the point where the optical axis is intersected by the light ray. As described above, this focal point is directly related to the refractive power of the element by using $\mathrm{P} = 1/\mathrm{f}$. We can calculate the distance of the ray's intersection point with the optical axis either as $r = R\sin(\alpha)$ or $r = \mathrm{f}\sin(\gamma)$. Using these equivalent formulations and the relation $\gamma = \alpha - \beta$ following from Fig. 7.7, we receive $R\sin(\alpha) = \mathrm{f}\sin(\alpha - \beta)$. Using the trigonometric relation $\sin(\alpha - \beta) = \sin(\alpha)\cos(\beta) - \cos(\alpha)\sin(\beta)$ and the paraxial approximation (small angles α and β), which yields $\cos(\alpha) \approx \cos(\beta) \approx 1$, we can deduce $\sin(\alpha - \beta) \approx \sin(\alpha) - \sin(\beta)$. From Eq. 7.3, we further know that $\mathrm{n}_1 \sin(\alpha) = \mathrm{n}_2 \sin(\beta)$. We can therefore create the paraxial description of focal point and refractive power respectively as:

$$\mathrm{P} = \frac{1}{\mathrm{f}} = \frac{\sin(\alpha - \beta)}{\sin(\alpha)R} \approx \frac{\sin(\alpha) - \sin(\beta)}{\sin(\alpha)R} = \frac{\sin(\alpha) - \sin(\alpha)\mathrm{n}_1/\mathrm{n}_2}{\sin(\alpha)R} = \frac{\mathrm{n}_2 - \mathrm{n}_1}{\mathrm{n}_2 R} \tag{7.4}$$

The found relation is therefore described by the refractive indices of the two optical media and the curvature radius of the surface boundary only. This relation will be helpful for the description of the eye's optical model in Sec. 7.2.2. We know that all incoming light rays of an object point have to focus onto one point within the created virtual image in order to receive proper imaging. When the location of this point is known, it is therefore sufficient to describe the propagation of one ray in order to describe the complete optical system. We will do so by

a very effective physical modeling concept, called ray transfer matrix analysis. This concept will be described within the next section.

7.2.2 Ray transfer matrix analysis

Within the concept of ray transfer matrix analysis (RTMA), every point of the light ray propagation process is described as a vector $\vec{v}$ with two quantities (Eq. 7.5): the distance r between the considered point of the ray and the corresponding closest point of the optical axis and the angle α of the current ray propagation direction at this point with the optical axis. Following the previous assumptions, we can now describe the ray propagation process in the optical system with two types of linear transformations applied onto the ray vector representation $\vec{v}$ to transform it into $\vec{v}'$, expressed by r' and α': one element to describe the propagation with constant direction within one optical medium and the change in direction at a surface boundary, described by the matrices **A** and **B** in Eq. 7.5 respectively. We will derive these matrices in the following.

$$\vec{v} = \begin{bmatrix} r \\ \alpha \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} 1 & \mathrm{L} \\ 0 & 1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 & 0 \\ \frac{(\mathrm{n}_1 - \mathrm{n}_2)}{\mathrm{n}_2 R} & \frac{\mathrm{n}_1}{\mathrm{n}_2} \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{\mathrm{f}} & 1 \end{bmatrix} \tag{7.5}$$

Within one optical medium, the angle α of the ray with the optical axis will not change during propagation from a point x_0 to a point x_1 with a distance L between them projected onto the optical axis, therefore $\alpha' = \alpha$. Following simple geometry, the value of r will change to $r' = r + \sin(\alpha) \cdot \mathrm{L} \approx r + \alpha \cdot \mathrm{L}$ (see Fig. 7.7A). Therefore, applying the linear transformation **A** of Eq. 7.5 to vector $\vec{v}$ will describe the new vector $\vec{v}' = \mathbf{A} \cdot \vec{v}$ at the end of the medium. At the surface boundary between two media, no effective propagation takes place, instead the propagation direction changes. The description of this process is more complex, since $r' = r$ stays constant, while α is modified following the laws of refraction. Within a human eye, the surfaces are defined by the optical elements of the cornea and human lens. We can approximate these boundaries as spherical surfaces with curvature radius R. Within Fig. 7.7B, the refraction at such a sphere is depicted. We know from Eq. 7.3 that the relation between the angles α_1 and α_2 is given by $\mathrm{n}_1\alpha_1 = \mathrm{n}_2\alpha_2$. At the same time, we can deduce the relationship $\alpha_1 - \alpha = -\alpha' + \alpha_2 = \gamma = r/R$ (α' is calculated counter-clockwise). Inserting this relation into Eq. 7.3 yields ([61]):

$$\mathrm{n}_1 \left(\frac{r}{R} + \alpha \right) = \mathrm{n}_2 \left(\frac{r}{R} + \alpha' \right) \quad \rightarrow \quad \alpha' = \frac{\mathrm{n}_1}{\mathrm{n}_2}\alpha + \frac{\mathrm{n}_1 - n_2}{\mathrm{n}_2}\frac{r}{R} \tag{7.6}$$

The linear transformation **B** of Eq. 7.5 is therefore describing the refraction process $\vec{v}' = \mathbf{B} \cdot \vec{v}$ at the surface boundary. It is interesting to see that the found relation within $\mathbf{B}[1,0]$, the lower left element of **B**, is directly linked to the expression found for the refractive power P in Eq. 7.4, namely $\mathbf{B}[1,0] = -\mathrm{P} = -1/\mathrm{f}$. In fact, for the relations derived within Sec. 7.2.2 and Sec. 7.2.3 we can extract the focal point and corresponding refractive power of the optical systems by performing the matrix multiplication on its individual elements and extract the lower left entry of the resulting matrix, as will be shown later.

In general, we describe an optical element like the human lens as two surface boundaries with curvature radius R incorporating an optical medium with deviating refractive index n_2 from

its surrounding with n_1 and a distance L between the two intersections of the surfaces with the optical axis. For simplicity, we assume that the anterior and posterior surfaces have the curvature radii R and $-R$ respectively. Following the derived relations in Eq. 7.5, we can describe the transformation **M** of ray vector $\vec{v}$ into $\vec{v}'$ by the optical element as:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ \frac{n_2 - n_1}{-n_1 R} & \frac{n_2}{n_1} \end{bmatrix} \begin{bmatrix} 1 & L \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{n_1 - n_2}{n_2 R} & \frac{n_1}{n_2} \end{bmatrix} = \begin{bmatrix} 1 + \frac{n_1 - n_2}{n_2 R} L & \frac{n_1}{n_2} L \\ \frac{2(n_1 - n_2)}{n_1 R} + \frac{(n_1 - n_2)^2}{n_1 n_2 R^2} L & 1 + \frac{n_1 - n_2}{n_2 R} L \end{bmatrix} \tag{7.7}$$

The negative entry $\mathbf{M}[1,0]$ directly corresponds to the refractive power P of a symmetric optical thick lens, defined by Eq. 7.8 [61]. This definition provides a useful relation between the refractive power of a lens and its curvature radius that we will rely on later:

$$P = -\mathbf{M}[1,0] = \frac{n_2 - n_1}{n_1} \cdot \left(\frac{2}{R} - \frac{(n_2 - n_1) \cdot L}{n_2 \cdot R^2} \right) \tag{7.8}$$

If the distance L between the two surfaces is small enough, we can approximate the optical element as a thin lens with $L \to 0$ mm. From Eq. 7.7, we can directly derive that this thin lens is described by a strongly simplified relation. Its application to the ray vector $\vec{v}$ is shown in Eq. 7.9.

$$\mathbf{M} \cdot \begin{bmatrix} r \\ \alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{2(n_1 - n_2)}{n_1 R} & 1 \end{bmatrix} \begin{bmatrix} r \\ \alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} r \\ \alpha \end{bmatrix} = \begin{bmatrix} r \\ -\frac{r}{f} + \alpha \end{bmatrix} \tag{7.9}$$

The description of a thin lens within RTMA can therefore be approximated as a linear transformation only described by an anti-proportional relation to the focal distance f of this element, also defined in matrix **C** in Eq. 7.5. We again extract this focal distance and optical refractive power P of a thin lens as $P = 1/f = 2(n_2 - n_1)/(n_1 R)$.

By an alternating application of the linear transformations **A** and **B** (or **C**) of Eq. 7.5, we can very precisely describe the propagation of a light ray through a complex optical system in paraxial approximation. We can now use this powerful concept to describe the propagation of a light ray through the optical system of the eye with an incorporated intraocular lens.

7.2.3 Single-ray raytracer design of human eye

We will leverage the powerful concept of RTMA to model the optical system of the human eye. Since the description effectively describes the propagation of one light ray through the system, we will refer to the designed model as a single-ray raytracer (SRR). The human eye consists of the surfaces of the cornea and the lens as well as the optical media in between them. Since we want to describe the optical system of the eye after the intraocular lens (IOL) has been implanted during cataract surgery, the lens here refers to the implanted IOL. As described in Sec. 7.2.1, the goal of the cataract surgeon is to implant an IOL into the optical system of the eye which leads to a corresponding target refractive error suitable for the patients needs. As described, the term target refraction corresponds to the necessary correction for the refractive power error of the eye after surgery. Normally, this error is chosen to 0 diopters (D) (emmotropia), so no visual correction would be necessary after surgery. Sometimes, the surgeon however aims for a slight myopia (nearsightedness), so that the patient can easily e.g.

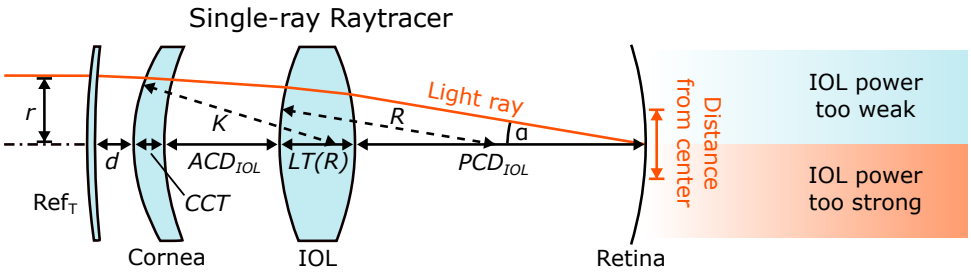


Fig. 7.8. Schematic depiction of the light ray propagation process within the human eye after surgery, the refractive target is mimicked as a refractive lens (e.g. glasses) in front of the eye. The light enters the optical system parallel to the optical axis and is refracted by the several refractive surfaces onto the retina. If the ray hits the retina above the center in this modeling (positive distance to the center), the IOL refractive power was chosen too weak, if it hits below the center (negative distance to the center) it was too strong. The shown refraction angles are only schematic.

read a book. Since the refractive power of the eye is therefore slightly to strong, the corrective target refraction is represented by a negative refractive power, e.g. -0.5 D. Within the model, the optical element corresponding to this correction is placed in front of the eye in a certain distances, the so-called vertex distance d, a concept directly relatable to glasses in front of the eye. Here, the correction is approximated as a thin lens, therefore describable by matrix **C** in Eq. 7.5. By incorporating this correction directly into the description, we can use the SRR for the calculation of an IOL refractive power that results in emmmetropia (0 D error) of the complete system and automatically receive an eye with the intended refractive error. The overall optical system within the SRR therefore consists of the surfaces of the cornea and human lens, the refractive correction as well as all optical media in between. The optical axis of the eye intersects all optical surfaces such that the boundaries' gradients are perpendicular to the optical axis, see Fig. 7.8.

As described in Sec. 7.2.1, in order to have an emmetropic eye, all incoming light rays from one point of an object of interest have to focus onto one point of the retina, otherwise the image is not seen sharply. Since visual tests refer to a larger distance to the object (around 6 m), we can approximate the incoming light rays as parallel. When the eye focuses on a point, the light rays enter the eye parallel to the optical axis, the optical system is therefore emmetropic, when all rays hit the center of the retina (the intersection of the optical axis and the retina). Within the terms of RTMA, this corresponds to an $r' = 0$ for the descriptive vector $\vec{v}$ of the ray intersecting the cornea independent from α', visualized in Fig. 7.8. The incentive of our SRR is therefore defined, we need to find an IOL refractive power for which a light ray parallel to the optical axis (r is arbitrary, $\alpha = 0$) intersects the retina at the same point as the optical axis ($r' = 0$, α' is arbitrary). In order to describe the complete system, we can rely on the biometric measurements of the human eye based on the OCT imaging scan shown in Fig. 7.5A. Here, we need to replace the human lens with the IOL, as shown in Fig. 7.8. Since we cannot measure the position of the IOL before the cataract surgery (the IOL is not implanted yet), this parameter is the one quantity in the system that needs to be predicted. The corresponding process will be described in Sec. 7.3.2.

In order the encode the different refractive elements of the eye within a RTMA approach, we require the following biometric and refractive quantities (visualized in Fig. 7.8): the refractive target value Ref_T corresponding to the planned deviation from emmetropia $\Delta\text{P}_\text{eye}$

(since $\Delta P_{eye} = Ref_T = 1/f_T$ in matrix C for Eq. 7.5 [61]), the central cornea thickness CCT, the cornea curvature radius K (called keratometry), the postoperative anterior chamber depth ACD_{IOL} (defined as the distance between the posterior cornea and anterior IOL surfaces), the curvature radius R of the IOL, the IOL thickness LT(R) (which is defined as a function of the curvature radius R), the axial length AL defined as the distance between the anterior surface of the cornea and the retina and the postoperative posterior chamber depth $PCD_{IOL} = AL - CCT - ACD_{IOL} - LT(R)$, which can be expressed by the other axial measurements. If the specific relation is unknown for a given IOL type, we approximate the IOL thickness with a constant value $LT(R) = 1$ mm. Additionally, we need to consider the refractive indices of the cornea n_C, aqueous and vitreous n_V (fluid in anterior and posterior chamber respectively, see Fig. 7.1) as well as the intraocular lens n_L. As described previously, within the SRR a light ray parallel to the optical axis (r is arbitrary, $\alpha = 0$) needs to be refracted onto the center of the retina ($r = 0$, α is arbitrary). We can therefore create the following matrix description of the complete refraction process Eq. 7.10

$$\begin{bmatrix} 0 \\ \alpha \end{bmatrix} = \begin{bmatrix} 1 & PCD_{IOL} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{(n_L - n_V)}{-n_V R} & \frac{n_L}{n_V} \end{bmatrix} \begin{bmatrix} 1 & LT(R) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{(n_V - n_L)}{n_L R} & \frac{n_V}{n_L} \end{bmatrix} \begin{bmatrix} 1 & ACD_{IOL} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ \frac{(n_C - n_V)}{n_V \frac{6.8}{7.7} K} & \frac{n_C}{n_V} \end{bmatrix} \begin{bmatrix} 1 & CCT \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{(1 - n_C)}{n_C K} & \frac{1}{n_C} \end{bmatrix} \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -Ref_T & 1 \end{bmatrix} \cdot \begin{bmatrix} r \\ 0 \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} r \\ 0 \end{bmatrix} \tag{7.10}$$

Here, the refractive index of the cornea $n_C = 1.376$ [175], the aqueous and vitreous $n_V = 1.336$ [175] are constant values for the human eye. The factor $\frac{6.8}{7.7}$ is the Gullstrand ratio, which provides the standard ratio of anterior and posterior cornea surface curvature [175]. The matrix **M** refers to the complete performed matrix multiplication of all elements. We can see that this equation does not explicitly contain the IOL refractive power, instead the quantity defining the refractive impact of the IOL is its curvature radius R. This curvature radius is however directly related to the IOL refractive power P_{IOL} using the thick lens equation defined in Eq. 7.8 [61]:

$$P_{IOL}(R) = \frac{n_L - n_V}{n_V} \cdot \left(\frac{2}{R} - \frac{(n_L - n_V) \cdot LT(R)}{n_L \cdot R^2} \right) \tag{7.11}$$

In combination with Eq. 7.10, we have therefore defined the physical modeling concepts to consistently relate the necessary IOL refractive power to the specific requirements of a human eye undergoing cataract surgery.

7.3 Development of Optical network

7.3.1 Design of unsupervised physical loss function

The purpose of the derived terminologies is to create a network capable of performing a precise IOL refractive power prediction considering detailed physical knowledge. To achieve such a network, we will now come back to the concepts of domain-specific loss design introduced in Sec. 6.1.2. The found relation in Eq. 7.10 provides a fully differentiable relation between the curvature radius (and respective refractive power) of an IOL inserted into an eye with given

biometry and the intersection point of the light ray passing this IOL with the retina. Since the model provides this differentiable calculation for any values of the biometric quantities involved, it covers the complete data domain with a very strong and informative domain prior. Since we are only interested in the intersection point with the retina (distance to optical axis r'), we can focus on the first of the two equations received by performing the matrix multiplication in Eq. 7.10, namely: $0 = r' = \mathbf{M}[0,0] \cdot r + \mathbf{M}[0,1] \cdot 0 = \mathbf{M}[0,0] \cdot r$. This is the equation which needs to be solved, while the target value of the equation is the intersection point with the retina r' (for which we want to achieve the value zero). Here, $\mathbf{M}[0,0]$ contains all biometric and refractive values of the eye as well as the curvature radius of the IOL R. Since this value directly describes the searched IOL power, the value for which $\mathbf{M}[0,0]$ needs to be solved is R, where R_{true} refers to the value for which $r' = 0$. Following Eq. 6.2, we define the target variable $\mathbf{Z}$ of the implicit physical equation to be $\mathbf{Z} = r'$, the intersection point with the retina. The variable $\mathbf{X} = [\text{AL},\ \text{ACD}_{\text{IOL}},\ \text{CCT},\ \text{K}_{\text{max}},\ \text{K}_{\text{min}}]$ is defined by the biometric measurements of the eye, $\mathbf{\Omega} = [\text{Ref}_{\text{T}}]$ is the refractive target value of the surgery and $\mathbf{Y} = R$ is the target value which we would like to predict. We can therefore define:

$$\begin{aligned} \mathbf{Z} = \mathbf{M}[0,0](R, \mathbf{X}, \text{Ref}_{\text{T}}) \cdot r \propto \mathbf{M}[0,0](R, \mathbf{X}, \text{Ref}_{\text{T}}) \\ \text{with } 0 = \mathbf{M}[0,0](R_{\text{true}}, \mathbf{X}, \text{Ref}_{\text{T}}) \end{aligned} \tag{7.12}$$

Since this equation only provides an implicit calculation of the searched value R_{true}, we cannot easily setup a training process for our network similar to Eq. 6.1. We therefore will create a domain-specific loss function following the laws of optical physics as derived in our general description for a physics-based loss in Eq. 6.7 and Eq. 6.8. This loss should force the network to learn a correct prediction of the IOL refractive power P_{IOL} for given eye biometry and refractive target, but Eq. 7.12 only provides a relation to the curvature radius R of the IOL. This problem can however easily be solved by the inversion of Eq. 7.11, given by:

$$R(\text{P}_{\text{IOL}}) = \frac{\text{n}_{\text{L}}}{(\text{n}_{\text{L}} - \text{n}_{\text{V}})\ \text{LT}(\text{P}_{\text{IOL}})} - \sqrt{\frac{\text{n}_{\text{L}}^2(\text{n}_{\text{L}} - \text{n}_{\text{V}}) - \text{P}_{\text{IOL}}\text{n}_{\text{L}}\text{LT}(\text{P}_{\text{IOL}})}{(\text{n}_{\text{L}} - \text{n}_{\text{V}})^2\ \text{LT}(\text{P}_{\text{IOL}})^2}} \tag{7.13}$$

By inserting Eq. 7.13 into Eq. 7.12, we receive a direct relation between the IOL refractive power and the intersection point with the retina which directly corresponds to the refractive error obtained after surgery. Following Eq. 6.7, we then formulate our physical loss function as:

$$\text{loss}_{phy}(\text{P}_{\text{IOL}}) = \mathbf{M}[0,0](R(\text{P}_{\text{IOL}}), \mathbf{X}, \text{Ref}_{\text{T}})^2 \tag{7.14}$$

It is notable that we used Eq. 6.7 instead of Eq. 6.8 for the loss setup. The reason for this is that the function $\mathbf{M}[0,0](R(\text{P}_{\text{IOL}}), \mathbf{X}, \text{Ref}_{\text{T}})$ is a fully differentiable monotone function which therefore provides only one root given by $\text{P}_{\text{IOL}}(R_{\text{true}})$. The function is depicted within Fig. 7.9A. Hence, Eq. 7.14 directly yields a function which provides the correct physical solution of IOL refractive power for a given biometric eye input as its minimum (Fig. 7.9C). By training a neural network using this function, it is possible to mirror the optical physics within the system completely unsupervised.

7.3.2 OpticNet - Training process

In order to incorporate and persistently enforce the optical knowledge encoded within the physical loss function given by Eq. 7.14, we separate the training of the neural network into

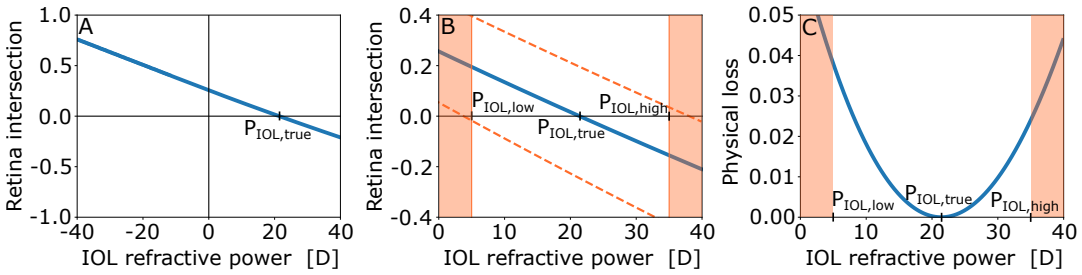


Fig. 7.9. Analysis of Eq. 7.12 and Eq. 7.14 for inputs $\hat{\mathbf{X}}$: AL = 24 mm, ACD_{IOL} = 4.5 mm, CCT = 0.5 mm, K_{max} = 8.0 mm, K_{min} = 7.80 mm, Ref_{Tar} = -0.5 D (standard biometric parameters for a human eye). Figure A shows the plot of Eq. 7.12, the root of the function marks the searched IOL refractive power marked as $P_{IOL,true}$ to reach perfect vision for an eye with $\hat{\mathbf{X}}$. For the physical loss, we want to consider only positive IOL refractive powers. By sampling an arbitrary amount of random input parameters $\hat{\mathbf{X}}_{simu}$ for our unsupervised pre-training, we inevitably receive a large variety of corresponding IOL refractive power $P_{IOL,true}$. We limit the area of valid refractive powers, shown in Figure B. For every $\hat{\mathbf{X}}$, we calculate the result of Eq. 7.12 when inserting the two IOL power boundaries $P_{IOL,low}$ and $P_{IOL,high}$. Here, we chose the valid IOL refractive power range of 5 D to 35 D. When the result is larger than zero for $P_{IOL,high}$ or lower than zero for $P_{IOL,low}$, it becomes clear from Figure B that we have entered the invalid radii area for $P_{IOL,true}$ (orange dashed plots, the function root is placed in the orange zone) and we therefore omit these samples (detailed description in Sec. 7.3.2). Figure C finally shows the resulting physical loss function that can be used for an unsupervised pre-training of our neural network.

two distinct parts. In a first step, the neural network is trained on a large amount of simulated artificial biometric data, which is used to incorporate the optical knowledge on the full data domain into the network. In a second step, this physics-informed network is fine-tuned on real patient cases to learn the deviation of reality from the physical model while continuously considering the physical constraint. Here, the constraint is kept within the fine-tuning process to prevent a potential occurrence of catastrophic forgetting. The chosen neural network type is a standard multilayer perceptron.

