UNIVERSITÀ
DI TORINO

Collateral-Free Learning of Deep Representations: From Natural Images to Biomedical Applications

Ph.D. Thesis

Carlo Alberto Barbano

Computer Science dept., University of Turin
LTCI, Télécom Paris, IP Paris

Jury Composition:

Rosa Meo Professor, University of Turin	President
Hervé Lombaert, Professor, ETS Montreal	Reviewer
Daniel Rueckert, Professor, Imperial College London	Reviewer
Enrico Magli, Professor, Politecnico di Torino	Examiner
Marco Grangetto, Professor, University of Turin	Thesis Supervisor
Isabelle Bloch, Professor, LTCI, Télécom Paris, IP Paris	Thesis Supervisor
Pietro Gori, Associate Professor, LTCI, Télécom Paris, IP Paris	Invited (Co-Supervisor)

Abstract

Deep Learning (DL) has become one of the predominant tools for solving a variety of tasks, often with superior performance compared to previous state-of-the-art methods. DL models are often able to learn meaningful and abstract representations of the underlying data. However, it has been shown that they might also learn additional features, which are not necessarily relevant or required for the desired task. This could pose a number of issues, as this additional information can contain bias, noise, or sensitive information, that should not be taken into account (e.g. gender, race, age, etc.) by the model. We refer to this information as *collateral*. The presence of collateral information translates into practical issues when deploying DL-based pipelines, especially if they involve private users' data. Learning robust representations that are free of collateral information can be highly relevant for a variety of fields and applications, like medical applications and decision support systems.

In this thesis, we introduce the concept of Collateral Learning, which refers to all those instances in which a model learns more information than intended. The aim of Collateral Learning is to bridge the gap between different fields in DL, such as robustness, debiasing, generalization in medical imaging, and privacy preservation. We propose different methods for achieving robust representations free of collateral information. Some of our contributions are based on regularization techniques, while others are represented by novel loss functions.

In the first part of the thesis, we lay the foundations of our work, by developing techniques for robust representation learning on natural images. We focus on one of the most important instances of Collateral Learning, namely biased data. Specifically, we focus on Contrastive Learning (CL), and we propose a unified metric learning framework that allows us to both easily analyze existing loss functions, and derive novel ones. Here, we propose a novel supervised contrastive loss function, ϵ-SupInfoNCE, and two debiasing regularization techniques, EnD and FairKL, that achieve state-of-the-art performance on a number of standard vision classification and debiasing benchmarks.

In the second part of the thesis, we focus on Collateral Learning in medical imaging, specifically on neuroimaging and chest X-ray images. For neuroimaging, we present a novel contrastive learning approach for brain age estimation. Our approach achieves state-of-the-art results on the OpenBHB dataset for age regression and shows increased robustness to the site effect. We also leverage this method to detect unhealthy brain aging patterns, showing promising results in the classification of brain conditions such as Mild Cognitive Impairment (MCI) and Alzheimer's Disease (AD). For chest X-ray images (CXR), we will target Covid-19 classification, showing how Collateral Learning can effectively hinder the reliability of such models. To tackle such issue, we propose a transfer learning approach that, combined with our regularization techniques, shows promising results on an original multi-site CXRs dataset.

Finally, we provide some hints about Collateral Learning and privacy preservation in DL models. We show that some of our proposed methods can be effective in preventing certain information from being learned by the model, thus avoiding potential data leakage.

Contents

Part I

Background

Chapter 1

Introduction

In the last two decades, Artificial Neural Network (ANN) models received huge interest from the research community. Nowadays, complex and even ill-posed problems can be tackled provided that one can train a deep enough ANN model with a large enough dataset. Furthermore, ANNs are quickly becoming a powerful tool helping us make a variety of decisions (Johnson et al., 2016; Kraus and Feuerriegel, 2017). ANNs are usually trained to process a desired output from some inputs. However, we do not have a clear idea of how information is represented inside of a network. This lack of understanding makes it difficult to interpret and explain the decisions made by ANNs. Recently, AI trustworthiness has been formally recognized as a major prerequisite for people and societies to use and accept such systems (AI HLEG, 2019; Zhang and Dafoe, 2019). In April 2019, the High-Level Expert Group on AI of the European Commission defined the three main aspects of trustworthy AI (AI HLEG, 2019): it should be lawful, ethical, and robust. Providing a warranty on this topic is currently a matter of study and discussion (Schramowski et al., 2020; Stock and Cissé, 2018; Teso and Kersting, 2019; Wang et al., 2020a).

This thesis focuses mainly on the aspect of robustness. While many different meanings can be attributed to the concept of robustness (Drenkow et al., 2021), in this thesis we refer to robustness as the ability to achieve and preserve good model performance under the presence of noise and biases in the data. A more comprehensive explanation will be provided in the following sections, by introducing the concept of Collateral Learning.

