TU WIEN Informatics

Self-Explaining Transformers for Cell Population Detection in Flow Cytometry Data

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Florian Kowarsch, Bsc.
Matrikelnummer 11777780

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Dr. tech. Michael Reiter
Mitwirkung: DI Lisa Weijler
Dr. tech. Florian Kleber

Wien, 26. April 2023

Florian Kowarsch | Michael Reiter

TU WIEN Informatics

Self-Explaining Transformers for Cell Population Detection in Flow Cytometry Data

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Florian Kowarsch, Bsc.
Registration Number 11777780

to the Faculty of Informatics

at the TU Wien

Advisor: Dr. tech. Michael Reiter
Assistance: DI Lisa Weijler
Dr. tech. Florian Kleber

Vienna, 26th April, 2023

Florian Kowarsch | Michael Reiter

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Florian Kowarsch, Bsc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 26. April 2023

Florian Kowarsch

Acknowledgements

I want to thank my mother for her dedication in my upbringing, which ignited my curiosity to explore new things.

I want to thank my father for cultivating and sharpening my mind in logical reasoning and critical thinking.

I want to thank Andreas Benisch, my high-school teacher, who sparked my interest in computer science and set me on this path.

I want to thank Roxane Licandro, my bachelor thesis advisor, for her dedication to training me in the craft of scientific writing and research. She also proposed me as a researcher at the Computer Vision Lab, for which I will always be thankful.

I want to thank Florian Kleber for his encouragement to write this thesis and for providing the structure and direction that has made it possible.

I want to thank Matthias Wödlinger and Lisa Weijler for their tireless support, insights, and discussions that helped me shape my ideas and refine my work as well as for the friendly team spirit they have created.

Finally, I want to thank my advisor Michael Reiter for his valuable feedback, his willingness to delve me into mathematical concepts in great detail, and for providing me with the opportunity to embark on this scientific journey as part of the MyeFlow project at the Computer Vision Lab.

Kurzfassung

Entscheidungen von automatisierten Systemen im Gesundheitswesen können weitreichende Folgen haben, wie z.B. eine verzögerte oder falsche Behandlung, und müssen daher für medizinische Experten erklärbar und nachvollziehbar sein. Dies gilt auch für den Bereich der automatisierten Flow Cytometry (FCM) Datenanalyse. Bei der Leukämietherapie werden FCM-Proben aus dem Knochenmark des Patienten gewonnen, um die Anzahl der verbleibenden Leukämiezellen zu bestimmen. In einem manuellem Prozess, dem so genannten Gating, zeichnen medizinische Experten mehrere Polygone um verschiedene Zellpopulationen in 2D-Projektionen, um eine Krebszellpopulation in einer FCM-Probe aufzuspüren. Es gibt mehrere Ansätze, die darauf abzielen, diese Aufgabe zu automatisieren. State-of-the-art Modelle zur automatischen zellweisen Klassifizierung agieren in ihrer Vorhersage als Black-Boxen und haben nicht die Erklärbarkeit von durch den Menschen erstellten Gating Hierarchien. In dieser Arbeit wird ein neuartiger transformer-basierter Ansatz vorgestellt, der Zellen in FCM-Daten klassifiziert, indem er den Entscheidungsprozess von medizinischen Experten nachahmt. Das Netzwerk berücksichtigt alle Messereignisse einer Probe auf einmal und sagt die entsprechenden Polygone der Gating-Hierarchie voraus, wodurch eine nachvollziehbare Visualisierung ensteht, ähnlich wie es ein menschlicher Operator tut. Das vorgeschlagene Modell wurde an drei öffentlich zugänglichen Datensätzen für akute lymphoblastische Leukämie (ALL) evaluiert. Im experimentellen Vergleich erreicht es mit großen Datensätzen ähnliche Genauigkeit bei der automatischen Identifizierung von Blastenzellen und liefert gleichzeitig erklärbare Visualisierungen für menschliche Experten.

Abstract

Decisions of automated systems in healthcare can have far- reaching consequences such as delayed or incorrect treatment and thus must be explainable and comprehensible for medical experts. This also applies to the field of automated Flow Cytometry (FCM) data analysis. In leukemic cancer therapy, FCM samples are obtained from the patient's bone marrow to determine the number of remaining leukemic cells. In a manual process, called gating, medical experts draw several polygons among different cell populations on 2D plots in order to hierarchically sub-select and track down cancer cell populations in an FCM sample. Several approaches exist that aim at automating this task. However, predictions of state-of-the-art models for automatic cell-wise classification act as black-boxes and lack the explainability of human-created gating hierarchies. In this thesis a novel transformer-based approach is proposed that classifies cells in FCM data by mimicking the decision process of medical experts. The network considers all events of a sample at once and predicts the corresponding polygons of the gating hierarchy, thus, producing a verifiable visualization in the same way a human operator does. The proposed model has been evaluated on three publicly available datasets for acute lymphoblastic leukemia (ALL). In experimental comparison it reaches state-of-the-art performance for automated blast cell identification while providing transparent results and explainable visualizations for human experts.

Contents

CHAPTER 1

Introduction

In this chapter, the thesis is first motivated, followed by a specification of the main research questions.

1.1 Motivation & Problem Statement

Deep Learning models are applicable to a variety of problems arising in healthcare. However, since wrong predictions can have severe consequences, the interpretability of models in this domain is crucial. The output produced by a model needs to be transparent, even for clinicians without any knowledge about the interior of the model. For example, Zech et al. [ZBL+18] underpins the urgency to develop explainable approaches in healthcare. Their cross-sectional study revealed that a Convolutional Neural Network (CNN), which was originally designed to identify pneumonia actually differentiated the equipment employed on patients in emergency situations. Using explainable [1] methods could have highlighted the model's wrong assumption early on [ABV+20] and therefore could have helped to verify if the correct concept is learned. This example demonstrates the need for explainability when deploying Deep Learning in healthcare.

The necessity of explainability of Deep Learning approaches also applies to the field of automated cell detection in Flow Cytometry (FCM) data. FCM measures the antigen expression levels of blood or bone marrow cells. It is used in research as well as in daily clinical routines for tasks such as immunophenotyping or for monitoring residual numbers of cancer cells (minimal residual disease, MRD) during chemotherapy. A typical sample contains 50k-500k cells per patient with up to 15 different features (markers) measured. Each feature corresponds to either the physical properties of a cell (cell size, granularity) or to the expression level of a specific antigen marker on the cell's surface [McK18]. While

[1] Note that the terms *explainability* and *interpretability* are used as interchangeable concepts in this work.

methods for automated MRD assessment already reach human expert level performance [WRW+22], they lack interpretability of their predictions. Regardless of a model's performance, clinicians have to manually verify the prediction in a time-consuming process. Using explainable methods could overcome this issue.

Molnar [Mol20] divides existing explainable AI methods into two categories:
Intrinsically interpretable models are interpretable due to their internal structures. Linear models, decision trees, or naive Bayes are common examples of this category. **Post-hoc interpretation methods** analyze a model after training in order to gather explainable insights. A common example of this category are saliency maps, which visualize inner structures of neural networks [NZP18]. In [Elt20] a third category **self-explaining AI** is described, according to which a self-explaining model yields two outputs: a decision and an explanation of that decision. Self-explaining models seem particularly suitable for MRD assessment in FCM data since there already exists a standardized procedure for communication and documentation of the FCM data analyses by medical experts. Predicting this procedure by a model would generate explainable solutions to the problem.

1.2 Aim of the Work

The aim of this work is to develop a novel method that outputs explainable predictions of MRD assessment in FCM samples. The model's explainability is achieved by predicting the entire sequence of polygons (called **gates**) that lead to the classification of the target population. This sequence mimics the procedure which is used to manually analyze MRD in FCM samples by human experts as pictured in Figure 1.1. In this process, several 2D projections of the FCM data are inspected and sub-populations are labeled by drawing gates around them. Gates drawn in specific projections are often applied in sequence, such that one plot only depicts the events selected by the previous plot's polygon. Sequentially applying these gates allows identifying cancer cell populations in the FCM sample. Gating enables the analysis of complex cell populations patterns by a sequence of simpler intermediate steps, which are interpretable by clinicians. Thus, gating is not only a way for finding biologically meaningful sub-populations but has also become the standard for communication and documentation of FCM sample assessment.

The method to develop extends the state-of-the-art set transformer [WRW+22] model. Similar to the model in [WRW+22] the input are events of a full FCM sample. However, instead of predicting the class label for each cell, the model predicts a sequence of 2D coordinates, that form the polygons of the gating hierarchy. The cell-wise class labels, as well as the MRD value, can then be obtained from this gating hierarchy. The model follows an encoder-decoder architecture similar to Facebook's Detection Transformer (DETR) [CMS+20]. For the encoder a set transformer similar to [WRW+22] is used. The decoder design is inspired by [CMS+20]: Object queries are learned for each predicted polygon. These object queries are applied to the encoder's output via cross attention. The goal of providing an interpretable method to assess MRD in FCM data is considered

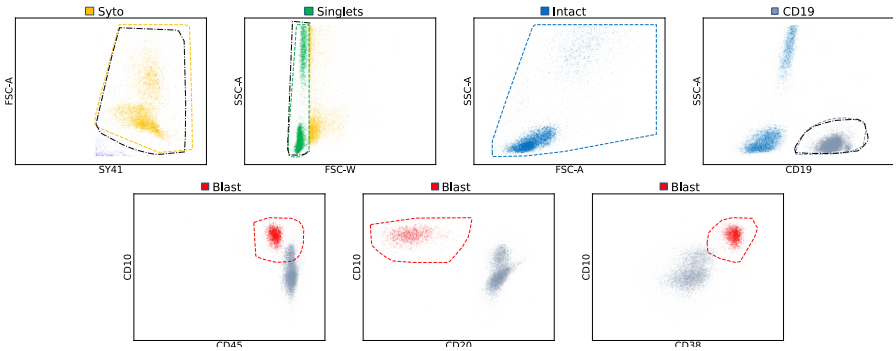

Figure 1.1: Example of a gating hierarchy for a single FCM sample. The gates are drawn in the 2D projection of the FCM data, which is obtained by plotting the expression levels of two markers against each other. The gates are drawn in the order of their application.

fulfilled when the predicted gating hierarchies reach a similar performance as the state-of-the-art set transformer on all experiments stated in [WRW+22]. The performance is measured in median per sample classification F1-Score.

1.2.1 Research Questions

As described above the aim of this work is to investigate development of an self-explainable model for MRD assessment in b-ALL FCM samples. In detail the research is split among the following three questions:

1. *What are the main building blocks that enable to predict FlowCytometry (FCM) gating hierarchies with set transformers?*
   This research question addresses the architectural extensions of the set transformer, which are needed to predict gating hierarchies. Several design choices should be explored: What number of object queries per predicted gate leads to the best performance? How many layers of encoder and decoder lead to the best performance while being computational feasible? How can the loss function be designed to work with a varying number of polygon points in the ground truth?

2. *Which **data augmentation** strategies and synthetic **data generation** strategies are beneficial for the given task?*
   As the model is trained on some rather small datasets (< 60 samples) data augmentation strategies as well as pretraining on artificially generated data should be investigated and their effect on the model performance should be examined.

3. *Which visualization techniques ease explainability of the proposed model's decision process?*
   In this question, additional insights about the model's inner workings should be explored. Common explainability methods for deep learning models include

attention visualization and gradient-based visualization techniques. The suitability of those techniques for FCM data models should be investigated.

1.3 Summary of Results

In the course of this thesis a deep learning method is created that predicts the gating hierarchy for B-ALL MRD assessment of FCM samples. Thereby the proposed method not only automatically identifies the cancer cells in the data, it also provides an comprehensible path of decisions that documents and explains the identification process. When trained on bigger datasets ($\geq$ 180 samples) the proposed method achieves equal performs as the state-of-the-art for direct cell classification [WRW+22]. However, when trained on smaller datasets ($\leq$ 60 samples) it fails to meet the performance of direct cell classification. The conducted experiments indicate that various data augmentation techniques benefit the model's performance, while pre-training on synthetic data only increase the convergence speed but not the performance. Visualizing the self-attention revealed that different heads specialize attending on different biologically plausible populations. Visualizing the gradients of the model's output with respect to the input data demonstrated how the learned relationship between predicted polygon and input data can be inspected.

1.4 Structure of the Thesis

The content of this master's thesis is divided among five chapters. In chapter 1, an introduction into this work is given. Chapter 2 describes the necessary medical background on Flow Cytometry and Acute Pediatric Leukemia. This is followed by chapter 3, which outlines existing methods for automated FCM analysis, object detection and explainability. Chapter 4 provides an overview of the transformer architecture and then describes the proposed method. The last chapter, 5 elaborates on the results of the conducted experiments, exploratory data analysis as well as concludes this thesis.

CHAPTER 2

Medical Background

This chapter provides an introduction to the medical background of this thesis describing the principles of FCM as well as basic knowledge regarding Pediatric Leukemia. In brief, FCM is a laser-based technique that allows us to measure both the physical properties of cells and antigen expression levels. It is used in research as well as in daily clinical tasks such as immunophenotyping or for monitoring residual numbers of cancer cells (Measurable Residual Disease (MRD)).

2.1 Principles of Flow Cytometry

The core underlying mechanism of FCM is to retrieve properties of particles in a fluid by measuring the amount of emitted scatter and fluorescence light when one particle passes a laser. In the case of biological cells, the scattered light reveals physical properties of the cell, such as cell size or granularity, and, under the use of chemicals, the emitted light allows to measure expression levels of antigens on the cell's surface [doi13a]. Modern cytometers are able to analyze hundreds-thousands cells in one sample measuring over 15 different features per cell. One recorded observation is referred to as *event* as acquisition artifacts can cause recordings without a concrete cell present.

2.1.1 The Flow Cytometer

A cytometer consists of three main parts [Dic02] as illustrated in Figure 2.1:

- *Fluidic System*
  This component aims to correctly present one cell after the other to the laser. A fast-moving stream of sheath fluid surrounds the sample fluid and focuses it in front of the laser. This process is called hydrodynamic focusing as pressure, velocity and density differences of the two fluids prevent them from mixing and allows to focus the sample fluid, forcing the cells to pass the laser individually [doi13a].

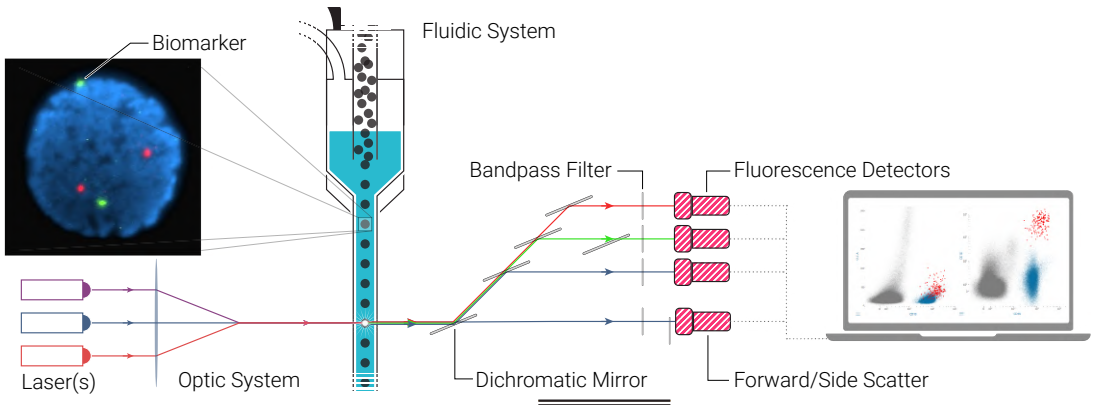

Figure 2.1: The main parts of a cytometer consisting of a fluidic system, optical system and electronic system.

- *Optical System*
  The optical system consists of lasers, focusing lenses, prisms, collection lenses, mirrors and filters. Modern cytometers use multiple lasers that transmit different wavelengths of light. When a cell is illuminated by the laser beam scatter light and fluorescence light is emitted. The scatter light is either diffracted at narrow angles (Forward Scattered (FSC) light) or 90° (Side Scattered (SSC)) to the laser beam. The amount of scatter light is measured by detectors in line and in 90° to the laser beam. In addition to the scatter light, specific dyes and fluorochromes are used to produce fluorescence light. Either dyes bind directly to a cell or fluorochromes are bound to antibodies that bind to the cell's surface antigens. These molecules absorb light of a specific wavelength band (absorption spectrum) and as a result emit light in another wavelength band (emission spectrum). A careful selection of fluorochromes is needed to minimize overlaps in the emission spectrum of different fluorescent compounds. Photon-multipliers together with optical bandpass filters are used to measure the amount of emitted light of different spectra [doi13a].

- *Electronics*
  This component is responsible for the conversion of optical to digital signals. The comparable strong signal of the FSC light is captured by photodiodes. While the less intensive SSC and fluorescent light is collected by more sensitive photon-multiplier tubes. In brief, photodetectors convert photons of incoming light to electrons, which generates a voltage pulse that is proportional to the number of detected photons [Giv01]. A analogue-to-digital converter (ADC) transforms the produced signal into digital numbers and records the height, width and area of the voltage pulse [doi13a].

2.1.2 Staining & Cluster of Differentiation

Scatter light is caused by laser light passing through the cell membrane and refracting and reflecting on cytoplasmic organelles or nucleus of the cell. In contrast, fluorescence light is emitted by special chemicals attached to or placed inside a cell. Usually, these fluorochromes are combined with specific antibodies to which they covalently bind. The antibodies anon bind to specific cell antigens. The process of combining the sampled cells with fluorochromes is called *staining*. The Cluster of Differentiation (CD) nomination gives the antigens unique and standardized names [EBB$^+$15, McK18]. For instance, cluster of differentiation number 19 (CD19) is a surface marker for B-lymphocyte cells [RRK$^+$16]. Besides these, there are also fluorochromes that do not utilize antibodies or target cell antigens. For instance, cell dyes like propidium iodide (PI), which binds to nucleic acids of a cell's DNA, are used to assess a cell's viability since it is blocked by an intact cell membrane but can pass in dying or dead cells [RN06]. For instance, there is the specialized DNA polymerase Terminal deoxynucleotidyl Transferase (TdT), a molecule, which is only present inside a cell's nucleus. In this case, before staining, the cell's surface and nucleus membrane must be permeablized, such that the fluorochromes can pass through [GTB$^+$09]. Fluorescent dyes can be categorized into single dyes and tandem dyes. While single dyes emit light at a single wavelength, tandem dyes emit multiple wavelengths simultaneous as they consists of multiple fluorescent dyes chemically linked together [LRVBACL09, BNRC12].

2.1.3 Data Visualization

The measured scatter and fluorescence signals are often displayed in two dimensional plots, where each measured event is represented by a point. The signal strength of two parameters define the point's location. According to the AIEOP-BFM consensus guidelines of 2016 [DBG$^+$18] cell populations can be classified into one of three categories based on the measured signal strength:

- *Negative*: Populations of low to no expression of a particular marker.
- *Dim Positive*: Populations of moderate measured signal intensity, but strong enough to exceed potential background variability.
- *Positive Bright*: Populations of intensive measured signal intensity.

Most antibody conjugated fluorochromes generate a large dynamic range of fluorescence intensity. Therefore, to view the wide-ranging FCM data, the displayed data is usually logarithmic or logical scaled as discussed in Subsection 2.1.5.

2.1.4 Compensation

Although the fluorochrome selection aims to minimize emission spectra overlap, complete avoidance is in practice often not possible. Figure 2.2 illustrates the emission spectral

profile of the two fluorochromes Fluorescein Isothiocyanate (FITC) and Phycoerythrin (PE). It displays how spectral profile overlap is recorded by opposing detectors. The red area shows the amount of FITC fluorescence detected in the PE channel. Because signal spills over from FITC to PE, this is called spillover [Roe02]. This effect can introduce misleading measurements in the obtained data. For instance, cells without any antigen for the PE stained compound expressed on their surface, can display a dim appearance of PE, since the emitted light of FITC spills over to the spectrum of the PE detector. Another common example of a spillover-related issue is the appearance of double-positive events in biological unreasonable settings. For instance, CD5 and CD19 are mutually exclusive markers in healthy cells and are only known to be double positive in B-cell Chronic lymphocytic leukemia (CLL) but due to spillover can be spotted in a healthy sample too [DGA+11]. To account for the unwanted side-effects a post-processing step called *compensation* is conducted. Knowing the proportion of light that spills over to another fluorochrome's detector, allows subtracting this amount from the measured signal in the other detector. As shown in Figure 2.2 the red area depicts the amount of FITC light detected in the PE detector. This area is always proportional to the amount of light in the FITC detector. Therefore, staining cells only with FITC reveals the spillover amount in the PE detector and can further be used to compensate the spillover by subtracting this proportion of FITC signal from the measured PE signal. This process only demonstrates the compensation for the spillover of one fluorochrome into one other's detector. However, in practice, the procedure is more complex:

1. Spillover usually occurs in both directions, for instance not only from FITC to PE but also from PE to FITC (see Figure 2.2).

2. Modern cytometers have multiple lasers with different excitation spectra and measure signals in 10-20 different channels resulting in a more complex spillover behavior.

3. Not only the used compounds emit light but also the cells themselves emit fluorescence light by a phenomenon called *autofluorescence*. The cytometer laser generates fluorescence emission from cell compounds such as the coenzymes Nicotinamide adenine dinucleotide phosphate (NADPH) or flavin, occurring in mitochondria and lysosomes [doi13a]. Autofluroscence is more likely for short wavelength lasers (488nm) [PR07], on aged cells [doi13a] and for specific cell types like myeloid cells due to their high content of granule-associated flavoproteins [Mon05].