Physical pre-training

The physical loss defined by Eq. 7.14 contains all necessary information to transfer the optical knowledge of the designed single-ray raytracer into a randomly initialized artificial neural network. The objective function that needs to be optimized is defined by Eq. 7.15

$$\min_{\mathbf{W}} \ \text{loss}_{phy}(\mathbf{\Phi}(\mathbf{X}, \text{Ref}_\text{T})) = \min_{\mathbf{W}} \ \mathbf{M}[0,0](R(\mathbf{\Phi}(\mathbf{X}, \text{Ref}_\text{T})), \mathbf{X}, \text{Ref}_\text{T})^2 \tag{7.15}$$

where W defines the learnable parameters of the network $\mathbf{\Phi}$. This optimization task does not require any additional information on the inputs provided. Whenever the biometric information $\mathbf{X}$ and Ref_T of an eye is provided as input to the network, the output of the network (corresponding to the predicted necessary IOL refractive power P_{IOL} to focus a light ray onto the retina center) will be evaluated within the physical loss with respect to reaching an emmetropic outcome (the deviation of the ray from the retina center and correspondingly the error of the network). Therefore, it enforces physical consistency of the network prediction in an unsupervised manner.

We will leverage this loss property by sampling a large amount (> 100,000) of artificial eyes and train our network on this simulated dataset. The simulation process is performed by setting a medically realistic range for every biometric parameter x defined by x_{min} and x_{max}

and sample the corresponding parameter from a uniform distribution limited within this range. The sampling distribution is therefore given by Eq. 7.16.

$$f(x) = \begin{cases} \frac{1}{x_{\text{max}} - x_{\text{min}}} & \text{for } x_{\text{min}} \leq x \leq x_{\text{max}} \\ 0 & \text{otherwise} \end{cases} \tag{7.16}$$

The uniform character of the sampling process and the large amount of samples created guarantees that the complete data domain is statistically sufficiently covered. However, depending on the provided simulated biometric input, the corresponding IOL refractive powers $\text{P}_{\text{IOL,true}}$ and resulting R_{true} which solve Eq. 7.12 can vary largely, also medically unrealistic combinations of biometric parameters might be sampled. For e.g. a very long eye with at the same time strong cornea curvature, which is already myopic without a human lens in place, it might be even required to use an IOL with negative curvature radius and therefore negative refractive power (a concave lens). On the other hand, short eyes with flat corneas might require very large IOL refractive powers. The sampling ranges defined for every parameter are only able to create a realistic network input, they allow by itself no direct control on the obtained output. We want to also control the range of IOL refractive powers $\text{P}_{\text{IOL}}(R_{\text{true}})$ which are the correct solution for the given simulated samples. A direct way to achieve this would be to calculate the numerical solution for Eq. 7.10 for every single sample and remove all samples with out-of-range IOL refractive powers, this process is however time-consuming and redundant, since the physical loss is already designed to train the network for performing this task. Instead, we follow the logic of Eq. 7.12. Within Fig. 7.9B, we can directly deduce that an IOL refractive power smaller than $\text{P}_{\text{IOL}}(R_{\text{true}})$ will result in a light ray that intersects with the retina above the optical axis (light ray is not sufficiently refracted), the corresponding value r' of the RTMA vector $\vec{v}'$ will therefore be positive. When increasing the refractive power (stronger bending of the light ray), the value r' will decrease, reach zero (center of the retina, $\text{P}_{\text{IOL}} = \text{P}_{\text{IOL,true}}$) and then move on into the negative space below the optical axis. The described propagation can be also seen within Fig. 7.9A. We can now set in interval of IOL refractive powers we want to consider within the simulation, defined by the interval boundaries $\text{P}_{\text{IOL,low}}$ and $\text{P}_{\text{IOL,high}}$. When the calculation of Eq. 7.12 for a $\mathbf{X}_{\text{simu}}$, $\text{Ref}_{\text{T,simu}}$ and $R(\text{P}_{\text{IOL,low}})$ has to a negative result, we can directly deduce that $\text{P}_{\text{IOL,true}} < \text{P}_{\text{IOL,low}}$. Similarly, if we receive a positive result for a $\mathbf{X}_{\text{simu}}$, $\text{Ref}_{\text{T,simu}}$ and $R(\text{P}_{\text{IOL,high}})$, it follows that $\text{P}_{\text{IOL,true}} > \text{P}_{\text{IOL,high}}$ (see also Fig. 7.9B). These considerations leave us with a clear mathematical relation to filter the simulated data pool, provided by Eq. 7.17:

$$\begin{aligned} 0 &\leq \mathbf{M}[0,0](R(\text{P}_{\text{IOL,low}}), \mathbf{X}_{\text{simu}}, \text{Ref}_{\text{T,simu}}) \\ 0 &\geq \mathbf{M}[0,0](R(\text{P}_{\text{IOL,high}}), \mathbf{X}_{\text{simu}}, \text{Ref}_{\text{T,simu}}) \end{aligned} \tag{7.17}$$

Compared to numerical solving procedures, this process only requires a uniform random sampling and two matrix multiplications to create a controlled simulated dataset that precisely and densely covers the considered data domain within seconds, not requiring any ground truth information. Using the designed physical loss function (Fig. 7.9C), we can now perform a standard gradient-descent based training approach to force the neural network into mirroring the behavior of the optical raytracing model, effectively forcing the RTMA description into the network weights. The resulting network can now be adapted to real patient data in the second part of the training process.

Data-based fine-tuning

The neural network received within the previous pre-training process encodes the eye's complete optical system described by a physical ray-tracing model. Resulting from the incorporated approximations and the general shortcomings of every modeling approach, the used physical model will show small deviations from reality. In order to remove these deviations, it is therefore required to fine-tune the neural network on real patient data. Since the pre-training process has already provided a huge knowledge benefit on the full data domain compared to an untrained network, we will however require only a small amount of real patient data to perform this fine-tuning step. Since it is a ubiquitous challenge in the medical domain to provide large amounts of training data, this knowledge advantage is especially beneficial.

For given biometry $\mathbf{X}$ and target refraction $\mathrm{Ref_T}$, we know that the pre-trained network $\mathbf{\Phi}$ has learned to provide the corresponding necessary IOL refractive power to achieve the requested target refraction within a physically consistent modeling. We now provide a training set of N real patient eyes with OCT imaging-based extracted biometric features $\mathbf{X}$, corresponding ground truth IOL refractive powers $\mathrm{P_{IOL,true}} = \mathbf{Y}_{\mathrm{GT}}$ and postoperatively measured eye refractive errors $\Delta\mathrm{P_{eye}}$, the respective deviation from emmetropia. The refractive error $\Delta\mathrm{P_{eye}}$ of an eye and the inserted IOL refractive power $\mathrm{P_{IOL,true}}$ form a ground truth pair for every patient, since the refractive error reflects the eye refraction that was actually achieved by inserting the IOL (ideally, this error should match the initially targeted refraction). Therefore, by using the measured refractive error after surgery as the target refraction $\mathrm{Ref_T} = \Delta\mathrm{P_{eye}}$ within our training dataset, we know that the inserted IOL is the correct choice for this target refraction, directly providing us with a label for the training data point. A similar setup was also proposed by [238], however only in the context of formula evaluation.

As described in Sec. 7.3.1, the input features of network Φ consist of $\mathrm{AL},\ \mathrm{ACD_{IOL}},\ \mathrm{CCT},\ \mathrm{K_{max}},\ \mathrm{K_{min}}$ and $\mathrm{Ref_T}$. From these input parameters, only $\mathrm{ACD_{IOL}}$, the distance of the IOL to the cornea cannot be measured preoperatively. It is therefore necessary to create a predictor for the quantity. The prediction of the postoperative IOL position is a research field on its own [73, 102, 174] and not focused on in detail within this work. We however performed an evaluation of the predictive importance of different preoperatively measured parameters. To reduce the risk of overfitting on the small amount of real data, we then performed a Partial Least Square (PLS) Regression on the found predictive biometries. The following regression is used:

$$\mathrm{ACD_{IOL}} = \mathrm{PLS}(\mathrm{AL},\ \mathrm{CCT},\ \mathrm{ACD_{HL}},\ \mathrm{LT_{HL}})\ , \tag{7.18}$$

where $\mathrm{ACD_{HL}}$ is the preoperative anterior chamber depth (the human lens is not removed yet) and $\mathrm{LT_{HL}}$ is the human lens thickness. Both quantities are preoperatively achievable and standard quantities reported by modern OCT devices. Additionally, they are considered as relevant input quantities by current IOL calculation formulas, the implicit inclusion into the prediction process can be therefore seen as beneficial. We can therefore create network input based on the biometric input vector $\mathbf{X}$ and Ω defined in Eq. 7.19:

$$\mathbf{X} = [\mathrm{AL},\ \mathrm{PLS}(\mathrm{AL},\ \mathrm{CCT},\ \mathrm{ACD_{HL}},\ \mathrm{LT_{HL}}),\ \mathrm{CCT},\ \mathrm{K_{max}},\ \mathrm{K_{min}}],\quad \mathbf{\Omega} = [\mathrm{Ref_T}] \tag{7.19}$$

For the transfer learning process on real data samples, two objectives need to be considered. First, the training process should learn to properly map the given inputs to correct IOL outputs

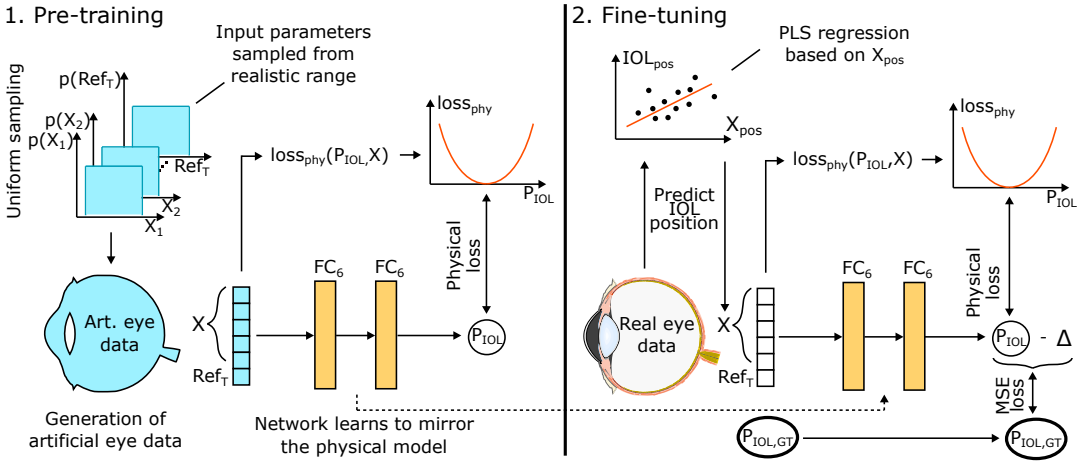


Fig. 7.10. Schematic description of OpticNet training process: Left) Using several uniform distributions defined on medically realistic ranges for each parameter, an arbitrary amount ($> 100,000$) of artificial eyes is sampled, their biometric values are used as input for an unsupervised training process of the neural network (FC_6 = fully connected layer with 6 hidden units) leveraging the physical loss function. During this pre-training, the physical concepts of human eye optics are enforced on the network. Right: The weights of the pre-trained network are transferred to the fine-tuning process. Here, a smaller amount of real patient cases is used to train the model further using a balanced loss function consisting of the MSE and physical loss.

for real data samples. At the same time, the already incorporated physical consistency should be preserved. To allow for this preservation, the transfer step needs to be both designed as smooth as possible and ideally still incorporate the constraints provided by the physical modeling. To achieve smooth adaption of the system, we manually remove the systematic offset of the physical model from the real data samples. By inserting $\mathbf{X}$ and Ω into Eq. 7.12 and solving the equation for $R(\mathrm{P}_{\mathrm{IOL,true}})$ numerically using the Newton-Raphson method shown in Eq. 6.5, we can calculate the average difference Δ between the physically calculated IOL refractive power and the ground truth values for the respective training dataset. Subtracting this value from the prediction outcome of Φ effectively removes the necessity for the system to learn this offset, keeping the representation closer to the physical model. Using the Mean Squared Error (MSE, Eq. 2.3) loss in Eq. 7.20, we can therefore train the network to adapt to the real data samples:

$$\mathrm{MSE}(\mathbf{\Phi}(\mathbf{X}, \mathrm{Ref_T}) - \Delta, \mathbf{Y}_{\mathrm{GT}}) \tag{7.20}$$

The offset supports the network in maintaining its originally learned representation. This is additionally supported by an early stopping routine of the training based on the results on a validation dataset for every training epoch. Nevertheless, it is possible that the training process substantially impacts the network performance with respect to physical consistency by e.g. erroneous data points or other influencing factors, leading to so-called catastrophic forgetting, the deletion of previously learned information. To prevent the deviation from physical consistency, we therefore add the unsupervised physical loss function as an additional weighted component to the used training loss. The neural network is punished when performing predictions which are strongly inconsistent with previously learned scientific knowledge. Both components of the loss function are weighted using an adjustable factor λ to balance the influence of real data on the network fine-tuning process, defined in Eq. 7.21:

$$\min_{\mathbf{W}} (1-\lambda)\ \mathrm{MSE}(\mathbf{\Phi}(\mathbf{X}, \mathrm{Ref_T}) - \Delta, \mathbf{Y}_{\mathrm{GT}}) + \lambda\ \mathrm{loss}_{phy}(\mathbf{\Phi}(\mathbf{X}, \mathrm{Ref_T})) \tag{7.21}$$

Tab. 7.1. Detailed specifications of used clinical datasets obtain from various surgery sites. For every dataset, the size, the origin of the dataset as well the covered ranges for important biometric and refractive parameters are provided (AL, ACD, P_{IOL}, ΔP_{eye}). Additionally, the IOL type as well as the status of their geometry and position (known or unknown for the dataset) is given.

Study	**1**	**2**	**3**	**4**	**5**
Size	130	77	83	952	553
Site	Site 1	Site 2	Site 3	varying	varying
AL [mm]	$20.58 - 27.38$	$20.32 - 26.03$	$20.38 - 24.78$	$20.99 - 28.77$	$21.31 - 28.64$
ACD [mm]	$2.30 - 3.87$	$2.34 - 4.23$	$2.12 - 4.23$	$1.96 - 4.62$	$2.00 - 4.72$
P_{IOL} [D]	$10.0 - 28.5$	$13.0 - 31.0$	$18.5 - 30.0$	$6.0 - 30.0$	$6.0 - 27.0$
ΔP_{eye} [D]	$-3.25 - 1.50$	$-2.50 - 1.25$	$-2.50 - 3.00$	$-6.0 - 1.125$	$-3.125 - 1.0$
IOL type	monofocal	monofocal	monofocal	monofocal	monofocal
Geometry	known	known	known	unknown	unknown
Position	known	known	known	unknown	unknown

After the fine-tuning of the network has been performed using the loss defined in Eq. 7.21, we receive the fully trained OpticNet capable of performing a strong and physically consistent prediction of the required IOL refractive power for a given patient eye and targeted refraction defined by $P_{IOL} = \mathbf{\Phi}(\mathbf{X}, \text{Ref}_T) - \Delta$. The results achieved with the designed network OpticNet will be evaluated extensively within the next sections.

7.4 Experimental evaluation

7.4.1 Experimental setup

Description of datasets

The following experiments are designed to evaluate all components of the designed training process to create OpticNet and compare its performance against the state of the art. Further, it will examine data dependencies as well as the achieved domain coverage and physical consistency.

For the pre-training process, we simulated in total 210,000 artificial eyes. 200,000 eyes were used for the unsupervised training set, for the validation and testing set, 5000 eyes were used each. The parameter ranges defining the uniform sampling distributions from Eq. 7.16 were defined to completely cover a medically realistic range of eyes. Since for validation and testing the ground truth (according to the physical model) needs to be known, we solved Eq. 7.12 for the 10,000 samples and evaluated the performance on the test set, using the validation set for an early stopping criterion.

The fine-tuning process and actual clinical evaluation is based on five clinical datasets achieved from various different surgery sites. Every site is characterized by an individual type of monofocal IOL, produced by different companies. Three of the datasets have a consistent site

for all surgeries, the other two datasets were obtained from varying sites. For all surgeries, preoperative information in form of OCT imaging scans and cornea keratometry as well as the implanted IOL refractive power was achieved. Additionally, the postoperative refractive outcomes of the surgeries were measured to obtain the required ground truth pairs for the training of OpticNet. For three of the five datasets, we further obtained postoperative biometric information, which allows us to extract the postoperative position of the IOL, important information to create the predictor described by Eq. 7.18. Additionally, we obtained detailed information on the relation between the IOL refractive power and the corresponding IOL thickness to achieve the relation $\mathrm{LT}(R(\mathrm{P}_{\mathrm{IOL}}))$. We will therefore use these smaller but more detailed datasets for performance analysis, the larger datasets for an estimation about the required amount of data. The extraction of biometric information from the OCT imaging scans was performed via licensed medical software [41]. The data is analyzed retrospectively, allowing to evaluate the performance of both OpticNet and other state-of-the-art IOL calculation methodologies. Specific details for all used medical datasets are given in Tab. 7.1.

Training specifications

The specifications for the training process are driven by two major objectives. We want to minimize the risk of overfitting on the data and loss of generalizability and at the same time create a strong representation on the full data domain. Since the amount of data available for training is limited (see Tab. 7.1), the neural network complexity and size should be kept as large as necessary but as small as possible. After performing a hyperparameter tuning, we found that our chosen multilayer perceptron with two hidden layers and six hidden units each showed the strongest and most reliable performance. While adapting nicely to the provided data in the pre-training process, it did not result in strong overfitting during the fine-tuning on real patient cases. For both the pre-training and fine-tuning process as described in Sec. 7.3.2, the learning rate 0.1 and the weight decay 0.005 was chosen. We used a Leaky ReLU activation function with an α-factor of 0.1 for stronger gradient propagation, no dropout was applied within the network during the training process. For fine-tuning, we set λ in Eq. 7.21 to 0.4, a value resulting from an analysis described in detail in Sec. 7.4.6. The training was stopped by an early-stopping criterion on the corresponding validation data set of each fold within the cross validation.

Evaluation procedure

The performed evaluations in the following sections are based on a 10-fold cross validation. Here, the data is split into ten separate folds of equal size. In overall ten independent training runs and evaluations, every fold is used as an unseen test set once. The respective remaining 90% of the dataset are randomly split into 70% training and 30% validation data, resulting in 63% training data, 27% validation data and 10% testing data for every of the ten training runs and evaluations performed. For the resulting predictions of IOL refractive power, we will evaluate for the metrics Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), number of refractive surprises (strong deviation from target refraction) as well as performance significance (Wilcoxon signed-rank test). For every evaluation, the results are reported on the corresponding unseen test set.

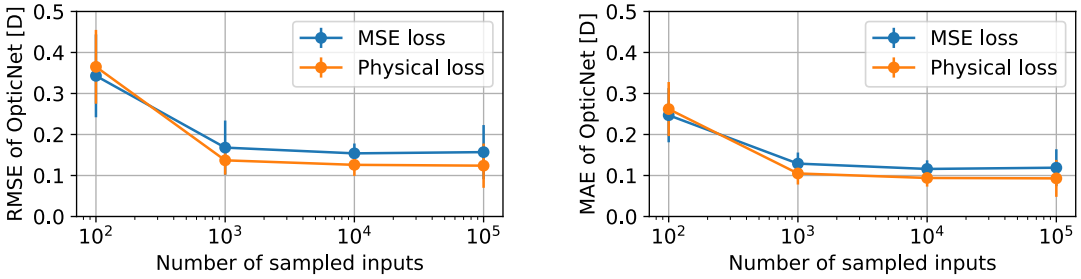


Fig. 7.11. Comparison of physical consistency of IOL refractive power prediction obtained after pre-training with a standard supervised MSE loss (blue) and the designed unsupervised physical loss (orange) for different amounts of simulated training data. The artificial data was sampled from the uniform distribution described by Eq. 7.16. For the supervised training process, a numerical calculation of the corresponding ground truth values was required, while the physical loss was directly applicable to the sampled data. The training was controlled with a validation set of 5,000 sampled independent eyes, another 5,000 eyes were used as unseen test set after training for performance reporting. The ground truth IOL refractive power for evaluation is given by the solution of the numerical solving. The performance on the test set visualized in the two plots (RMSE and MAE) clearly shows that the unsupervised physical loss provides an equal and for larger amounts of data even better performance compared to a potentially time-consuming standard supervised approach, therefore validating our developed methodology.

7.4.2 Evaluation of physical loss performance

Our designed physical loss promises an elegant, fast and unsupervised methodology to import knowledge of the eye's optical system into a neural network. It is however necessary to validate, that the developed method actually provides the same or better physical consistency compared to standard training techniques and does not result in any performance drawbacks. Additionally, we need an estimation of the required amount of simulated data for the pre-training. We therefore perform a detailed analysis of the loss functions for different amounts of simulation data.

To achieve this, we specify four different amounts of data $N = [10^2,\ 10^3,\ 10^4,\ 10^5]$ and simulate five different artificial datasets with OCT imaging-based biometric parameters X_{simu} and target refractions $Ref_{T,simu}$ each, therefore in total 20 datasets. Additionally, we are simulating 5,000 independent eyes for validation and another 5,000 unseen test eyes for performance evaluation. Since we require the actual ground truth for performance evaluation, we numerically solve Eq. 7.12 for these eyes to obtain the IOL refractive power leading to emmetropia with respect to the physical model. The closer the predictions of the network are to these ground truth values, the stronger is the physical consistency of the model. We can therefore evaluate the physical consistency as a deviation from the simulation solution in form of the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). To obtain a statistically more meaningful result, we average the results for the five datasets sampled for each data amount.