1.1 Aim of this work

The aim of the work in this thesis is to deal with the task of learning data representations, which, ideally, should satisfy the following requirements. They should be:

1. General enough in order to capture a meaningful variability of the data;

2. Discriminative enough for solving a desired downstream task;

3. Robust to confounding factors and spurious information in the data.

The first and second points can be referred to as representation learning: a process in which machine learning algorithms and models (e.g. neural networks) extract meaningful information from the data, which can then be used, for example, for classification. Representation learning can be supervised, unsupervised, or self-supervised. Recently, contrastive learning has become one of the most relevant approaches for representation learning, both in self-supervised and supervised forms (Chen et al., 2020; Khosla et al., 2020). Representation learning in the form of Contrastive Learning (CL) will be the subject of extensive analysis in this work, with a focus on the supervised case. A detailed introduction to representation learning and contrastive learning can be found in Section 2.1.
About the third and last point, in the rest of this manuscript, a better explanation will be provided by introducing the concept of Collateral Learning. This represents the core of this work, as finding a way to mitigate the impact of spurious information such as biases in the data has proven to be necessary for the development and deployment of real-world deep learning applications.

The methods presented in this work will be developed with the goal of generalizability. For this reason, the first chapters will focus on natural images. This allows us to exploit established benchmarks and larger datasets for assessing the efficacy of our contributions. Later in the work, the proposed methods will be tailored for tackling specific biomedical applications, such as brain imaging and chest X-ray diagnosis.

We will also briefly mention how Collateral Learning can pose an issue from the point of view of privacy. In fact, it can be shown that ANNs can learn different information from a given dataset, some of which might be considered sensitive (e.g. gender, sex, age, etc.). When sharing a model, or even just its output, it might be possible to retrieve and disclose such sensitive information.

1.2 Collateral Learning

Collateral learning, conceptualized by John Dewey, describes the accidental learning that occurs in and outside the classroom (Dewey, 1997).

> "Perhaps the greatest of all pedagogical fallacies is the notion that a person learns only the particular thing he is studying at the time. Collateral learning in the way of formation of enduring attitudes, of likes and dislikes, may be and often is much more important than the spelling lesson or lesson in geography or history that is learned. For these attitudes are fundamentally what count in the future. The most important attitude that can be formed is that of desire to go on learning."
> (Dewey, *Experience and Education*)

Based on this definition, and extending it to the Deep Learning (DL) context, we say that collateral learning occurs when a model learns more information than intended. In the following sections, we will present cases in which such a phenomenon occurs, but here we will provide some basic intuitions. Consider a very common computer vision dataset such as ImageNet (Deng et al., 2009), used by the vast majority of works in the computer vision domain. It contains one thousand classes of different

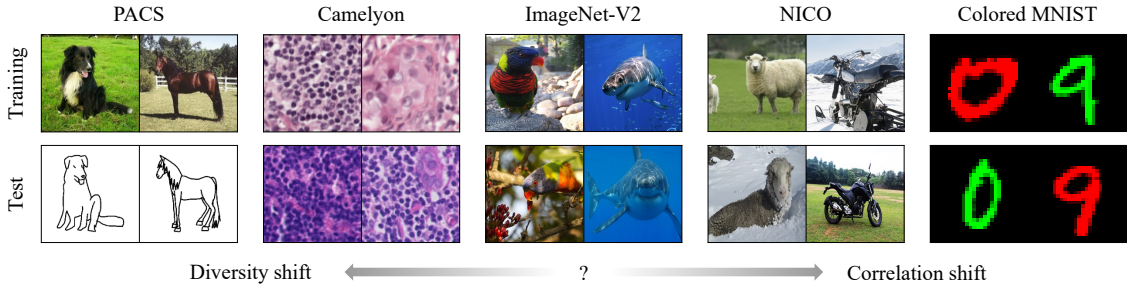


Figure 1.2.1: Examples of distribution shifts in image classification datasets. Datasets at both ends show apparent differences (diversity and correlation shifts). However, many realistic datasets lie in the middle and can be affected by collateral learning to some extent. Credits to Ye et al. (2022).

objects and subjects, such as animals, and vehicles, for example, cows and airplanes. It is reasonable to predict that, given the nature of such subjects, images containing cows and similar animals will exhibit common scenes (e.g. grass fields) in the vast majority of cases. Similarly, with plane images, we will be probably looking at a good portion of the sky as well. Given such a correlation between the subjects and the contexts they appear in, a model trained on this data will probably rely on landscape clues for recognizing the subject class. If presented with the same subjects but in a completely different context, for example, a cow on a beach (Beery et al., 2018), will the model be actually robust enough to provide the correct classification?