Nevertheless, also in more complex cases compensation is based on the principle of subtracting the amount of spilled over signal per channel. The parameters defining the compensation are therefore stored in a $C \times C$ matrix, where C denotes the number of channels. To address the complexity the compensation procedure is often aided by software and special chemicals are stained to measure spillover amounts. In general, it is still preferable to minimize spillover in the first place by carefully selecting fluorochromes tailored for the intended use. Spectral overlap is still one of the most frequent sources

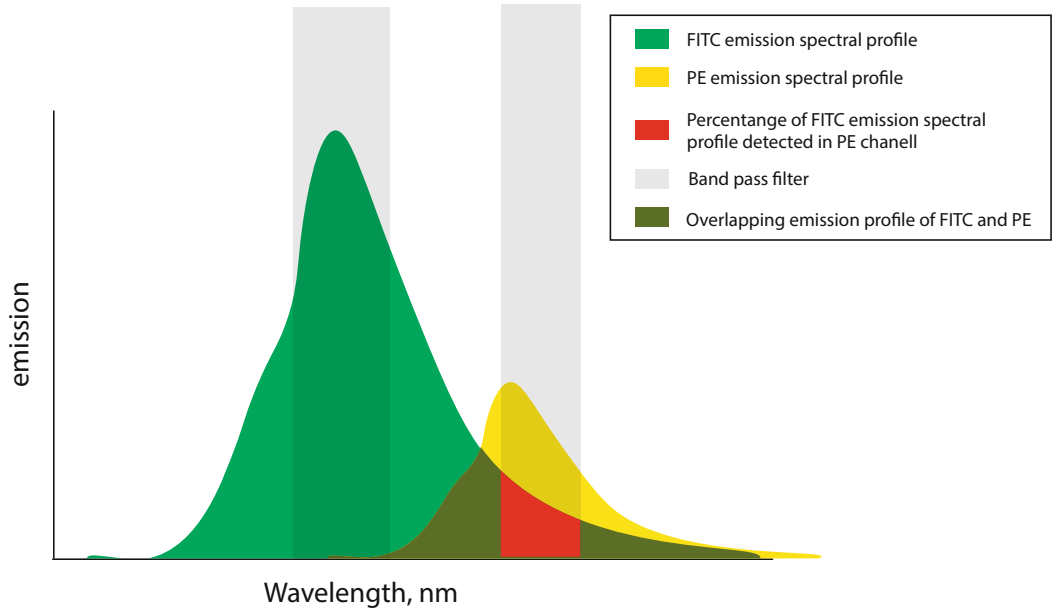

Figure 2.2: The emission spectral profile of the FITC and PE and the bandwidth of two detectors. Figure inspired by [BR].

of error in flow signal and requires experienced operators [doi13a, Roe01]. For more in depth discussion about compensation in FCM the reader is referred to [Roe02].

2.1.5 Logicle Scale

After compensation, some cell populations can have low mean and include events with negative data values. Both, low mean and negative values cause issues on the logarithmic scale, as displayed in Figure 2.3. The Figure shows that the population with a low mean is disturbed on the log scale, as the location of the median (red cross) differs from the visual center of the data [PRM06]. Parks et el. [PRM06] proposed a display method called *logicle* scaling, which addresses these issues and aims to be advantageous over both linear and logarithmic scaling. Logicle scaling combines properties of both logarithmic functions and linear functions. For large data values it is logarithmic, to ensure a wide dynamic range and provide good visualization for population at high fluorescence intensities. Near zero the function is linear and also extends to negative values to correctly display populations with low mean fluorescence intensity as well as negative values after compensation. The logicle scaling is based on the hyperbolic sine function (sinh):

$$sinh(x) = \frac{e^x - e^{-x}}{2}. \tag{2.1}$$

Parks et el. refers to the generalization of sinh as the family of biexpoential functions:

$$S(x; a, b, c, d, f) = ae^{bx} - ce^{-dx} + f. \tag{2.2}$$

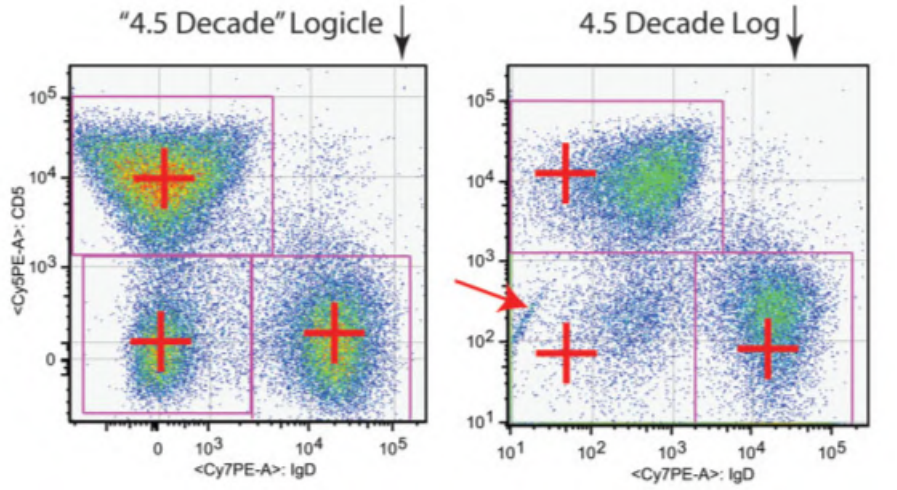

Figure 2.3: Two scatter plots of the same FCM sample. The right plot is scaled with *logicle* scale and the left plot is logarithmic. The red arrows indicate the median of the three populations. Graphic taken from [PRM06].

For a given FCM sample the best function's parameters are computed dynamically. For a more in depth discussion about details on the logicle scaling the reader is refered to Parks et el. [PRM06].

2.1.6 Gating

The conventional procedure to analyze FCM data in the clinical routine is to look at 2D projections of the FCM data and label sub-populations of events by drawing polygons around them [McK18]. This procedure is called **gating** and the polygons are called **gates**. As illustrated in Figure 2.4, gates act as filters by defining the events that are subject to further analysis in other 2D projections (events inside a gate) and the events that will be discarded (events outside the gate). The target population can then be identified by a boolean combination of gates. Gates drawn in specific projections are often applied in sequence, such that one plot only depicts the events selected by the previous plot's polygon. Sequentially applying these gates allows to identify cancer cell populations in the FCM sample. The 2D plots of the data space allow to explicitly depict antigen expressions of the cells in the sample, which are known to be relevant in particular diseases. For example, among other characteristics, CD19 is known to be higher expressed for B-cells [McK18]. Gating allows analyzing complex patterns of cell populations by a sequence of simpler intermediate steps, which are interpretable by clinicians. Thus, gating is not only a way for finding biologically meaningful sub-populations but has also become the standard for the communication and documentation of FCM sample assessment.

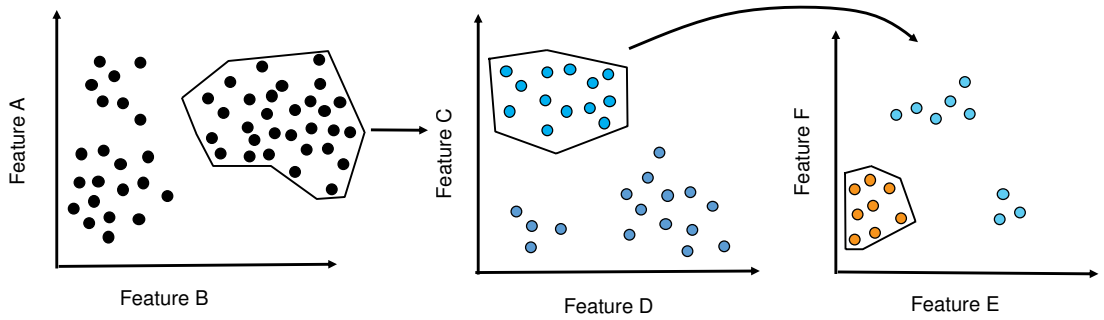

Figure 2.4: Gating Hierachy: By sequentially sub-selecting individual cell clusters in different 2D projections, medical experts can track down cancer cells.

2.2 Hematology and the Human Immune System

This section provides a short overview of the human immune system and the primary hematologic cells involved.

2.2.1 Human Immune System

Human blood essential provides three functionalities:

1. The transportation and distribution of nutriment throughout the body as well as regulation of body temperature, pH and water balance. This is mainly achieved by erythrocytes (red blood cells).

2. Blood protects the body from physical damages to tissues. The thrombocytes in blood are responsible to close and heal wounds.

3. Various other cells circulating in the blood are responsible to defend the body against infections and invaders.

The immune system can be divided into two parts the *innate immune system*, that consists of all counter measurements that are unspecific to any particular invader, and the *adaptive immune system*, which consists of the cells that specialize for specific pathogens [MWWK18].

Besides physical and chemical barriers the innate immune system consists of *macrophages*, *neutrophils*, *natural killer cells* and *dendritic cells*. An invading pathogen to the human body will most likely first encounter macrophages and neutrophils. Macrophages are rather big cells that attack and swallow foreign bacteria. Neutrophils join the battlefield guided by chemical signals released by cells in response to the presence of a pathogen. They attack pathogens by releasing toxic substances, swallowing or forming web-like structures that trap and kill pathogens [YL15].

The consistent activity of macrophages and neutrophils increases inflammation, a body response that floods the attacked area with blood and deploys *complement proteins*. This effect is usually perceived as a warm and red swelling of a wound. The complement proteins coat pathogens to assists other immune cells to recognize those. Some also directly cut wholes into pathogen's cell membrane and increase the inflammation response. *Denditric cells* collect parts of the observed pathogens at the site of the wound and then move through the lymph system in search of a matching *helper-T-cell*. The body constantly produces T-cells with different specialized receptors. Once found, the *denditric cell* activities the *helper-T-cell* by presenting the found pathogen parts to specific receptors of the *helper-T-cell*.

The *helper-T-cell* immediately starts to divide itself. Some of its clones enter the battlefield, where they reactivate and intensify the immune response of the macrophages. While other clones aim to find and activate corresponding *B-cells*. The *B-cells* produce specialized antibodies for the observed pathogen and flood the battlefield with these antibodies. The antibodies attach to the pathogens and thereby damp their activity and mark them for the other immune cells. Some of the helper-T-cells and B-cells remain and turn into memory cells, which accelerates future immune response of the same and similar pathogens [MWWK18, YL15].

2.2.2 Blood Cell Maturation

Blood cells develop from haematopoietic stem cells (HSC) in the bone marrow (BM). These immature cells can differentiate into progenitors of any lineage, are $CD34^+$ and do not express any lineage-related antigens (for instance markers for lymphoid or myeloid differentiation) [doi13b]. Figure 2.5 depicts the maturation of blood cells, showing the different possible lineages from a hematopoietic stem cell to fully developed blood cells. The HSC further develops into two major progenitors:

- *Common myeloid progenitors*, which, once fully develop, become neutrophils, eosinophils, basophils, monocytes, erythrocytes, megakaryocytes, mast cells, marcophages or myeloid dentiric cells and are defined by the following expression pattern: $CD34^+CD117^+CD45^{dim}CD13^+$ [doi13b].
- *Common lymphoid progentiors*, which can develop into natural killer cells, lymphoid dendritic cells, B and T-cells and are defined in contrast to common myeloid progenitors by the following expression pattern $CD34^+CD117^-CD45^{dim}CD13^-$ [doi13b].

Myeloid Maturation

This lineage describes the maturation of common myeloid progenitors that emerge from HSC. Myeloid progenitors can develop into megakaryoblasts, mast cells, proerythroblasts, or myeloblasts. Megakryoblasts will finally develop into megakaryocytes, which produce thrombocytes. Proerythroblasts develop into erythtocytes, which transport oxygen in

the blood. Myeloblast develops into granulocytes (basophils, neutrophils and eosinophils) or monocytes (macrophages and myeloid dendritic cells).

Granulocyte maturation describes the progression from myeloblasts into promyelocytes, myelocytes, metamyelocytes, bands, and finally, the three granulocytes: basophils, neutrophils and eosinophils. CD34, CD117 and HLA-DR expression initially accompany the maturation. However, when maturing into promyelocytes HLA-DR and CD34 expression gets lost and CD117 gets lost when maturing into myelocytes. A full-grown neutrophil expresses CD45, CD13, CD33, CD11b, CD15, CD16. In promyelocyte and myelocyte state they can express CD64 [doi13b, Hof09]. Fully developed basophils express CD45, CD13, CD33, CD38, CD123, CD25, CD9 and CD22 [HJB+08]. Matured eosinophils have intense granularity and therefore show high SSC signals. They express CD45, CD13, CD11b, CD66 and CD16 [doi13b].

Monocyte maturation, which also starts with myeloblasts, manifests itself, similar to granulocyte maturation, by CD34 and CD117 expression. In contrast to granulocyte maturation, HLA-DR remains expressed during the whole monocyte development. Mature monocytes are characterized by CD4, CD64, CD14 and CD15. While CD15 is also expressed on neutrophils and CD4 on T-cells and CD64 on myeloblasts, dendritic cells and activated neutrophils, CD14 is specific to monocytes [doi13b].

Erythroid maturation describes the development from common myeloid progenitors to proerythroblasts, erythroblasts, polychromatic erythtocyte and finally to erythrocytes the red blood cells. Erythroblasts express CD34, CD38, CD117 and CD45. These markers get lost during maturation and expression of CD72, CD235a and CD36 establish [doi13b].

Megakaryocitc maturation describes the development from common myeloid progenitors to megakaryoblasts, promegakaryoctes to megakaryocytes. During maturation, precursor markers are lost and expression of CD41, CD42 and CD61 is gained [doi13a].

Lymphoid Maturation

The lymphoid maturation compromises the development of the common lymphoid progenitor, which are unspecific cells of the lymphoid lineage. They can develop into natural killer cells, T- or B-lymphocytes. Common lymphoid progenitors express CD45RA and CD127 [MAS+03].

B-Cells and NK cells develop in the bone marrow. Immature NK cells express low CD16 and high CD56, which progresses into high CD16 and low CD56 over the maturation process. Throughout their complete development B-cells express CD19 and CD38. Early B-cell express CD10, Cd34 and TdT, which vanishes over maturation. In contrast, mature B-cells express CD22, CD20, CD79a and surface immunoglobulin [DFF+97, VLWVDB+00].

In contrast to the other lineages, most T-cells' maturation occurs in the thymus. CD7 serves as a T-cell marker throughout the complete development. Early T-cells express Tdt and CD34, while mature T-cells express CD45, CD2 and CD4 or CD8 [doi13b].

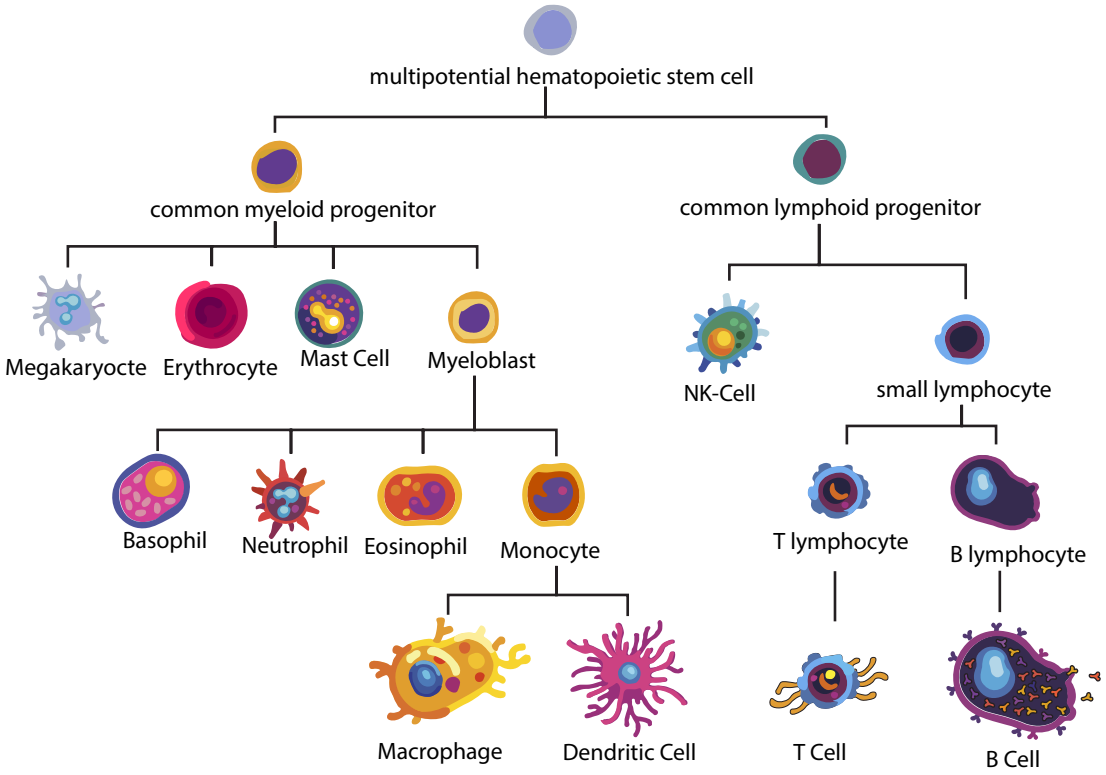

Figure 2.5: The different lineages from a hematopoietic stem cell to fully developed blood cells. In Acute Myeloid Leukaemia (AML) the lineage of common myeloid progenitors is affected, while Acute Lymphoblastic Leukemia (ALL) entails the common lymphoid progenitors lingeage.

2.3 Targeted Disease: Acute Pediatric Leukemia

Acute leukemia is an abnormal proliferation of partially developed blood cells [OK20]. Since these cells are not yet developed unmatured blood cells, they do not function as effectively as matured blood cells, but they still consume nutrition and space, which is disparately needed for healthy matured blood cells [OK20].

According to which kind of precursor cells are dividing in an uncontrolled manner, acute leukaemia can be classified into AML, where myeloid cell precursors are affected, and ALL, where lymphoid cell precursors proliferate uncontrollably [OK20]. See Figure 2.5 for the bifurcation in the hematopoietic development. ALL is more common and is associated with about 25% of all childhood cancer cases [KGS19, KDPV+13]. Most cases of ALL in children are reported in patients aged 2 to 5 years [IGM13]. While ALL is more common in childhood, AML is more common in adulthood. The number of registered AML cases increases with the patients' age [OK20]. The following symptoms can be directly related to the suppression of normal blood cell formation, called hematopoiesis,

and the overflood of unmatured blasts in an uncontrolled way: Pain in lymph nodes is caused by the deposition of vast amounts of blast cells in the lymphatic system [OK20], which is comparable to pain in bones originating from the growth of leukemic cells in the medullary space [RLS+05, Onc09]. Fatigue is caused by anemia (due to the suppression of healthy hematopoiesis) [RLS+05, OK20]. Fever is related to the reduced amount of neutrophil granulocytes and easier bleeding results from a shortage of thrombocytes [RLS+05, Onc09].

Normal precursor cell populations in the BM can be easily confused with leukemic blast cells since they share many antigens of immaturity and morphological similarities. However, normal precursors are usually less frequent than leukemic blast cells [doi13b]. For instance $CD34^+$ cells are usually only up to 1-2% of all BM cells [BBL+02, NNSP18]. Although, in stages of regenerating, for instance, after chemotherapy or transplantation, BM can express proportionally higher levels of progenitor cell antigens than in steady state [DFF+97, VLWVDB+00].

2.3.1 Treatment of childhood ALL

The main goal of ALL treatment is to decrease the number of blast cells. The treatment can be divided into three different phases: (1) remission-induction, (2) consolidation and (3) maintenance [CB15, PRL08]. The initial phase, called remission-induction, aims to achieve a remission, which means that nearly no leukemia cells are observable in the BM and normal healthy blood cells regenerate [PRL08, CB15]. In practice, remission is defined if less than 5% blasts are detectable in the peripheral blood by microscopic morphology assessment [BVU+05, SHP+12]. The chemotherapy in the remission-induction phase is carefully adjusted according to the leukemia subtypes, risk potential, genetic prepositions and the patient's response [CB15]. This first phase usually takes 4-6 weeks and approximately 95% of all patients achieve remission in this period [CB15]. Consolidation is the second phase. It lasts 6-9 months and aims to further reduce the post-remission remaining blasts cells, also called MRD [CB15, PRL08]. The third phase of treatment is called maintenance. This phase usually lasts at least two years. Here much lower doses of chemotherapy are given to the patient. The goal of the maintenance phase is to lower the risk of relapse [CB15].

In addition, intrathecal chemotherapy is applied to remove any blasts in the brain and spinal cord. Before the 1970s, additional intrathecal chemotherapy was not part of standard treatment protocols, which led to several cases of central nervous system relapses after BM remission [EGZ70]. As an alternative to intrathecal chemotherapy in the past, it was common to use cranial radiation. However, since the risk of intellectual disabilities emerged, the use of cranial radiation has been minimized [PCP+09, CB15].

One remaining issue in ALL treatment is the risk of relapse. While in 1998 a relapse rate of 25-30% has been assumed [MKS+98], more recent reports speak of ALL relapse rate of 15-20% [CB15]. Nevertheless, more sensitive techniques are needed to identify potential relapsing cases early. The number of children with ALL who's treatment fails

is similar to the number of new AML cases [Gay05]. FCM is a commonly used method, which enables sensitive MRD assessment [BVV+09].

2.4 Flow Cytometry in Pediatric Leukemia

The clinical application of FCM in Leukemia is twofold: Firstly FCM is used to diagnose and classify leukemia types (Immunophenotyping). Secondly, during and after therapy, FCM serves to asses the treatment response [doi13c].