We now train two neural networks on each dataset, for one using the standard MSE loss, for the other one our designed physical loss. The target value of the training is the physically correct IOL refractive power $P_{IOL,true}$. Since the MSE is supervised, we additionally have to provide it with the numerical solution of Eq. 7.12 for every eye we sample, a potentially time-consuming process, and then train the network following Eq. 6.6. The unsupervised physical loss elegantly performs an automatic adaption of the network without this requirement,

following the optimization given in Eq. 7.15. Within Fig. 7.11, the results of the training and corresponding evaluation are reported. As a first result, we can deduce that the performance of the network saturates for an increasing amount of training data. This already gives us a good estimate of the amount of data required during pre-training to obtain a best prepared model for the fine-tuning process. For an amount of over 100,000 eyes no further performance improvement is expected. For the evaluations performed in the next sections, we choose a dataset size of 200,000 sampled artificial eyes for every pre-training for safety. Additionally, from Fig. 7.11 it becomes clear that the physical loss shows an equal and for larger amount of data even better performance compared to the MSE loss, although no supervision takes place. The physical prior knowledge therefore effectively has been transferred into the network by the explicit connection to the mathematical formulation of the optical model and its gradients. It is interesting to see that the error of the network does not approach zero for large amounts of data. Within the fine-tuning process, we are focusing on small amounts of data, therefore we limited the complexity (two layers with six hidden neurons each) and therefore adaptability of the model in order to reduce the risk of overfitting even further.

7.4.3 Performance evaluation against state of the art

After evaluating the pre-training process of OpticNet, we will now analyze the actual performance on real medical datasets. Within this evaluation, we will evaluate OpticNet in an ablation study and compare it to several IOL calculation formulas currently used within medical procedures. Since for study 1, 2 and 3 detailed information on the IOL geometry, the correlation $\mathrm{LT}(\mathrm{P_{IOL}})$ and the postoperative position $\mathrm{ACD_{IOL}}$ is given, while at the same time only small amounts of data (130, 77 and 83 patients respectively) are available for fine-tuning, we will focus on these datasets within the evaluation. Using the postoperative measurements of the IOL position, we can create individual predictors for the position using Eq. 7.18. For the pre-training process, 200,000 artificial eyes were created, the fine-tuning process on the medical data was performed with $\lambda = 0.4$ in Eq. 7.21 to minimize catastrophic forgetting of physical prior information.

As described in Sec. 7.4.1, all reported results are the outcomes on the respective test sets of a 10-fold cross-validation. For the compared IOL calculation methods, we use the latest constants provided by the IOLCon database [274] optimized for usage on the implanted IOL types. Similar to OpticNet, we further calculate an average offset to the IOL refractive power ground truth for all compared methods on the respective training set of the fold and subtract this offset from the predictions on the test set. All used methods predict the required IOL refractive power necessary for a given target refraction, therefore the comparison of the eye refractive error of the performed predictions requires the transfer of the IOL refractive error to the resulting eye refractive error. We follow the approximation of [149] given in Eq. 7.22 to receive this transfer:

$$\Delta \mathrm{P_{eye}} = -0.04 - 0.65 \cdot \Delta \mathrm{P_{IOL}} \tag{7.22}$$

Using this relation, we can convert the IOL power error $\Delta \mathrm{P_{IOL}}$ into the approximated eye refractive error after surgery $\Delta \mathrm{P_{eye}}$. Based on $\Delta \mathrm{P_{eye}}$, we report the RMSE and MAE and RS (refractive surprises, cases with refractive error larger 1 D) of all analyzed methods in Tab. 7.2. For every compared method, we additionally provide the significance level of the performance improvement of OpticNet using the Wilcoxon signed-rank test. Additionally, within Fig. 7.12

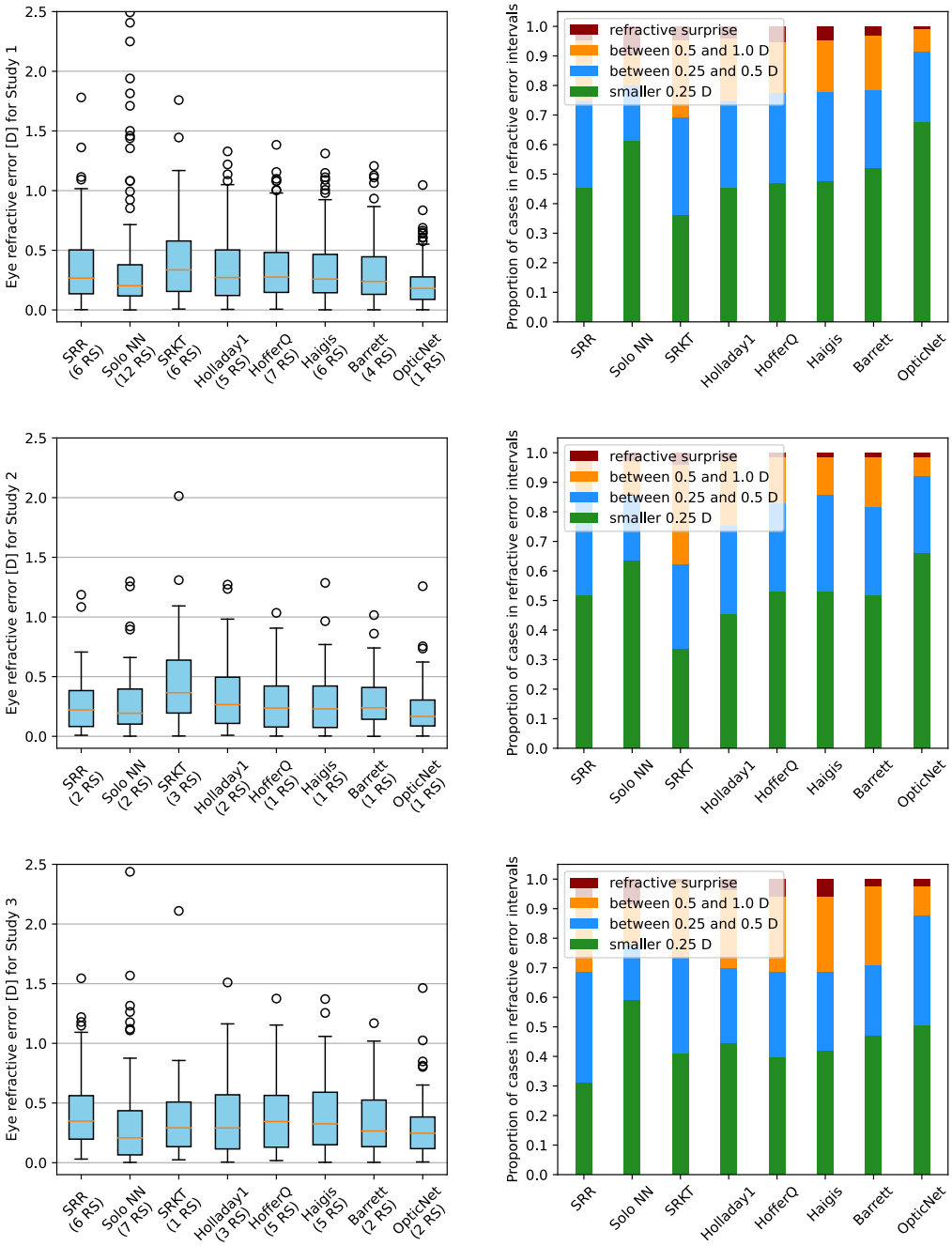


Fig. 7.12. Performance comparison against ablation study and state-of-the-art IOL calculation methods. The validation was performed on the three medical datasets from study 1, 2 and 3 with 130, 77 and 83 patients respectively (see Tab. 7.1 for details). For each dataset, the IOL type was constant within the patient population. Within the left column, a boxplot of the resulting eye refrative errors is given, in the right column a bar plot visualizes the percentage of cases which are within certain refractive error ranges. First, we compare the performance of OpticNet in an ablation study against the single-ray raytracer (SRR) defined by Eq. 7.10 and a neural network without any pre-training procedure (Solo NN). As expected, SRR is not able to adapt to the data effectively, while Solo NN shows a substantially less stable behavior, since no prior physical information was incorporated into the network, resulting in strong overfitting on the provided data especially for study 1 and 3. Then, OpticNet is compared against multiple state-of-the-art IOL calculation methods, including the popular Barrett Universal II formula. OpticNet is significantly outperforming these methods. Finally, for every formula the number of refractive surprises (RS, cases with refractive error larger 1 D) is shown.

Tab. 7.2. Root mean squared error and mean absolute error of resulting eye outcome refraction on three biometric datasets (p-value: ≤ 0.05 $*$, ≤ 0.01 $**$, ≤ 0.001 $***$). The loss weighting factor $\lambda = 0.4$ was used. The absence of significance visible for certain comparisons especially for study 2 can be explained by the very small amount of 77 patients and the variance within the results.

Data	Method	RMSE P_{eye}	MAE P_{eye}	RS P_{eye}	p-v.
St. 1	Raytracer	0.454±0.110	0.355±0.076	6	$***$
St. 1	Solo NN	0.578±0.212	0.374±0.115	12	$**$
St. 1	SRK/T	0.513±0.079	0.409±0.055	6	$***$
St. 1	Holladay1	0.438±0.097	0.347±0.073	5	$***$
St. 1	HofferQ	0.443±0.104	0.356±0.075	7	$***$
St. 1	Haigis	0.433±0.098	0.344±0.072	6	$***$
St. 1	Barrett U II	0.391±0.096	0.313±0.072	4	$***$
St. 1	**OpticNet**	**0.275±0.082**	**0.219±0.059**	**1**	/
St. 2	Raytracer	0.347±0.082	0.276±0.043	2	$**$
St. 2	Solo NN	0.354±0.120	0.270±0.076	2	
St. 2	SRK/T	0.536±0.151	0.444±0.118	3	$***$
St. 2	Holladay1	0.411±0.121	0.327±0.093	2	$**$
St. 2	HofferQ	0.372±0.082	0.284±0.063	1	
St. 2	Haigis	0.360±0.090	0.275±0.062	1	
St. 2	Barrett U II	0.354±0.098	0.294±0.089	1	$**$
St. 2	**OpticNet**	**0.279±0.114**	**0.222±0.087**	**1**	/
St. 3	Raytracer	0.501±0.147	0.417±0.128	6	$***$
St. 3	Solo NN	0.499±0.212	0.344±0.104	7	
St. 3	SRK/T	0.451±0.143	0.364±0.098	1	$**$
St. 3	Holladay1	0.471±0.113	0.376±0.092	3	$**$
St. 3	HofferQ	0.498±0.102	0.401±0.086	5	$***$
St. 3	Haigis	0.497±0.105	0.399±0.091	5	$**$
St. 3	Barrett U II	0.414±0.098	0.336±0.095	2	$*$
St. 3	**OpticNet**	**0.358±0.128**	**0.291±0.092**	2	/

the boxplots of the resulting errors are shown for every method as well as the percentage of cases within certain error ranges depicted in a bar plot. Both Tab. 7.2 and Fig. 7.12 clearly demonstrate the superiority of OpticNet compared to the other used methodologies.

Within the ablation study compared against the single-ray raytracer (SRR) and a neural network trained without any prior physical information (Solo NN) the drawbacks of both approaches alone become visible. While the SRR, which constitutes the back bone of the physical pre-training process, cannot effectively adapt to the real data provided to the system, the Solo NN is strongly overfitting on the data, especially visible for study 1 and 3. Here, large refractive surprises occur as a result of the poor generalizability of the learned representation. For medical purposes, this unstable performance is not acceptable. Only for study 2, a strong performing dataset in general, the performance of Solo NN is on an acceptable level. OpticNet, which represents the effective combination of both concepts into one approach, overcomes

the problems of both methodologies alone. It can nicely adapt to the medical data provided while relying on the rich physical domain prior information given by the designed physical loss function.

In the second step, we compare the performance of OpticNet against state-of-the-art IOL calculation methodologies. Due to proprietary interest already reported in Sec. 7.1.3, for most of the few current ML-based methods no detailed information on implementation or training is available. For some of the approaches, online masks are provided which allow the sending of medical information and return the prediction result. The usage of this option was however prohibited to us due to data sharing agreements with our medical partners and licensing aspects of the online masks. Since methods like the Hill RBF formula are based on training on large amounts of data (30,000 eyes for version 3.0), a comparison against these approaches is not in our focus. We want to demonstrate strong prediction performance for small amounts of medical data. Still, we will provide an evaluation against other machine learning approaches as well as a RBF network in the next section to come close to a comparison against Hill RBF. Nonetheless, we were able to compare against several IOL calculation methods including the state-of-the-art Barrett Universal II formula, which was further improved over the years and shows comparable performance to methods like Hill RBF [120, 195]. OpticNet significantly outperforms all compared methods and results in both smaller overall RMSE and MAE as well as lesser refractive surprises (Tab. 7.2). Within Fig. 7.12, the boxplots nicely visualize this strong performance improvement further. Additionally, for the medical context the outcome distribution with respect to the percentage of eyes in a given outcome range is of interest when comparing against other state-of-the-art methods. The right column shows the error distribution of the compared methods within certain ranges (smaller 0.25 D, between 0.25 and 0.5, between 0.5 and 1.0 D, larger 1.0 D (RS)). Also within this comparison, the overall performance boost which OpticNet provides is clearly visible, providing around 90% of cases below an error of 0.5 D for all studies.

Following from the performed analysis, we can state that OpticNet proves itself as a versatile methodology outperforming all compared methods on the three evaluated medical studies. As expected, the combination of rich physical domain prior information and nice domain adaptability given by machine learning results in a strong IOL calculation algorithm.

7.4.4 Performance evaluation against other ML approaches

Independent from the medical context, we are also interested in the performance of our developed methodology from a technical perspective. Although neural networks are powerful machine learning approaches, especially on numerical data other machine learning methods are known to perform strongly as well, namely approaches like random forests (RFs) [27] or support vector machines (SVMs) [52]. These approaches have the additional advantage that by design they are less prone to overfitting on small amounts of data. A random forest (RF) is created as a set of decision trees which are optimized towards a prediction task on an individual subset of the dataset and features. For a new unseen sample, the prediction of the RF is then performed by a voting of all learned decision trees. The random forest therefore has an integrated ensemble method, which minimizes overfitting due to the accumulated impact of different predictors. On the other hand, support vector machines are trained to provide a

decision boundary function on the training data. Normally, this boundary function is used for classification tasks. When trained for regression, the boundary is given by an ϵ-tube which provides a distance to the learned underlying function. As long as a predicted data point stays within the tube, its deviation from the function is not penalized. During the optimization of the function on the given dataset, this tube therefore minimizes the risk of overfitting, since from an optimization perspective all data points within the tube are considered equally correct. Here, the underlying function is based on a radial basis function kernel.

Additionally, there are a few other machine learning based IOL calculation approaches, one of the most known being the Hill RBF method [100], which is based on a radial basis function network (RBFN). A RBFN constitutes a neural network using radial basis functions as their activations [163]. Its output is therefore provided by Eq. 7.23:

$$\mathbf{y} = \sum_{i=0}^{K} w_i \ \phi(\mathbf{x} - \mathbf{c}_i) \ \text{ with } \ \phi(\mathbf{x}) = \exp(-\mathbf{x}^2/\sigma_i^2) \tag{7.23}$$

where K defines the number of RBF kernels used, $\mathbf{c}_i$ is the center vector of the i$^{\text{th}}$ kernel (the mean of the Gaussian distribution), σ_i is the scalar standard deviation of the kernel (equal for all dimensions) and w_i is the network weight which determines the contribution of the kernel outcome to the final prediction. We would have liked to show a comparison against the Hill RBF method, since in our opinion it would have provided further insights into the performance strength of OpticNet. Due to agreements with our medical partners as well as licensing aspects of the available Hill RBF online calculator it was however unfortunately not possible to perform this comparison. To still allow for a conceptual and methodological comparison, we have implemented an RBFN and trained it on our provided biometric patient datasets.

For our evaluation, all parameters $\mathbf{c}$, σ and w are learnable by the network. For the initialization, $\mathbf{c}$ and w are randomly sampled from a normal distribution and σ is initialized as 1.0. For the number of kernels, we have chosen $K = 50$, following a hyperparameter analysis. The parameters for the RF and SVM are chosen as follows (for all settings a hyperparameter tuning was performed):

- **RF**: number of trees: 500, maximum depth: 15, number of features per tree: 5, minimum samples to split: 2, minimum samples per leaf: 1, ratio of data per tree: 1.0

- **SVM**: C (antiprop. to regularization): 100.0, ϵ (tube width): 0.01, γ (kernel width): 0.005, kernel: rbf

In general, machine and deep learning approaches are more prone to overfitting, we will therefore compare our methodology on studies containing more than 100 samples. Study 4 and 5 contain a larger amount of data with 952 and 553 cases respectively, at the same time as described no detailed lens information was available. In consequence, OpticNet has to use default values for the IOL thickness and position prediction. Within Fig. 7.13 and Tab. 7.3, the comparison of the different methodologies is shown. As described before, a hyperparameter tuning was performed for all methods. The resulting SMV shows a rather strong performance on the task, the RF however suffers from stronger overfitting behavior. We applied different

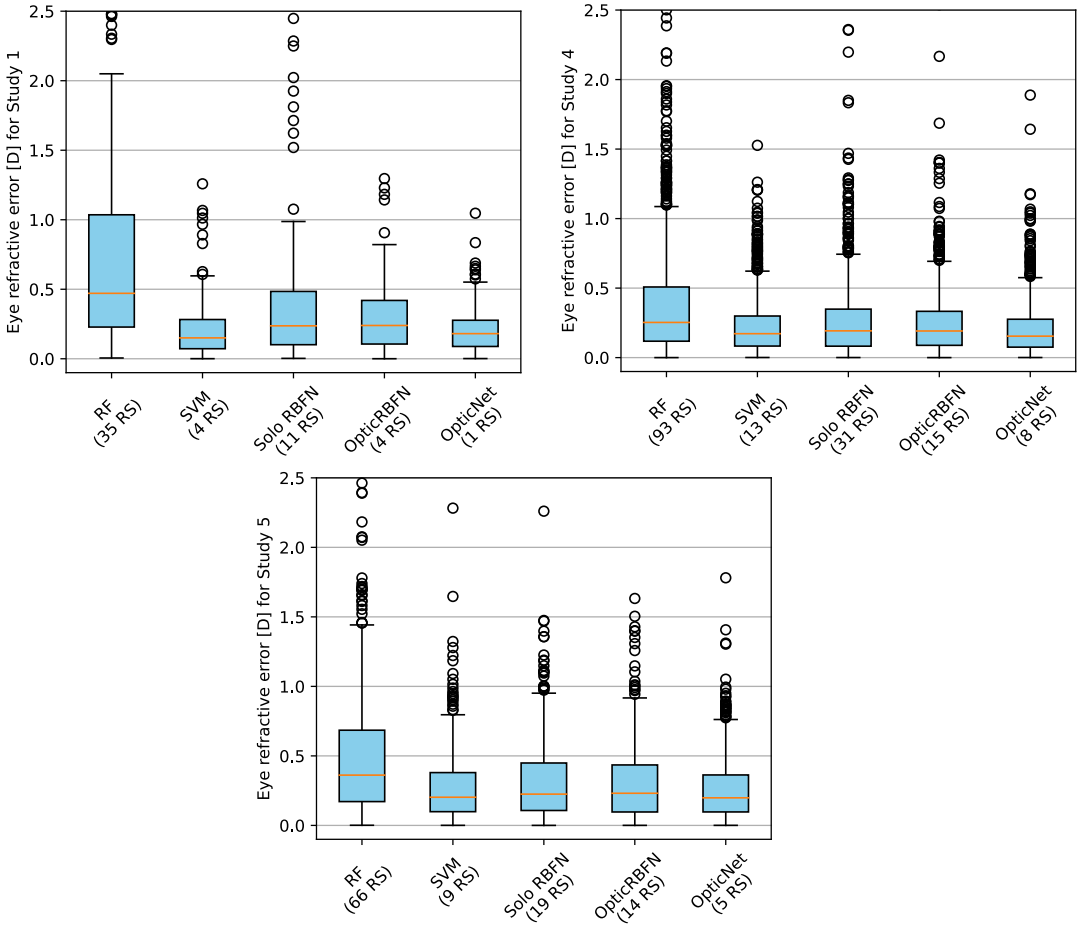


Fig. 7.13. Performance comparison against different ML approaches. A random forest (RF), support vector machine (SVM) and radial basis function network in a randomly initialized state (SoloRBFN) as well as pre-trained (OpticRBFN) are compared. The evaluation is performed on the three medical studies 1, 4 and 5 with over 100 patients cases (see Tab. 7.1 for details). For every study, a boxplot of the resulting eye refractive errors is given. While the SVM shows strong performance, OpticNet is still outperforming all other approaches, leading to the smallest number of refractive surprises on all studies. Additionally, it is nicely visible that the designed domain-specific pre-training concept is also leading to a performance improvement for the RBFN approach (see also Tab. 7.3).

measures to reduce this overfitting behavior like the reduction of maximum tree depth and number of features used per tree. These attempts reduced the performance or showed no major effect. Referring to the underlying logic of the two approaches, one explanation could be found in the working principle of the methodologies. An RF is a non-parametric method which does not learn a function to described relationships within the data, but instead hard decision boundaries. These boundaries allow for more uncontrolled behavior in poorly covered data areas, where a boundary has to be learned from a few provided samples and at the same time instead of a hard classification result a continuous regression output is expected. The approach of the SVM to learn a functional relationship, while reducing the overfitting risk via the penalty-free ϵ-tube conceptually matches better the requirements of the regression prediction task. Both approaches showed however a strong performance dependency on hyperparameter choice, for the SVM the most crucial parameter being the kernel width γ. Further, we can see that OpticNet still outperforms both methods and produces a smaller number of refractive surprises on all compared studies. Since the SVM with RBF kernels

Tab. 7.3. Root mean squared error and mean absolute error of resulting eye outcome refraction on the biometric datasets of studies 1, 4 and 5 with more than 100 patients. (p-value: ≤ 0.05 $*$, ≤ 0.01 $**$, ≤ 0.001 $***$). OpticNet as well as OpticRBFN used a loss weighting factor $\lambda = 0.4$.