It is important to note that no single marker exists, which is specific for any Acute Leukemia (AL), rather patterns of expression are used to spot AL blasts [doi13d]. Common expression patterns on AL include:

- *Aberrant Expression*
  For instance, lymphoid marker expression on myeloid cells is a sign for AML and can be defined as an abnormal expression of foreign-lineage antigens.
- *Abnormal Co-expression*
  The expression of maturity and immaturity is an example of this expression pattern.
- *Abnormal Expression*
  This expression pattern includes abnormally increased or decreased expression or unusually homogeneous expression of a usually heterogeneous expressed antigen.

In general leukemic blast cell populations show heterogeneous expression patterns, but AML are considered to be more heterogeneous than ALL blasts [doi13d].

2.4.1 Diagnosis

Based on clinical symptoms and indices in the peripheral blood bone marrow sampling is conducted. The aim of the FCM analysis is to find hallmarks of specific AL blasts to derive a diagnosis. While leukemic blast cells are heterogeneous and therefore exhibit different appearances in the same FCM sample, only the bulk population of leukemic cells is of interest for clinical diagnosis. In MRD an exact knowledge of the blast characteristics at diagnosis is prerequisite to allow precise enumeration of very small numbers of blast cells.

2.4.2 Minimal Residual Disease

Monitoring the response during and after the treatment via MRD is an important prognostic indicator as it can influence decisions on the duration, type and intensity of the treatment [doi13c, PMER12]. MRD can be asses with Polymerase Chain Reaction (PCR) or FCM, while light microscopy is considered to be too insensitive [doi13c]. The estimated sensitivity in cell-based ALL blast identification is defined as the maximum amount of

normal cells among which one blast cell can be detected. For FCM this sensitivity is estimated as 10^3-10^4 and that of PCR is estimated as 10^4-10^6 [vDvdVBO15]. However, FCM is considered to be more cost and time efficient than PCR [GCV+12, vDvdVBO15]. For AML PCR is only applicable for a specific gene fusion, which is only presented for 30% of AML patients [doi13c].

Although gating strategies vary depending on the used antibody panels, they follow similar steps for B-ALL [doi13d]:

1. *Select actual cells*
   First all measured events that are not actual single living cells, like debris, doublets or air bubbles are gated out. Usually projections in FSC-A, SSC-A and Syto-41 are used to perform this omission.

2. *Identify B-cells*
   The second step is to pin down the analysis on B-cells by utilizing common markers for B-cells like CD19.

3. *Separate blasts from healthy B-cells*
   In the last step B-ALL blasts are searched within the B-cell population. Common markers in this step are CD10, CD45, CD20 and CD38. The percentage of the found blasts in relation to all measured actual cells is usually reported as MRD.

CHAPTER 3

State of the Art

Numerous approaches have been established to automate the detection of cell populations in FCM data. The reader is referred to [CCW+21] for a more comprehensive review of current trends in automated FCM data analysis. In this chapter I firstly outline the landscape of automated approaches for FCM data, followed by a characterization of explainable methods for deep learning models and machine learning in the context of FCM data.

3.1 Targeted analysis of FCM data

In this work methods for the targeted analysis of FCM data are divided into event-wise and holistic approaches. Approaches that classify each event solely based on the presented information of this event are referred to as event-wise approaches. In contrast, holistic approaches process a whole FCM sample and, therefore can account for inter-sample variations, which has been identified as crucial for the correct classification of cell populations with high variability such as leukemic cells [WDRMG21].

Event-wise Approaches In [AvUH+19] linear discriminant analysis is proposed for the classification of cell populations as it allows for interpretable performance and reproducibility. Authors in [LKB+17] and [JNQ+18] use a table of marker expression patterns in different cell types as a reference dictionary. Methods based on neural networks include [LSR+18, LSS+17].

Holistic Approaches FlowDensity [MTC+14] and FlowLearn [LBC+18] use an operator's 2D gating strategy as a guideline for detecting cell populations. Recently, a one-class classification approach based on Uniform Manifold Approximation was introduced [WKW+22]. Further, Gaussian mixture models (GMM) have proven to be well suited to model cell populations in FCM data [CHL+15, RDS+19]. Reiter et al.

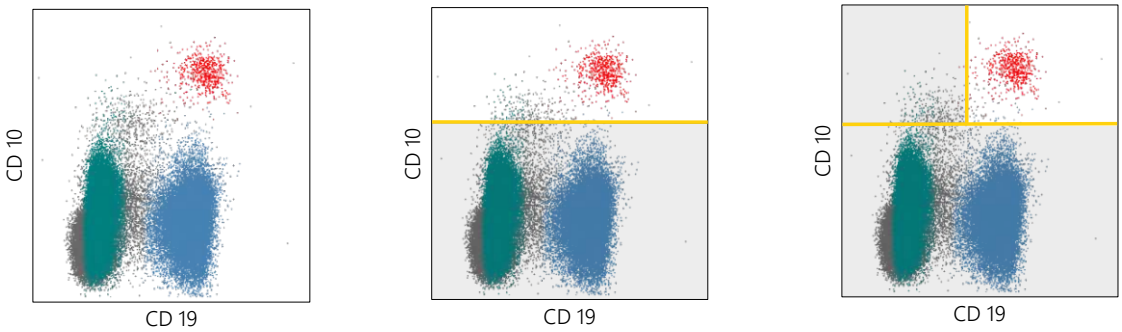

Figure 3.1: 2D Projection of one patient's FCM sample. The red dots indicate the position of blast cells. A simple decision tree, based on the event features can be constructed to classify blast and non-blast cells.

[RDS+19] fit a linear combination of GMMs with labeled components to an unseen sample by Expectation Maximization (EM). [ZMH+20, AC17, WRW+22] are approaches based on neural networks that can process a whole sample at once. Authors in [ZMH+20] use self-organized maps to obtain a 2D image that a CNN further processes. CellCNN [AC17] automatically learns a concise cell population representation with a 1D-convolution layer followed by a pooling layer to aggregate information. More recently, Wödlinger et al. [WRW+22] presented a method based on the transformer architecture [VSP+17] that performs classification on single-cell level, while processing an entire sample in a single neural network forward pass. The attention mechanism of the original transformer architecture [VSP+17] entails a quadratic complexity in the input length $\mathcal{O}(n^2)$ of both memory and time, which is unfavorable in the context of FCM data as one sample can contain up to millions of events. Wödlinger et al. thus use the concept of the Induced Set Attention Block (ISAB) as introduced in the set-transformer [LLK+19] that reduces the complexity to $\mathcal{O}(n)$.

Necessity of Holistic Approaches Holistic approaches typically outperform event-wise approaches, because they can better deal with inter-sample and inter-patient shifts. Consider Figure 3.1, which depicts a 2D Projection of an FCM sample with pediatric B-ALL. For this sample, a discriminative approach, for instance, a decision tree, can yield reasonable performance for the classification of blast and non-blast cells. For comparison, Figure 3.2 shows the same 2D plot for two other patients. The Figure demonstrates, that the number of blast cells as well as the position of the blast cell cluster can drastically differ from one patient to another. The decision tree from Figure 3.1 would not be applicable for the two samples in Figure 3.2. It is therefore not possible to solely rely on a cell's feature values for accurate blast identification. Instead, the whole sample must be considered in order to draw a decision based on the relative relationship between different cell clusters.

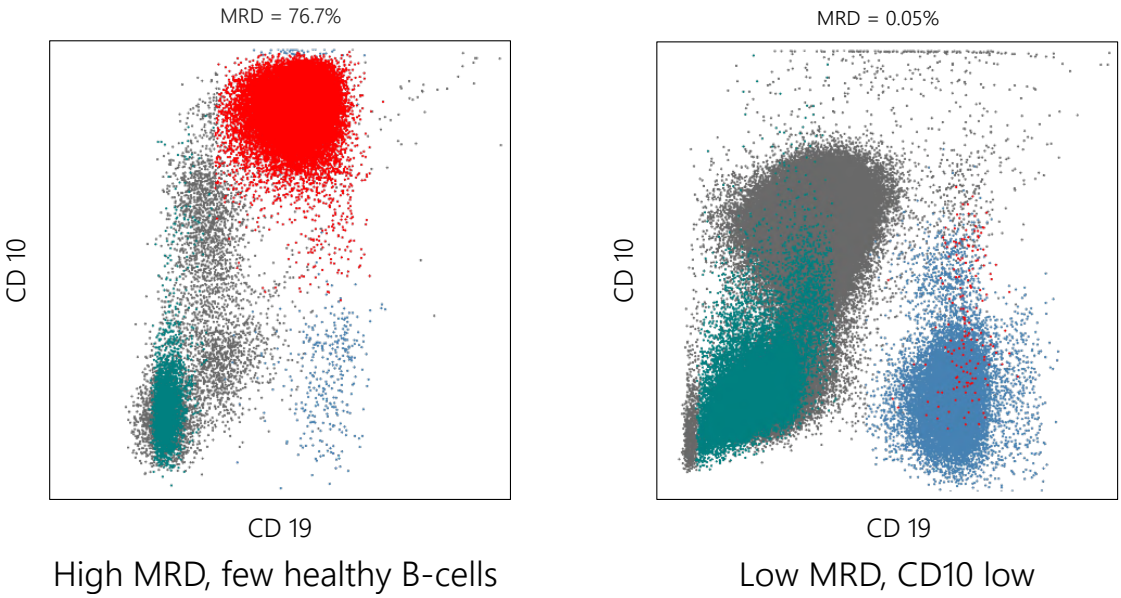

Figure 3.2: 2D Projection of two patient's FCM samples. The red dots indicate the position of blast cells. The number of blast cells as well as the position of the blast cell cluster can drastically differ from one patient to another.

3.2 Object Detection Approaches

The task of predicting polygons that surround cell clusters in FCM data can be loosely related to object detection in computer vision. The goal of object detection is to locate and identify objects in an image or video. Various approaches have been proposed, including sliding windows, Region-Based Convolutional Neural Networks (R-CNN) [GDDM15], and You Only Look Once (YOLO) [RDGF16]. YOLO is a real-time object detection algorithm that divides an image into patches and uses a single network to predict bounding boxes and class probabilities for objects in each patch.

In recent years, there has been growing interest in using transformers for object detection. Transformers are a type of neural network originally developed for natural language processing tasks and have shown remarkable success in various domains [VSP+17]. Detection Transformer (DETR) [CMS+20] is a object detection algorithm that utilizes transformers. Unlike YOLO and other grid-based approaches, DETR operates on a set of learned object queries. The model uses a transformer network to process a sequence of input pixels and generates class and bounding box predictions for each object query.

Evaluations on benchmark datasets [CMS+20] have shown that DETR outperforms the state-of-the-art object detection methods, such as Faster R-CNN [Gir15] or Mask R-CNN [HGDG17]. The results indicate the potential of using transformers for object detection.

3.3 Towards Explainability

In this section methods for explainability and model inspection that all aim to enhance the transparency of a model's decision in the context of FCM data and deep learning are visited. First, related explainable approaches for automated FCM processing are presented. Followed by a description of two commonly used explainability visualization techniques for deep learning models (gradient-based visualization) and transformers (attention visualization).

Explainable Approaches With respect to explainability of results, [JNQ+18, MTC+14, LBC+18] can be listed as their results rely on predicted thresholds and hence are interpretable. Algorithmic Population Descriptions (ALPODS), as proposed in [UHR+21], is designed to provide explainability by fuzzy reasoning rules in a Bayes decision network expressed in visualizations similar to those generated by domain experts. Simonson et al. [SWW+21] aim to identify impactful cell populations in FCM data with SHAP values [Sha53]. They employ an ensemble of CNNs trained on different 2D histograms of FCM data to detect classic Hodgkin lymphoma on a sample level. The SHAP values were used to identify the 2D histograms and the specific regions within them that had the greatest impact. Another approach related to explainable AI and the method presented in this work is GateFinder [ASK+18]. Its goal is to find the shortest yet most discriminative series of 2D polygon gates that lead to a previously specified target population. Although the goal of GateFinder is not targeted analysis, the underlying idea of mimicking the gating strategy of domain experts is similar to the approach presented.

Attention Visualization A common post-hoc method to interpret a transformer's decisions is to visualize how the model attends to different parts of the input data [BCB14, BBS+18, Vig19], often called **attention visualization**. For instance, Jesse Vig [Vig19] proposed an open-source tool to investigate self-attention between word-tokens by visualizing weighted edges between words and coloring the words based on the attention magnitude. The visualization allows to focus on individual words, as well as differentiates between the attention of different heads over multiple layers. For computer vision tasks, overlaying the input image with a heatmap is commonly used to visualize the attention [DBK+20, ADT+22, CMS+20]. For the task of object detection, Carion et al. [CMS+20] demonstrate how, in an image with multiple objects, one pixel of an object mainly attends to the other pixels belonging to the same object. Besides self-attention, the authors in [CMS+20] visualize the cross attention of a learned object query to the input image, revealing which pixels in the image are important to locate an object's bounding box as well as for assigning a class label to that object. For instance, the authors showed an example where the object query's attention is concentrated on an elephant's outlines and especially on characteristic features such as the elephant's trunk. Attention visualization facilitates model interpretability since it reveals what part of the data and which relationship among the data is considered important by the model. It

can be used to verify that the model learned desired concepts or to spot an unconscious bias like attending to the background of an image to classify an object in the foreground.

Gradient-based Visualization While attention visualization is a model-specific technique for attention-based architectures such as transformers, other common post-hoc explainability techniques for deep learning models, which are not restricted to models using attention, are **gradient based methods** such as Saliency Maps [SVZ13] or Gradient-weighted Class Activation Maps (Grad-CAM) [SCD+17]. These and related methods rely on computing the model's gradients of a specific class output with respect to the input image. Pixels with a low gradient norm are considered unimportant for the model's class prediction, as they have minimal impact on the output value. On the other hand, pixels with a high gradient norm are important for the prediction, as changing their values leads to a significant change in the class output. Saliency Maps were initially designed for Convolutional Neural Networks (CNNs) [SVZ13] but are also used for vision-based transformers [LZW+21, AGBD21], mixed modality vision and text transformers [ADT+22] as well as pure text transformers [ASLA20].

CHAPTER 4

Methodology

In this chapter, the proposed method is described. The selection of the method is motivated by the following discussion about the application of neural networks to FCM MRD assessment: For that purpose, a FCM sample is defined as a set of events $E \in \mathbb{R}^{N \times m}$. N defines the number of events ($50 - 500 \times 10^5$) and m denotes the number of markers (typically $10 - 20$). Furthermore $Y \in \{0,1\}^N$ denotes the set of class labels for which $\forall e \in E \exists y \in Y$ holds. A reasonable idea is to apply neural networks by processing each event of a FCM sample on its own and let the network classify each event between healthy and blast. In this case the neural network is a function $f_{eventwise} : \mathbb{R}^m \to \{0,1\}$. This approach is refered as *event-wise classification*. However, as described in Chapter 3, automated MRD assessment is not solvable by pure discriminative approaches and therefore demands holistic approaches. Therefore a legitimate adaption towards holistic approaches is, instead of processing each event separately, to use a fully connected neural network in which all events of an FCM sample are provided as input. In this case the neural network is a function $f_{fullyconnected} : \mathbb{R}^{N \times m} \to \{0,1\}^N$. This approach comes with two obstacles: First, the network size of such a fully connected network would be infeasible. Even one layer would exceed the memory of state-of-the-art consumer GPUs. For instance connecting 500×10^3 events to each other results into $(5 \times 10^5)^2 = 25^{10}$ connections. Secondly, since the events of a FCM sample have no inner order, we would expect the predicted class labels to change according to changes in the input. Consider a neural network as a function f that transforms a set $X = \{x_1, ..., x_M\}, x_m \in \mathbb{X}$ into a set $Y = \{y_1, ..., y_M\}, y_m \in \mathbb{Y}$, such that each instance x_m has an associated label y_m. For any permutation of the input instances $\pi : f([x_{\pi(1)}, ..., x_{\pi(M)}]) = [f_{\pi(1)}(x), ..., f_{\pi(M)}(x)]$ we expect the output labels to permute accordingly [ZKR+17]. This property is known as permutation equivariance. Fully connected neural networks do not fulfill this property, since they are sensitive to the order of input instances. For the same reason, Recurrent Neural Network (RNN) are not applicable to FCM data, since they require processing the input instances as sequence. To sum it up, processing FCM data by a neural network

requires the following properties:

- **holistically**: The classification decision for an event cannot be based purely on this event's features. Instead, the whole FCM sample must be taken into account [RDS+19].
- **transduction**: Each input event x_m has an associated label y_m [ZKR+17].
- **equivariance**: Changes to the order of the input instances must be reflected accordingly at the model's output [ZKR+17].

A family of neural networks that fulfills these properties is the *Transformer*.

4.1 Transformer & Efficient Attention-based Models

Transformer is a neural network architecture originally designed for Natural Language Processing (NLP) [VSP+17]. Transformers are capable of capturing global information among a set or sequence since all tokens are compared to each other in a process called self-attention. In contrast to RNNs, Transformers process the entire input at once. The vanilla transformer follows an encoder-decoder architecture. Essentially the architecture consists of several transformer blocks, where each uses an attention mechanism, normalization and linear layers. While the linear layers process each token independently and equally, information can flow between individual tokens due to the attention mechanism. The attention is calculated between every token and defines the most relevant other tokens for each token. For instance, Figure 4.1 illustrates the attention of the word token *it* among the other word tokens in the sentence. It shows that according to this attention the words *The animal* are more relevant to the word *it* than the others. Each transformer block embeds every token such that the embedding contains information about the token itself as well as a weighted combination of other relevant tokens.

For each attention unit [1], three different linear layers are applied to the input. Each linear layer has independent weights of dimension d and transforms a given token into one of three different embedded vectors query, key or value vector:

$$Q = XW_Q,\ K = XW_K,\ V = XW_V, \quad \text{with} \quad Q, K, V \in \mathbb{R}^{N \times d} \tag{4.1}$$

then the attention mechanism given the three matrices Queries (Q), Keys (K) and Values (V) is computed

$$Z = \text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \quad \text{with} \quad Z \in \mathbb{R}^{N \times d}. \tag{4.2}$$

[1] In this work we distinguish between *transformer block*, *attention unit* and *layer*. A *layer* represents any atomic operation, which is applied to the data in a neural network. An *attention unit* consists of all necessary layers that perform self- or cross-attention. Together with layer normalization, skip connections and fully-connected layers the attention unit forms a *transformer block*.

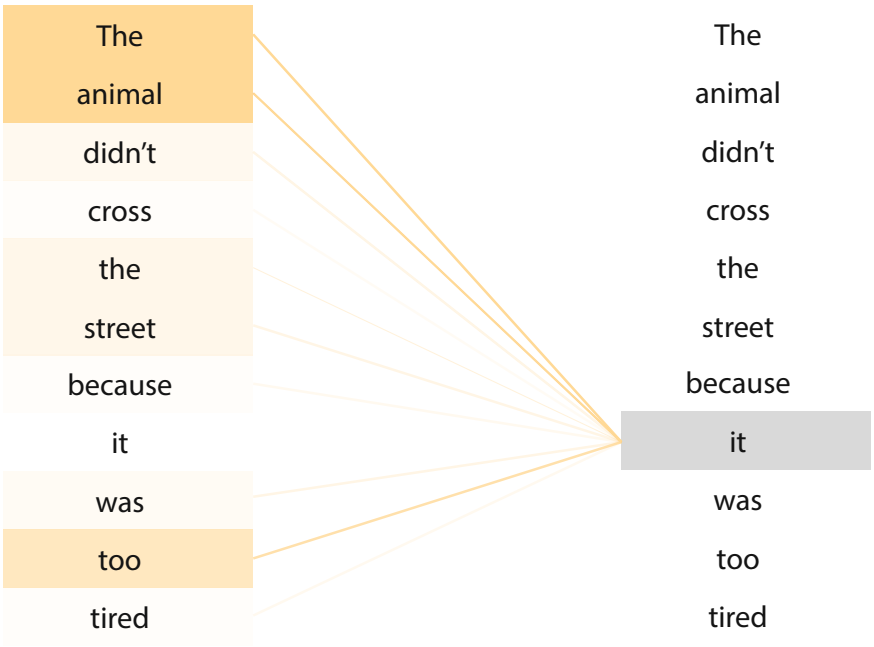

Figure 4.1: The self-attention weight of a given input sentence. Figure inspired by [Ala].

The attention between *queries* and *keys* are computed via the dot-product, scaled by $\sqrt{(d)}$ and normalized by the softmax function. The scaling term $\sqrt{(d)}$ is used to stabilize the model's gradients[VSP+17]. The computed attention values are used to weight the vectors of the value matrix V. Finally, a skip connection is introduced by adding the input X and Z before applying layer normalization [BKH16]:

$$X_A = \text{LayerNorm}(Z + X). \tag{4.3}$$

Similar to the different kernels of Convolutional Neural Network (CNN)s that are applied in parallel at the same layer, transformers utilize a concept called *multi-head attention* [VSP+17]. This means, instead of learning the set of linear attention weights (W_Q, W_K, W_V) for one attention unit, k such sets of weights are learned, where each set of weights is referred to as one attention head. Combining the computed values from the k different heads is achieved by concatenating the values and projecting it to the dimension of a single Z matrix using another linear layer with the weights W_O:

$$\text{MHAttn}(Q, K, V) = \text{concat}(Z_0, Z_1, ..., Z_k) \times W_O \tag{4.4}$$

$$\text{with } Z_i = \text{Attn}(Q_i, K_i, V_i). \tag{4.5}$$

All these described steps form one *transformer block* and can be summarized as:

$$X_A = \text{LayerNorm}(\text{MHAttn}(X, X) + X) \tag{4.6}$$

$$X_B = \text{LayerNorm}(\text{FNN}(X_A) + X_A). \tag{4.7}$$

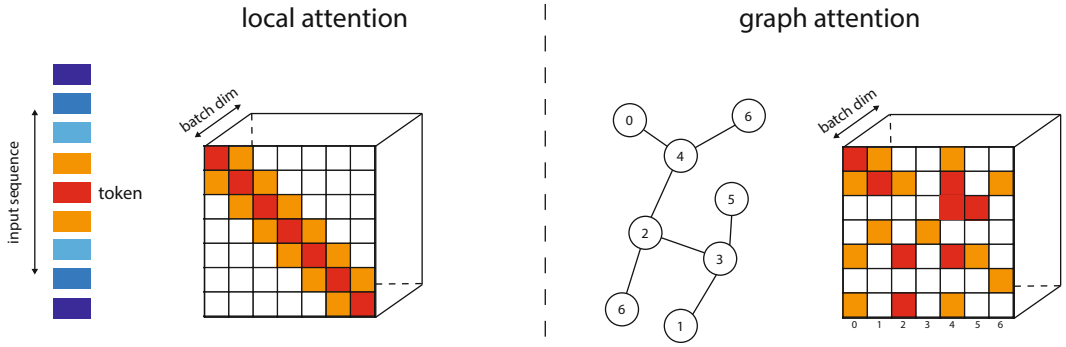

Figure 4.2: Efficient transformer variants often aim to reduce the quadratic complexity by sparsification of the attention matrix. For instance, by restricting the attention to local tokens in a sequence (left) or graph structure (right). The depicted illustration is inspired by [CC20].

In the case of self-attention Q, K, V are derived from the same input matrix X, while in the case of cross-attention the input matrix for Q and K,V differ.

The above description of transformers only considers the network architecture as operating on sets of input instances. Since all tokens are processed equally and the attention is computed between every token combination, transformers are inherently equivariant [ZKR+17]. However, for some tasks, such as language translation, the order of input instances reflects important information. To include the order information a positional encoding is added to the input. The positional encoding is a sequence of vectors that follow a specific pattern such as the sine and cosine functions of different frequencies [VSP+17]. The new resulting values of a token therefore not only depend on the initial token value but also on the position of the token in the input sequence. This allows the network to utilize sequencing information if necessary.

4.1.1 Overcoming quadratic complexity

Since all input tokens are compared to each other in the self-attention process, the transformer has a quadratic complexity in the input length for both memory and time $\mathcal{O}(N^2)$. While some tasks, such as sentence-to-sentence translation, do not require long input sequences, processing images, video or raw audio would become infeasible with pure self-attention. Due to this, several more efficient approaches exist that target to overcome the quadratic complexity of transformers. Efficient transformer variants often aim to reduce the quadratic complexity by sparsification of the attention matrix (see Figure 4.2). According to Tay et al. survey [TDBM20] the strategies to overcome quadratic complexity of self-attention in transformers can be divided into eight categories by their core techniques and primary use case. Whereas many models can be counted to more than one of the described categories. Here we discuss each category and introduce prominent examples for each.

- *Fixed Patterns*
  This category includes limiting the field of view to fixed, predefined patterns such as local windows and block patterns of fixed strides. BlockBert [QML+19] divides the input sequence into fixed-sized blocks of k tokens. Self-attention is only computed among a block. However, the position of the blocks changes to allow modeling long-sequence relationships. Image Transformer [PVU+18] restricts the attention to the local neighborhood and thereby introduces a similar bias as CNNs. The memory complexity depends on the chosen neighborhood size M as well as input size N: $\mathcal{O}(N \cdot M)$. Nevertheless restriction to local attention imposes limitations in tasks where global information is crucial.

- *Combination of (fixed) Patterns*
  Several approaches exist that combine two or more distinct access patterns e.g. strided and local attention. The underlying motivation is, while still reducing the memory complexity, aggregation and combination of multiple patterns the overall coverage of the self-attention mechanism improves [TDBM20]. Sparse Transformer [CGRS19] reduces the quadratic complexity by only computing the attention between a sparse number of token pairs. Half of the heads compute attention among a local neighborhood of tokens, while the other half of the heads compute the attention among strides of tokens. The computational complexity is reduced to $\mathcal{O}(N\sqrt{N})$. Longformer [BPC20] extends this concept by using heads with different amounts of dilation between the tokens considered for attention. Heads with none or little dilation focus on local context, while stronger dilation focuses more on long-range context.

- *Learnable Patterns*
  Approaches of this category aim to learn access-patterns in a data-driven fashion. Reformer [KKL20] addresses the transformer's quadratic complexity problem by utilizing Local Sensitify Hashing (LSH) attention. First, the authors show that there is no significant performance drop if the same linear layer is used to encode *Queries* and *Keys* such that $Q = K$ holds. Secondly, they argue only considering the query-key pairs that are similar to each other regarding dot-product gives a sufficient approximation of the attention since the softmax computation softmax(QK^T) is dominated by its largest components. LSH applied to these tokens allows obtaining buckets of similar tokens among which the attention is computed. The memory complexity of Reformer is $\mathcal{O}(Nlog(N))$. Sinkhorn Transformer [TBY+20] learns to sort the input sequence before computing local attention. This enables efficient quasi-global attention computation. For both models (Reformer & Sinkhorn Transformer), the similarity function is trained end-to-end jointly with the rest of the network [TDBM20].

- *Neural Memory*
  Instead of learning the access pattern, another category of approaches aims to improve efficiency by learning a side memory that can access multiple tokens at

once. The set transformer [LLK+19] architecture first introduced this concept with the usage of *inducing points*. To avoid the quadratic complexity the attention is not computed between each pair of input instances but rather between the input instances and a fixed set of learned vectors, the so-called *inducing points*. Information cannot flow directly from one token to another but by using the inducing points as proxies. First, the k inducing points serve as *Query* for cross-attention computation with the input instances as *Keys* and *Values*. The attention direction is swapped in the next step as the resulting k vectors from the previous attention-computation are then used as *Keys* and *Values* with the input instances as *Query*. The computational complexity is thereby reduced to $\mathcal{O}(kN)$. Perceiver [JGB+21] is similar to the set transformer as is alternates between cross-attention in both direction. But Perceiver, in contrast to set transformer, only uses one set of learned vectors in the first layer. The output of cross-attending these learned vectors in the first layer is then reused as input for the following layers.

- *Downsampling*
  Several models aim to tackle the quadratic complexity problem by reducing the resolution of the input sequence. The DEtection TRansformer (DETR) [CMS+20] is designed for object detection in images. The model reduces the sequence resolution first by applying convolution to the image. This typically reduces the image width and height by a factor 32. These obtained *superpixels* are then rearranged to a 1D sequence that serves as input to a vanilla self-attention based transformer. This sequence of superpixels is much shorter than a sequence of original pixels and therefore feasible for vanilla self-attention.

- *Low-Rank Methods*
  Approaches of this category reduce the computational costs by utilizing low-rank approximations of the self-attention matrix. For instance, Linformer [WLK+20] reduces the Values and Keys matrix via a linear projection among the sequence dimension. With Keys $K'(k \times d)$ and Queries $Q(N \times d)$ the attention matrix Softmax(QK') has a dimension of $N \times k$. When multiplied with Values $V'(k \times d)$ we obtain the usual $N \times d$ matrix as result. This trick allows reducing the memory complexity to $\mathcal{O}(n)$. Since the Linformer compress along the sequence dimension, it is not possible to hinder the model to mix past and future information when computing attention scores [TDBM20]. In addition, compressing along the sequence dimension violates the equivariance property of vanilla transformers. Since the weights of the transformations are applied to different positions in the sequence, reordering the tokens would affect the compressed representation.

- *Kernels*
  The idea behind the approaches of this category is to utilize kernels to avoid explicitly computing the $N \times N$ attention matrix. For instance, Performer [CLD+20] uses a *Generalized Attention mechanism* with random Kernels called FAVOR+ mechanism. In the original attention mechanism, it is not possible to decompose

the attention matrix Softmax(QK') after passing it into the nonlinear softmax function. Nevertheless, it is possible to decompose the attention matrix into a product of random nonlinear functions of Q and K. cosFormer [QSD+22] overcomes the quadratic complexity by linearization of self-attention. While the original transformer uses the non-decomposable similarity function $S(Q, K) = \exp(QK^T)$ the authors argue when using a decomposable similarity function such that $S(Q_i, K_j) = \phi(Q_i)\phi(K_j)^T$ we can exploit a matrix product property and compute $\phi(K)^T V$ before we multiple the result with $\phi(Q)$

$$(\phi(Q)\phi(K)^T)V = \phi(Q)(\phi(K)^T V), \tag{4.8}$$

which avoids the necessity of materializing the N^2-sized attention matrix $A = QK^T$. Since the softmax entails some essential properties that enable the transformer performance, Qin et al. aim to approximate the main properties of the softmax by applying ReLU [NH10] to ensure non-negativity and a cosine-based re-weighting mechanism that enforces locality. Figure 4.3 compare the matrix computations of the linearized self attention to the vanilla self attention.

- *Recurrence*
  Approaches in this category are united by the idea to process tokens blockwise but introduce some recurrent connections in order to enable information flow among the blocks. Transformer-XL [DYY+19] processes the input tokens in segments and utilizes the hidden states from previous segments to compute the hidden state of the current segments. The reused hidden states serve as a memory for the current segment, creating a recurrent connection between the segments.

4.1.2 Transformer for FCM

The transformer fulfills required properties for FCM data processing by neural networks: Firstly, it is *holistic* since it processes a whole sample at once thereby takes the whole sample into account when decided on one event's predicted class. We can view the set transformer as classifying each event on its own under the use of global sample-wide information aggregated by the attention mechanism. Secondly, it is *transductive* since we obtain a predicted class label for each event of the input sample.

However, the typical input size of $50 - 500 \times 10^5$ of an FCM sample makes the direct utilization of vanilla transformer infeasible. Moreover, many of the above described efficient transformer approaches are not applicable for input sets such as FCM data, since the events of an FCM sample have no ordering. For instance, the *Low-Rank Method* Linformer [WLK+20] utilized ordering to compress along the sequence dimension and can therefore not be used for sets. Similarly, the approaches of the *Downsampling* category that aim to reduce the resolution of the input sequence are not suitable for sets. Also models of the categories *Fixed Patterns* and *Combination of (fixed) Patterns*, such as BlockBert [QML+19] or Longformer [BPC20] are not suitable if they rely on computing

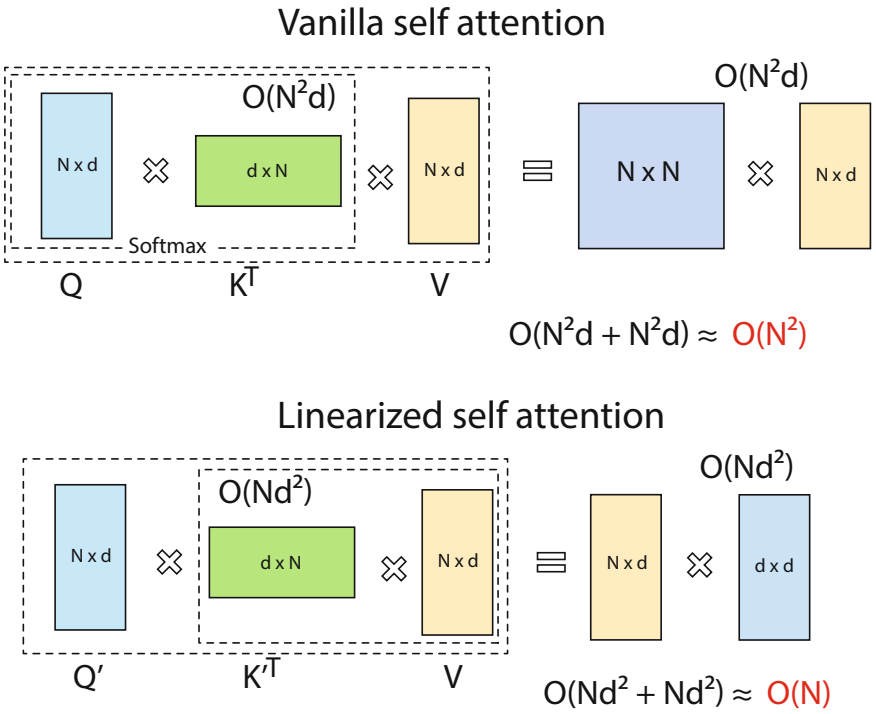

Figure 4.3: The difference between computing vanilla self attention and linearized self attention. By using a decomposable similarity function, we can multiple the matrix K and V before we incorporate the matrix Q, which prevents the materialization of a NxN-sized attention matrix and therefore reduced the computational footprint. The illustration is inspired by [QSD+22].

attention of nearby tokens in the sequence (tokens with nearby indices). Approximating attention via such locality constraints introduces the inductive bias that tokens near to each other in the sequence are more important for each other than tokens far away. All approaches of efficient transformer architectures that rely on this bias are not applicable to FCM data. For this reason also the discussed model Transformer XL [DYY+19] is not suitable for FCM data. Only models of the categories *Learnable Patterns* or *Neural Memory* or *Kernels* such as Reformer [KKL20], set transformer [LLK+19] or Performer [CLD+20] are suitable for FCM data.

As mentioned in Chapter 3, Wödlinger et al. [WRW+22] proposed a model based on the set transformer for blast cell classification of FCM data. This architecture, which uses inducing points to reduce the computational complexity, is also *equivariant* since all computations performed by the network are order invariant to the input set meaning that changes in the order of the input are reflected by corresponding changes of the output order. It therefore forms the basis upon which the proposed gate prediction model is built.

4.2 Gating Polygon Prediction for FCM

Since the proposed model gains explainability by predicting the polygons of the gating hierarchy, the objective function changes in comparison to direct cell classification approaches. While the model of Wödlinger et al. [WRW+22] produces predictions in the same shape as the ground truth labels $Y \in \{0,1\}^N$, the proposed approach must predict k polygons of t points $p \in \mathbb{R}^2$ each. The task of predicting a set of polygons from a sequence or set of tokens as input is related to the task of object detection. In objection detection a set of bounding boxes (and object classes) should be predicted from an input image, which can be viewed as a sequence of tokens. DETR [CMS+20] is a transformer based object detection model. It is the first architecture that managed to perform object detection without a post-processing step. Previous pipelines suffered from the problem of predicting multiple bounding boxes of the same object in an image. They required a post processing step called Non-Maximum Suppression [NVG06] to remove redundant bounding boxes. DETR overcomes this issue by utilizing self-attention among the latent representation of the bounding boxes. This flow of information allows the bounding boxes to be placed without redundancy. The proposed model model is fusion of DETR and set transformer to enable the prediction of polygons for FCM data.

4.2.1 The Model

The proposed method consists of a trained neural network that is based on the transformer architecture. The model expects a single FCM sample as input, i.e. a set of events $E \in \mathbb{R}^{N \times m}$. N defines the number of events $(50 - 500 \times 10^5)$ and m denotes the number of markers (typically $10 - 20$). The network's output are 7 polygons defined by $P = 20$ 2D points each. The polygons describe the gating hierarchy for MRD assessment in B-cell ALL data, which implies the cell's class membership.

Architecture

As depicted in Figure 4.4, the model's architecture follows an encoder-decoder schema as in [CMS+20]. A set transformer similar to Wödlinger et al. [WRW+22] is used for the encoder, consisting of two set transformer blocks. The decoder design is inspired the DETR model [CMS+20]: for each predicted polygon, four static object queries are learned. The object queries are applied to the encoder's output via cross-attention, which is followed by a self-attention layer. Each element of the 7-element long decoder output set is passed through a two-layer fully connected neural network called the prediction head. The resulting 20 2D points per element are used as gate polygon for each of the 7 gates in the ALL gating hierarchy. I empirically evaluated that 20 points are most suitable for the given task. More than 20 points only slightly increase the performance (max 1% median F1-Score) while drastically increasing the network size (see Table 5.4 in Result Chapter 5).

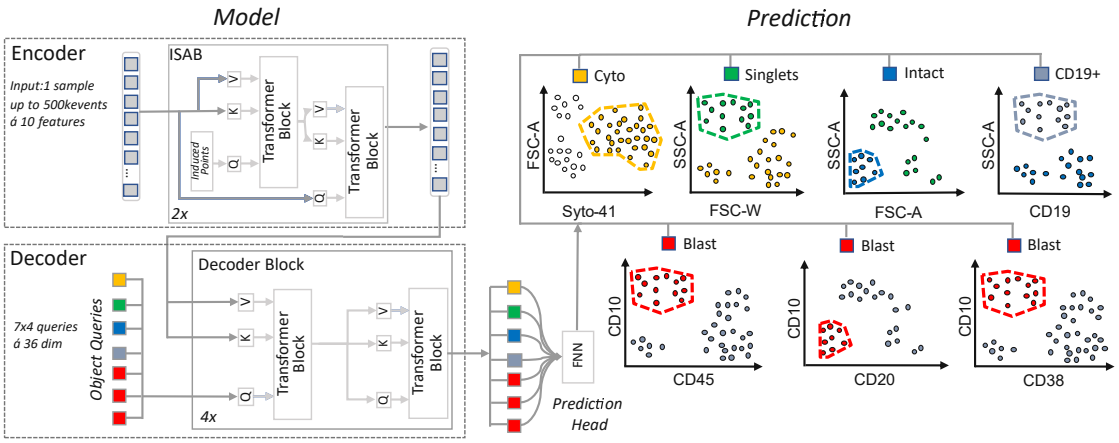

Figure 4.4: The network architecture consists of the encoder, decoder, prediction head and the resulting polygons that form the gating hierarchy for a given input FCM sample.

4.3 Preprocessing

The operator-annotated polygons comprise two issues regarding their usage as ground truth for training:

1. Polygons are typically only roughly estimated, with borders often far away from the nearest events inside the polygon. While this does not affect the effectiveness of the procedure during clinical routine, it introduces a source of ambiguity in the gating process by perturbing the relationship between polygon position and data points.

2. For different FCM samples different feature combinations for some of the plots in the gating hierarchy were used by the operator since different operators may use slightly different strategies to track down blast events. However, the model predicts the polygons for a statically predefined set of 2D plot feature combinations. The selected set reflects the most common feature combinations for each gate in the given datasets.

I address both issues by computing the convex hull of all events inside the polygon during preprocessing for each gate. The resulting hull serves as adapted training ground truth, which can be created for any required combination of 2D plot features while tightly enclosing the events inside.

However, this solution gives rise to another issue: Some events that are inside an operator-drawn polygon can be far apart from the selected cluster when projecting them on a 2D plot using a different combination of features. The resulting convex hull is unnecessarily widespread, which violates the relation between polygon and data cluster position and hence may include unwanted events. This issue is resolved by excluding outlier events

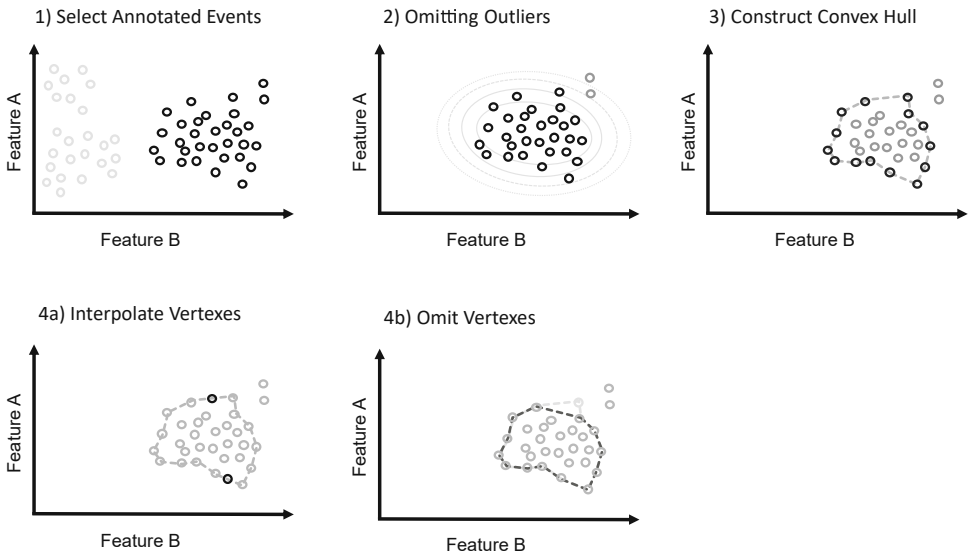

Figure 4.5: The four preprocessing steps to construct the polygons are used as training ground truth.

from the hull computation measured by the Mahalanobis distance [Mah36]. Assuming a normal distribution, events are excluded using a Chi-squared test with two degrees of freedom at a significance level of 1×10^{-5}. Figure 4.5 illustrates the four preprocessing steps to construct the ground truth polygons. Depending on the number of vertexes resulting from the convex hull construction, vertexes are either inserted or removed in the fourth step. For both, inserting or omitting vertexes, the position is evenly selected, such that no accumulation of vertexes on one side of the polygon appears.

4.4 Training

$$\mathcal{L}_{poly}(\hat{p}, p) = \sum_{i}^{P} \|\hat{p}_{\hat{\sigma}(i)}, p_i\|_1 \text{ with } \hat{\sigma} = \underset{\sigma \in \mathcal{S}_P}{\operatorname{argmin}} \sum_{i}^{P} \|p_i, \hat{p}_i\|_1 \tag{4.9}$$

The model is trained in a supervised manner. Since the number of polygon vertices differs from sample to sample in the ground truth but is fixed to $P = 20$ for the model prediction, we artificially insert or remove points in the ground truth polygons to obtain P points. Equation 4.9 states the loss for a predicted polygon $\hat{p}$ where $\hat{\sigma} \in \mathcal{S}_P$ defines a permutation of the polygon points such that every predicted point is matched to one corresponding ground truth point using the Hungarian method [Kuh55]. The distance between two points is calculated via L1 norm. Similar to [CMS+20, ARCC+19] I experienced, an auxiliary loss benefits the model convergence. The auxiliary loss performs the same

computation as the main loss but after each intermediate layer the following intermediate layers are skipped.