Data	Method	RMSE P_{eye}	MAE P_{eye}	RS P_{eye}	p-v.
St. 1	RF	1.034±0.267	0.750±0.173	35	$***$
St. 1	SVM	0.302±0.127	0.226±0.085	4	
St. 1	RBFN	0.686±0.271	0.443±0.154	11	$***$
St. 1	OpticRBFN	0.382±0.092	0.299±0.061	4	$***$
St. 1	**OpticNet**	**0.275±0.082**	**0.219±0.059**	1	/
St. 4	RF	0.595±0.094	0.399±0.046	93	$***$
St. 4	SVM	0.333±0.076	0.234±0.030	13	$***$
St. 4	RBFN	0.484±0.171	0.290±0.037	31	$***$
St. 4	OpticRBFN	0.445±0.232	0.271±0.030	15	$***$
St. 4	**OpticNet**	**0.287±0.038**	**0.208±0.021**	8	/
St. 5	RF	0.721±0.137	0.51±0.058	66	$***$
St. 5	SVM	0.368±0.052	0.274±0.029	9	
St. 5	RBFN	0.476±0.143	0.327±0.054	19	$***$
St. 5	OpticRBFN	0.415±0.086	0.305±0.047	14	$***$
St. 5	**OpticNet**	**0.358±0.047**	**0.268±0.030**	5	/

uses local approximators, it can be assumed that performance quickly deteriorates in training domain boundary cases, since no physical understanding of the complete data domain was learned.

As described before, an RBFN is implemented and trained on the datasets to motivate a comparison against the methodological concept of the Hill RBF formula. Here, the RBFN was trained in two different scenarios. "Solo RBFN" refers to an RBFN trained on the provided data alone, while "OpticRBFN" was undergoing the same pre-training concept as OpticNet on 200,000 simulated patient eyes. Therefore, especially study 4, which provides 1,000 patients, combined with a pre-training on 200,000 simulated patients shows characteristics of a big-data approach. It is interesting to see within Fig. 7.13 that for all three provided studies the RBFN shows stronger overfitting behavior while the pre-training process of OpticRBFN consistently yields a performance improvement. This is especially interesting for study 1 with only 130 patients, see also Tab. 7.3. At the same time, the approach is inferior to our OpticNet approach. As already discussed for the SVM approach, a potential reason could be that radial basis functions as present in the RBFN correspond to local approximators and therefore quickly deteriorate in performance when a data point is provided outside an area of good domain coverage. We investigated this and indeed found a data point for which an unusal combination of minimum and maximum keratometry values was present. This input resulted in an absurd prediction error for the RBFN larger than 10D, which is not even visible in Fig. 7.13. OpticNet also created a refractive surprise for this case, but ended up with a substantially lower error

value. Its underlying fully connected layers are global approximators and therefore less prone to out-of-domain behavior. The provided example data point therefore demonstrates both the vulnerability of learning approaches on such domain boundary cases as well as the nicely increased stability of our methodological approach OpticNet. The fact that OpticNet produces the smallest amount of refractive surprises of all compared methods further corroborates this assessment. The strength of the Hill RBF algorithm stems from the large amount of training data used (more than 30,000 patient eyes for version 3.0) which we could provide only indirectly via 1,000 real and 200,000 simulated eyes using study 4. Our evaluation still allows us some insights for the comparison against this method based on these findings.

7.4.5 Analysis of domain coverage

Within Sec. 7.4.3, we have evaluated the performance of OpticNet on a general scale. To analyze further, where the performance boost of OpticNet manifests itself within the data domain, we will now show a performance analysis with respect to different input parameter areas. This is interesting since the behavior of the eye's optical system can vary depending on the parameter range. Measurement errors of e.g. the axial length as well as errors in the estimation of the postoperative IOL position and other disturbances have a larger impact on the eye's refractive error in small eyes (small axial length), since the relative differences are larger [64]. In a study conducted by [215], it was further determined that current IOL calculation formulas provide a sub-optimal performance for small eyes and very small or long anterior chamber depths. Data in this parameter area is rare [107], which influences the quality of the methodology fits for this part of the data domain. The combination of physical modeling and strong data adaption performed within OpticNet could therefore show an especially strong performance enhancement for shorter eyes compared to standard methods.

We are evaluating this hypothesis for the same datasets analyzed within Sec. 7.4.3. Both the Barrett Universal II formula and OpticNet are used for the prediction of the required IOL refractive power for given target refractions. The errors of the predictions are transferred into eye refractive errors using Eq. 7.22. We can now evaluate the difference in created refractive errors for varying parts of the data domain. The results are depicted within Fig. 7.14. Here, we have plotted the difference in eye refractive error $z = \Delta P_{\text{eye,Barrett}} - \Delta P_{\text{eye,OpticNet}}$ for every patient predicted by the Barrett Universal II formula and OpticNet against the corresponding values of axial length and anterior chamber depth of every patient. The plot therefore allows a visual demonstration of the performance differences for different parameter areas. For every patient, who has a positive z value, OpticNet provided a better prediction (smaller refractive error) compared to Barrett, for negative values it is vice versa. For a better visual interpretability, a linear plane is fitted to the calculated differences to show the overall trend of the comparison. Additionally, we added a color coding, red areas indicating the superiority of OpticNet, blue areas the superiority of the Barrett Universal II formula.

From the performed analysis in Fig. 7.14, we can first deduce that OpticNet shows a strong performance on the overall data domain (values mostly larger zero, shift to red color). Further, for study 1 and 3 we can actually see the behavior expected within our hypothesis of more accurate predictions for smaller eyes compared to standard approaches. For study 2, the smallest study, the performance is enhanced overall, the trend is not as pronounced. The

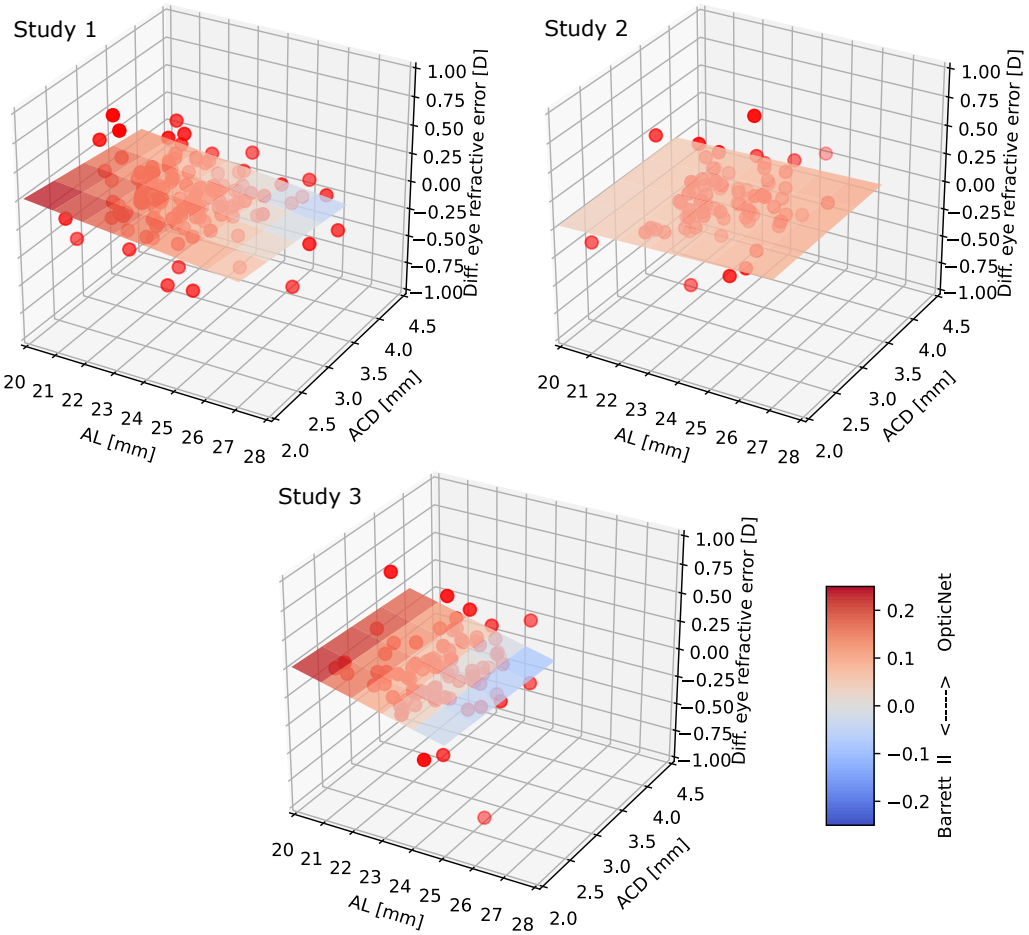


Fig. 7.14. Domain coverage for Barrett Universal II and OpticNet (trained with $\lambda = 0.4$ in Eq. 7.21). We compare the difference in eye refractive error $\Delta P_{\text{eye,Barrett}} - \Delta P_{\text{eye,OpticNet}}$ obtained for the predictions of both methods plotted against the biometric parameters axial length and anterior chamber depth. Positive values correspond to a smaller error and superiority of OpticNet, while negative values correspond to superiority of Barrett. A linear plane is fitted into the data to show the overall performance trend, color-coded for better visualization (red = superior OpticNet, blue = superior Barrett). Especially for studies 1 and 3, the prediction on short eyes (short axial length), a parameter area known for less available medical data and larger error susceptibility, shows a stronger performance of OpticNet.

results found especially for datasets 1 and 3 therefore corroborate the hypothesis that the chosen training approach of OpticNet leads to a more robust prediction performance in underrepresented parameter areas.

7.4.6 Analysis of physical consistency

During the fine-tuning process of OpticNet, the balance between the mean squared error loss (MSE) and the physical loss component is the essential criterion which controls to which extend the network will deviate from its originally learned representation to adapt to the provided real data samples. This balance is depended on the provided weighting factor λ within Eq. 7.21. For an increasing λ, the controlling influence of the physical loss gets more dominant, forcing the network to stay within its learned physical constraints during fine-tuning, but also prohibits strong adaption to the data samples. It becomes clear that an

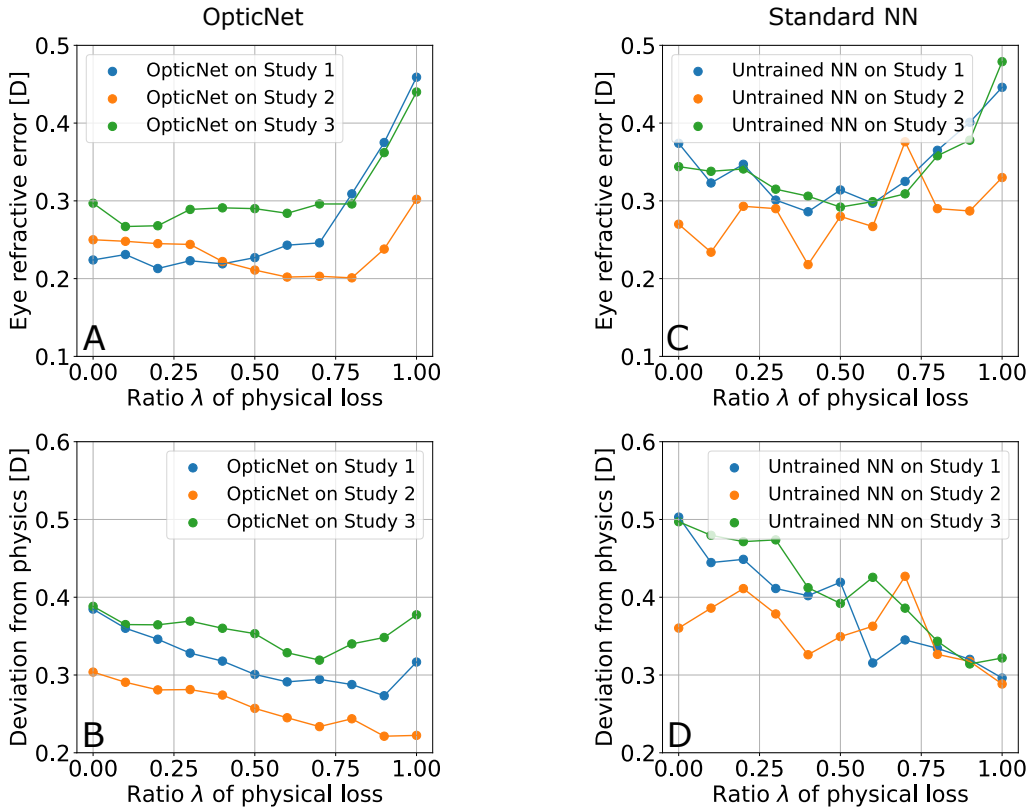


Fig. 7.15. **A and B:** Evaluation of balancing between influence of real data and physical consistency within training loss (Eq. 7.14) for OpticNet fine-tuning process. For small values of the factor $\lambda > 0$ and a corresponding increased influence of the physical loss, the performance of OpticNet gets improved since the impact of the physical loss prohibits that previously learned physical knowledge during the pre-training process is forgotten (**A**). At the same time, with an increasing λ value the physical consistency (proximity to the physically correct solution) of the results gets substantially increased (**B**). Only for $\lambda \approx 1$ the performance and the physical consistency drop substantially, which directly results from the fact that the early stopping criterion of the training is focusing on the performance improvement on real data. Therefore, by removing the impact of the MSE loss (which provides the feedback regarding the real data) within the loss function in Eq. 7.14 with $\lambda \approx 1$ the fine-tuning process becomes meaningless. **C and D:** The same evaluation is performed on a standard neural network which has not been pre-trained using the unsupervised physical loss. For an increasing λ a similar behavior as shown for OpticNet can be registered, it however becomes clearly visible that the performance of the standard NN is substantially less stable and strongly decreased compared to OpticNet.

analysis of the training behavior for different values of λ is required to evaluate the necessary balancing between the two impacts for the fine-tuning process.

This evaluation is shown within Fig. 7.15. In Fig. 7.15A, we can see the impact that a varying λ factor has on the prediction performance of OpticNet, given in resulting mean absolute refractive error. For $\lambda = 0$, the performance is already strong, however it still improves for small values of $\lambda > 0$. Especially study 2, which has the smallest number of real patient cases (77, see tab. 7.1), benefits from the influence of the physical constraint. This behavior is intuitive, since the study should be most prone to overfitting due to the small number of cases. A regulating factor which controls that the learned representation stays within a given physically realistic boundary boosts the performance of the network as a result. Also study 3 with 83 patients shows a performance improvement for a small value of lambda. The physical loss assures that the forgetting of previously learned domain knowledge is minimized. This

concept is however only effective to a certain extent, for $\lambda > 0.8$ the performance starts to deteriorate rapidly. The fine-tuning process of OpticNet gets increasingly disturbed since the impact of the MSE loss looses influence, prohibiting a proper adaption to the real data domain. It is interesting to see that for $\lambda = 1.0$, the performance collapses drastically even below the physical model performance. This directly follows from the fact that the fine-tuning process and its stopping criterion is focusing on minimizing the mean squared error of OpticNet on the validation data, directly corresponding to the minimization task of the MSE loss. If the impact of this MSE reaches 0 ($\lambda = 1.0$), the training process becomes meaningless.

Apart from model performance on real data, we are evaluating the physical consistency of the model predictions. We define this consistency as the deviation from the singe-ray raytracer predictions, given in refractive outcome deviation. The larger the deviation, the stronger the model disagrees with the underlying physical assumptions. Following our expectations, within Fig. 7.15B we clearly see that the physical consistency improves by up to 40 % for an increasing value of λ. It however does not approach 0 D (a strong alignment with the physical model) for $\lambda = 1.0$ but instead stagnates or decreases slightly in consistency. This again can be explained by the optimization process of the fine-tuning, which focuses on the MSE and becomes meaningless for negligible impact of the MSE loss. Additionally, as shown in Fig. 7.11 due to the chosen model setup the network is not approaching the physical representation perfectly.

We can use the performed analysis to also again evaluate the importance of the pre-training step of OpticNet. We would expect that both the performance as well as the physical consistency of a standard neural network (which has not been pre-trained) are decreased compared to OpticNet which already has incorporated a large prior knowledge benefit through domain-specific training. The corresponding evaluation is shown in Fig. 7.15C and 7.15D. As expected, the standard neural network does benefit from the influence of the physical loss during the training process ($\lambda > 0.0$), especially visible for study 1 and 3 by improving performance in Fig. 7.15C. Also the physical consistency gets strongly improved by up to 40 %, when the physical loss is more active (Fig. 7.15D). It however becomes visible that the overall performance of the standard neural network is very unstable, fluctuating especially for study 2 with the smallest amount of patients. Additionally, the performance is overall decreased compared to OpticNet. Here, the advantage of our designed pre-training process is clearly highlighted.

7.4.7 Analysis of data requirements

The modeling approach we designed for OpticNet allows the training of an IOL calculation model with very limited amounts of data. We want to evaluate how the performance of OpticNet varies with the respective amount of available training data. Additionally, it cannot in general be assumed that IOL geometry details are available as well as IOL postoperative position has been collected within a surgery site. Especially the measurement of the postoperative position requires a postoperative biometry, not a standard measurement practice. We therefore want to analyze the performance of OpticNet, when this detailed information is not provided. To achieve this, we perform an evaluation on the two largest studies, namely study 4 and 5. The studies only provide preoperative data, the IOL power and the refractive outcome ground

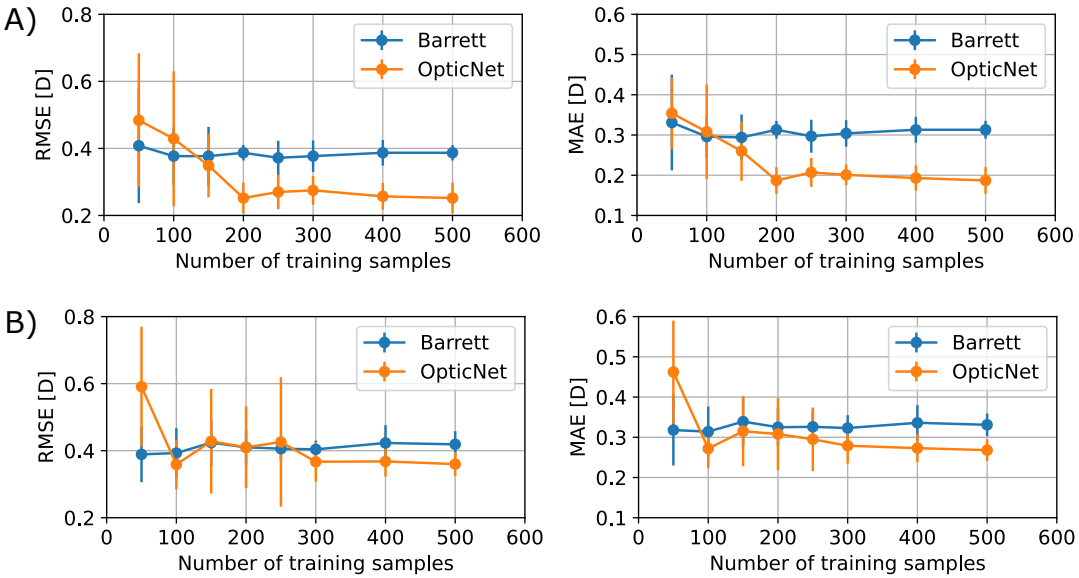


Fig. 7.16. Comparison of OpticNet performance for different amounts of available training data. The number of samples here corresponds to the complete data available for evaluation, therefore, only 63 % of the data is used for training (as described in Sec. 7.4.1), e.g. 63 training cases for 100 total cases. The evaluation is performed on the two studies 4 and 5 specified in Tab. 7.1, since they provide a sufficient amount of data for this evaluation. At the same time, the data does not provide detailed lens information about its geometry or postoperative position and the data quality is inferior compared to the other three studies in general. Before starting the evaluation, we therefore remove every data sample for which the Barrett Universal II formula shows an error $\Delta P_{IOL} > 2$ D, assuming that such a strong deviation should result from an erroneous data point. The Barrett formula therefore receives an advantage within the evaluation. The two upper plots (A) show the performance of OpticNet on study 4, it becomes visible that OpticNet has problems for very limited amounts of data, overfitting on the few provided points. With an increasing amount of training data, the model however quickly improves, clearly outperforming the Barrett Universal formula. The same behavior is also visible for the performance on study 5 (B), showing that OpticNet outperforms the Barrett formula already on small amounts of data.

truth. A further influencing factor is the collection of the data from various surgery sites, increasing the inhomogeneity of the datasets. As discussed, the lens thickness is approximated with the constant value LT $= 1$ mm. Since also no information on the postoperative position is given, the learned relation of Eq. 7.18 for the prediction of the postoperative IOL position based on study 1 was used to approximate the predictor. These approximations should overall lead to a weaker performance of the model. In correspondence, it is interesting to evaluate the required amount of data for training to receive a strong outcome given these difficulties.

In general, large medical datasets collected within various sites and from various surgeons will always contain transfer errors and erroneously reported values. A neural network based approach like OpticNet is better in adapting to such a situation, effectively learning from the deviating behavior. We try to counteract this effect by removing every data point from the dataset for which the Barrett Universal II formula resulted in an outcome error of larger than 2.0 D. This procedure will give an initial advantage to the Barrett Universal II formula, since essentially cases where it performed very poorly are automatically removed. We therefore see this as a balancing factor for the evaluation. For the experiment, we successively increase the amount of training data provided to the network and report the corresponding performance. Here, for every run we use 60 % of the reported data amount for training. Within Fig. 7.16, the results of this evaluation are depicted. Fig. 7.16 A-B clear shows that with an increasing

amount of data, OpticNet quickly starts to outperform the Barrett Universal II formula. Only for very small amounts (60 or less training samples), the training procedure cannot prevent an overfitting on the data. From these results, we can therefore expect that OpticNet would be also usable for patient cases without specific lens information. Via the machine learning based optimization, model insufficiencies are compensated, the drawbacks of both approaches alone are removed, providing a versatile IOL calculation algorithm.

7.5 Conclusion

Within this chapter, we have created a methodology for the effective inclusion of scientific prior knowledge in the form of physical functions for improved calculation of IOL refractive power in cataract surgery. This was achieved via the design of a physics-informed loss function as well as the creation of an unsupervised training process on arbitrary amounts of simulated artificial eye data. Here, the simulation process covered the full range of medically realistic parameters in a uniform way to ensure that the whole domain is properly sampled. A physically-informed constraint was applied onto the simulation process to prevent the creation of samples with medically unrealistic outcome, therefore applying a multivariate control on the simulation process. Via the physically motivated loss function, the domain covering physical model is enforced onto the neural network, a standard multilayer perceptron, using this simulated dataset. In the fine-tuning step, only a smaller amount of real data samples is required to learn the deviation of the learned physical representation from reality. During this transfer step, the learning process is still controlled by the unsupervised physical loss function to maintain physical consistency, while a balancing factor ensures that the real data samples achieve sufficient impact on the fine-tuning.