4.4.1 Regularization

The high capacity of deep learning models make them prone to overfitting, especially on small train datasets. To still facilitate generalization to real-world tasks deep learning heavily relies on regularization techniques [GBC16a]. Goodfellow et al. [GBC16b] defines regularization as *any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error*. Besides regularization techniques that alternate the model (e.g. dropout [SHK+14] or convolutional networks [LB+95]), or alternate the optimization process (e.g. weight decay or early stopping [GBC16b]), there also exists regularization techniques that alternate the training data. While data augmentation refers to the alternation of existing data (e.g. horizontal flip of an image), data synthesis describes drawing artificial data from a generation process. In fact the border between data augmentation and synthetic data generation is smooth since the more information the synthesis process entails about the data to more it is closer to alternation of existing data.

I employ both data augmentation and pretraining on synthetic data to address the low number of training samples (e.g.: ≤ 60 for the *BUE* dataset), to overcome inter-laboratory differences and to facilitate learning the relationship between polygon and cell cluster position.

Data Augmentation

Four different data augmentation steps are applied to the FCM samples during training: For all events and polygons random linear translations of randomly selected features are applied. For randomly selected gates linear scaling (stretching and squeezing in relation to the center), linear translation and shearing of polygons and their corresponding events are used. Figure 4.6 displays all four different data augmentation operations. The following equation defines the event and polygon scaling data augmentation:

$$\hat{x} = (x - c) \cdot (1 \pm s) + c \tag{4.10}$$

with $c = \min(x) + \frac{\Delta x}{2}$, where $\Delta x = \max(x) - \min(x)$ and $s \sim \mathcal{U}(0, 0.3)$

Data Synthesis

Another branch of options to overcome the problem of limited training data is the utilization of synthetic data. The core idea is to generate data by simulating processes, or sampling from distributions. Often the generation method allows to directly access label information, in contrast to the otherwise expensively human-crafted labels. For instance, when rendering scenes in 3D engines, one can obtain pixel-exact labels for panoptic segmentation from the 3D engine [Nik21]. A common problem of training on synthetic

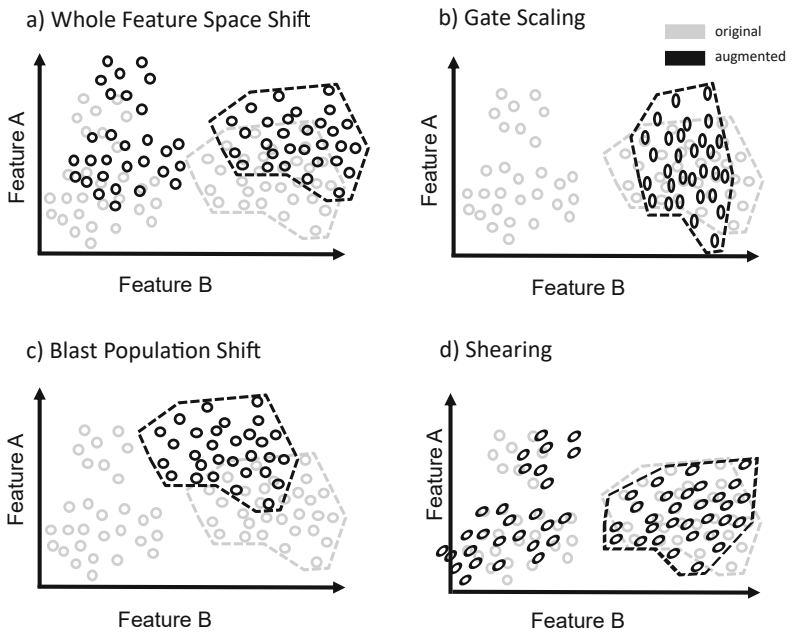

Figure 4.6: The different augmentation steps applied to an FCM sample: a) Random linear shifts of the whole feature space. b) Scaling of the blast population's shape. c) Random linear shifts of the blast events. d) Shearing of gates and events along single features.

data is domain transfer [Nik21]. Since the synthetic data often does not perfectly reflect the real-world data, a model purely trained on synthetic data can fail to generalize on the test set. Therefore, when using synthetic data, models can be pre-trained on the synthetic data and then fine-tuned on the limited training data.

As discussed later in the Result Chapter 5, when training on the small dataset *BLN* or *BUE* the model tends to overfit and fails to generalize on the validation set. Observing that although the position of the cell clusters were often captured by the predicted polygons, they failed to form the shapes to entail all cells in those clusters, leads to the hypothesis that some of those shapes are never presented to the model during training when using the small datasets. This motivates the usage of synthetic data. The idea is to utilize synthetic data to first train the model to predict polygons that surround all kind of different shaped clusters. After that, the model is fine-tuned on the real FCM training data, where it now mainly learns which of the presented clusters to select in order to track down the blast cells. Since the goal of the synthetic data training is to present the model differently shaped clusters, I keep the process simple by employing gaussian distribution to sample synthetic FCM cell data. More sophisticated synthesizing processes, such as multi modal distributions may reduce the domain gap to the real FCM data, but also bare the risk of restricting the model to made assumptions about FCM data. The synthesis process consists of sampling 3×10^5 vectors from a gaussian distribution $x_{syn} \sim N(\mu, \Sigma)$ where μ and Σ are sampled from a uniform distribution

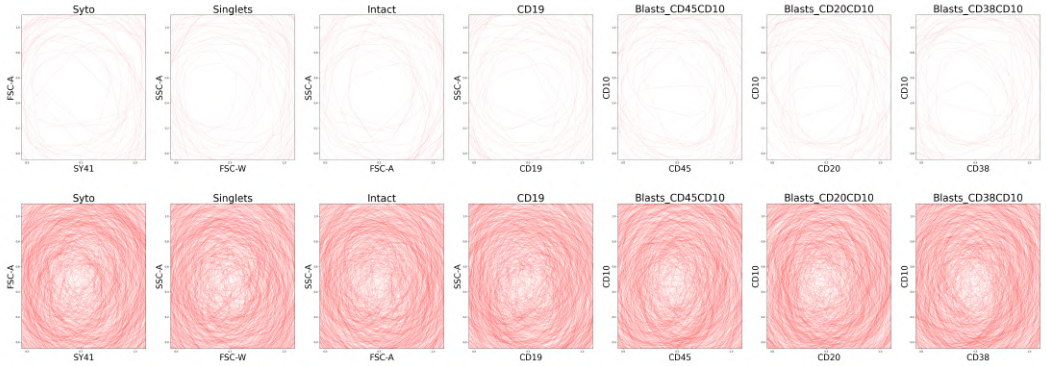

Figure 4.7: Polygon gates of 50 (first row) and 500 (second row) samples of synthetically generated data.

for each synthetic sample. In addition, I apply the data augmentation pipeline to the synthetic data. Similar to the real FCM data I compute convex hulls of the synthetic data for the same 7 2D projections as the real FCM data (as described in Section 5.1). Figure 4.7 pictures the polygon gates of 50 (first row) and 500 (second row) synthetic samples.

4.5 Explainability Visualization

To answer the third research question I investigated several common explainability techniques for deep learning models and tailored them for FCM data. First the techniques are examined for a cell classification transformer as presented in [WRW+22]. Then they are applied to the polygon prediction model of this thesis. Figure 4.8 provides an overview on the investigated explainability visualization techniques.

4.5.1 Attention Visualization

We are interested in visualizing the attention mechanism of the transformer model. As defined in Equation 4.1, for self-attention the matrix of input tokens X is linearly projected via the learned weight matrices W_Q, W_K, W_V to obtain three matrices Queries (Q), Keys (K) and Values (V). Then, the dot-product attention with Softmax normalization, as defined in Equation 4.2 is the dot-product between Q and K, scaled by the square root of the dimension of K $\sqrt{(d)}$ and normalized by the Softmax function. Usually, the result of the Softmax-normalized dot-product QK^T is used to visualize the attention. For instance, to visualize the attention score between the i^{th} input token and all other tokens, the result of $Q_i K^T$ can be plotted as a heatmap or weighted graph. For cross-attention, Q originates from different input matrices than K and V. Carion et al. [CMS+20] visualized the cross-attention between the learned object queries and the input image as overlay heatmap on the input image, which reveals which parts of the image are relevant for the model to locate a particular bounding box as well as to predict the corresponding object

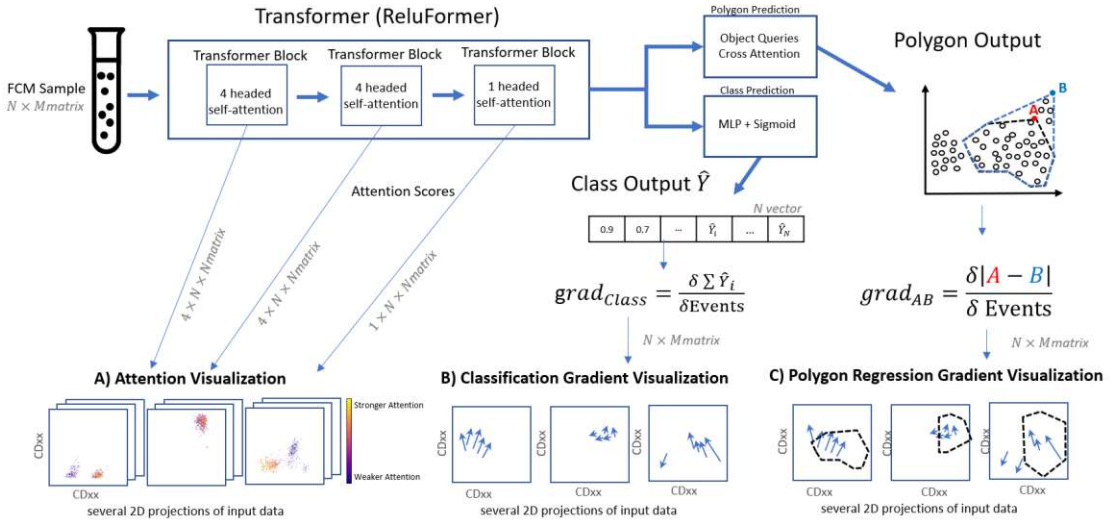

Figure 4.8: The three visualization techniques: A) The attention scores are retrieved from the intermediate layers of the model. I plot the top 500 events with strongest attention scores in different 2D projections of the data. B) The gradients of the summed event-wise class predictions with respect to the input data is computed. I plot the vectors of the top 100 gradients are displayed, which point in the direction of fastest change of the predicted class. C) The gradients of the distance between the predicted polygon to a desired target polygon with respect to the input data is computed. I depict the vectors of the top 100 gradients, which indicate the direction the input should change to minimize the different between predicted and target polygon.

class. With multi-head attention the different heads can learn to attend to different parts of the data. Common attention visualization tools either allow to switch between different heads in different layers of the model [Vig19] or visualize the last layer's attention [ADT+22, CMS+20].

Attention Visualization for FCM Data

To visualize transformer attention in FCM data, the possibility of obtaining attention scores between individual events of an FCM sample is needed. However, the set transformer does not directly compute the attention between the individual tokens, it calculates the attention between tokens and the k prototype vectors and vice versa. Thus, in this work, we utilize another efficient transformer called **cosFormer** [QSD+22], which allows us to directly compute attention between individual events while still having a feasible complexity, as described in Subsection 4.1. cosFormer overcomes the quadratic complexity by linearization of self-attention and applying a cosine re-weighting. However, the re-weighting mechanism assumes that the input forms a sequence, which is not given for the set of events forming an FCM sample, we, therefore, omit this part of the cosFormer. In the following, we refer to this simplified version of the cosFormer as **ReluFormer**.

To obtain attention scores between any events of interest we can simply compute the attention matrix $A_{viz} = ReLU(\check{Q})ReLU(\check{K})^T$, where $\check{Q}$ and $\check{K}$ represents the queries and keys for the selected events respectively and $\check{A}_{viz}$ represents the row-normalized matrix. When computing the self-attention between all events of an FCM sample, we obtain these interactions as a N^2-sized matrix. In the following, we present three ways to visualize the information entailed in this matrix as well as one way to visualize cross-attention between learned object queries and the input events.

Single Event Attention 2D plots A common way to inspect FCM data is by plotting several 2D projections of the high-dimensional data. In clinical practice, this emerged as a common standard to document FCM data and its analysis as experienced clinicians can consistently spot different biological phenoms in these plots. It, therefore, seems natural to visualize the attention of a single event to all other events by coloring the attention score as heatmap in these 2D plots. This visualization is comparable to the self-attention heatmap of individual pixels to the whole images as demonstrated in[CMS+20].

Aggregated Attention 2D plots Focusing on the attention of a single event can provide too much detail and thereby miss to depict more sample-wide data relations. Often we are interested in inspecting the attention of a particular biologically reasonable sub-population. To do so, we can aggregate the attention of all cancer cells to all other events in an FCM sample and again visualize the attention scores as colors on several 2D plots. By aggregating the attention of a group of events, we can gain insight into how the model handles these events. However, this approach is still limited because it does not reflect the attention of all events in the sample. Additionally, the aggregation process may cancel out some of the effects of individual events.

UMAP-based full Attention plots A possible way to exploit the whole attention matrix is to use Uniform Manifold Approximation and Projection (UMAP) [MHM18], a graph-based non-linear dimensionality reduction method. UMAP creates a graph based on the distances between data points in a high-dimensional space and tries to find a lower-dimensional representation of the data points, such that the graph in the embedded space is similar to the original graph in high dimension. By using the following function as a distance measure between events a and b in UMAP

$$\mathrm{d}(a,b) = \frac{1}{\mathrm{attn}(a,b)}, \tag{4.11}$$

we obtain a visualization in which events with strong attention among each other are clustered together and events that hardly attend to each other are pushed apart.

Object Queries Cross-Attention While approaches mentioned above are all tailored for self-attention among events aiming to visualize the N^2-sized self-attention matrix, it is also possible to visualize the cross-attention of k learned object queries entailed in a $k \times N$-sized matrix. Similar to [CMS+20] we obtain attention scores per object query and

input token. Each object query corresponds to a specific predicted gate. It is therefore reasonable to display the attention by coloring the events in the gate's corresponding 2D projects of the input data. It thereby shows which events have been focused on in order to predict the polygon in a particular 2D projection.

4.5.2 Gradient Fields

During the training of deep neural networks, gradients of the loss $\mathcal{L}$ are computed with respect to the model weights:

$$\text{grad}_W = \frac{\partial \mathcal{L}}{\partial W} \tag{4.12}$$

The gradients point in the direction of the steepest ascent, such that a gradient descent based optimizer takes a step in the opposing direction aiming to reduce the loss [GBC16c, GBC16d]. In contrast, gradient-based explainability methods usually compute the gradients of a particular output value with respect to the input data [SVZ13] or intermediate representation [SCD+17]. For instance, the gradients of $\hat{Y}_i$, the binary classification output of the i^{th} event of the network T, with respect to the input events E is defined by

$$\text{grad}_{E_i} = \frac{\partial T(E)_i}{\partial E} = \frac{\partial \hat{Y}_i}{\partial E} \tag{4.13}$$

Although the gradients are only computed of the i^{th} classification output with respect to the input sample, we obtain a gradient vector for each event since the class prediction of one event depends not only on the position of this event in hyperspace but also on all other events of that sample. Gradient-based explainability methods for images such as Saliency Map take the norm of gradients of each input pixel and visualize them as a heatmap on top of the original image. A high gradient norm indicates that these pixels have a strong impact on the prediction of the corresponding class. For FCM data we can use different 2D projections of one sample to plot the obtained gradients, similar as described above for attention visualization. Either we use the gradient norms to highlight the most influential events or we use the gradient vectors themselves plotted as vector fields on the 2D plots. But in general, the latter is preferable as it not only indicates which input events are important for a particular prediction but also reveals the direction in hyperspace that leads to the greatest change in the prediction. In [STK+17] Smilkov et al. proposed **SmoothGrad**, an extension of the standard Saliency Map that aims to reduce visual noise in Saliency Maps. SmoothGrad creates a visualization by averaging the gradients of multiple noised versions of the same input image and thereby smooths out local permutations. I observed that the same procedure leads to more paralleled gradients in FCM samples. I, therefore, add noise $s \sim \mathcal{U}(0, 0.1)$ to the events and compute gradients. This action is repeated 10 times (empirically determined) and the average of the gradients is plotted.

While the above-described gradient-based explainability visualizations are solely for classification tasks, the same concept can be applied to various downstream tasks such

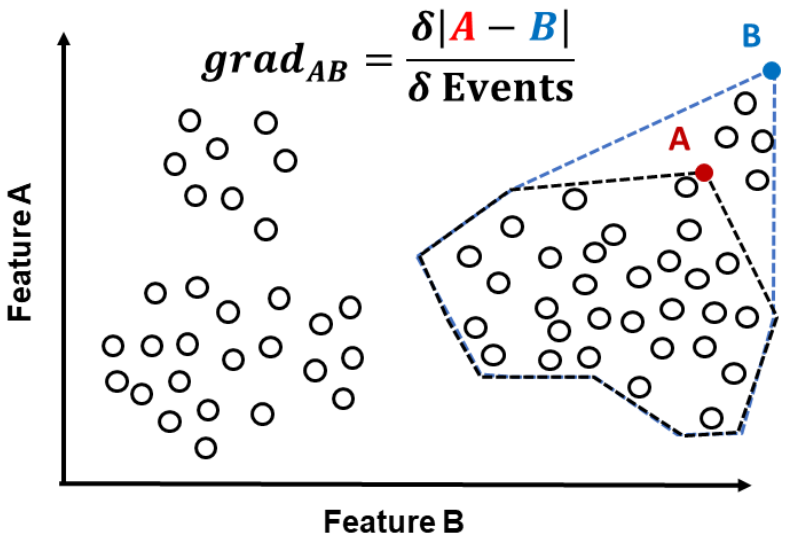

Figure 4.9: Illustration of the gradient computation for polygon regression. The gradients are computed of the difference between predicted point A and desired target point B with respect to the input events.

as polygon regression. If we define P as a polygon regression model, A as the i^{th} vertex of a predicted polygon and B as the actual desired location of that point, then we can calculate the gradients of the norm of A - B with respect to the input events E

$$grad_{AB} = \frac{\partial \|P(E)_i - B\|}{\partial E} = \frac{\partial \|A - B\|}{\partial E}, \tag{4.14}$$

which describes the direction in which the events should shift in order to move the predicted point A to the location of B as illustrated in Figure 4.9.

CHAPTER 5

Results

This chapter starts with introducing the operated data and then dives into the evaluation of the experiment results as well as answering the stated research questions. Finally the conclusion summaries the contribution and insights of this thesis. Figure 5.1 provides an insight of the polygons predicted by the model in comparison to the ground truth polygons.

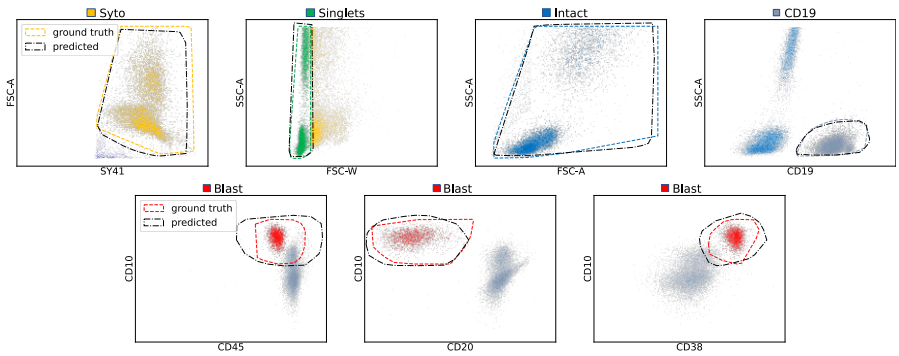

Figure 5.1: Example of a gating hierarchy for a single FCM sample. The gates are drawn in the 2D projection of the FCM data, which is obtained by plotting the expression levels of two markers against each other. The predicted gates are drawn in black.

5.1 Exploratory Analysis & Preprocessing

While Section 4.3 described the utilized preprocessing steps this Section introduces the used data as well as the outcomes of apply the preprocessing to this data.

Table 5.1: Due to missing intermediate or blast gates, not all samples as in [WRW+22] could be used in this work. This table compares the number of used samples per dataset to [WRW+22]. In Table 5.3 the same samples were used to evaluate all 3 methods.

Dataset	# **Transformer** [WRW+22]	# **Proposed**
VIE14	200	186
VIE20	319	291
BLN	72	70
BUE	65	60

5.1.1 Datasets

The proposed model is evaluated on four different datasets collected across three distinct institutions, measured on three different FCM devices, consisting of over 600 samples in total. From all four datasets, the three datasets VIE14, BLN, BUE are publicly available[1]. All samples have been obtained from the bone marrow of pediatric B-ALL patients on day 15 after induction therapy. The following markers are used in the experiments as they are shared upon all samples: CD10, CD19, CD20, CD34, CD38, CD45 and Syto41 as well as FSC-A, FSC-W and SSC-A. For a detailed dataset description, the reader is referred to [RDS+19] for VIE14, BLN and BUE, and to [WRW+22] for VIE20. Table 5.1 states how many samples per dataset are used for the proposed approach.

5.1.2 Gating Hierarchy

Since the utilized datasets originate from different institutions, different strategies were applied to execute manual gating. This results incompatible 2D projections used for the ground truth polygons. As described in Section 4.3 preprocessing was applied to calculate the same gating hierarchy for all samples. Table 5.2 displays the predicted gates and the used markers. In Figure 5.2 the polygons of all used FCM samples are displayed. We can see that the first three Gates (Syto, Singlets and Intact) depict more consistent shapes among the datasets than the Blast Gates. This is especially true for the Singlets. The Figures also reveals shifts from one dataset to another. For instance, the Singlet Gates in *BLN* dataset are on a lower FSC-W position than in the *BUE* dataset. It is also noticeable that *VIE14* and *VIE20* have more variety in the Blast Gates than the *BLN* and *BUE*. While the Blast Gates in CD45/CD10 of *BLN* are dominated by shapes, does *VIE14* contain a lot of horizontal Gates. I believe that those differences show possible challenges when generalizing from dataset to another.