The unsupervised physical loss provides a differentiable feedback how close the prediction for a given data point is to the correct physical solution in an unsupervised fashion. Within Sec. 7.4.2, we demonstrated that this approach is as effective or even more effective than a training of a fully supervised setup using the MSE loss. Since the unsupervised loss design essentially allows a training on the physics itself, it is easily transferable as domain knowledge incorporating constraint into the fine-tuning process of the network. In the next step, we evaluated our approach on three medical datasets, which were consistent of preoperative patient data, IOL lens geometry details, IOL refractive power and patient refractive outcome groundtruth. Additionally, postoperative information about the position of the IOL after surgery was given to train a corresponding predictor. First, an ablative setup was evaluated, then a comparison against multiple IOL calculation formulas including the state-of-the-art Barrett Universal II formula was performed. The proposed methodology OpticNet was able to outperform all compared methods in IOL refractive power calculation on all datasets. Additionally, the performance against other machine learning based approaches showed the versatility of our designed approach. Here, especially the low number of created refractive surprises compared to the not pre-trained model (SoloNN) highlights the stability that was gained with the incorporation of physical prior knowledge. This was further evaluated with an analysis of the model domain coverage which corroborated the assumption that domain coverage also for underrepresented areas was achieved. Additionally, the increased physical consistency of the results as well as the required amount of fine-tuning data was evaluated. It was shown that the network outperforms competing methods for already small amounts of

fine-tuning data. Our methodology therefore has the potential to provide an IOL refractive power algorithm that is adaptable to individual surgeon requirements with small amounts of medical patient cases.

Further analysis of the methodology should focus on an evaluation within a clinical study setting on larger amounts of data. Here, a deeper analysis of model dependencies on statistically sufficient data size could be performed. One aspect for evaluation could be the just described personalization of the network to site-specific or even surgeon specific cases. Another interesting aspect would be an evaluation of the impact of target refraction planning on the prediction process, a quantity not addressed in this evaluation due to limited availability within our data. Since for a standard cataract surgery near-emmetropic target refractions are preferred, the corresponding refractive outcome could reflect this behavior. An evaluation of these relations within a larger data cohort which allows statistically meaningful findings would be very interesting. Additionally, the current version of OpticNet is only focusing on the lens type of monofocal IOLs. An extension to toric and multifocal IOLs could be very promising, since these medical cases are still more rare and specific. Incorporation of physical prior knowledge therefore could highly improve the prediction performance in this area.

Part IV

Conclusion

8 Conclusion

8.1 Overview of contributions

Within this thesis, we have highlighted a path for the effective incorporation of domain prior information into the learning process of deep neural networks. Here, we focused on the three domain prior forms of relational knowledge, logical knowledge and scientific knowledge, divided into empirical priors and model priors. We have already provided a discussion of the developed approaches for the respective chapters, we still would like to briefly summarize our contributions.

Empirical priors

For the consideration of empirical priors, we have introduced the strong learning capabilities of graph neural networks (GNNs) with respect to underlying structures within the data domain modeled as graphs. First, an extensive review of the historical development of graph neural networks as well as current state of the art of GNN approaches has been performed. In a first work, we have then developed a methodology CNNGAT for the effective incorporation of relational knowledge into an end-to-end processing of imaging data using graph convolutional networks (CGNNs), specifically graph attention networks (GAT). Here, the relational knowledge is given in the form of similarity between the meta information of different image data samples. While much work was previously performed on the processing of already extracted image features, to the best of our knowledge, we developed one of the first approaches which incorporated the feature extraction process itself into the image learning pipeline. We were able to demonstrate that the feature learning process can be adapted via the inclusion of graph information and improved towards the classification task. We motivate that this is achievable through our methodology on the one hand via the extraction of feature information only relevant in graph context and on the other hand via the inclusion of important inter-class connections. Our creation of a hybrid representation based on GAT-extracted and purely CNN-extracted features allows our methodology to leverage class-specific inter-class connections for a graph-based feature enrichment. The received new representation assists in the distinction of classes which are difficult to separate based on purely CNN-extracted features. The methodology is evaluated on an MNIST-based toy dataset as well as the NIH Chest-Xray14 dataset. CNNGAT shows significantly improved performance compared to ablated methodologies under the constraint that meaningful graph information is provided. While for the MNIST toy-dataset, this was the case, the NIH Chest-Xray14 dataset could not provide meaningful graph information. Within an occlusion shift experiment, it was still possible to demonstrate the gained stability of the prediction via the graph connections. The CNNGAT approach was pushed further via the development of our methodology U-GAT for improved outcome prediction for COVID-19 patients, where we gave special focus to a detailed analysis of the graph creation process [127].

After this effective incorporation of empirical priors in form of relational knowledge, we further advanced by including also the second form of empirical priors, namely logical knowledge in the form of expert information. This is given in the medical realm in the form of e.g. a symptom diagnosis linked to certain medical conditions. Normally, the knowledge therefore occurs in the form of if-then rules and can be described in the literature as e.g. a set of symptoms occurring when a certain disease or ailment is present. We therefore developed a methodology which is capable of encoding this logical knowledge in hard-coded and differentiable form within a neural network approach. At the same time, the usage of a graph neural network basis allows us to still consider the previously discussed relational meta information of the considered cases. Our system is demonstrated on the task of intoxication prediction, a very important medical field where fast and precise decision making is crucial and at the same time a vast amount of literature information in the form of symptom to poison matchings is present. Additionally, next to the patient symptoms a larger amount of meta information about each patient is provided, allowing for the incorporation of this relational information into the graph approach. We first performed an extensive cleaning of the provided data as well as literature information to create a medical dataset usable for our prediction task. We then were able to demonstrate that the combined inclusion of logical and relational knowledge within our methodology led to an improved performance and was able to outperform medical experts on the task of poison prediction, being very stable in easy cases (determined via low inter-rater variability) while improving against the experts in tougher cases.

Model priors

In the second part of the thesis, we focused on the second form of considered domain priors, namely scientific knowledge. As already discussed within the introduction in Sec. 1.2, we again would like to highlight that the term "scientific" does not reflect on the quality of this form of domain priors compared to the other two, the terminology is adapted from [201] and refers to the fact that an underlying and full domain covering theory like a physical equation is given for this form of domain prior knowledge. Nevertheless, also the other two knowledge forms can be based on highly scientific findings. The property of a full domain covering theoretical relation like a physical formula provides a very strong domain prior, since it essentially yields a broad prior understanding of the full data domain. We developed a methodology which can leverage this type of domain prior information in form of physical functions to full extend for the challenging task of intraocular lens (IOL) calculation for cataract surgery. Here, a precise choice of the optical properties of the implanted IOL, especially the refractive power, is necessary to prevent the patient from the requirement of visual aids like glasses after surgery. We designed a domain specific optical loss function, which encodes a paraxial ray tracing model of the human eye into a differentiable penalty term. Via the controlled simulation of a vast amount of artificial eye samples and the unsupervised training on this simulated dataset, we effectively performed a pre-training of our network on the domain prior itself, incorporating the full physical knowledge into the network. In a second step, a fine-tuning process on real data samples is performed to learn the slight deviation of our model assumptions from reality. During fine-tuning, the optical loss function is applied in weighted form to still control the outcome of the system for physical consistency. We were able to show that this training concept allows an effective training on small amounts of data and improves the physical consistency of the prediction results. At the same time, we were able to outperform given state-of-the-art IOL calculation formulas.

8.2 Important findings

During the development of our methodologies, multiple important findings became apparent we would like to highlight here.

Importance of domain prior quality

First, especially for empirical priors, which are by design data-based, the quality of this data was of high importance for the value it could add to the learning task. For the inclusion of relational knowledge in form of meta information used for the graph creation based on meta data similarity, the informativeness of the corresponding formed edges was of high importance to the system. Within the results shown in Sec. 4.5.1 and 4.5.2, it became apparent that for the absence of an informative structure any performance improvement is immediately lost. This could be e.g. clearly seen for the random graph setting in Sec. 4.5.1. By changing the meaningful graph of the MNIST toy example to a random setting, the performance significantly decreases back to the CNN baseline level. A major incentive for the design of our methodologies on empirical priors was therefore that the performance is robust to uninformative priors and remains its baseline performance. This was e.g. effectively achieved for CNNGAT via the skip connection setup, since it becomes visible that the performance does not drop below the performance of a standard CNN approach, even when completely random and therefore useless information is provided by the graph.

The same holds up for the designed inclusion of logical knowledge into the prediction task. For our chosen case of intoxication prediction, we could show that the used literature information provided a certain amount of orthogonal information to the information extractable from the standard symptom features, at the same time a prediction purely based on the literature-based rules performed rather poorly. Our designed literature matching branch was still able to strongly improve on the performance of the Naive Matching approach. This was directly resulting from the fact that the chosen hard-coding provides a literature-based initialization, allowing the network to start the learning process in a nicely setup stage. At the same time, it still has the capability to adapt itself based on the data during the learning process via the off-diagonal elements. Like this, flaws in the provided literature data can be corrected. Additionally, a parallel branch completely independent from the literature information is applied, so that the information can be combined most effectively. Looking at the purely literature based poor results of the Naive matching approach, this proved to be a good choice.

It can therefore be seen as a general guideline that the information provided by empirical priors should be seen as an additional source of information which should at the same time be incorporated into the learning process in a save and effective way, which is performance-preserving. To a certain extend this holds also true for the incorporation of model priors, in this thesis performed via the inclusion of physically motivated priors for the improvement of IOL refractive power prediction. Here, the characteristics are slightly different, since the physical model of the domain can be seen as a fundamental mathematical description. Nevertheless, real data points collected in the corresponding medical scenario can deviate from the theory due to measurement deviations not addressed by the model or simply model insufficiencies resulting from e.g. approximations. It is therefore essential that the training process allows

the freedom of deviation from the underlying model prior when required by the data domain. Again, this is addressed within our methodological setup in two ways. First, a systematic offset correction between the predictions of the physical model and the real ground truth data is calculated. This allows for a systematic deviation of the data domain from the physical prior model without the necessity for the learned representation to change, effectively narrowing the gap between the optimum of the designed physical loss function and the MSE loss. Second, a weighting between the physical loss and the MSE loss is performed. This allows for a controlled inclusion of the domain prior which is adaptable to the needs of the data domain.

Domain prior integration

A key challenge of the thesis was the proper integration of the domain prior information in its different forms into the learning process. Since neural networks are trained using a gradient-based optimization process, the essential aspect was the differentiability of the inclusion. Overall, the main ways of differentiable domain prior inclusion are given via the input of the neural network, the structure / setup of the network itself, the post-processing of the output or the inclusion of knowledge via the loss function. Interestingly, these possibilities nicely match with the definition provided by [12] for the four possible ways of the inclusion of physical information. Within our developed approaches, we have essentially addressed all of these possibilities. CNNGAT uses its structural setup, the graph neural network, for the inclusion of relational information. Additionally, the inclusion is performed as an output post-processing of the graph attention network via the designed CNN/GAT hybrid representation. Similarly, ToxNet includes both relational and logical knowledge via the design of its parallel graph and literature branches. OpticNet on the other hand includes the physical prior information via the input of vast amounts of domain covering simulated data and a physically motivated loss function. We were therefore able to evaluate several ways of differentiable domain prior incorporation and demonstrate their effectiveness.

Requirements of domain prior creation

Another important aspect are the requirements for the creation of a proper domain prior itself. This aspect is especially important with respect to the generalization of our approaches to other applications. For a medical application relying on the developed concepts, it is necessary to determine which prior information is available and how it can be transferred into usable form. A methodological concept like OpticNet requires that an underlying domain theory can be defined. There are a lot of examples where this is possible (Physical absorption concepts of X-ray imaging dependent on the structure as shown in Sec. 4.2.1, wave propagation models of ultra-sound, physical light reflection properties for skin lesions as demonstrated in [47]). The corresponding expertise needs to be involved for the incorporation process. Considering the approaches for the integration of empirical priors, requirements towards the clinical process can be formulated. To use an approach like CNNGAT on a medical imaging dataset most effectively, important parameters regarding patient demographic and clinical status need to be determined and systematically collected in order to maximize the effect of the approach. The importance of this additional information as also e.g. highlighted in [110] needs to become a matter of course in the data collection process.

Secondly, increasing the independence from the domain prior and therefore decreasing requirements for its creation could be another suitable approach. We would like to highlight

that the thesis focuses on the incorporation of domain prior information under the assumption of its existence, since the idea of the approaches is the incorporation of additional information into the training process. Still, approaches presented in e.g. Sec. 3.3.4 for the learning of the graph could be very valuable extensions to our setup. The motivation here would be less to incorporate orthogonal information into the learning process not yet present within the data, but instead extract as much information from the data as possible. The usage of feature information similarity as in [51] already goes in this direction.

8.3 Outlook

Future work motivated by this thesis can go into multiple directions. Given the large impact, which the graph structure informativeness has on prediction performance, a deeper analysis can be started into this direction. A first work tackling this challenge has already been performed by us in [127] as discussed also in the discussion section. Since the developed methodology CNNGAT and ToxNet are both easily adaptable to changes and extensions in the datasets, the exploration about better ways of graph construction and meta data incorporation could be of major interest.

A second direction is given by a further extension of our ToxNet approach. Within our work, we have shown a very promising performance on a group of 10 important toxins, where our system was able to outperform even the medical experts. To gain more relevance with respect to a clinical daily routine, the number of toxins would need to be further extended. Additionally, a major potential lies in the improvement of the literature information. So far, the Naive Matching benchmark performed with Chap. 5 clearly showed that the literature was not strongly representative of the task. Within our group, we have already started work in this direction and performed an evaluation of our approach on an extended dataset of 29 toxins representing the generally most important groups of toxins which would allow a first application of the approach in the daily routine. At the same time, our involved medical experts have performed a extensive review of the literature knowledge base and optimized the keyword matching. First results on the new pipeline and dataset look very promising. Additionally, a project towards the development of an app usable in clinical context was started.

Our developed methodology OpticNet demonstrates a way how to effectively incorporate physical domain prior information into a learning approach within the medical context of IOL refractive power calculation for cataract surgery. The results of the methodology looked very promising against current state-of-the-art approaches on several medical datasets. It would be therefore worth continuing the evaluation of the approach on larger amounts of data and within a study context to find remaining weaknesses and improve the methodology further. Steps in this direction are already very actively ongoing.

Another interesting direction to move forward is the combination of our proposed approaches. For ToxNet, this was already done to a certain extend via the combination of a graph branch incorporating relation information and a literature branch incorporating expert information. It would be possible to pre-train the system on a simulated dataset based on the now improved literature data before the fine-tuning process on real information. In the same way, relational

information for datasets of cataract patients with a richer collection of meta information could be used for the setup of a graph structure to help further boost performance.

Finally, an extension of our developed approaches to new applications would be a very interesting and clearly feasible next step. The developed methodologies are on purpose all designed to be applicable on various medical tasks. CNNGAT can be applied to any image classification problem where the image data contains informative meta information with respect to the data structure. The incorporation of literature information is not limited to toxicological problems. And physical modeling assumptions can be found as fundamental concepts of many medical procedures in clinical practice today.

Part V

Appendix

A List of Authored and Co-authored Publications

2022

[130] Marcel Kollovieh, Matthias Keicher, Stephan Wunderlich, **Hendrik Burwinkel**, Thomas Wendler, Nassir Navab, U-PET: MRI-based dementia detection with joint generation of synthetic FDG PET images, *Pre-print (2022): arXiv:2206.08078*

2021

[35] **Hendrik Burwinkel**, Holger Matz, Stefan Saur, Christoph Hauger, Michael Trost, Nino Hirnschall, Oliver Findl, Nassir Navab, Seyed-Ahmad Ahmadi, *Physics-aware learning and domain-specific loss design in ophthalmology, Medical Image Analysis, 2022, 76, 102314*

[127] Matthias Keicher*, **Hendrik Burwinkel***, David Bani-Harouni*, Magdalini Paschali, Tobias Czempiel, Egon Burian, Marcus R. Makowski, Rickmer Braren, Nassir Navab, Thomas Wendler, U-GAT: Multimodal Graph Attention Network for COVID-19 Outcome Prediction, *In revision for Medical Image Analysis (2021), pre-print available in arXiv:2108.00860* (* shared first authorship)

[234] Gerome Vivar, Anees Kazi, **Hendrik Burwinkel**, Andreas Zwergal, Nassir Navab, Seyed- Ahmad Ahmadi, Simultaneous imputation and classification using Multigraph Geometric Matrix Completion (MGMC): Application to neurodegenerative disease classification, *Artificial Intelligence in Medicine, Volume 117, 2021*

[261] Tobias Zellner, **Hendrik Burwinkel**, Matthias Keicher, David Bani-Harouni, Nassir Navab, Seyed-Ahmad Ahmadi, Florian Eyer, Decision support for toxin prediction using artificial intelligence, *41st International Congress of the European Association of Poisons Centres and Clinical Toxicologists (EAPCCT) 2021, Clinical Toxicology, 59:6, 537-602* (**Winner of the Young Investigator Award**)

[98] Dennis M. Hedderich, Matthias Keicher, Benedikt Wiestler, Martin J. Gruber, **Hendrik Burwinkel**, Florian Hinterwimmer, Tobias Czempiel, Judith E. Spiro, Daniel Pinto dos Santos, Dominik Heim, Claus Zimmer, Daniel Rückert, Jan S. Kirschke, Nassir Navab, AI for doctors - A course to educate medical professionals in artificial intelligence for medical imaging, *Healthcare (Switzerland), 2021, 9:10*

2020

[34] **Hendrik Burwinkel**, Holger Matz, Stefan Saur, Christoph Hauger, Ayşe Mine Evren, Nino Hirnschall, Oliver Findl, Nassir Navab, Seyed-Ahmad Ahmadi, Domain-Specific Loss Design for Unsupervised Physical Training: A New Approach to Modeling Medical ML Solutions, *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2020*

[33] **Hendrik Burwinkel***, Matthias Keicher*, David Bani-Harouni, Tobias Zellner, Florian Eyer, Nassir Navab, Seyed-Ahmad Ahmadi, Decision Support for Intoxication Prediction Using Graph Convolutional Networks, *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2020* (* shared first authorship)

2019

[32] **Hendrik Burwinkel**, Anees Kazi, Gerome Vivar, Shadi Albarqouni, Guillaume Zahnd, Nassir Navab, Seyed-Ahmad Ahmadi, *Adaptive Image-Feature Learning for Disease Classification Using Inductive Graph Networks, MICCAI 2019*

[126] Anees Kazi, Shayan Shekarforoush, S. Arvind Krishna, **Hendrik Burwinkel**, Gerome Vivar, Karsten Kortüm, Seyed-Ahmad Ahmadi, Shadi Albarqouni, Nassir Navab, InceptionGCN: Receptive Field Aware Graph Convolutional Network for Disease Prediction, *International Conference on Information Processing in Medical Imaging (IPMI) 2019*

[125] Anees Kazi, Shayan Shekarforoush, S. Arvind Krishna, **Hendrik Burwinkel**, Gerome Vivar, Benedikt Wiestler, Karsten Kortüm, Seyed-Ahmad Ahmadi, Shadi Albarqouni, Nassir Navab, Graph Convolution Based Attention Model for Personalized Disease Prediction, *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2019*

[233] Gerome Vivar, **Hendrik Burwinkel**, Anees Kazi, Andreas Zwergal, Nassir Navab, Seyed- Ahmad Ahmadi, Multi-modal Graph Fusion for Inductive Disease Classification in Incomplete Datasets, *Pre-print (2019): arXiv:1905.03053*

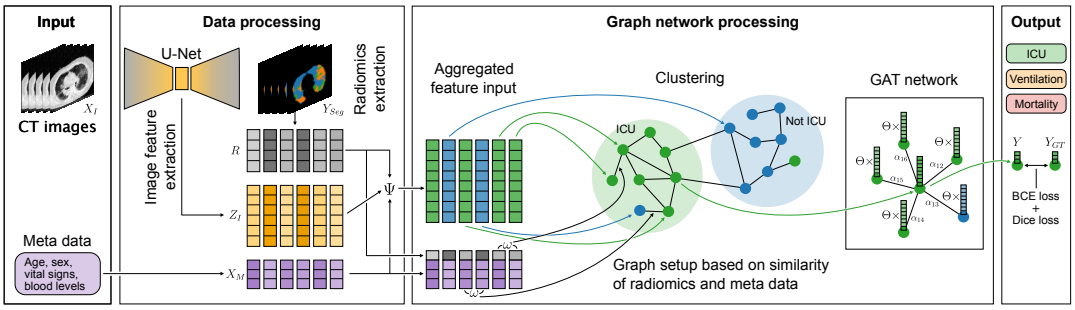
B

Abstracts of Publications not Discussed in this Thesis

U-GAT: Multimodal Graph Attention Network for COVID-19 Outcome Prediction

Matthias Keicher*, Hendrik Burwinkel*, David Bani-Harouni*, Magdalini Paschali, Tobias Czempiel, Egon Burian, Marcus R. Makowski, Rickmer Braren, Nassir Navab, Thomas Wendler

* Authors contributed equally to this work



U-GAT is an end-to-end trained model that combines image and radiomic features (ZI and R respectively) with clinical metadata XM like age, sex, vital signs and blood levels to predict the patient outcome of a disease. The segmentation YS eg of disease-affected areas in a CT image XI is used as an auxiliary task to extract the radiomic features R from the segmentation output and to regularize the extraction of the image feature ZI from the bottleneck of the U-Net [127].

During the first wave of COVID-19, hospitals were overwhelmed with the high number of admitted patients. An accurate prediction of the most likely individual disease progression can improve the planning of limited resources and finding the optimal treatment for patients. However, when dealing with a newly emerging disease such as COVID-19, the impact of patient- and disease-specific factors (e.g. body weight or known co-morbidities) on the immediate course of disease is by and large unknown. In the case of COVID-19, the need for intensive care unit (ICU) admission of pneumonia patients is often determined only by acute indicators such as vital signs (e.g. breathing rate, blood oxygen levels), whereas statistical analysis and decision support systems that integrate all of the available data could enable an earlier prognosis. To this end, we propose a holistic graph-based approach combining both imaging and non-imaging information. Specifically, we introduce a multimodal similarity metric to build a population graph for clustering patients and an image-based end-to-end Graph Attention Network to process this graph and predict the COVID-19 patient outcomes: admission to ICU, need for ventilation and mortality. Additionally, the network segments chest CT images as an auxiliary task and extracts image features and radiomics for feature

fusion with the available metadata. Results on a dataset collected in Klinikum rechts der Isar in Munich, Germany show that our approach outperforms single modality and non-graph baselines. Moreover, our clustering and graph attention allow for increased understanding of the patient relationships within the population graph and provide insight into the network's decision-making process.

In revision for Medical Image Analysis (2021), pre-print in arXiv:2108.00860

U-PET: MRI-based dementia detection with joint generation of synthetic FDG PET images

Marcel Kollovieh, Matthias Keicher, Stephan Wunderlich, Hendrik Burwinkel, Thomas Wendler, Nassir Navab

Alzheimer's disease (AD) is the most common cause of dementia. An early detection is crucial for slowing down the disease and mitigating risks related to the progression. While the combination of MRI and FDG-PET is the best image-based tool for diagnosis, FDG-PET is not always available. The reliable detection of Alzheimer's disease with only MRI could be beneficial, especially in regions where FDG-PET might not be affordable for all patients. To this end, we propose a multi-task method based on U-Net that takes T1-weighted MR images as an input to generate synthetic FDG-PET images and classifies the dementia progression of the patient into cognitive normal (CN), cognitive impairment (MCI), and AD. The attention gates used in both task heads can visualize the most relevant parts of the brain, guiding the examiner and adding interpretability. Results show the successful generation of synthetic FDG-PET images and a performance increase in disease classification over the naive single-task baseline.

Pre-print available in arXiv:2206.08078 (2022)

Simultaneous imputation and classification using Multigraph Geometric Matrix Completion (MGMC): Application to neurodegenerative disease classification

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

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 feature-complete data, which is often not the case 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 Multi-graph 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 multi-graph 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 and robust 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.

Artificial Intelligence in Medicine, Volume 117, 2021

AI for doctors - A course to educate medical professionals in artificial intelligence for medical imaging

Dennis M. Hedderich, Matthias Keicher, Benedikt Wiestler, Martin J. Gruber, Hendrik Burwinkel, Florian Hinterwimmer, Tobias Czempiel, Judith E. Spiro, Daniel Pinto dos Santos, Dominik Heim, Claus Zimmer, Daniel Rückert, Jan S. Kirschke, Nassir Navab

Successful adoption of artificial intelligence (AI) in medical imaging requires medical professionals to understand underlying principles and techniques. However, educational offerings tailored to the need of medical professionals are scarce. To fill this gap, we created the course "AI for Doctors: Medical Imaging". An analysis of participants' opinions on AI and self-perceived skills rated on a five-point Likert scale was conducted before and after the course. The participants' attitude towards AI in medical imaging was very optimistic before and after the course. However, deeper knowledge of AI and the process for validating and deploying it resulted in significantly less overoptimism with respect to perceivable patient benefits through AI ($p = 0.020$). Self-assessed skill ratings significantly improved after the course, and the appreciation of the course content was very positive. However, we observed a substantial drop-out rate, mostly attributed to the lack of time of medical professionals. There is a high demand for educational offerings regarding AI in medical imaging among medical professionals, and better education may lead to a more realistic appreciation of clinical adoption. However, time constraints imposed by a busy clinical schedule need to be taken into account for successful education of medical professionals.

Healthcare (Switzerland), 2021, 9:10, 1-9

InceptionGCN: Receptive Field Aware Graph Convolutional Network for Disease Prediction

Anees Kazi, Shayan Shekarforoush, S. Arvind Krishna, Hendrik Burwinkel, Gerome Vivar, Karsten Kortüm, Seyed-Ahmad Ahmadi, Shadi Albarqouni, Nassir Navab

Geometric deep learning provides a principled and versatile manner for integration of imaging and non-imaging modalities in the medical domain. Graph Convolutional Networks (GCNs)

in particular have been explored on a wide variety of problems such as disease prediction, segmentation, and matrix completion by leveraging large, multi-modal datasets. In this paper, we introduce a new spectral domain architecture for deep learning on graphs for disease prediction. The novelty lies in defining geometric 'inception modules' which are capable of capturing intra- and inter-graph structural heterogeneity during convolutions. We design filters with different kernel sizes to build our architecture. We show our disease prediction results on two publicly available datasets. Further, we provide insights on the behaviour of regular GCNs and our proposed model under varying input scenarios on simulated data.

International Conference on Information Processing in Medical Imaging (IPMI) 2019

Graph Convolution Based Attention Model for Personalized Disease Prediction

Anees Kazi, Shayan Shekarforoush, S. Arvind Krishna, Hendrik Burwinkel, Gerome Vivar, Benedikt Wiestler, Karsten Kortüm, Seyed-Ahmad Ahmadi, Shadi Albarqouni, Nassir Navab

Geometric deep learning provides a principled and versatile manner for integration of imaging and non-imaging modalities in the medical domain. Graph Convolutional Networks (GCNs) in particular have been explored on a wide variety of problems such as disease prediction, segmentation, and matrix completion by leveraging large, multi-modal datasets. In this paper, we introduce a new spectral domain architecture for deep learning on graphs for disease prediction. The novelty lies in defining geometric 'inception modules' which are capable of capturing intra- and inter-graph structural heterogeneity during convolutions. We design filters with different kernel sizes to build our architecture. We show our disease prediction results on two publicly available datasets. Further, we provide insights on the behaviour of regular GCNs and our proposed model under varying input scenarios on simulated data.

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

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

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

Clinical diagnostic decision making and population-based studies often rely on multi-modal data which is noisy and incomplete. Recently, several works proposed geometric deep learning approaches to solve disease classification, by modeling patients as nodes in a graph, along with graph signal processing of multi-modal features. Many of these approaches are limited by assuming modality- and feature-completeness, and by transductive inference, which requires re-training of the entire model for each new test sample. In this work, we propose a novel inductive graph-based approach that can generalize to out-of-sample patients, despite missing features from entire modalities per patient. We propose multi-modal graph fusion which

is trained end-to-end towards node-level classification. We demonstrate the fundamental working principle of this method on a simplified MNIST toy dataset. In experiments on medical data, our method outperforms single static graph approach in multi-modal disease classification.

Pre-print (2019): arXiv:1905.03053

Bibliography

[1] N. Abu-Jaradeh and D. L. Powers. “Heat conduction on graphs”. In: *Discrete Applied Mathematics* 52.1 (1994), pp. 1–16 (cit. on p. 28).

[2] A. Achiron, Z. Gur, U. Aviv, et al. “Predicting Refractive Surgery Outcome: Machine Learning Approach With Big Data”. In: *Journal of Refractive Surgery* 33.9 (2017), pp. 592–597 (cit. on p. 132).

[3] D. Allen and A. Vasavada. “Cataract and surgery for cataract”. In: *British Medical Journal* 333.7559 (2006), pp. 128–132 (cit. on pp. 123, 124).

[4] K. D. Althoff, R. Bergmann, S. Wess, et al. “Case-based reasoning for medical decision support tasks: The Inreca approach”. In: *Artificial Intelligence in Medicine* 12.1 (1998), pp. 25–41 (cit. on pp. 89–91).

[5] R. Anirudh and J. J. Thiagarajan. “Bootstrapping Graph Convolutional Neural Networks for Autism Spectrum Disorder Classification”. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2019-May (2019), pp. 3197–3201 (cit. on p. 56).

[6] S. G. Armato, G. McLennan, L. Bidaut, et al. “The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A Completed Reference Database of Lung Nodules on CT Scans”. In: *Medical Physics* 38.2 (2011), pp. 915–931 (cit. on p. 64).

[7] S. Arslan, S. I. Ktena, B. Glocker, and D. Rueckert. “Graph Saliency Maps Through Spectral Convolutional Networks: Application to Sex Classification with Brain Connectivity”. In: *Graphs in Biomedical Image Analysis and Integrating Medical Imaging and Non-Imaging Modalities*. 2018, pp. 3–13 (cit. on p. 57).

[8] M. Asada, M. Miwa, and Y. Sasaki. “Enhancing Drug-Drug Interaction Extraction from Texts by Molecular Structure Information”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2. 2014. Stroudsburg, PA, USA: Association for Computational Linguistics, 2018, pp. 680–685 (cit. on pp. 39, 58).

[9] D. A. Atchison, A. Bradley, L. N. Thibos, and G. Smith. “Useful Variations of the Badal Optometer”. In: *Optometry and Vision Science* 72.4 (1995), pp. 279–284 (cit. on p. 125).

[10] D. A. Atchison and W. N. Charman. “Thomas Young’s contribution to visual optics: The Bakerian lecture "On the mechanism of the eye"”. In: *Journal of Vision* 10.12 (2010), pp. 1–16 (cit. on p. 125).

[11] J. Atwood and D. Towsley. “Diffusion-Convolutional Neural Networks”. In: *30th Conference on Neural Information Processing Systems (NIPS 2016)* (2016) (cit. on p. 49).

[12] Y. Ba, G. Zhao, and A. Kadambi. “Blending Diverse Physical Priors with Neural Networks”. In: *arXiv* (2019), pp. 1–15. arXiv: 1910.00201 (cit. on pp. 113, 120, 168).

[13] D. Bacciu, F. Errica, and A. Micheli. “Contextual graph markov model: A deep and generative approach to graph processing”. In: *35th International Conference on Machine Learning (ICML 2018)* (2018), pp. 495–504 (cit. on p. 49).

[14] R. Bajcsy and S. Kovačič. “Multiresolution elastic matching”. In: *Computer Vision, Graphics, and Image Processing* 46.1 (1989), pp. 1–21 (cit. on p. 118).

[15] G. D. Barrett. “An improved universal theoretical formula for intraocular lens power prediction”. In: *Journal of Cataract and Refractive Surgery* 19.6 (1993), pp. 713–720 (cit. on p. 132).

[16] G. D. Barrett. “Intraocular lens calculation formulas for new intraocular lens implants”. In: *Journal of Cataract and Refractive Surgery* 13.4 (1987), pp. 389–396 (cit. on p. 132).

[17] R. T. B. Batista-Navarro, D. A. Bandojo, M. A. K. Gatapia, et al. “ESP: An expert system for poisoning diagnosis and management”. In: *Informatics for Health and Social Care* 35.2 (2010), pp. 53–63 (cit. on pp. 90, 91).

[18] P. Battaglia, R. Pascanu, M. Lai, D. Rezende, and K. Kavukcuoglu. “Interaction networks for learning about objects, relations and physics”. In: *30th Conference on Neural Information Processing Systems (NIPS 2016)* (2016), pp. 4509–4517 (cit. on pp. 45, 50).

[19] M. Belkin and P. Niyogi. “Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering”. In: *Advances in Neural Information Processing Systems 14*. The MIT Press, 2001 (cit. on p. 28).

[20] A. R. Benson, D. F. Gleich, and J. Leskovec. “Higher-order organization of complex networks”. In: *Science* 353.6295 (2016), pp. 163–166 (cit. on p. 44).

[21] R. van den Berg, T. N. Kipf, and M. Welling. “Graph Convolutional Matrix Completion”. In: *KDD’18 Deep Learning Day* (2018), pp. 1–7 (cit. on p. 47).

[22] R. D. Binkhorst. “Intraocular Lens Power Calculation”. In: *International Ophthalmology Clinics* 19.4 (1979), pp. 237–254 (cit. on p. 131).

[23] A. Bojchevski and S. Günnemann. “Certifiable robustness to graph perturbations”. In: *Advances in Neural Information Processing Systems* 32.NeurIPS (2019), pp. 1–12 (cit. on p. 44).

[24] A. Bojchevski, O. Shchur, D. Zugner, and S. Gunnemann. “NetGAN: Generating Graphs via random walks”. In: *35th International Conference on Machine Learning, ICML 2018* 2 (2018), pp. 973–988 (cit. on pp. 26, 54, 55).

[25] D. Boscaini, J. Masci, E. Rodolà, and M. M. Bronstein. “Learning shape correspondence with anisotropic convolutional neural networks”. In: *30th Conference on Neural Information Processing Systems (NIPS 2016)* (2016), pp. 1–9 (cit. on p. 51).

[26] J. T. Botticelli and P. G. Pierpaoli. “Louis Gdalman, pioneer in hospital pharmacy poison information services”. In: *American Journal of Health-System Pharmacy* 49.6 (1992), pp. 1445–1450 (cit. on p. 87).

[27] L. Breiman. “Random Forests”. In: *Machine Learning* 45 (2001), pp. 5–32 (cit. on p. 151).

[28] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. “Geometric Deep Learning: Going beyond Euclidean data”. In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42 (cit. on pp. 26, 38, 39).

[29] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. “Spectral Networks and Locally Connected Networks on Graphs”. In: *2nd International Conference on Learning Representations, ICLR 2014* (2014), pp. 1–14 (cit. on pp. 26, 28, 40, 45, 47, 48, 50, 66, 92).

[30] J. Bühren and T. Kohnen. “Anwendung der Wellenfrontanalyse in Klinik und Wissenschaft”. In: *Der Ophthalmologe* 104.10 (2007), pp. 909–925 (cit. on p. 126).

[31] S. Buoso, T. Joyce, and S. Kozerke. “Personalising left-ventricular biophysical models of the heart using parametric physics-informed neural networks”. In: *Medical Image Analysis* 71 (2021), p. 102066 (cit. on p. 121).

[32] H. Burwinkel, A. Kazi, G. Vivar, et al. “Adaptive Image-Feature Learning for Disease Classification Using Inductive Graph Networks”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. Ed. by D. Shen, T. Liu, T. M. Peters, et al. Vol. 11769. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 640–648 (cit. on pp. 26, 60, 66, 174).

[33] H. Burwinkel, M. Keicher, D. Bani-Harouni, et al. “Decision Support for Intoxication Prediction Using Graph Convolutional Networks”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. Springer International Publishing, 2020, pp. 633–642 (cit. on pp. 93, 108, 174).

[34] H. Burwinkel, H. Matz, S. Saur, et al. “Domain-Specific Loss Design for Unsupervised Physical Training: A New Approach to Modeling Medical ML Solutions”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. Vol. 12262 LNCS. 2020, pp. 540–550 (cit. on pp. 26, 125, 174).

[35] H. Burwinkel, H. Matz, S. Saur, et al. “Physics-aware learning and domain-specific loss design in ophthalmology”. In: *Medical Image Analysis* 76 (2022), p. 102314 (cit. on pp. 125, 173).

[36] C. Bycroft, C. Freeman, D. Petkova, et al. “The UK Biobank resource with deep phenotyping and genomic data”. In: *Nature* 562.7726 (2018), pp. 203–209 (cit. on p. 57).

[37] E. Çallı, E. Sogancioglu, B. van Ginneken, K. G. van Leeuwen, and K. Murphy. “Deep learning for chest X-ray analysis: A survey”. In: *Medical Image Analysis* 72 (2021), p. 102125. arXiv: 2103.08700 (cit. on p. 64).

[38] G. L. Campbell. *Phoroptors: early american instruments of refraction and those who used them*. 2008 (cit. on p. 125).

[39] E. J. Candès and B. Recht. “Exact matrix completion via convex optimization”. In: *Foundations of Computational Mathematics* 9.6 (2009), pp. 717–772 (cit. on p. 46).

[40] S. Cao, W. Lu, and Q. Xu. “Deep neural networks for learning graph representations”. In: *30th AAAI Conference on Artificial Intelligence, AAAI 2016* (2016), pp. 1145–1152 (cit. on pp. 26, 54, 55).

[41] Carl Zeiss Meditec AG. *Getting fewer refractive surprises*. 2020 (cit. on pp. 130, 146).

[42] C. H. Chao, Z. Zhu, D. Guo, et al. “Lymph Node Gross Tumor Volume Detection in Oncology Imaging via Relationship Learning Using Graph Neural Network”. In: *International Conference on Medical Image Computing and Computer Assisted Intervention, MICCAI 2020* 12267 LNCS (2020), pp. 772–782 (cit. on p. 64).

[43] Z. Chen, X. S. Wei, P. Wang, and Y. Guo. “Learning Graph Convolutional Networks for Multi-Label Recognition and Applications”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021) (cit. on p. 45).

[44] J. Cheng, L. Dong, and M. Lapata. “Long Short-Term Memory-Networks for Machine Reading”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 551–561 (cit. on p. 92).

[45] H. Choi, G. Moon, and K. M. Lee. “Pose2Mesh: Graph Convolutional Network for 3D Human Pose and Mesh Recovery from a 2D Human Pose”. In: *European Conference on Computer Vision, ECCV 2020* 12352 LNCS (2020), pp. 769–787 (cit. on p. 59).

[46] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”. In: *NIPS 2014 Workshop on Deep Learning* (2014), pp. 1–9 (cit. on p. 54).

[47] E. Claridge, S. Cotton, P. Hall, and M. Moncrieff. “From colour to tissue histology: Physics-based interpretation of images of pigmented skin lesions”. In: *Medical Image Analysis* 7.4 (2003), pp. 489–502 (cit. on pp. 118, 119, 168).

[48] G. P. Clarke and J. Burmeister. “Comparison of intraocular lens computations using a neural network versus the Holladay formula”. In: *Journal of Cataract and Refractive Surgery* 23.10 (1997), pp. 1585–1589 (cit. on p. 132).

[49] Committee on Poison Prevention and Control Board on Health Promotion and Disease Prevention. *Forging a poison prevention and control system*. 2004 (cit. on pp. 86, 87).

[50] B. J. Connell and J. X. Kane. “Comparison of the Kane formula with existing formulas for intraocular lens power selection”. In: *BMJ Open Ophthalmology* 4.1 (2019), pp. 1–6 (cit. on p. 132).

[51] L. Cosmo, A. Kazi, S.-A. Ahmadi, N. Navab, and M. Bronstein. “Latent-Graph Learning for Disease Prediction”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. 2020, pp. 643–653 (cit. on pp. 45, 169).

[52] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000 (cit. on p. 151).

[53] H. Dai, Z. L. Kozareva, B. Dai, A. J. Smola, and L. Song. “Learning steady-states of iterative algorithms over graphs”. In: *35th International Conference on Machine Learning, ICML 2018* 3 (2018), pp. 1788–1798 (cit. on p. 35).

[54] H. Dai, Y. Tian, B. Dai, S. Skiena, and L. Song. “Syntax-Directed Variational Autoencoder for Structured Data”. In: *Sixth International Conference on Learning Representations, ICLR 2018* (2018), pp. 1–17 (cit. on p. 54).

[55] S. J. Darmoni, P. Massari, J. M. Droy, et al. “SETH: an expert system for the management on acute drug poisoning in adults”. In: *Computer Methods and Programs in Biomedicine* 43.3-4 (1994), pp. 171–176 (cit. on pp. 89, 91).

[56] F. Daxecker. “Christoph Scheiner’s eye studies”. In: *History of Ophthalmology 5*. Dordrecht: Springer Netherlands, 1992, pp. 27–35 (cit. on p. 125).

[57] N. De Cao and T. Kipf. “MolGAN: An implicit generative model for small molecular graphs”. In: *ICML Workshop Theor. Found. Appl. Deep Generative Models* (2018), pp. 1–11 (cit. on pp. 26, 54, 55).

[58] M. Defferrard, X. Bresson, and P. Vandergheynst. “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering”. In: *30th Conference on Neural Information Processing Systems (NIPS 2016)* (2016), pp. 1–9 (cit. on pp. 26, 29, 41–44, 48, 50, 54, 57, 91).

[59] L. Degenhardt, F. Charlson, A. Ferrari, et al. “The global burden of disease attributable to alcohol and drug use in 195 countries and territories, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016”. In: *The Lancet Psychiatry* 5.12 (2018), pp. 987–1012 (cit. on p. 86).

[60] D. Demner-Fushman, M. D. Kohli, M. B. Rosenman, et al. “Preparing a collection of radiology examinations for distribution and retrieval”. In: *Journal of the American Medical Informatics Association* 23.2 (2016), pp. 304–310 (cit. on p. 63).

[61] W. Demtröder. *Experimentalphysik 2*. Springer-Lehrbuch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013 (cit. on pp. 128, 133–137, 139).

[62] W. Demtröder. *Experimentalphysik 3*. Springer-Lehrbuch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016 (cit. on pp. 62, 63).

[63] L. Deng. “The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142 (cit. on pp. 10, 61, 72).

[64] U. Devgan. “Pearls for cataract surgery in small eyes”. In: *Cataract and Refractive Surgery Today Europe* May (2015), pp. 48–50 (cit. on p. 155).

[65] D. Di, F. Shi, F. Yan, et al. “Hypergraph learning for identification of COVID-19 with CT imaging”. In: *Medical Image Analysis* 68.January (2021), p. 101910 (cit. on p. 59).

[66] A. Di Martino, C. G. Yan, Q. Li, et al. “The autism brain imaging data exchange: Towards a large-scale evaluation of the intrinsic brain architecture in autism”. In: *Molecular Psychiatry* 19.6 (2014), pp. 659–667 (cit. on pp. 56, 74).

[67] H. Du, J. Feng, and M. Feng. “Zoom in to where it matters: A hierarchical graph based model for mammogram analysis”. In: *arXiv* (2019). arXiv: 1912.07517 (cit. on pp. 39, 56).

[68] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, Inc., 1973 (cit. on p. 36).

[69] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, et al. “Convolutional Networks on Graphs for Learning Molecular Fingerprints”. In: *Advances in Neural Information Processing Systems 28 (NIPS 2015)* (2015), pp. 1–9 (cit. on pp. 50, 58).

[70] D. B. Elliott. “Determination of the Refractive Correction”. In: *Clinical Procedures in Primary Eye Care*. Elsevier, 2007, pp. 83–150 (cit. on pp. 126, 127).

[71] F. Feng, W. Huang, X. He, X. Xin, Q. Wang, and T. S. Chua. “Should Graph Convolution Trust Neighbors A Simple Causal Inference Method”. In: *SIGIR 2021 - Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* 1.1 (2021), pp. 1208–1218 (cit. on p. 45).

[72] M. Fey, J. E. Lenssen, F. Weichert, and H. Muller. “SplineCNN: Fast Geometric Deep Learning with Continuous B-Spline Kernels”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018), pp. 869–877 (cit. on p. 51).

[73] O. Findl, W. Struhal, G. Dorffner, and W. Drexler. “Analysis of nonlinear systems to estimate intraocular lens position after cataract surgery”. In: *Journal of Cataract and Refractive Surgery* 30.4 (2004), pp. 863–866 (cit. on p. 143).

[74] U. Forssell and P. Lindskog. “Combining Semi-Physical and Neural Network Modeling: An Example of its Usefulness”. In: *IFAC Proceedings Volumes* 30.11 (1997), pp. 767–770 (cit. on p. 118).

[75] S. N. Fyodorov, M. A. Galin, and A. Linksz. “Calculation of the optical power of intraocular lenses”. In: *Investigative Ophthalmology and Visual Science August* 14.8 (1975), pp. 625–628 (cit. on p. 131).

[76] M. L. Gabriele, G. Wollstein, H. Ishikawa, et al. “Optical coherence tomography: History, current status, and laboratory work”. In: *Investigative Ophthalmology and Visual Science* 52.5 (2011), pp. 2425–2436 (cit. on pp. 128, 129).

[77] N. Gaggion, L. Mansilla, D. H. Milone, and E. Ferrante. “Hybrid Graph Convolutional Neural Networks for Landmark-Based Anatomical Segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI 2021* 12901 LNCS (2021), pp. 600–610 (cit. on p. 59).

[78] C. Gallicchio and A. Micheli. “Graph echo state networks”. In: *Proceedings of the International Joint Conference on Neural Networks* (2010) (cit. on p. 35).

[79] H. Gao, Z. Wang, and S. Ji. “Large-scale learnable graph convolutional networks”. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2018), pp. 1416–1424 (cit. on p. 51).