5.2 Evaluation Setup

The same experiments as in [WRW+22] have been conducted. In all experiments the proposed model's ability to generalize to new unseen FCM samples (in most cases from

[1] flowrepository.org

Table 5.2: The gates and their used features of the predicted gating hierarchy

Name	Syto	Singlets	Intact	CD19	Blast-A	Blast-B	Blast-C
Marker iy-Axis	FSC-A	SSC-A	SSC-A	SSC-A	CD10	CD10	CD10
Marker x-Axis	Syto41	FSC-W	FSC-A	CD19	CD45	CD20	CD38

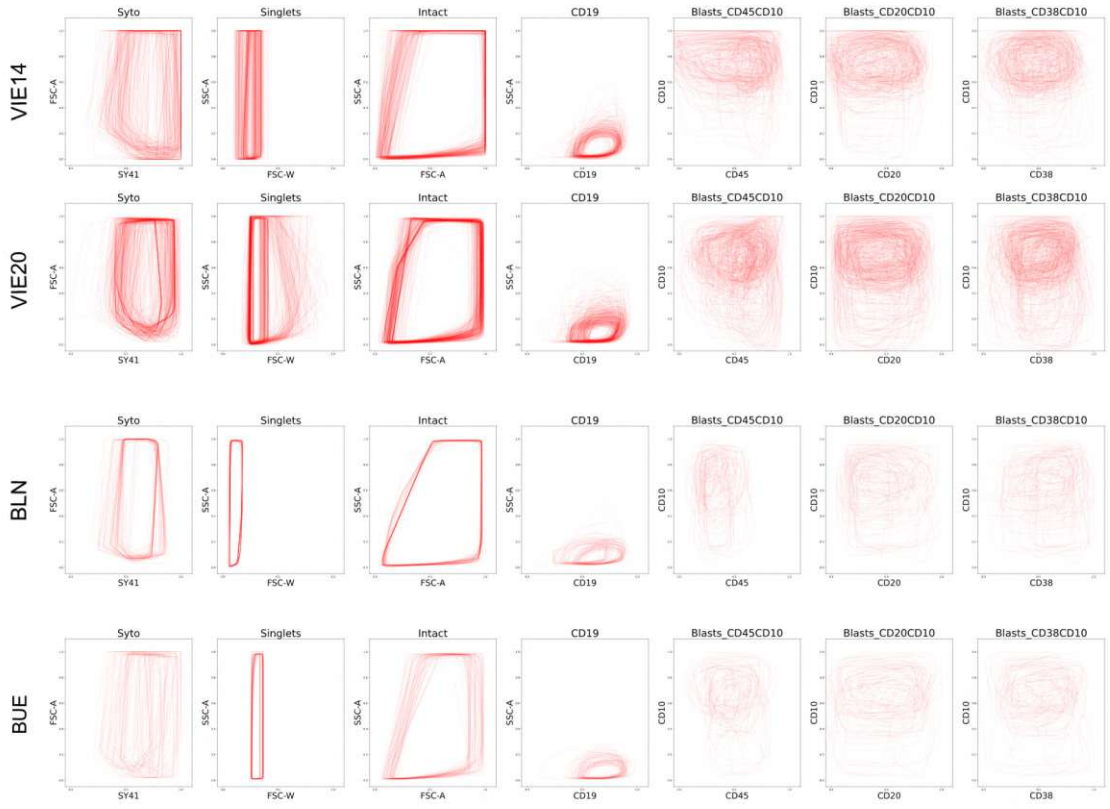

Figure 5.2: The ground truth polygons constructed from convex data hulls. Each row shows the Gates of one dataset. Each column shows one of the 7 Gates of the used ALL gating hierarchy. This plot highlights the cell clusters shifts among the different laboratories.

Table 5.3: Experiment results of the proposed method compared to GMM [RDS+19] and set transformer [WRW+22]. The table reports mean F1-Score / median F1-Score.

Train	Test	GMM	Transformer	Proposed
VIE14	BLN	0.72/0.81	0.77/**0.90**	0.79/0.88
	BUE	0.75/0.90	0.82/**0.95**	0.78/0.89
	VIE20	0.77/0.90	0.80/**0.91**	0.78/0.87
VIE20	BLN	0.53/0.58	0.68/0.83	0.73/**0.85**
	BUE	0.74/0.88	0.75/0.88	0.82/**0.92**
	VIE14	0.80/0.91 T	0.84/**0.93**	0.73/0.88
BLN	BUE	0.65/0.76	0.66/**0.87**	0.69/0.84
	VIE14	0.48/0.48	0.82/**0.92**	0.58/0.73
	VIE20	0.53/0.60	0.82/**0.91**	0.50/0.55
BUE	BLN	0.62/0.73	0.64/**0.78**	0.57/0.69
	VIE14	0.66/0.73	0.83/**0.92**	0.62/0.69
	VIE20	0.67/0.78	0.79/**0.90**	0.65/0.75

different institutes) is tested. The model is implemented in Pytorch 1.10 [PGM+19] and trained using the Adam optimizer with a batch size of 12 and a learning rate of 1×10^{-3}. It consists of 32892 parameters and has been trained on a NVIDIA Gefore RTX 2080 Ti. One model forward pass takes $\approx 400ms$ on the used GPU and $\approx 3000ms$ on an Intel i7-10750H CPU. Details about the training setup can be found in the provided code on GitHub[2].

Table 5.3 displays the results compared to [RDS+19] and [WRW+22]. For each experiment the cell classification performance (blast cell vs. non-blast cell) of each sample is summarized with the mean and median F1-Score of all samples in the corresponding test set. The results show that the proposed model is able to reach state-of-the-art performance for blast identification tested on data across different institutes. However, the model under-performs on small training datasets such as *BLN* and *BUE* with 70 and 60 training samples. In these cases, the model overfitted during training and was not able to generalize well onto new samples from different sources: Qualitatively inspections revealed that while the cluster positions were mostly correctly predicted, the model failed to predict the correct form of unseen polygon shapes.

5.3 Ablation Study

This section summarize the results of the conducted ablation study.

[2]Github Repository

Table 5.4: Median F1-Score of the artificially generated convex hull polygons compared to the operator ground-truth for different polygon lengths.

Dataset	5	10	20	30	40	60
VIE14	72.02	92.65	94.81	94.98	95.07	95.38
VIE20	67.90	92.57	92.96	93.35	93.51	94.07
BLN	61.38	88.97	90.35	90.41	90.79	91.18
BUE	72.27	96.75	97.54	97.46	97.71	97.97

5.3.1 Optimal Polygonsize

As described in the Preprocessing Section 4.3, when constructing the ground truth polygons vertexes are either inserted or omitted to meet the proposed model's fixed number of predicted points per polygon. A higher number of vertexes can unnecessarily increase the model's capacity and memory footprint. On the otherside, a low number of vertexes can result in performance reduction since the vertex omission changed the polygon shape too drastically. In order to find the optimal number of vertexes that the model should predict, we compared the classification output of ground truth polygons with a different number of vertexes against the classification output of the original operator-defined ground truth. Table 5.4 displays the F1-Score for a different number of vertexes and datasets. Based on these results we have chosen a polygon size of 20 vertexes. As the table shows, a higher number than 20 does not substantially increase the classification performance: While the difference in F1-Score between 5 vertexes and 20 vertexes is at least 20%, it is at most 1% between 20 vertexes and 60 vertexes. The thereby obtained results can be seen as the upper bound for the performance of the proposed model. No matter how well the model learned to predict the polygons, its performance will not surpass the performance difference between the operator polygons and the constructed polygons.

5.3.2 Number of Object Queries

An important parameter of the proposed model is the number of learned object queries. For DETR the number of object queries have an influences of the prediction performance, since the number represents the upper limit of possible detections. While DETR uses the learned object queries to represent the individual predicted bounding boxes in image object detection, the proposed models uses the object queries for the individual predicted polygons. In the latter case,the number of object queries mainly influence the model's capacity. The model can either learn one object query per polygon or divide the points of the polygon among different object queries, such that one object query corresponds to e.g. 4 points of the predicted polygon. Figure 5.3 plots the average validation F1-Score during 600 epochs of training. The Figure shows, that more object queries per polygon lead to better performance on the validation set. However, increasing the number of object queries also impacts the model's memory foot print. I, therefore, opt for 5 objects-queries although it under performs compared to 10 on the first 600 epochs.

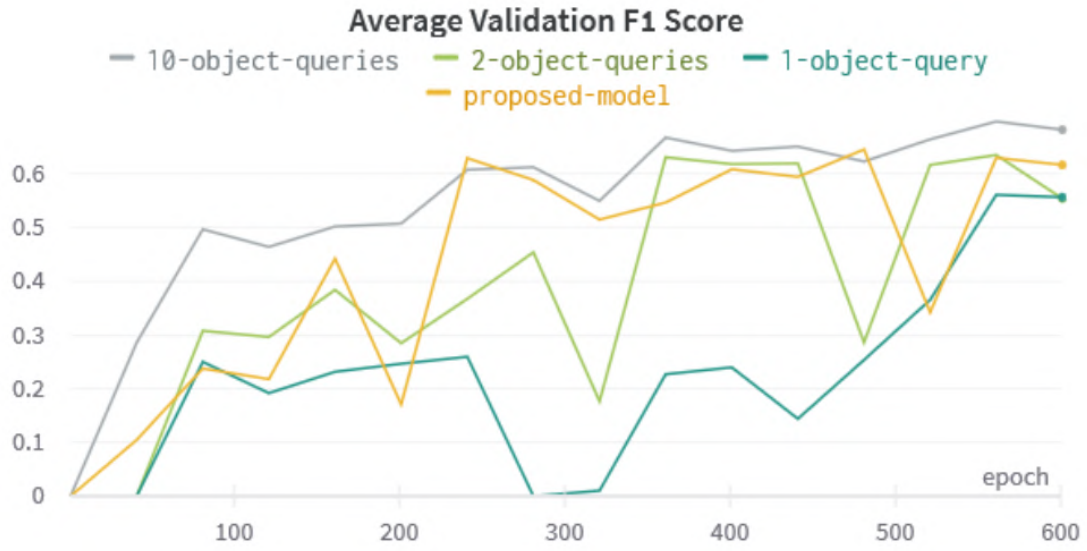

Figure 5.3: 10 Objects queries per polygon lead to best performance on the validation set compared to using 5 (proposed model), 2 or 1 object queries.

5.3.3 Number of Layers

I investigated the generalization performance of the proposed model for different number of layers. Specifically, if more encoder layers or more decoder layers are beneficial. Figure 5.4 shows the average F1-Score of the validation set over 800 epochs of training. Although different settings overlap multiple times during these 800 epochs, 3 encoder layers seem more beneficial than 3 decoder layers.

5.3.4 Data Augmentation

Figure 5.5 depicts the average F1-Score on the validation set during 600 epochs of training. The plot shows that augmentation improves the generalization capabilities, since it outperforms the same model without augmentation. Scale augmentation seems to be the most effective augmentation, although additional scale and shear augmentation still improve the performance.

5.4 Pretraining with Synthetic FCM data

For pretraining a synthetic dataset of 2000 samples has been generated using the process described in Section 4.4.1. Two pretrained models were obtained one after 50 epochs and one after 150 epochs of synthetic data training. Figure 5.7 displays that synthetic pretraining already shows an improvement after the first epoch. In addition, using pretraining speed up the model convergence. In Figure 5.6 the training loss using different amount of pretraining is depicted. The Figure shows, that the loss faster

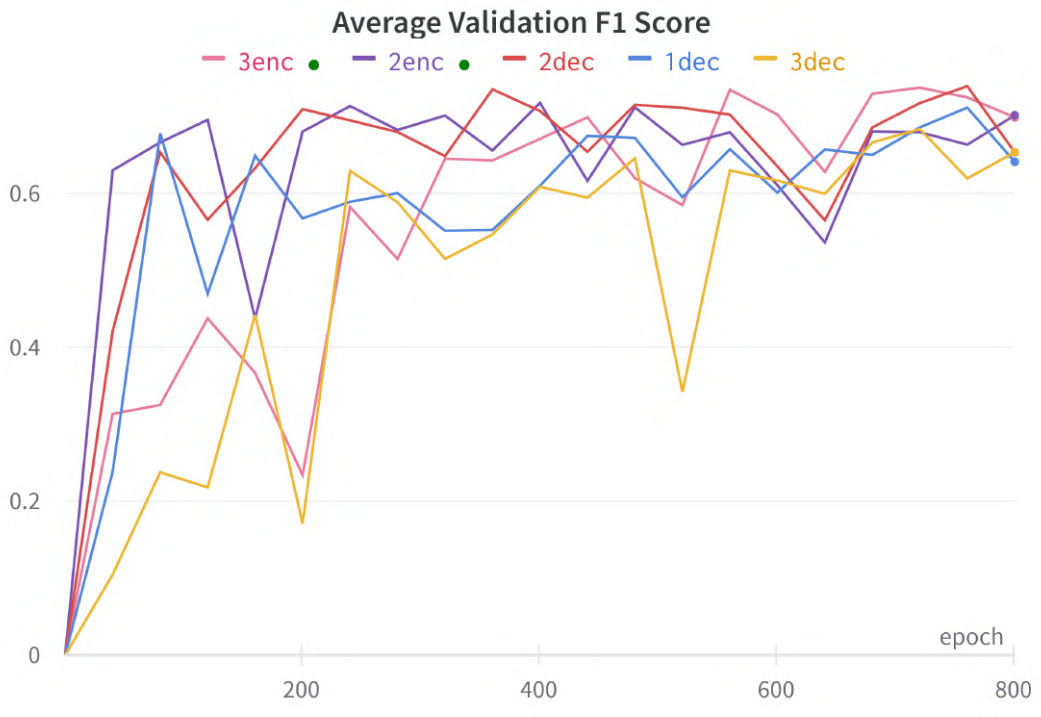

Figure 5.4: Average Validation F1 Score over 800 epochs of training. 3 encoder layers seem more useful than 3 decoder layers.

converge to its minimum as well as ends up at a smaller value after 300 epochs, compared to without pretraining. However, the synthetic pretraining could not avoid the failed generalization from *BLN* to *VIE14*.

5.5 Visualization Techniques

In this section, we demonstrate the proposed visualization techniques. For that purpose we use the ReluFormer trained on direct cell classification B-ALL FCM samples, as we as the proposed polygon regression model. We observed that the ReluFormer performs on the same data similar as in [WRW+22], while allowing to compute attention scores between events for visualization purposes directly. First, we show how the visualizations help to reveal different aspects of the model's decision process. Then we use the visualization techniques to analyze failure cases, FCM samples for which the model failed to predict the cancer cells correctly.

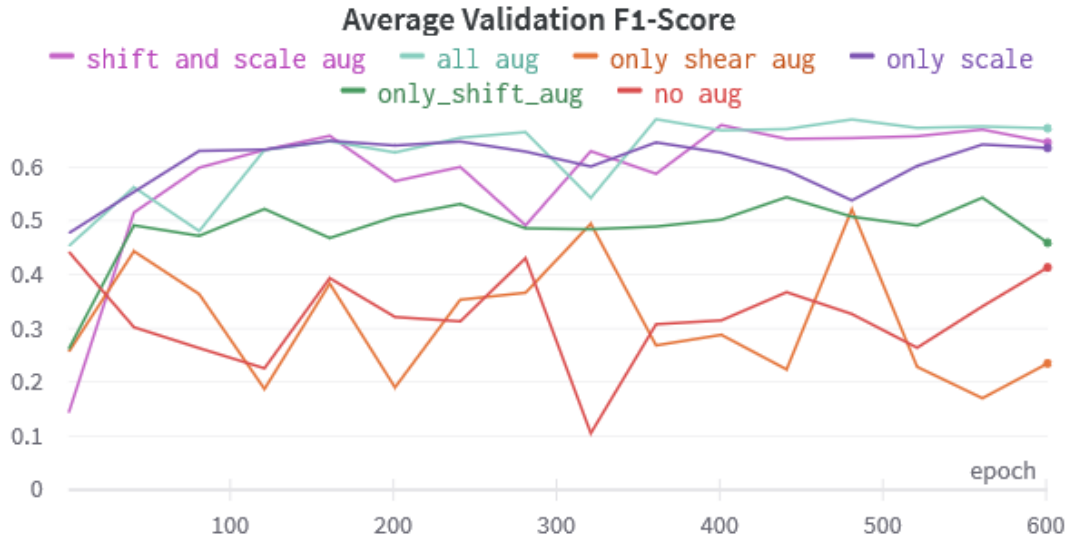

Figure 5.5: Using all described augmentation strategies leads to the best average F1-Score on the validation set. The Figure shows the average F1-Score on the validation set on 600 epochs during training. Only shear augmentation and no augmentation performs worst, while applying all augmentation methods performs best.

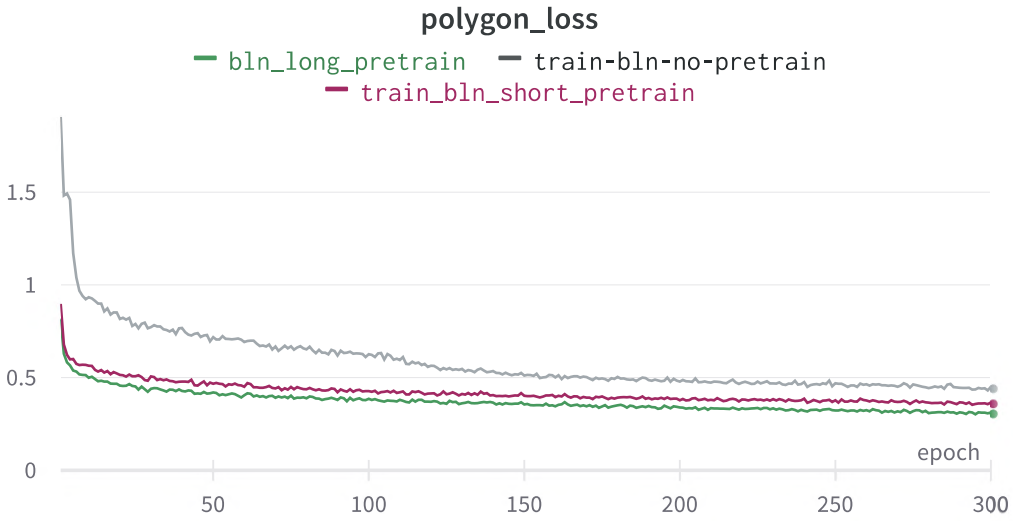

Figure 5.6: The training loss on the BLN dataset without pretraining (gray), with 50 epochs of pretraining (magenta), with 150 epochs of pretraining (green).

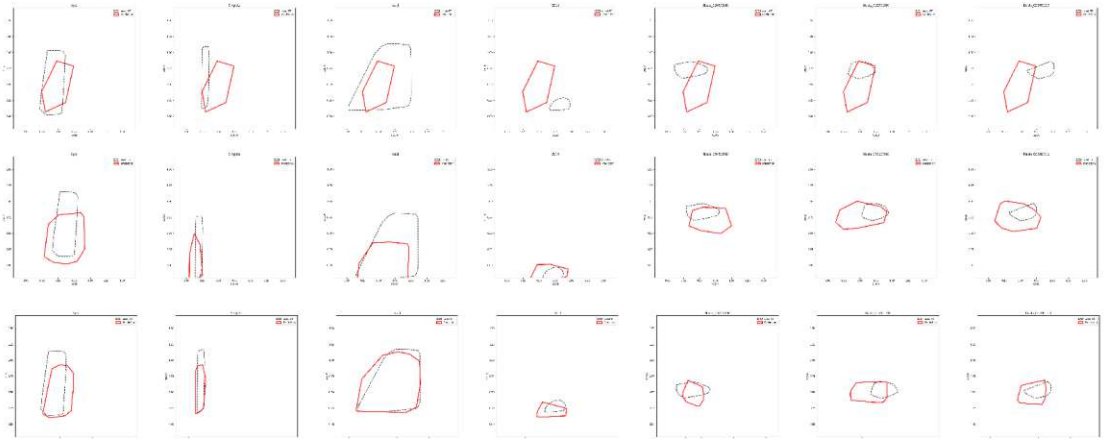

Figure 5.7: Synthetic pretraining already shows an improvement after the first epoch. The Figure shows the predicted ground truth and predicted polygons for one FCM simple using three different training regimes. The first row represents a model without any pretraining of synthetic data. In the second row displays the prediction of a model that was pretrained for 50 epochs on synthetic data. While the last row depicts the polygons of a model pretrained on 150 epochs of synthetic data.

5.5.1 Attention Visualization

Cell Classification Figure 5.8 visualizes to which events 1000 randomly sampled cancer cells (blasts) of an arbitrarily selected B-ALL sample attend the most. Although the network is trained solely for binary classification (cancer cells vs. non-cancer cells), we can see, that the heads focus on different biologically meaningful populations such as CD19- CD45- (likely erythroblasts) or CD19+ CD45+ (likely healthy B-cells). The right-most column pictures the UMAP embeddings using the attention weights of the corresponding head as a distance measure between the events. While the aggregation of 1000 randomly sampled blasts only reveals the attention from the perspective of the cancer cells, the UMAP accounts for the interaction of all events. In the first row, the blasts mainly attend to themselves. Similarly, in the UMAP plot the blast cells form a cluster among themselves, while most other cells form another distinct cluster. For the other heads, the blasts attend to different populations and the UMAP shows more mixed interactions between blast and non-blast cells. The fact that the model could identify meaningful biological structures in the data without being explicitly told to do so suggests that it has a deep understanding of the data modality and is not simply relying on biased shortcuts for learning.