[80] Z. Gao, S. Wu, Z. Liu, et al. “Learning the implicit strain reconstruction in ultrasound elastography using privileged information”. In: *Medical Image Analysis* 58 (2019), p. 101534 (cit. on p. 121).

[81] J. Gasteiger, S. Weißenberger, and S. Günnemann. “Diffusion improves graph learning”. In: *Advances in Neural Information Processing Systems* 32.NeurIPS (2019), pp. 1–13 (cit. on p. 49).

[82] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. “Neural message passing for quantum chemistry”. In: *34th International Conference on Machine Learning, ICML 2017* 3 (2017), pp. 2053–2070 (cit. on p. 50).

[83] H. V. Gimbel and T. Neuhann. “Development, advantages, and methods of the continuous circular capsulorhexis technique”. In: *Journal of Cataract and Refractive Surgery* 16.1 (1990), pp. 31–37 (cit. on p. 124).

[84] B. van Ginneken, M. B. Stegmann, and M. Loog. “Segmentation of anatomical structures in chest radiographs using supervised methods: A comparative study on a public database”. In: *Medical Image Analysis* 10.1 (2006), pp. 19–40 (cit. on p. 63).

[85] D. Goldblum. “Physics Lesson : Differences in PCI and OLCR Optical Biometry”. In: *Cataract and Refractive Surgery Today Europe* August (2015), pp. 41–43 (cit. on p. 129).

[86] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, et al. “Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules”. In: *ACS Central Science* 4.2 (2018), pp. 268–276 (cit. on pp. 26, 54).

[87] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. The MIT Press, 2017, p. 775 (cit. on pp. 3, 7–12, 15, 36, 37).

[88] K. Gopinath, C. Desrosiers, and H. Lombaert. “Adaptive Graph Convolution Pooling for Brain Surface Analysis”. In: *International Conference on Information Processing in Medical Imaging, IPMI 2019*. Vol. 11492 LNCS. Springer International Publishing, 2019, pp. 86–98 (cit. on p. 56).

[89] M. Gori, M. Maggini, and L. Sarti. “A Recursive Neural Network model for processing Directed Acyclic Graphs with labeled edges”. In: *Proceedings of the International Joint Conference on Neural Networks* 2 (2003), pp. 1351–1355 (cit. on pp. 26, 32, 33).

[90] M. Gori, G. Monfardini, and F. Scarselli. “A new model for learning in graph domains”. In: *Proceedings of the International Joint Conference on Neural Networks* 2.May 2014 (2005), pp. 729–734 (cit. on pp. 26, 32, 34, 35).

[91] T. Goswami. “Impact of Deep Learning in Image Processing and Computer Vision”. In: *Lecture Notes in Electrical Engineering* 471 (2018), pp. 475–485 (cit. on p. 3).

[92] M. Govaerts. “Poison Control in Europe”. In: *Pediatric Clinics of North America* 17.3 (1970), pp. 729–739 (cit. on p. 87).

[93] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan. “Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (2019), pp. 922–929 (cit. on p. 53).

[94] W. L. Hamilton, R. Ying, and J. Leskovec. “Inductive Representation Learning on Large Graphs”. In: *Advances in Neural Information Processing Systems 30 (NIPS 2017)* (2017), pp. 1–19 (cit. on pp. 26, 48–50, 52, 55, 58, 66, 67, 70).

[95] W. L. Hamilton, R. Ying, and J. Leskovec. “Representation Learning on Graphs: Methods and Applications”. In: *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* (2017), pp. 1–24 (cit. on p. 50).

[96] D. K. Hammond, P. Vandergheynst, and R. Gribonval. “Wavelets on graphs via spectral graph theory”. In: *Applied and Computational Harmonic Analysis* 30.2 (2011), pp. 129–150 (cit. on p. 41).

[97] L. Hansen and M. P. Heinrich. “GraphRegNet: Deep Graph Regularisation Networks on Sparse Keypoints for Dense Registration of 3D Lung CTs”. In: *IEEE Transactions on Medical Imaging* 40.9 (2021), pp. 2246–2257 (cit. on p. 59).

[98] D. M. Hedderich, M. Keicher, B. Wiestler, et al. “Ai for doctors—a course to educate medical professionals in artificial intelligence for medical imaging”. In: *Healthcare (Switzerland)* 9.10 (2021), pp. 1–9 (cit. on p. 173).

[99] M. Henaff, J. Bruna, and Y. LeCun. “Deep Convolutional Networks on Graph-Structured Data”. In: *arXiv* (2015), pp. 1–10. arXiv: 1506.05163 (cit. on p. 45).

[100] W. E. Hill. “Hill-RBF Method”. In: *Haag-Streit White Paper* (2017) (cit. on pp. 132, 152).

[101] N. Hirnschall, T. Buehren, M. Trost, and O. Findl. “Pilot evaluation of refractive prediction errors associated with a new method for ray-tracing–based intraocular lens power calculation”. In: *Journal of Cartaract and Refractive Surgery* 45.6 (2019), pp. 738–744 (cit. on p. 132).

[102] N. Hirnschall, S. Farrokhi, S. Amir-Asgari, J. Hienert, and O. Findl. “Intraoperative optical coherence tomography measurements of aphakic eyes to predict postoperative position of 2 intraocular lens designs”. In: *Journal of Cataract and Refractive Surgery* 44.11 (2018), pp. 1310–1316 (cit. on pp. 132, 143).

[103] N. Hirnschall and O. Findl. “Intraocular Lens Power Calculation – Still Searching for the Holy Grail”. In: *European Ophthalmic Review* 09.01 (2015), p. 13 (cit. on p. 131).

[104] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780 (cit. on p. 47).

[105] K. J. Hoffer. “The Hoffer Q formula: A comparison of theoretic and regression formulas”. In: *Journal of Cataract and Refractive Surgery* 19.6 (1993), pp. 700–712 (cit. on p. 132).

[106] P. C. Hoffmann, J. Wahl, W. W. Hütz, and P.-R. Preußner. “A Ray Tracing Approach to Calculate Toric Intraocular Lenses”. In: *Journal of Refractive Surgery* 29.6 (2013), pp. 402–408 (cit. on p. 132).

[107] J. T. Holladay. “Standardizing constants for ultrasonic biometry, keratometry, and intraocular lens power calculations”. In: *Journal of Cataract and Refractive Surgery* 23.9 (1997), pp. 1356–1370 (cit. on p. 155).

[108] J. T. Holladay, K. H. Musgrove, T. C. Prager, J. W. Lewis, T. Y. Chandler, and R. S. Ruiz. “A three-part system for refining intraocular lens power calculations”. In: *Journal of Cataract and Refractive Surgery* 14.1 (1988), pp. 17–24 (cit. on p. 132).

[109] R. Horaud. *A Short Tutorial on Graph Laplacians, Laplacian Embedding, and Spectral Clustering*. Tech. rep. INRIA Grenoble Rhone-Alpes, France, 2012, p. 43 (cit. on p. 28).

[110] S. C. Huang, A. Pareek, S. Seyyedi, I. Banerjee, and M. P. Lungren. “Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines”. In: *npj Digital Medicine* 3.1 (2020) (cit. on pp. 59, 64, 168).

[111] Z. Huang, X. Zhou, D. Gao, et al. “A Novel Label-Free Supervision Learning Method for Lithium-ion Battery RUL Prediction”. In: *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE, 2019, pp. 3150–3156 (cit. on p. 119).

[112] M. T. Islam, M. A. Aowal, A. T. Minhaz, and K. Ashraf. “Abnormality Detection and Localization in Chest X-Rays using Deep Convolutional Neural Networks”. In: *arXiv* (2017). arXiv: 1705.09850 (cit. on p. 63).

[113] C. R. Jack, M. A. Bernstein, N. C. Fox, et al. “The Alzheimer’s disease neuroimaging initiative (ADNI): MRI methods”. In: *Journal of Magnetic Resonance Imaging* 27.4 (2008), pp. 685–691 (cit. on pp. 56, 74).

[114] S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wáng, P.-X. Lu, and G. Thoma. “Two public chest X-ray datasets for computer-aided screening of pulmonary diseases.” In: *Quantitative imaging in medicine and surgery* 4.6 (2014), pp. 475–7 (cit. on p. 63).

[115] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. “Structural-RNN: Deep learning on spatio-temporal graphs”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem (2016), pp. 5308–5317 (cit. on pp. 26, 54).

[116] S. Ji, W. Xu, M. Yang, and K. Yu. “3D Convolutional neural networks for human action recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1 (2013), pp. 221–231 (cit. on p. 60).

[117] X. Jia, J. Willard, A. Karpatne, et al. “Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles”. In: *SIAM International Conference on Data Mining, SDM 2019* (2019), pp. 558–566 (cit. on p. 120).

[118] A. Kadambi. “Blending physics with artificial intelligence”. In: *Computational Imaging V* 11396 (2020) (cit. on p. 120).

[119] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst. “Matrix Completion on Graphs”. In: *NIPS 2014 workshop "Out of the Box: Robustness in High Dimension"* (2014), pp. 1–9 (cit. on p. 46).

[120] J. X. Kane, A. Van Heerden, A. Atik, and C. Petsoglou. “Accuracy of 3 new methods for intraocular lens power selection.” In: *Journal of cataract and refractive surgery* 43.3 (2017), pp. 333–339 (cit. on p. 151).

[121] A. Karpatne, W. Watkins, J. Read, and V. Kumar. “Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling”. In: *arXiv* (2017). arXiv: 1710.11431 (cit. on pp. 118–120).

[122] J. Kawahara, C. J. Brown, S. P. Miller, et al. “BrainNetCNN: Convolutional neural networks for brain networks; towards predicting neurodevelopment”. In: *NeuroImage* 146 (2017), pp. 1038–1049 (cit. on pp. 39, 57).

[123] A. Kazi, L. Cosmo, S.-A. Ahmadi, N. Navab, and M. Bronstein. “Differentiable Graph Module (DGM) for Graph Convolutional Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), pp. 1–1. arXiv: 20 (cit. on p. 45).

[124] A. Kazi, S. A. Krishna, S. Shekarforoush, K. Kortuem, S. Albarqouni, and N. Navab. “Self-Attention Equipped Graph Convolutions for Disease Prediction”. In: *arXiv* (2018), pp. 4–7. arXiv: 1812.09954 (cit. on p. 57).

[125] A. Kazi, S. 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 (MICCAI 2019)*. Vol. 1. 2019, pp. 122–130 (cit. on pp. 57, 174).

[126] A. Kazi, S. Shekarforoush, S. Arvind Krishna, et al. “InceptionGCN: Receptive Field Aware Graph Convolutional Network for Disease Prediction”. In: *Information Processing in Medical Imaging (IPMI 2019)* 11492 LNCS (2019), pp. 73–85 (cit. on pp. 39, 57, 61, 174).

[127] M. Keicher, H. Burwinkel, D. Bani-Harouni, et al. “U-GAT: Multimodal Graph Attention Network for COVID-19 Outcome Prediction”. In: *In revision for "Medical Image Analysis"* (2021). arXiv: 2108.00860 (cit. on pp. 67, 80, 81, 83, 165, 169, 173, 175).

[128] T. N. Kipf and M. Welling. “Semi-Supervised Classification with Graph Convolutional Networks”. In: *5th International Conference on Learning Representations, ICLR 2017*. 2017, pp. 1–14 (cit. on pp. 26, 43, 51, 55, 56, 60, 91).

[129] T. N. Kipf and M. Welling. “Variational Graph Auto-Encoders”. In: *Proc. NIPS Workshop Bayesian Deep Learn* (2016), pp. 1–3 (cit. on pp. 26, 47, 50, 54, 55).

[130] M. Kollovieh, M. Keicher, S. Wunderlich, H. Burwinkel, T. Wendler, and N. Navab. “U-PET: MRI-based Dementia Detection with Joint Generation of Synthetic FDG-PET Images”. In: *arXiv* (2022). arXiv: 2206.08078 (cit. on p. 173).

[131] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet classification with deep convolutional neural networks”. In: *Advances in Neural Information Processing Systems (NIPS 2012)* (2012), pp. 1–9 (cit. on p. 75).

[132] S. I. Ktena, S. Parisot, E. Ferrante, et al. “Distance Metric Learning Using Graph Convolutional Networks: Application to Functional Brain Networks”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2017)* 10433 LNCS (2017), pp. 469–477 (cit. on pp. 39, 57).

[133] S. I. Ktena, S. Parisot, E. Ferrante, et al. “Metric learning with spectral graph convolutions on brain connectivity networks”. In: *NeuroImage* 169 (2018), pp. 431–442 (cit. on pp. 39, 57).

[134] P. Kulling and H. Persson. “Role of the intensive care unit in the management of the poisoned patient”. In: *Medical toxicology* 1.5 (1986), pp. 375–386 (cit. on p. 88).

[135] S. Kurras. “Variants of the Graph Laplacian with Applications in Machine Learning”. PhD thesis. 2016, p. 221 (cit. on p. 28).

[136] M. J. Kusner, B. Paige, and J. M. Hemández-Lobato. “Grammar variational autoencoder”. In: *34th International Conference on Machine Learning, ICML 2017* 4 (2017), pp. 3072–3084 (cit. on pp. 26, 54).

[137] O. Larichev, A. Asanov, Y. Naryzhny, and S. Strahov. “ESTHER - expert system for the diagnostics of acute drug poisonings”. In: *Applications and Innovations in Intelligent Systems IX* (2002), pp. 159–168 (cit. on pp. 90, 91).

[138] Y. LeCun. “Generalization and Network Design Strategies”. In: *Technical Report CRG-TR-89-4, University of Toronto* (1989) (cit. on p. 37).

[139] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein. “CayleyNets: Graph Convolutional Neural Networks with Complex Rational Spectral Filters”. In: *IEEE Transactions on Signal Processing* 67.1 (2019), pp. 97–109 (cit. on pp. 42, 43).

[140] C. Lewis. “The ’Poison Squad’ and the advent of food and drug regulation”. In: *FDA Consum* 36.6 (2002) (cit. on p. 87).

[141] R. Li, S. Wang, F. Zhu, and J. Huang. “Adaptive Graph Convolutional Neural Networks”. In: *Thirty-Second AAAI Conference on Artificial Intelligence* (2018) (cit. on p. 45).

[142] X. Li, Y. Zhou, N. Dvornek, et al. “BrainGNN: Interpretable Brain Graph Neural Network for fMRI Analysis”. In: *Medical Image Analysis* 74 (2021) (cit. on p. 57).

[143] Y. Li, R. Yu, C. Shahabi, and Y. Liu. “Diffusion Convolutional Recurrent Neural Network”. In: *International Conference on Learning Representations* 1090.2014 (2018), pp. 1–15 (cit. on pp. 26, 49, 54).

[144] Y. Li, Z. Zhang, C. Dai, Q. Dong, and S. Badrigilan. “Accuracy of deep learning for automated detection of pneumonia using chest X-Ray images: A systematic review and meta-analysis”. In: *Computers in Biology and Medicine* 123.July (2020), p. 103898 (cit. on p. 63).

[145] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. “Gated Graph Sequence Neural Networks”. In: *International Conference on Learning Representations (ICLR 2016)* (2016) (cit. on pp. 35, 50, 58).

[146] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. Battaglia. “Learning Deep Generative Models of Graphs”. In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018* (2018), pp. 1–21 (cit. on p. 54).

[147] X. Liang, Y. Zhang, J. Wang, Q. Ye, Y. Liu, and J. Tong. “Diagnosis of COVID-19 Pneumonia Based on Graph Convolutional Network”. In: *Frontiers in Medicine* 7.January (2021), pp. 1–13 (cit. on pp. 59, 60).

[148] Z. Lin, M. Feng, C. N. dos Santos, et al. “A Structured Self-attentive Sentence Embedding”. In: *5th International Conference on Learning Representations, ICLR 2017* (2017) (cit. on p. 92).

[149] Y. Liu, Z. Wang, and G. Mu. “Effects of measurement errors on refractive outcomes for pseudophakic eye based on eye model”. In: *Optik* 121.15 (2010), pp. 1347–1354 (cit. on pp. 132, 148).

[150] Y. Liu, F. Zhang, Q. Zhang, S. Wang, Y. Wang, and Y. Yu. “Cross-view correspondence reasoning based on bipartite graph convolutional network for mammogram mass detection”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2020), pp. 3811–3821 (cit. on p. 64).

[151] Z. Liu, C. Chen, L. Li, et al. “GeniePath: Graph Neural Networks with Adaptive Receptive Paths”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.0 (2019), pp. 4424–4431 (cit. on p. 53).

[152] L. Ljung, J. Sjöberg, and H. Hjalmarsson. “On neural network model structures in system identification”. In: *In: Identification, Adaption, Learning (S. Bittanti and G. Picci, Eds.). Vol. 153 of Computer and Systems Sciences*. Springer, 1996, pp. 366–399 (cit. on p. 118).

[153] J. B. Long, Y. Zhang, V. Brusic, L. Chitkushev, and G. Zhang. “Antidote Application: An Educational System for Treatment of Common Toxin Overdose”. In: *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*. ACM Press, 2017, pp. 442–448 (cit. on pp. 90, 91).

[154] X. Luo, D. Zhang, and X. Zhu. “Deep learning based forecasting of photovoltaic power generation by incorporating domain knowledge”. In: *Energy* 225 (2021), p. 120240 (cit. on p. 120).

[155] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu. “Disentangled graph convolutional networks”. In: *36th International Conference on Machine Learning, ICML 2019* 2019-June (2019), pp. 7454–7463 (cit. on p. 45).

[156] J. Ma, X. Zhu, D. Yang, J. Chen, and G. Wu. “Attention-Guided Deep Graph Neural Network for Longitudinal Alzheimer’s Disease Analysis”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI 2020* 12267 LNCS (2020), pp. 387–396 (cit. on pp. 39, 56).

[157] T. Ma, J. Chen, and C. Xiao. “Constrained generation of semantically valid graphs via regularizing variational autoencoders”. In: *Advances in Neural Information Processing Systems, NeurIPS 2018* (2018), pp. 7113–7124. arXiv: 1809.02630 (cit. on pp. 54, 55).

[158] A. Marafioti, M. Hayoz, M. Gallardo, et al. “CataNet: Predicting Remaining Cataract Surgery Duration”. In: *International Conference on Medical Image Computing and Computer Assisted Intervention, MICCAI 2021* 12904 LNCS (2021), pp. 426–435 (cit. on p. 124).

[159] R. V. Marinescu, N. P. Oxtoby, A. L. Young, et al. “TADPOLE Challenge: Accurate Alzheimer’s Disease Prediction Through Crowdsourced Forecasting of Future Data”. In: *PRIME 2019: Predictive Intelligence in Medicine, Lecture Notes in Computer Science* 11843 LNCS.Icv (2019), pp. 1–10 (cit. on p. 57).

[160] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst. “Geodesic Convolutional Neural Networks on Riemannian Manifolds”. In: *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 2015, pp. 832–840 (cit. on p. 51).

[161] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. “Automating the construction of internet portals with machine learning”. In: *Information Retrieval* 3.2 (2000), pp. 127–163 (cit. on p. 55).

[162] Y. Meng, M. Wei, D. Gao, et al. “CNN-GCN Aggregation Enabled Boundary Regression for Biomedical Image Segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI 2020* 12264 LNCS (2020), pp. 352–362 (cit. on p. 58).

[163] A. Meyer-Baese and V. Schmid. “Foundations of Neural Networks”. In: *Pattern Recognition and Signal Analysis in Medical Imaging*. Elsevier, 2014, pp. 197–243 (cit. on p. 152).

[164] A. Micheli. “Neural network for graphs: A contextual constructive approach”. In: *IEEE Transactions on Neural Networks* 20.3 (2009), pp. 498–511 (cit. on p. 49).

[165] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein. “Geometric deep learning on graphs and manifolds using mixture model CNNs”. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-Janua (2017), pp. 5425–5434 (cit. on p. 51).

[166] F. Monti, M. M. Bronstein, and X. Bresson. “Geometric matrix completion with recurrent multi-graph neural networks”. In: *Advances in Neural Information Processing Systems, NIPS 2017* Nips (2017), pp. 3698–3708 (cit. on pp. 46, 47).

[167] F. Monti, K. Otness, and M. M. Bronstein. “MotifNet: A motif-based graph convolutional network for directed graphs”. In: *2018 IEEE Data Science Workshop (DSW)*. IEEE, 2018, pp. 225–228 (cit. on p. 44).

[168] M. R. Munk, T. Kurmann, P. Márquez-Neila, M. S. Zinkernagel, S. Wolf, and R. Sznitman. “Assessment of patient specific information in the wild on fundus photography and optical coherence tomography”. In: *Scientific Reports* 11.1 (2021), pp. 1–11 (cit. on p. 56).

[169] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne, and N. Ramakrishnan. “Incorporating Prior Domain Knowledge into Deep Neural Networks”. In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 36–45 (cit. on p. 120).

[170] M. Nakao, F. Tong, M. Nakamura, and T. Matsuda. “Image-to-Graph Convolutional Network for Deformable Shape Reconstruction from a Single Projection Image”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI 2021* 12904 LNCS (2021), pp. 259–268 (cit. on p. 59).

[171] National Institute of Biomedical Imaging and Bioengineering. *X-Rays*. Tech. rep. 2022, pp. 1–2 (cit. on p. 63).

[172] M. Niepert, M. Ahmed, and K. Kutzkov. “Learning Convolutional Neural Networks for Graphs”. In: *Proceedings of The 33rd International Conference on Machine Learning* 48 (2016), pp. 2014–2023 (cit. on p. 49).

[173] B. Nikolic. “Acceleration of Non-Linear Minimisation with PyTorch”. In: *arXiv* (2018), pp. 1–5. arXiv: 1805.07439 (cit. on p. 120).

[174] S. Norrby, R. Bergman, N. Hirnschall, Y. Nishi, and O. Findl. “Prediction of the true IOL position”. In: *British Journal of Ophthalmology* 101.10 (2017), pp. 1440–1446 (cit. on pp. 132, 143).

[175] T. Olsen. “Prediction of intraocular lens position after cataract extraction”. In: *Journal of Cataract and Refractive Surgery* 12.4 (1986), pp. 376–379 (cit. on p. 139).

[176] T. Olsen and P. Hoffmann. “C constant: New concept for ray tracing-assisted intraocular lens power calculation”. In: *Journal of Cataract and Refractive Surgery* 40.5 (2014), pp. 764–773 (cit. on p. 132).

[177] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. “Learning and transferring mid-level image representations using convolutional neural networks”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2014), pp. 1717–1724 (cit. on p. 115).

[178] A. Ortega, P. Frossard, J. Kovacevic, J. M. Moura, and P. Vandergheynst. “Graph Signal Processing: Overview, Challenges, and Applications”. In: *Proceedings of the IEEE* 106.5 (2018), pp. 808–828 (cit. on pp. 26–30).

[179] G. Ou, G. Yu, C. Domeniconi, X. Lu, and X. Zhang. “Multi-label zero-shot learning with graph convolutional networks”. In: *Neural Networks* 132 (2020), pp. 333–341 (cit. on p. 45).

[180] Oxford University Press. *A Dictionary of Physics*. Oxford University Press, 2009 (cit. on p. 28).

[181] N. Padoy, T. Blum, H. Feussner, M. O. Berger, and N. Navab. “On-line recognition of surgical activity for monitoring in the operating room”. In: *Proceedings of the National Conference on Artificial Intelligence* 3 (2008), pp. 1718–1724 (cit. on p. 124).

[182] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang. “Adversarially regularized graph autoencoder for graph embedding”. In: *IJCAI International Joint Conference on Artificial Intelligence* 2018-July (2018), pp. 2609–2615 (cit. on pp. 54, 55).

[183] S. J. Pan and Q. Yang. “A survey on transfer learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359 (cit. on p. 115).

[184] S. Parisot, S. I. Ktena, E. Ferrante, et al. “Disease prediction using graph convolutional networks: Application to Autism Spectrum Disorder and Alzheimer’s disease”. In: *Medical Image Analysis* 48 (2018), pp. 117–130 (cit. on pp. 44, 47, 56, 57, 61, 64, 67, 69).

[185] S. Parisot, S. I. Ktena, E. Ferrante, et al. “Spectral Graph Convolutions for Population-Based Disease Prediction”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2017)* 10435 LNCS.319456 (2017), pp. 177–185 (cit. on pp. 39, 45, 47, 56, 57, 59, 61, 64, 73–75, 91).

[186] A. Paszke, S. Gross, F. Massa, et al. “PyTorch: An imperative style, high-performance deep learning library”. In: *Advances in Neural Information Processing Systems* 32.NeurIPS (2019) (cit. on p. 120).

[187] R. Poplin, A. V. Varadarajan, K. Blumer, et al. “Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning”. In: *Nature Biomedical Engineering* 2.3 (2018), pp. 158–164 (cit. on p. 56).

[188] P.-R. Preussner, J. Wahl, H. Lahdo, B. Dick, and O. Findl. “Ray tracing for intraocular lens calculation”. In: *Journal of Cataract and Refractive Surgery* 28.8 (2002), pp. 1412–1419 (cit. on p. 132).

[189] M. Raissi, P. Perdikaris, and G. E. Karniadakis. “Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations”. In: *arXiv* (2017), pp. 1–19. arXiv: 1711.10566 (cit. on p. 119).

[190] P. Rajpurkar, J. Irvin, K. Zhu, et al. “CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning”. In: *arXiv* (2017), pp. 3–9. arXiv: 1711.05225 (cit. on pp. 62, 63).

[191] N. Rao, H.-F. Yu, P. Ravikumar, and I. S. Dhillon. “Collaborative Filtering with Graph Information: Consistency and Scalable Methods”. In: *Advances in Neural Information Processing Systems, NIPS 2015* (2015), pp. 1–9 (cit. on p. 46).

[192] M. I. Razzak, S. Naz, and A. Zaib. “Deep learning for medical image processing: Overview, challenges and the future”. In: *Lecture Notes in Computational Vision and Biomechanics* 26 (2018), pp. 323–350 (cit. on p. 3).

[193] J. A. Retzlaff, D. R. Sanders, and M. C. Kraff. “Development of the SRK/T intraocular lens implant power calculation formula”. In: *Journal of Cataract and Refractive Surgery* 16.3 (1990), pp. 333–340 (cit. on p. 132).

[194] D. Rivoir, S. Bodenstedt, F. von Bechtolsheim, M. Distler, J. Weitz, and S. Speidel. “Unsupervised Temporal Video Segmentation as an Auxiliary Task for Predicting the Remaining Surgery Duration”. In: *OR 2.0 Context-Aware Operating Theaters and Machine Learning in Clinical Neuroimaging. OR 2.0 MLCN 2019*. Vol. 11796 LNCS. 2019, pp. 29–37 (cit. on p. 124).

[195] T. V. Roberts, C. Hodge, G. Sutton, and M. Lawless. “Comparison of Hill-radial basis function, Barrett Universal and current third generation formulas for the calculation of intraocular lens power during cataract surgery”. In: *Clinical and Experimental Ophthalmology* 46.3 (2018), pp. 240–246 (cit. on p. 151).

[196] L. Robman and H. Taylor. “External factors in the development of cataract”. In: *Eye* 19.10 (2005), pp. 1074–1082 (cit. on p. 124).

[197] C. E. Rubens. “A rule-based poison control expert system”. In: *ACM SIGBIO Newsletter* 7.1 (1985), pp. 33–36 (cit. on pp. 89, 91).

[198] L. von Rueden, S. Mayer, R. Sifa, C. Bauckhage, and J. Garcke. “Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions”. In: *Advances in Intelligent Data Analysis XVIII. IDA 2020. Lecture Notes in Computer Science*. Vol. 12080 LNCS. Springer International Publishing, 2020, pp. 548–560 (cit. on p. 121).

[199] B. H. Rumack. “POISINDEX: an emergency poison management system”. In: *Drug Information Journal* 9.2-3 (1975), pp. 171–180 (cit. on pp. 89, 91).

[200] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning Internal Representations by Error Propagation”. In: *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence* V (1985), pp. 399–421 (cit. on pp. 26, 32).

[201] R. Salakhutdinov. “Integrating Domain-Knowledge into Deep Learning”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2019, pp. 3176–3176 (cit. on pp. 4, 7, 16, 17, 166).

[202] M. Salehi, R. Mohammadi, H. Ghaffari, N. Sadighi, and R. Reiazi. “Automated detection of pneumonia cases using deep transfer learning with paediatric chest X-ray images”. In: *The British Journal of Radiology* 94.1121 (2021), p. 20201263 (cit. on p. 62).

[203] D. R. Sanders and M. C. Kraff. “Improvement of intraocular lens power calculation using empirical data”. In: *American Intra-Ocular Implant Society Journal* 6.3 (1980), pp. 263–267 (cit. on p. 132).

[204] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. “The graph neural network model.” In: *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council* 20.1 (2009), pp. 61–80 (cit. on pp. 26, 35).

[205] R. G. Scherz and W. O. Robertson. “The history of poison control centers in the United states”. In: *Clinical Toxicology* 12.3 (1978), pp. 291–296 (cit. on p. 87).

[206] M. S. Schlichtkrull, N. De Cao, and I. Titov. “Interpreting graph neural networks for NLP with differentiable edge masking”. In: *International Conference on Learning Representations, ICLR 2021* 1 (2021), pp. 1–13 (cit. on p. 45).

[207] S. K. Scholtz and A. Langenbucher. “Die Berechnung des menschlichen Auges – zur Evolution der Biometrie in der Kataraktchirurgie”. In: *Klinische Monatsblätter für Augenheilkunde* 237.08 (2020), pp. 933–937 (cit. on p. 127).

[208] J. Schwiegerling. *Field guide to visual and ophthalmic optics*. SPIE Press, Bellingham, WA, 2004 (cit. on pp. 126, 127).

[209] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson. “Structured sequence modeling with graph convolutional recurrent networks”. In: *International Conference on Neural Information Processing, ICONIP 2018* 11301 LNCS (2018), pp. 362–373 (cit. on pp. 26, 54).

[210] A. A. A. Setio, A. Traverso, T. de Bel, et al. “Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: The LUNA16 challenge”. In: *Medical Image Analysis* 42 (2017), pp. 1–13 (cit. on p. 64).

[211] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. “Weisfeiler-Lehman graph kernels”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2539–2561 (cit. on p. 51).

[212] H.-C. Shin, K. Roberts, L. Lu, D. Demner-Fushman, J. Yao, and R. M. Summers. “Learning to Read Chest X-Rays: Recurrent Neural Cascade Model for Automated Image Annotation”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2016-Decem. IEEE, 2016, pp. 2497–2506 (cit. on p. 63).

[213] J. Shiraishi, S. Katsuragawa, J. Ikezoe, et al. “Development of a digital image database for chest radiographs with and without a lung nodule: Receiver operating characteristic analysis of radiologists’ detection of pulmonary nodules”. In: *American Journal of Roentgenology* 174.1 (2000), pp. 71–74 (cit. on p. 63).

[214] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. 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. 29).

[215] A. Siddiqui and U. Devgan. “Intraocular lens calculations in atypical eyes”. In: *Indian Journal of Ophthalmology* 65.12 (2017), p. 1289 (cit. on p. 155).

[216] M. Simonovsky and N. Komodakis. “GraphVAE: Towards generation of small graphs using variational autoencoders”. In: *International Conference on Artificial Neural Networks, ICANN 2018* 11139 LNCS (2018), pp. 412–422 (cit. on p. 54).

[217] R. D. Soberanis-Mukul, N. Navab, and S. Albarqouni. “Uncertainty-based graph convolutional networks for organ segmentation refinement”. In: *Medical Imaging with Deep Learning, MIDL 2019* 1 (2019), pp. 1–15 (cit. on pp. 58, 59).

[218] X. Song, H. Li, W. Gao, et al. “Augmented Multicenter Graph Convolutional Network for COVID-19 Diagnosis”. In: *IEEE Transactions on Industrial Informatics* 17.9 (2021), pp. 6499–6509 (cit. on pp. 59, 60).

[219] A. Sperduti and A. Starita. “Supervised neural networks for the classification of structures”. In: *IEEE Transactions on Neural Networks* 8.3 (1997), pp. 714–735 (cit. on pp. 26, 32).

[220] M. Sramka and A. Vlachynska. “Artificial Neural Networks Application in Intraocular Lens Power Calculation”. In: *Proceedings of The 9th EUROSIM Congress on Modelling and Simulation, EUROSIM 2016, The 57th SIMS Conference on Simulation and Modelling SIMS 2016*. Vol. 142. X. 2018, pp. 25–30 (cit. on p. 132).

[221] L. Stanković, M. Daković, and E. Sejdić. “Introduction to Graph Signal Processing”. In: *Vertex-Frequency Analysis of Graph Signals*. 2019 (cit. on pp. 25, 27, 28, 30, 31, 40).

[222] R. Stewart and S. Ermon. “Label-free supervision of neural networks with physics and domain knowledge”. In: *31st AAAI Conference on Artificial Intelligence, AAAI 2017* 1.1 (2017), pp. 2576–2582 (cit. on pp. 119, 120).

[223] V. P. Sudarshan, U. Upadhyay, G. F. Egan, Z. Chen, and S. P. Awate. “Towards lower-dose PET using physics-based uncertainty-aware multimodal learning with robustness to out-of-distribution data”. In: *Medical Image Analysis* 73 (2021), p. 102187 (cit. on p. 121).

[224] Z. Tan, J. Feng, and J. Zhou. “SGNet: Structure-Aware Graph-Based Network for Airway Semantic Segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI 2021* 12901 LNCS (2021), pp. 153–163 (cit. on p. 59).

[225] H. Tercan, A. Guajardo, J. Heinisch, T. Thiele, C. Hopmann, and T. Meisen. “Transfer-Learning: Bridging the Gap between Real and Simulation Data for Machine Learning in Injection Molding”. In: *Procedia CIRP* 72 (2018), pp. 185–190 (cit. on p. 120).

[226] Z. Tian, X. Li, Y. Zheng, et al. “Graph-convolutional-network-based interactive prostate segmentation in MR images”. In: *Medical Physics* 47.9 (2020), pp. 4164–4176 (cit. on p. 58).

[227] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin. “Accelerating Eulerian Fluid Simulation With Convolutional Networks”. In: *Proceedings of the 34th International Conference on Machine Learning* 70 (2017), pp. 3424–3433 (cit. on p. 118).

[228] D. V. Tran, N. Navarin, and A. Sperduti. “On Filter Size in Graph Convolutional Networks”. In: *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018* (2018), pp. 1534–1541 (cit. on pp. 50, 51).

[229] K. Tu, P. Cui, X. Wang, P. S. Yu, and W. Zhu. “Deep recursive network embedding with regular equivalence”. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2018), pp. 2357–2366 (cit. on pp. 54, 55).

[230] M. Turczynowska, K. Koźlik-Nowakowska, M. Gaca-Wysocka, and A. Grzybowski. “Effective Ocular Biometry and Intraocular Lens Power Calculation”. In: *European Ophthalmic Review* 10.2 (2016), pp. 94–100 (cit. on p. 132).

[231] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio. “Graph attention networks”. In: *6th International Conference on Learning Representations, ICLR 2018*. 2018, pp. 1–12 (cit. on pp. 26, 47, 49, 52, 68, 75, 92).

[232] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm. “Deep Graph Infomax”. In: *International Conference on Learning Representations, ICLR 2019* (2019), pp. 1–17 (cit. on pp. 26, 54, 55).

[233] G. Vivar, H. Burwinkel, A. Kazi, A. Zwergal, N. Navab, and S.-A. Ahmadi. “Multi-modal Graph Fusion for Inductive Disease Classification in Incomplete Datasets”. In: *arXiv* (2019), pp. 1–9. arXiv: 1905.03053 (cit. on pp. 57, 61, 174).

[234] G. Vivar, A. Kazi, H. Burwinkel, A. Zwergal, N. Navab, and S.-A. Ahmadi. “Simultaneous imputation and classification using Multigraph Geometric Matrix Completion (MGMC): Application to neurodegenerative disease classification”. In: *Artificial Intelligence in Medicine* 117 (2021), p. 102097 (cit. on pp. 46, 47, 173).

[235] G. Vivar, A. Zwergal, N. Navab, and S.-A. Ahmadi. “Multi-modal Disease Classification in Incomplete Datasets Using Geometric Matrix Completion”. In: *GRAIL Workshop on International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2018)* (2018), pp. 24–31 (cit. on p. 57).

[236] N. J. Wade. “Porterfield and Wells on the motions of our eyes”. In: *Perception* 29.2 (2000), pp. 221–239 (cit. on p. 125).

[237] D. Wang, P. Cui, and W. Zhu. “Structural deep network embedding”. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 13-17-Augu (2016), pp. 1225–1234 (cit. on p. 55).

[238] L. Wang, M. a. Booth, and D. D. Koch. “Comparison of intraocular lens power calculation methods in eyes that have undergone laser-assisted in-situ keratomileusis.” In: *Transactions of the American Ophthalmological Society* 102 (2004), 189–96; discussion 196–7 (cit. on p. 143).

[239] S.-H. Wang, V. V. Govindaraj, J. M. Górriz, X. Zhang, and Y.-D. Zhang. “Covid-19 classification by FGCNet with deep feature fusion from graph convolutional network and convolutional neural network”. In: *Information Fusion* 67 (2021), pp. 208–229 (cit. on pp. 59, 60).

[240] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. “ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3462–3471 (cit. on pp. 61–63, 72–74).

[241] Y. Wang and L. H. Staib. “Physical model-based non-rigid registration incorporating statistical shape information”. In: *Medical Image Analysis* 4.1 (2000), pp. 7–20 (cit. on p. 118).

[242] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. “Dynamic Graph CNN for Learning on Point Clouds”. In: *ACM Trans. Graph* 1.1 (2019), pp. 1–13 (cit. on p. 45).

[243] W. A. Watson, T. L. Litovitz, G. C. Rodgers, et al. “2002 annual report of the American association of poison control centers toxic exposure surveillance system”. In: *The American Journal of Emergency Medicine* 21.5 (2003), pp. 353–421 (cit. on pp. 88, 89).

[244] D. Weinshall, G. Cohen, and D. Amir. “Curriculum Learning by Transfer Learning: Theory and Experiments with Deep Networks”. In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018* 80 (2018) (cit. on p. 115).

[245] M. Wojciechowska, S. Malacrino, N. Garcia Martin, H. Fehri, and J. Rittscher. “Early Detection of Liver Fibrosis Using Graph Convolutional Networks”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI 2021*. Vol. 12908 LNCS. Springer International Publishing, 2021, pp. 217–226 (cit. on p. 56).

[246] J. M. Wolterink, T. Leiner, and I. Išgum. “Graph Convolutional Networks for Coronary Artery Segmentation in Cardiac CT Angiography”. In: *MICCAI 2019 Workshop on Graph Learning in Medical Image (GLMI)*. Lecture Notes in Computer Science 11849 (2019). Ed. by D. Zhang, L. Zhou, B. Jie, and M. Liu, pp. 62–69 (cit. on p. 58).

[247] World Health Organization. “Standardization of interpretation of chest radiographs for the diagnosis of pneumonia in children”. In: *World Health Organization Pneumonia Vaccine Trial Investigators' Group* (2001) (cit. on p. 62).

[248] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. “A Comprehensive Survey on Graph Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (2021), pp. 4–24 (cit. on pp. 24–26, 35, 38, 49–51, 53–55).

[249] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang. “Graph wavenet for deep spatial-temporal graph modeling”. In: *IJCAI International Joint Conference on Artificial Intelligence* (2019), pp. 1907–1913 (cit. on p. 54).

[250] X. Xie, J. Niu, X. Liu, Z. Chen, S. Tang, and S. Yu. “A survey on incorporating domain knowledge into deep learning for medical image analysis”. In: *Medical Image Analysis* 69 (2021), p. 101985 (cit. on pp. 3, 7, 19, 64, 121).

[251] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. “How Powerful are Graph Neural Networks?” In: *Int. Conf. on Learning Representations* (2019), pp. 1–17 (cit. on p. 51).

[252] J. Yan, Y. Chen, S. Yang, et al. “Multi-head GAGNN: A Multi-head Guided Attention Graph Neural Network for Modeling Spatio-temporal Patterns of Holistic Brain Functional Networks”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI 2021* 12907 LNCS (2021), pp. 564–573 (cit. on p. 58).

[253] S. Yan, Y. Xiong, and D. Lin. “Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition”. In: *32nd AAAI Conference on Artificial Intelligence*. AAAI press, 2018, pp. 7444–7452 (cit. on p. 54).

[254] D. Yao, J. Sui, M. Wang, et al. “A Mutual Multi-Scale Triplet Graph Convolutional Network for Classification of Brain Disorders Using Functional or Structural Connectivity”. In: *IEEE Transactions on Medical Imaging* 40.4 (2021), pp. 1279–1289 (cit. on p. 57).

[255] K.-L. Yao and W.-J. Li. “Convolutional Geometric Matrix Completion”. In: *arXiv* (2018). arXiv: 1803.00754 (cit. on p. 47).

[256] M. Yarmahmoodi, H. Arabalibeik, M. Mokhtaran, and A. Shojaei. “Intraocular Lens Power Formula Selection Using Support Vector Machines”. In: *Frontiers in Biomedical Technologies* 2.1 (2015), pp. 36–44 (cit. on p. 132).

[257] J. You, R. Ying, X. Ren, W. L. Hamilton, and J. Leskovec. “GraphRNN: Generating realistic graphs with deep auto-regressive models”. In: *35th International Conference on Machine Learning, ICML 2018* 13 (2018), pp. 9072–9081 (cit. on p. 54).

[258] B. Yu, H. Yin, and Z. Zhu. “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting”. In: *IJCAI International Joint Conference on Artificial Intelligence* 2018-July (2018), pp. 3634–3640 (cit. on pp. 26, 54).

[259] T. Yu, E. J. Canales-Rodríguez, M. Pizzolato, et al. “Model-informed machine learning for multi-component T2 relaxometry”. In: *Medical Image Analysis* 69 (2021), p. 101940 (cit. on pp. 59, 121).

[260] W. Yu, C. Zheng, W. Cheng, et al. “Learning Deep Network Representations with Adversarially Regularized Autoencoders”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2018, pp. 2663–2671 (cit. on pp. 26, 54, 55).

[261] T. Zellner, H. Burwinkel, M. Keicher, et al. “Decision support for toxin prediction using artificial intelligence”. In: *41st International Congress of the European Association of Poisons Centres and Clinical Toxicologists (EAPCCT) 2021, Clinical Toxicology* 59.6 (2021), p. 541 (cit. on p. 173).

[262] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D. Y. Yeung. “GaAN: Gated attention networks for learning on large and spatiotemporal graphs”. In: *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018* 1 (2018), pp. 339–349 (cit. on pp. 53, 54).

[263] J. Zhang. “Basic Neural Units of the Brain: Neurons, Synapses and Action Potential”. In: *arXiv* (2019), pp. 1–38. arXiv: 1906.01703 (cit. on p. 7).

[264] Y. Zhang, X. Wang, Z. Xu, Q. Yu, A. Yuille, and D. Xu. “When radiology report generation meets knowledge graph”. In: *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence* (2020), pp. 12910–12917 (cit. on p. 64).

[265] Z. Zhang and Z. Liang. “Representation Learning on Networks”. In: *35th Conference on Neural Information Processing Systems (NeurIPS 2021)* (2021), pp. 1–16 (cit. on p. 51).

[266] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. “Learning Deep Features for Discriminative Localization”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 2921–2929 (cit. on pp. 57, 63).

[267] J. Zhu, E. Zhang, and K. Rio-Tsonis. “Eye Anatomy”. In: *eLS*. Wiley, 2012 (cit. on p. 123).

[268] C. Zhuang and Q. Ma. “Dual graph convolutional networks for graph-based semi-supervised classification”. In: *The Web Conference 2018 - Proceedings of the World Wide Web Conference, WWW 2018* 3 (2018), pp. 499–508 (cit. on p. 44).

[269] D. Zügner, A. Akbarnejad, and S. Günnemann. “Adversarial Attacks on Neural Networks for Graph Data”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2018), pp. 2847–2856 (cit. on pp. 44, 47, 56, 69).

Online Resources

[270] Arcscan Inc. *Arcscan*. 2022. URL: https://www.arcscan.com/ (visited on June 21, 2022) (cit. on p. 127).

[271] Centers for Disease Control and Prevention (CDC). *Leading Causes of Death*. 2022. URL: https://www.cdc.gov/nchs/fastats/leading-causes-of-death.htm (visited on June 21, 2022) (cit. on p. 62).

[272] Johnson Space Center (NASA). *CLIPS: A Tool for Building Expert Systems*. 1996. URL: https://www.clipsrules.net/ (visited on June 21, 2022) (cit. on p. 90).

[273] NIH. *Open-i: An open access biomedical search engine.* 2022. URL: https://openi.nlm.nih.gov (visited on June 21, 2022) (cit. on p. 63).

[274] Steinbeis Transfer Center - Vision Research. *IOL Con*. 2022. URL: https://iolcon.org/ (visited on June 21, 2022) (cit. on p. 148).

[275] U.S. Food and Drug Administration (FDA). *Medical X-ray Imaging*. 2022. URL: https://www.fda.gov/radiation-emitting-products/medical-imaging/medical-x-ray-imaging (visited on June 21, 2022) (cit. on p. 63).

List of Figures

List of Tables