To quantitatively support this observation I evaluated the amount of blast cells among the top attention cells. For each head the top 5% of cells according to the aggregated blast attention are calculated. Then the amount of blast cells among these are computed:

$$\frac{\mathrm{n}_{\mathrm{blasts}}@\mathrm{top5\%}}{min(\mathrm{n}_{\mathrm{cells}}@\mathrm{top5\%}, n_{blasts})} \tag{5.1}$$

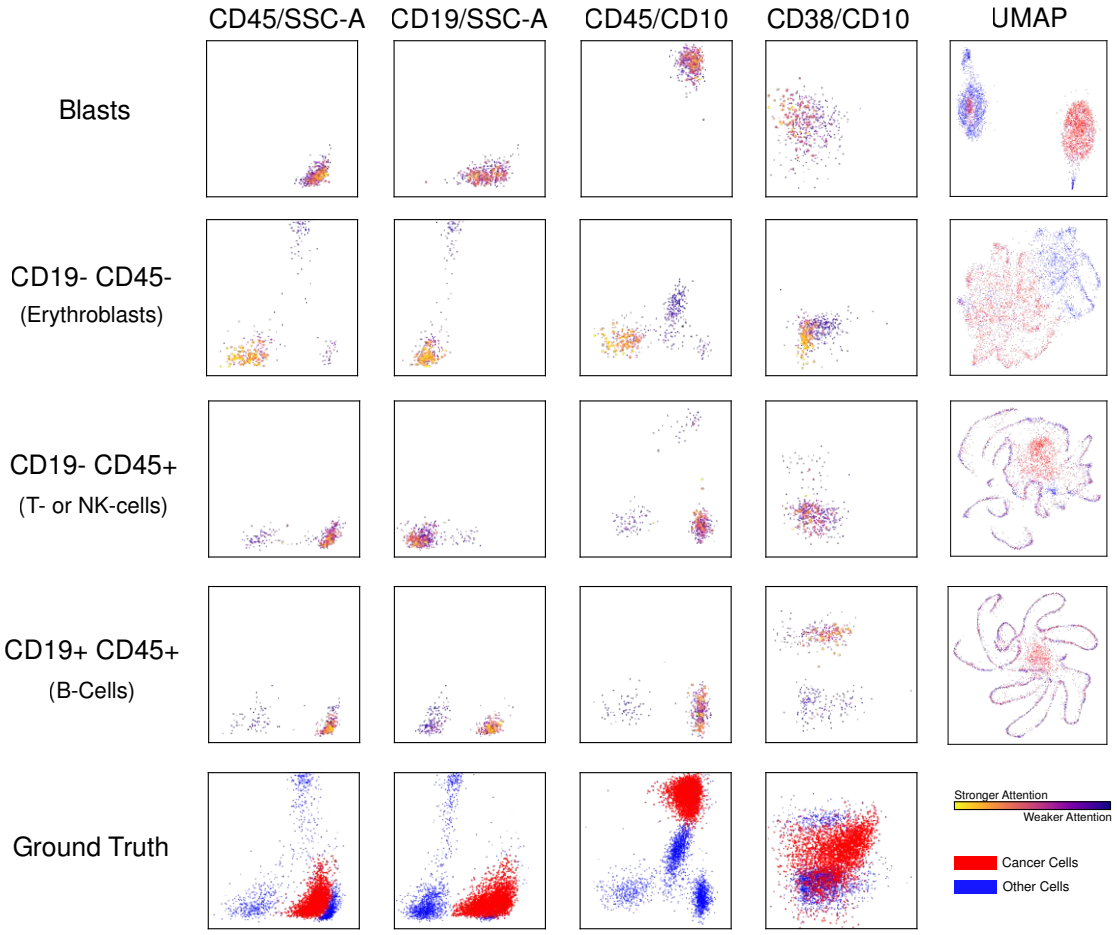

Figure 5.8: Different attention heads find biologically meaningful populations in the data, when trained on supervised binary classification. The first four rows show the attention of four different heads for one B-ALL FCM sample. Each column displays a different 2D projection of the data. For each row we visualize the 500 events with the strongest attention. Color codes the attention strength for rows and columns 1-4 and the class-membership cancer cells (red), and other cells (blue) in the remaining plots.

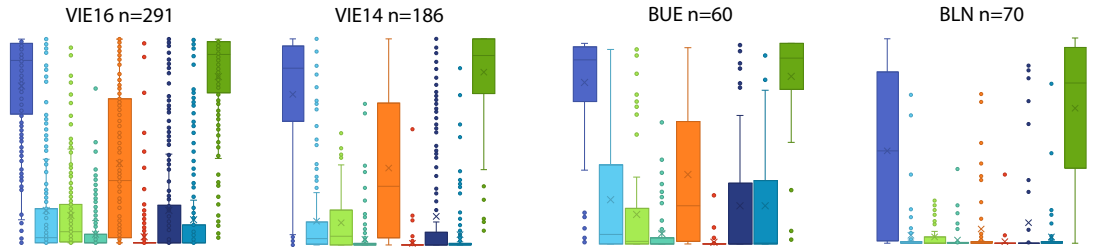

Figure 5.9: Head 1 (blue) and Head 9 (green) show higher amount of cancer cells then the other heads across all three datasets.

This metric evaluates to 0 if no blast cells are among the top5% and it returns 1 if either all blasts cells are among the top5% or if the top5% solely consists of blasts. This switch is needed in case that the number of blast cells in the sample is higher then the number of cells forming the top5%. Figure 5.9 shows boxplots for this metric computed on *VIE14*, *BLN* and *BUE*. Head 1 (blue) and Head 9 (green) show higher amount of cancer cells then the other heads across all three datasets.

Polygon Prediction Figure 5.10 visualizes to which input events the learned object queries attend the most. The first row shows all input events as well as the predicted and ground truth polygons for each gate. The other rows display the 500 events to which the object queries for each gate attend the most. We can see that different heads attend to different regions inside a polygon. For example consider the Syto Gate: Head 4 focus on events on the right bottom half of the Gate, while Head 5 and Head 6 attend to different locations on the left bottom half. Surprisingly, for the three Blast Gates many heads focus on events outside the Blast polygon. This could indicate that the model learned detect the Blast population

5.5.2 Gradients Visualization

Figure 5.11 depicts two cases in which the gradients of the difference between the prediction polygon (black) and a slightly shifted **target polygon** (blue) with respect to the input data are visualized. We plot the top 100 biggest gradients and color code them by their length as well. The gradients indicate in which direction which input events should change in order to cause the model to shift its prediction to the blue target polygon. In Figure 5.11 A the gradients are mainly defined by the events inside the predicted polygon and point in the direction of the polygon shift. It thereby verifies that the model learned the correct relationship between the position of specific events and the gate polygon, since a shift of the polygon is mainly caused by a shift of the events inside the polygon in the same direction. However, similar to the Saliency Map for images, the visualization is not noise-free as gradients of a few events outside the polygon are among the top 100 biggest gradients and point in an arbitrary direction.

Figure 5.11 B shows the shifts of the **Singlets Gate** for one B-ALL FCM sample.

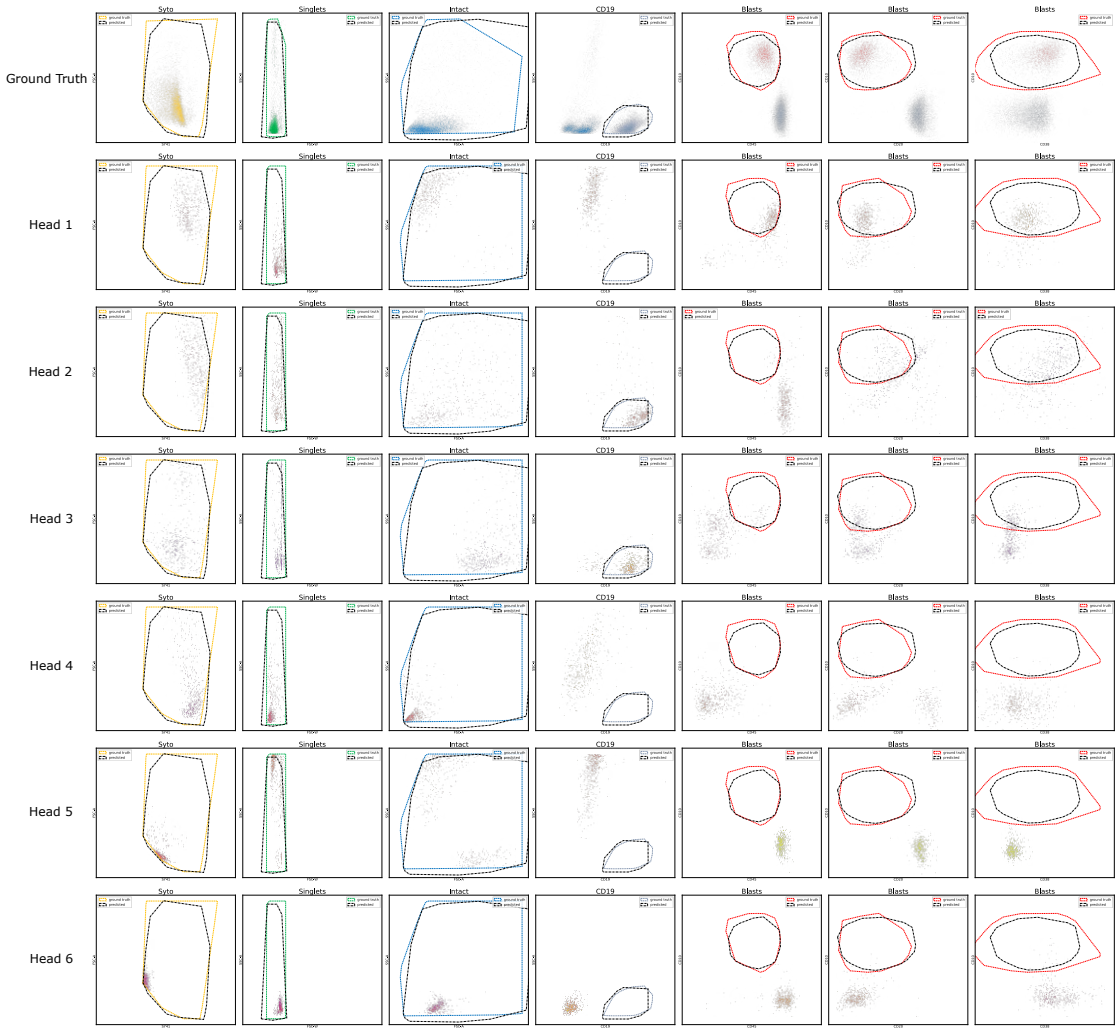

Figure 5.10: Different heads attend to different locations among the input events. For the task of polygon regression, the first row depicts the ground truth and the other rows show the attention of six different heads for one B-ALL FCM sample. Each column displays a different 2D projection of the data corresponding to different Gates. For each row we visualize the 500 events with the strongest attention. Color codes the attention strength.

Although the predicted polygon position is correct, the gradients suggest that the model has learned an incorrect relationship between the event positions and the polygon position. The figure shows that when the polygon is shifted in any direction, the gradients are mainly at the bottom left corner of the event cluster, rather than being distributed evenly among all events within the polygon. This indicates that the model has learned to rely on the position of the bottom corner to predict the polygon, since the Singlets Gate mainly differs in position, not shape, among different samples in the dataset (see Figure 5.2 for a visualization of different Singlets Gates). However, this learned shortcut is problematic, as it may cause the model to miss samples with events located outside of typical polygon shapes.

5.6 Answering Research Questions

In this section the results are observed from the lens of the different research questions.

5.6.1 *What are the main building blocks that enable to predict FCM gating hierarchies with set transformers?*

This research question aims to cover the main architectural design choices of the proposed model. The model is built upon the set transformer [WRW+22] and DETR [CMS+20]. While DETR uses learned objects queries to represent the individual predicted bounding boxes in image object detection, the proposed models uses the object queries for the individual predicted polygons. The model can either learn exactly one object query per polygon or divide the points in the polygon among different object queries, such that one object query corresponds to e.g. 4 points in the polygon. I experimented with different amount of object queries per polygons. The

One necessity to enable the prediction of gating hierarchies is the Hungarian matching based polygon loss, as described in Chapter 4. The utilization of other losses, such as Mean Squared Error (MSE) did not lead to convergence of the model. My assumption is that the requirement of MSE to compute the loss between points of the same index (e.g. between the second point in the predicted polygon and the second point in the ground truth polygon) is contradictory for learning.

5.6.2 *Which synthetic data generation strategies and which data augmentation strategies are beneficial for the given task?*

To answer this question I investigated the effect of different data augmentations, linear shifts of the whole feature space, scaling of individual cell clusters, linear shifts of individual cell clusters and shearing of the data from different 2D projections (as illustrated in Figure 4.6). Without data augmentation the model fails to generalize from the trained dataset. All the presented data augmentation methods improved the model's capability to generalize, but scaling individual cell populations generated the greatest gain in performance on the validation set. Shearing should the least performance improvement.

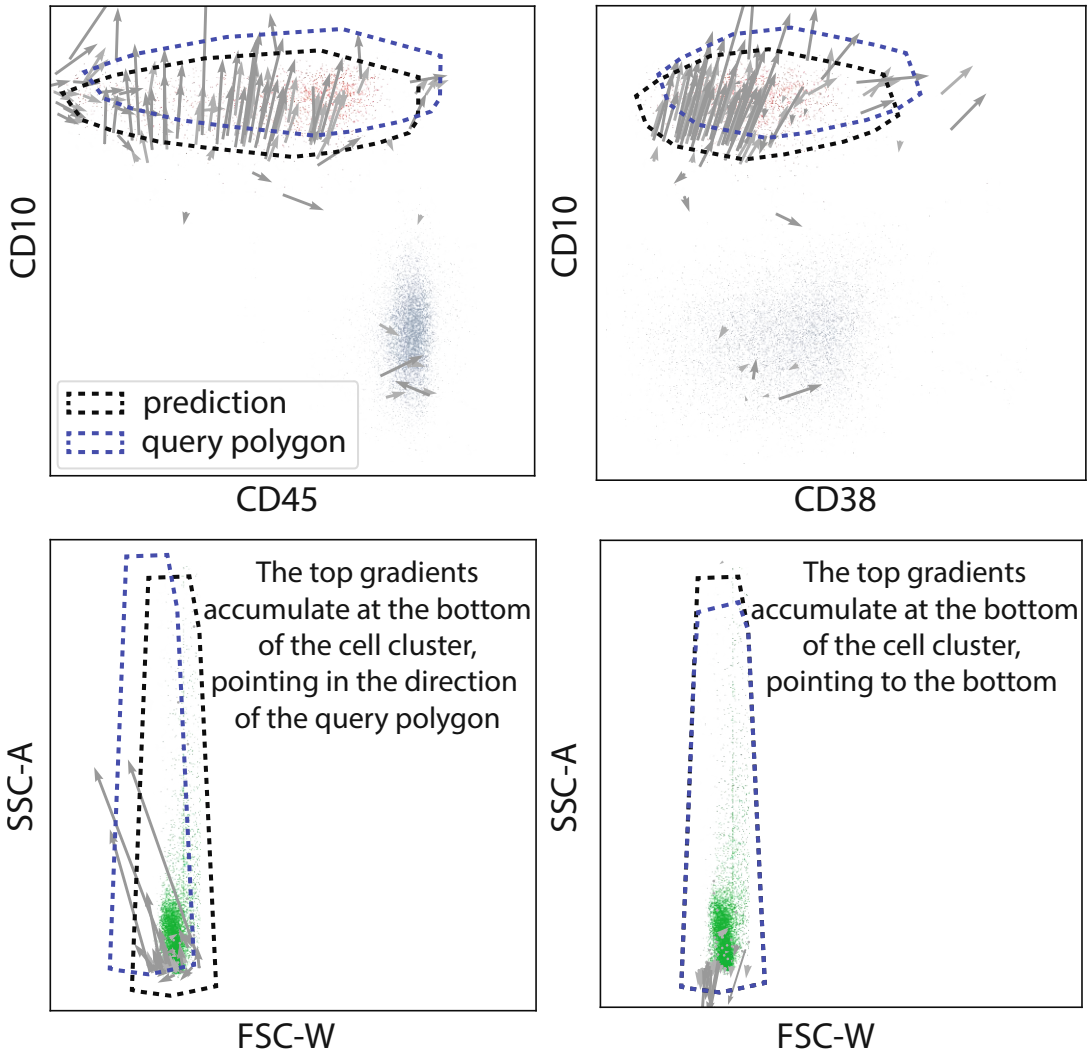

Figure 5.11: Gradients for Polygon regression can be used to confirm correctly learned relationships (first row) and to spot overfitting behavior (second row). We plot the top 100 gradients of the difference between predicted polygon and query polygon.

Synthetic data generation speed up the model's convergence, but it did not improve the model performance. Maybe more advanced synthetic data generation processes could improve on the performance and enable generalization when trained on the small datasets.

5.6.3 *Which visualization techniques ease explainability of the proposed model's decision process?*

To answer this research question I investigated the application of two common explainability techniques *attention visualization* and *gradient-based visualization* to the field of FCM:

Attention visualization proved to be especially useful, when analyzing self-attention in the direct cell classification setting. It revealed that the model learned to attend to different biologically meaningful subpopulations in the data, although it was solely trained on binary blast cell classification. This phenomenon not only emerged in the selected qualitative analyzed samples, quantitative evaluations could demonstrate that the same two heads consistently show higher attention for blast cells than non-blast cells over samples in all datasets (*VIE14, BLN, BUE*).

Gradient-based visualizations were usefully deployed to analyze the proposed polygon prediction model. Forming artificial target polygons allows to query the model how certain predictions would change. Computing the gradients of the difference between the predicted polygon and the artificial target polygon enables to question certain behaviors of the models. For instance, an artificial target polygon created from linear shifts of the predicted polygons could demonstrate that the model learned the true relationship between cell cluster position and predicted polygon. Also the gradient visualization revealed the overfitting behavior of the model when predicting the position of the *Singlets* Gates. Even when the artificial target polygon only differs from the predicted polygon in height, the top gradients are accumulated at bottom instead as expected on top where both polygons differ. The application of gradient-based visualizations to FCM also confirmed that averaging the repeated noised execution of gradient computation leads to smoother visualizations similar as proposed in [STK+17].

5.7 Conclusion

In this thesis I propose a novel transformer-based approach for blast cell detection in FCM samples of ALL patients. The model visually reveals which cells it identifies as blast cells by predicting the polygons of the gating hierarchy for a given FCM sample. This imitates the construction of a gating hierarchy by a human expert in clinical practice and therefore explains why certain events are detected as blast cells. While the proposed model fails to generalize well when trained on small datasets ($\leq$ 70 samples), its performance is comparable to non-explainable state-of-the-art approaches on more populated datasets ($\geq$ 180 samples). Since the model mimics the decision process of domain experts, it is more suitable to be included in the clinical gating routines as direct cell classification models.

The proposed model is designed and evaluated for pediatric ALL, but the underlying concept could be applied to any disease for which standardized FCM gating hierarchies exist.

The proposed method gives rise to more explainability in the context automated FCM analysis. Especially in combination with the gradient visualization for polygon regression in-depth insights about the model's decision process can be obtained. The investigated attention visualization helps to understand on which cell population the model focuses. For instance, the fact that the model can identify meaningful biological structures in the data without being explicitly told to do so suggests that it has a deep understanding of the data modality and is not simply relying on biased shortcuts for learning. The proposed gradient-based visualization demonstrated how to inspect the model's learned relationship between input cell data and predicted polygons.

Future work could utilize the proposed FCM-tailored interpretability techniques to introduce an inductive bias by imposing gradient-based regularization term in model training similar to [MPR22] or [SLDV98]. This inductive bias in combination with more sophisticated data synthesis strategies could overcome the problem of generalization when trained on small datasets.

Summarized, this thesis demonstrates how explainability can be achieved solely by reframing the task's objective. It serves as an example that self-explainable models are an alternative to other explainability methods, not only in theory, but also in practice.

List of Figures

List of Tables

Acronyms

AL Acute Leukemia. 16

ALL Acute Lymphoblastic Leukemia. 14–17, 33, 57, 59

AML Acute Myeloid Leukaemia. 14, 16, 17, 59

BM bone marrow. 12, 15

CD Cluster of Differentiation. 7

CLL Chronic lymphocytic leukemia. 8

CNN Convolutional Neural Network. 27, 29

DETR Detection Transformer. 21, 47, 55

FCM FlowCytometry. 3, 5, 9, 10, 16, 17, 21, 25, 33, 38, 55, 57, 58

FITC Fluorescein Isothiocyanate. 8, 9, 59

FSC Forward Scattered. 6

HSC haematopoietic stem cells. 12

LSH Local Sensitify Hashing. 29

MRD Measurable Residual Disease. 5, 15–17, 25, 33

MSE Mean Squared Error. 55

NADPH Nicotinamide adenine dinucleotide phosphate. 8

NLP Natural Language Processing. 26

PCR Polymerase Chain Reaction. 16, 17

PE Phycoerythrin. 8, 9, 59

PI propidium iodide. 7

R-CNN Region-Based Convolutional Neural Networks. 21

RNN Recurrent Neural Network. 25, 26

SSC Side Scattered. 6, 13

TdT Terminal deoxynucleotidyl Transferase. 7

YOLO You Only Look Once. 21

Bibliography

[ABV+20] Julia Amann, Alessandro Blasimme, Effy Vayena, Dietmar Frey, and Vince I Madai. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC Medical Informatics and Decision Making*, 20(1):1–9, 2020.

[AC17] Eirini Arvaniti and Manfred Claassen. Sensitive detection of rare disease-associated cell subsets via representation learning. *Nature Communications*, 8(14825):2041–1723, 2017.

[ADT+22] Estelle Aflalo, Meng Du, Shao-Yen Tseng, Yongfei Liu, Chenfei Wu, Nan Duan, and Vasudev Lal. Vl-interpret: An interactive visualization tool for interpreting vision-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21406–21415, 2022.

[AGBD21] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021.

[Ala] Jay Alammar. The illustrated transformer.

[ARCC+19] Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3159–3166, 2019.

[ASK+18] Nima Aghaeepour, Erin F Simonds, David J H F Knapp, Robert V Bruggner, Karen Sachs, Anthony Culos, Pier Federico Gherardini, Nikolay Samusik, Gabriela K Fragiadakis, Sean C Bendall, Brice Gaudilliere, Martin S Angst, Connie J Eaves, William A Weiss, Wendy J Fantl, and Garry P Nolan. GateFinder: projection-based gating strategy optimization for flow and mass cytometry. *Bioinformatics*, 34(23):4131–4133, 05 2018.

[ASLA20] Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. A diagnostic study of explainability techniques for text classification. *arXiv preprint arXiv:2009.13295*, 2020.

[AvUH+19] Tamim Abdelaal, Vincent van Unen, Thomas Höllt, Frits Koning, Marcel J.T. Reinders, and Ahmed Mahfouz. Predicting cell populations in single cell mass cytometry data. *Cytometry Part A*, 95(7):769–781, 2019.

[BBL+02] BJ Bain, D Barnett, D Linch, E Matutes, and JT Reilly. Revised guideline on immunophenotyping in acute leukaemias and chronic lymphoproliferative disorders. *Clinical & Laboratory Haematology*, 24(1):1–13, 2002.

[BBS+18] Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Identifying and controlling important neurons in neural machine translation. *arXiv preprint arXiv:1811.01157*, 2018.

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[BNRC12] Sean C Bendall, Garry P Nolan, Mario Roederer, and Pratip K Chattopadhyay. A deep profiler's guide to cytometry. *Trends in immunology*, 33(7):323–332, 2012.

[BPC20] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[BR] Bio-Rad. www.bio-rad-antibodies.com fluorescent compensation - flow cytometry guide: Bio-rad.

[BVU+05] Adriana Balduzzi, Maria Grazia Valsecchi, Cornelio Uderzo, Paola De Lorenzo, Thomas Klingebiel, Christina Peters, Jan Stary, Maria Felice, Edina Magyarosy, Valentino Conter, et al. Chemotherapy versus allogeneic transplantation for very-high-risk childhood acute lymphoblastic leukaemia in first complete remission: comparison by genetic randomisation in an international prospective study. *The Lancet*, 366(9486):635–642, 2005.

[BVV+09] Giuseppe Basso, Marinella Veltroni, Maria Grazia Valsecchi, Michael Dworzak, Richard Ratei, Daniela Silvestri, Alessandra Benetello, Barbara Buldini, Oscar Maglia, Giuseppe Masera, et al. Risk of relapse of childhood acute lymphoblastic leukemia is predicted by flow cytometric measurement of residual disease on day 15 bone marrow. *Journal of Clinical Oncology*, 27(31):5168–5174, 2009.

[CB15] Stacy Lorine Cooper and Patrick Andrew Brown. Treatment of pediatric acute lymphoblastic leukemia. *Pediatric Clinics*, 62(1):61–73, 2015.

[CC20] Krzysztof Choromanski and Lucy Colwell. Rethinking attention with performers, Oct 2020.

[CCW+21] Melissa Cheung, Jonathan J. Campbell, Liam Whitby, Robert J. Thomas, Julian Braybrook, and Jon Petzing. Current trends in flow cytometry automated data analysis software. *Cytometry Part A*, pages 1–15, 2021.

[CGRS19] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[CHL+15] Xiaoyi Chen, Milena Hasan, Valentina Libri, Alejandra Urrutia, Benoît Beitz, Vincent Rouilly, Darragh Duffy, Étienne Patin, Bernard Chalmond, and Lars Rogge. Automated flow cytometric analysis across large numbers of samples and cell types. *Clinical Immunology*, 157(2):249–260, 2015.

[CLD+20] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[CMS+20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[DBG+18] Michael N Dworzak, Barbara Buldini, Giuseppe Gaipa, Richard Ratei, Ondrej Hrusak, Drorit Luria, Eti Rosenthal, Jean-Pierre Bourquin, Mary Sartor, Angela Schumich, et al. Aieop-bfm consensus guidelines 2016 for flow cytometric immunophenotyping of pediatric acute lymphoblastic leukemia. *Cytometry Part B: Clinical Cytometry*, 94(1):82–93, 2018.

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[DFF+97] MN Dworzak, G Fritsch, C Fleischer, D Printz, G Fröschl, P Buchinger, G Mann, and H Gadner. Multiparameter phenotype mapping of normal and post-chemotherapy b lymphopoiesis in pediatric bone marrow. *Leukemia*, 11(8):1266–1273, 1997.

[DGA+11] Françoise Durrieu, Franck Genevieve, Christine Arnoulet, Caren Brumpt, Jean-Claude Capiod, Michel Degenne, Jean Feuillard, Richard Garand, Amina Kara-Terki, Emilienne Kulhein, et al. Normal levels of peripheral cd19+ cd5+ cll-like cells: Toward a defined threshold for cll follow-up—a geil-goelams study. *Cytometry Part B: Clinical Cytometry*, 80(6):346–353, 2011.

[Dic02] Becton Dickinson. Company. introduction to flow cytometry: A learning guide, 2002.

[doi13a] *Principles of Flow Cytometry*, chapter 2, pages 3–19. John Wiley & Sons, Ltd, 2013.

[doi13b] *Principles of Flow Cytometry*, chapter 4, pages 31–42. John Wiley & Sons, Ltd, 2013.

[doi13c] *Principles of Flow Cytometry*, chapter 9, pages 43–99. John Wiley & Sons, Ltd, 2013.

[doi13d] *Principles of Flow Cytometry*, chapter 5, pages 43–99. John Wiley & Sons, Ltd, 2013.

[DYY+19] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[EBB+15] Pablo Engel, Laurence Boumsell, Robert Balderas, Armand Bensussan, Valter Gattei, Vaclav Horejsi, Bo-Quan Jin, Fabio Malavasi, Frank Mortari, Reinhard Schwartz-Albiez, et al. Cd nomenclature 2015: human leukocyte differentiation antigen workshops as a driving force in immunology. *The Journal of Immunology*, 195(10):4555–4563, 2015.

[EGZ70] Audrey Elizabeth Evans, Ethel Gilbert, and Richard Zandstra. The increasing incidence of central nervous system leukemia in children.(children's cancer study group a). *Cancer*, 26(2):404–409, 1970.

[Elt20] Daniel C Elton. Self-explaining ai as an alternative to interpretable ai. In *International conference on artificial general intelligence*, pages 95–106. Springer, 2020.

[Gay05] Paul Gaynon. Childhood acute lymphoblastic leukaemia and relapse. *British Journal of Haematology*, 131(5):579–587, 2005.

[GBC16a] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. [GBC16e]. `http://www.deeplearningbook.org`.

[GBC16b] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. [GBC16e]. `http://www.deeplearningbook.org`.

[GBC16c] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. [GBC16e]. http://www.deeplearningbook.org.

[GBC16d] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. [GBC16e]. http://www.deeplearningbook.org.

[GBC16e] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[GCV+12] Giuseppe Gaipa, Giovanni Cazzaniga, Maria Grazia Valsecchi, Renate Panzer-Grümayer, Barbara Buldini, Daniela Silvestri, Leonid Karawajew, Oscar Maglia, Richard Ratei, Alessandra Benetello, et al. Time point-dependent concordance of flow cytometry and real-time quantitative polymerase chain reaction for minimal residual disease detection in childhood acute lymphoblastic leukemia. *haematologica*, 97(10):1582–1593, 2012.

[GDDM15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015.

[Gir15] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[Giv01] Alice L Givan. Principles of flow cytometry: an overview. *Methods in cell biology*, 63:19–50, 2001.

[GTB+09] Sumeet Gujral, Prashant Tembhare, Y Badrinath, PG Subramanian, Ashok Kumar, Kunal Sehgal, et al. Intracytoplasmic antigen study by flow cytometry in hematolymphoid neoplasm. *Indian Journal of Pathology and Microbiology*, 52(2):135, 2009.

[HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[HJB+08] Xiaohong Han, Jeffrey L Jorgensen, Archana Brahmandam, Ellen Schlette, Yang O Huh, Yuankai Shi, Sylvester Awagu, and Weina Chen. Immunophenotypic study of basophils by multiparameter flow cytometry. *Archives of pathology & laboratory medicine*, 132(5):813–819, 2008.

[Hof09] Johannes JML Hoffmann. Neutrophil cd64: a diagnostic marker for infection and sepsis. *Clinical chemistry and laboratory medicine*, 47(8):903–916, 2009.

[IGM13] Hiroto Inaba, Mel Greaves, and Charles G Mullighan. Acute lymphoblastic leukaemia. *The Lancet*, 381(9881):1943–1955, 2013.

[JGB+21] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

[JNQ+18] Disi Ji, Eric Nalisnick, Yu Qian, Richard H. Scheuermann, and Padhraic Smyth. Bayesian trees for automated cytometry data analysis. *bioRxiv*, 2018.

[KDPV+13] Mawar Karsa, Luciano Dalla Pozza, Nicola Venn, Tamara Law, Rachael Shi, Jodie Giles, Anita Bahar, Shamira Cross, Daniel Catchpoole, Michelle Haber, Glenn Marshall, Murray Norris, and Rosemary Sutton. Improving the identification of high risk precursor b acute lymphoblastic leukemia patients with earlier quantification of minimal residual disease. *PLOS ONE*, 8(10):1–6, 10 2013.

[KGS19] Peter Kaatsch, Desiree Grabow, and Claudia Spix. German childhood cancer registry - annual report 2018 (1980-2017). *Institute of Medical Biostatistic, Epidemiology and Informatics (IMBEI) at the University Medical Center of the Johannes Gutenberg University Mainz*, 2019.

[KKL20] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

[Kuh55] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[LB+95] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[LBC+18] Markus Lux, Ryan Remy Brinkman, Cedric Chauve, Adam Laing, Anna Lorenc, Lucie Abeler-Dörner, and Barbara Hammer. flowlearn: fast and precise identification and quality checking of cell populations in flow cytometry. *Bioinformatics*, 34(13):2245–2253, feb 2018.

[LKB+17] Hao-Chih Lee, Roman Kosoy, Christine E Becker, Joel T Dudley, and Brian A Kidd. Automated cell type discovery and classification through knowledge transfer. *Bioinformatics*, 33(11):1689–1695, jan 2017.

[LLK+19] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753. PMLR, 2019.

[LRVBACL09] Christine Le Roy, Nadine Varin-Blank, Florence Ajchenbaum-Cymbalista, and Rémi Letestu. Flow cytometry apc-tandem dyes are

degraded through a cell-dependent mechanism. *Cytometry Part A: The Journal of the International Society for Advancement of Cytometry*, 75(10):882–890, 2009.

[LSR+18] Roxane Licandro, Thomas Schlegl, Michael Reiter, Markus Diem, Michael Dworzak, Angela Schumich, Georg Langs, and Martin Kampel. Wgan latent space embeddings for blast identification in childhood acute myeloid leukaemia. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3868–3873. IEEE, 2018.

[LSS+17] Huamin Li, Uri Shaham, Kelly P Stanton, Yi Yao, Ruth R Montgomery, and Yuval Kluger. Gating mass cytometry data by deep learning. *Bioinformatics*, 33(21):3423–3430, 2017.

[LZW+21] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4722–4732, 2021.

[Mah36] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936.

[MAS+03] Colin H Martin, Iannis Aifantis, M Lucila Scimone, Ulrich H von Andrian, Boris Reizis, Harald von Boehmer, and Fotini Gounari. Efficient thymic immigration of b220+ lymphoid-restricted bone marrow cells with t precursor potential. *Nature immunology*, 4(9):866–873, 2003.

[McK18] Katherine McKinnon. Flow cytometry: An overview. *Current protocols in immunology*, 120(1):5–1, 2018.

[MHM18] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[MKS+98] Tomoko Matsumura, Masahiro Kami, Toshiki Saito, Hisashi Sakamaki, and Hisamaru Hirai. Prognostic value of minimal residual disease in acute lymphoblastic leukaemia in childhood. *The Lancet*, 352(9142):1731–1738, 1998.

[Mol20] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[Mon05] Monica Monici. Cell and tissue autofluorescence research and diagnostic applications. *Biotechnology annual review*, 11:227–256, 2005.

[MPR22] Dwarikanath Mahapatra, Alexander Poellinger, and Mauricio Reyes. Interpretability-guided inductive bias for deep learning based medical image. *Medical image analysis*, 81:102551, 2022.

[MTC+14] Mehrnoush Malek, Mohammad Jafar Taghiyar, Lauren Chong, Greg Finak, Raphael Gottardo, and Ryan R. Brinkman. flowDensity: reproducing manual gating of flow cytometry data by automated density-based cell population identification. *Bioinformatics*, 31(4):606–607, oct 2014.

[MWWK18] Jean S Marshall, Richard Warrington, Wade Watson, and Harold L Kim. An introduction to immunology and immunopathology. *Allergy, Asthma & Clinical Immunology*, 14(2):1–10, 2018.

[NH10] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[Nik21] Sergey I Nikolenko. *Synthetic data for deep learning*, volume 174. Springer, 2021.

[NNSP18] Faramarz Naeim, P. Nagesh Rao, Sophie X. Song, and Ryan T. Phan. Chapter 2 - principles of immunophenotyping. In Faramarz Naeim, P. Nagesh Rao, Sophie X. Song, and Ryan T. Phan, editors, *Atlas of Hematopathology (Second Edition)*, pages 29–56. Academic Press, second edition edition, 2018.

[NVG06] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006.

[NZP18] Weili Nie, Yang Zhang, and Ankit Patel. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *International Conference on Machine Learning*, pages 3809–3818. PMLR, 2018.

[OK20] Reeba Omman and Ameet Kini. Rodak's hematology chapter 31 - acute leukemias. In Elaine M. Keohane, Catherine N. Otto, and Jeanine M. Walenga, editors, *Rodak's Hematology (Sixth Edition)*, pages 540 – 554. Content Repository Only!, St. Louis (MO), sixth edition edition, 2020.

[Onc09] Mihaela Onciu. Acute lymphoblastic leukemia. *Hematology/oncology clinics of North America*, 23(4):655–674, 2009.

[PCP+09] Ching-Hon Pui, Dario Campana, Deqing Pei, Paul Bowman, John Torrey Sandlund, Sue Kaste, Raul Ribeiro, Jeffrey Rubnitz, Susana Raimondi, Mihaela Onciu, et al. Treating childhood acute lymphoblastic leukemia without cranial irradiation. *New England Journal of Medicine*, 360(26):2730–2741, 2009.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep

learning library. *Advances in Neural Information Processing Systems*, 32:8026–8037, 2019.

[PMER12] Ching-Hon Pui, Charles Mullighan, William Evans, and Mary Relling. Pediatric acute lymphoblastic leukemia: Where are we going and how do we get there? *Blood*, 120:1165–74, 06 2012.

[PR07] Stephen P Perfetto and Mario Roederer. Increased immunofluorescence sensitivity using 532 nm laser excitation. *Cytometry Part A: The Journal of the International Society for Analytical Cytology*, 71(2):73–79, 2007.

[PRL08] Ching-Hon Pui, Leslie Robison, and Thomas Look. Acute lymphoblastic leukaemia. *The Lancet*, 371(9617):1030 – 1043, 2008.

[PRM06] David R Parks, Mario Roederer, and Wayne A Moore. A new "logicle" display method avoids deceptive effects of logarithmic scaling for low signals and compensated data. *Cytometry Part A: The Journal of the International Society for Analytical Cytology*, 69(6):541–551, 2006.

[PVU+18] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International conference on machine learning*, pages 4055–4064. PMLR, 2018.

[QML+19] Jiezhong Qiu, Hao Ma, Omer Levy, Scott Wen-tau Yih, Sinong Wang, and Jie Tang. Blockwise self-attention for long document understanding. *arXiv preprint arXiv:1911.02972*, 2019.

[QSD+22] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. *arXiv preprint arXiv:2202.08791*, 2022.

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[RDS+19] Michael Reiter, Markus Diem, Angela Schumich, Margarita Maurer-Granofszky, Leonid Karawajew, Jorge G Rossi, Richard Ratei, Stefanie Groeneveld-Krentz, Elisa O Sajaroff, Susanne Suhendra, et al. Automated flow cytometric mrd assessment in childhood acute b-lymphoblastic leukemia using supervised machine learning. *Cytometry Part A*, 95(9):966–975, 2019.

[RLS+05] Alberto Redaelli, Benjamin Laskin, JM Stephens, Marc Botteman, and CL Pashos. A systematic literature review of the clinical and epidemiological burden of acute lymphoblastic leukaemia (all). *European journal of cancer care*, 14(1):53–62, 2005.

[RN06] Carlo Riccardi and Ildo Nicoletti. Analysis of apoptosis by propidium iodide staining and flow cytometry. *Nature protocols*, 1(3):1458–1461, 2006.

[Roe01] Mario Roederer. Spectral compensation for flow cytometry: visualization artifacts, limitations, and caveats. *Cytometry: The Journal of the International Society for Analytical Cytology*, 45(3):194–205, 2001.

[Roe02] Mario Roederer. Compensation in flow cytometry. *Current protocols in cytometry*, 22(1):1–14, 2002.

[RRK+16] Michael Reiter, Paolo Rota, Florian Kleber, Markus Diem, Stefanie Groeneveld-Krentz, and Michael Dworzak. Clustering of cell populations in flow cytometry data using a combination of gaussian mixtures. *Pattern Recognition*, 60:1029–1040, 2016.

[SCD+17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[Sha53] L Shapley. Quota solutions op n-person games1. *Edited by Emil Artin and Marston Morse*, page 343, 1953.

[SHK+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[SHP+12] Martin Schrappe, Stephen Hunger, Ching-Hon Pui, Vaskar Saha, Paul S Gaynon, André Baruchel, Valentino Conter, Jacques Otten, Akira Ohara, Anne Birgitta Versluys, et al. Outcomes after induction failure in childhood acute lymphoblastic leukemia. *New England Journal of Medicine*, 366(15):1371–1381, 2012.

[SLDV98] Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. Transformation invariance in pattern recognition—tangent distance and tangent propagation. In *Neural networks: tricks of the trade*, pages 239–274. Springer, 1998.

[STK+17] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[SVZ13] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[SWW+21] Paul D Simonson, Yue Wu, David Wu, Jonathan R Fromm, and Aaron Y Lee. De novo identification and visualization of important cell populations for classic hodgkin lymphoma using flow cytometry and machine learning. *American Journal of Clinical Pathology*, 156(6):1092–1102, 2021.

[TBY+20] Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *International Conference on Machine Learning*, pages 9438–9447. PMLR, 2020.

[TDBM20] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys (CSUR)*, 2020.

[UHR+21] Alfred Ultsch, Jörg Hoffmann, Maximilian Röhnert, Malte Von Bonin, Uta Oelschlägel, Cornelia Brendel, and Michael C. Thrun. An Explainable AI System for the Diagnosis of High Dimensional Biomedical Data. *arXiv e-prints*, page arXiv:2107.01820, July 2021.

[vDvdVBO15] Jacques van Dongen, Vincent van der Velden, Monika Brüggemann, and Alberto Orfao. Minimal residual disease diagnostics in acute lymphoblastic leukemia: need for sensitive, fast, and standardized technologies. *Blood*, 125(26):3996–4009, 2015.

[Vig19] Jesse Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019.

[VLWVDB+00] EG Van Lochem, YM Wiegers, R Van Den Beemd, K Hählen, JJM Van Dongen, and Herbert Hooijkaas. Regeneration pattern of precursor-b-cells in bone marrow of acute lymphoblastic leukemia patients depends on the type of preceding chemotherapy. *Leukemia*, 14(4):688–695, 2000.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[WDRMG21] Lisa Weijler, Markus Diem, Michael Reiter, and Margarita Maurer-Granofszky. Detecting rare cell populations in flow cytometry data using umap. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4903–4909, 2021.

[WKW+22] Lisa Weijler, Florian Kowarsch, Matthias Wödlinger, Michael Reiter, Margarita Maurer-Granofszky, Angela Schumich, and Michael N. Dworzak. Umap based anomaly detection for minimal residual disease quantification within acute myeloid leukemia. *Cancers*, 14(4), 2022.

[WLK+20] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[WRW+22] Matthias Wodlinger, Michael Reiter, Lisa Weijler, Margarita Maurer-Granofszky, Angela Schumich, Stefanie Groeneveld-Krentz, Richard Ratei, Leonid Karawajew, Elisa Sajaroff, Jorge Rossi, and Michael N. Dworzak. Automated identification of cell populations in flow cytometry data with transformers. *Computers in Biology and Medicine*, page 105314, 2022.

[YL15] Karim M Yatim and Fadi G Lakkis. A brief journey through the immune system. *Clinical Journal of the American Society of Nephrology*, 10(7):1274–1281, 2015.

[ZBL+18] John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683, 2018.

[ZKR+17] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

[ZMH+20] Max Zhao, Nanditha Mallesh, Alexander Höllein, Richard Schabath, Claudia Haferlach, Torsten Haferlach, Franz Elsner, Hannes Lüling, Peter Krawitz, and Wolfgang Kern. Hematologist-level classification of mature b-cell neoplasm using deep learning on multiparameter flow cytometry data. *Cytometry Part A*, 97(10):1073–1080, 2020.