Focusing on the concept of robustness for AI, Attenberg et al. (2015) discussed the problem of finding the so-called "unknown unknowns" in data. These unknown unknowns relate to the case when the deep model elaborates information in an unintended way, but shows high confidence on its predictions. In recent years, during the Covid-19 pandemic, this behavior affected many works proposing DL-based solutions for Covid-19 detection from Chest X-ray images. Unfortunately, the available datasets, especially at the beginning of the pandemic, were heavily biased. This often resulted in models mistakenly predicting a Covid-19 diagnosis with high confidence, due to the presence of unwanted biases, for example by detecting the catheters or medical devices for positive patients, their age (at the beginning of the pandemic, most ill patients were elderly people), or even by recognizing the origin of the data itself (when negative cases were augmented borrowing samples from other datasets) (Apostolopoulos and Mpesiana, 2020; Sethy and Behera, 2020; Tartaglione et al., 2020). The latter phenomenon, also commonly known in medical imaging as site effect (Glocker et al., 2019; Howard et al., 2021), is particularly relevant in neuroimaging (Chen et al., 2022; Dufumier et al., 2022; Nguyen et al., 2018).

In summary, in order to be *robust*, DL models should not be affected by the collateral learning problem. For a more detailed analysis, in the scope of this thesis, we distinguish some specific scenarios where collateral learning can be a problem, namely biased data and multi-site medical imaging. Furthermore, this thesis will also provide some hints in the context of privacy preservation.

1.2.1 Biased data

Learning good representations for a given dataset is usually achieved by minimizing some loss function of a given model, as explained in Section 2.1. However, in practice, this does not guarantee the ability of the model to generalize to new and unseen data. Often, testing data exhibit differences with respect to the training data, a phenomenon which is commonly known as domain gap or domain shift (Ganin and Lempitsky, 2015; Luo et al., 2019; Quinonero-Candela et al., 2008). While there are many different ways in which this can happen, with whole research fields dedicated to them, in this thesis we are mainly interested in two cases: diversity shift and correlation shift (Ye et al., 2022). To provide a formal explanation of these two different types of domain shifts, let us define a sample x as the composition of different signal sources $\mathbb{S}$, which we can model as random variables. A diversity shift happens when the distribution of a certain source $\mathbb{S}_i \in \mathbb{S}$ changes from a source domain S to a target domain T: $\exists i \,|\, p_S(\mathbb{S}_i) \neq p_T(\mathbb{S}_i)$; while a correlation shift can be defined as $\exists i, j \neq i \,|\, p_S(\mathbb{S}_i|\mathbb{S}_j) \neq p_T(\mathbb{S}_i|\mathbb{S}_j)$, meaning that the distribution of the source $\mathbb{S}_i$, conditioned on another source $\mathbb{S}_j$, varies between the source and target domain. In both these cases, collateral learning means that the model captures the spurious correlation/information in the data. An example of these kinds of domain shifts is illustrated in Figure 1.2.1.

Diversity shift

To provide a more intuitive understanding, let us consider an example. Imagine we are working with a dataset that contains images of different animals. In this scenario, $\mathbb{S}_i$ could represent a specific attribute of the images, such as their color distribution. A diversity shift would occur if, for instance, in the source domain S, the images predominantly feature animals with brown fur, while in the target domain T, the images primarily depict animals with white fur.
This change in the distribution of the attribute (in this case, fur color) across the source and target domains can have a significant impact on the performance of a machine learning model. It may lead to the model making inaccurate predictions or classifications on data from the target domain, as it has been primarily trained on data from the source domain.

Addressing diversity shift is crucial in machine learning tasks, particularly when the goal is to deploy models in real-world applications where the distribution of data can vary over time or across different environments. Techniques to mitigate diversity shift typically involve strategies like domain adaptation, where the model is trained to align the distributions of relevant features across different domains, allowing it to better generalize to unseen data.

Correlation shift

Many datasets are biased, namely they contain easy-to-learn features that are highly correlated with the target class only in the dataset but not in the true underlying distribution of the data. In the latest years, it has become increasingly evident how neural networks tend to rely on simple patterns in the data (Geirhos et al., 2019; Li et al., 2021). As deep neural networks grow in size and complexity, guaranteeing

that they do not learn spurious elements in the training set is becoming a pressuring issue to tackle. It is indeed a known fact that most of the commonly-used datasets, such as ImageNet (Deng et al., 2009), are biased (Geirhos et al., 2019; Torralba et al., 2011) and that this affects the learned models (Tommasi et al., 2017). In particular, when the biases correlate very well with the target task, it is hard to obtain predictions that are independent of the biases. Furthermore, if the bias is also easier to learn than the desired features (e.g. a simple pattern or color), we will most likely obtain a biased model, whose predictions majorly rely on these spurious attributes and not on the true, generalizable, and discriminative features. In fact, based on these observations, a bias can be characterized on its "malignancy" as either *benign* or *malignant*, as done by Arpit et al. (2017); Nam et al. (2020) In order to be malignant, a bias must:

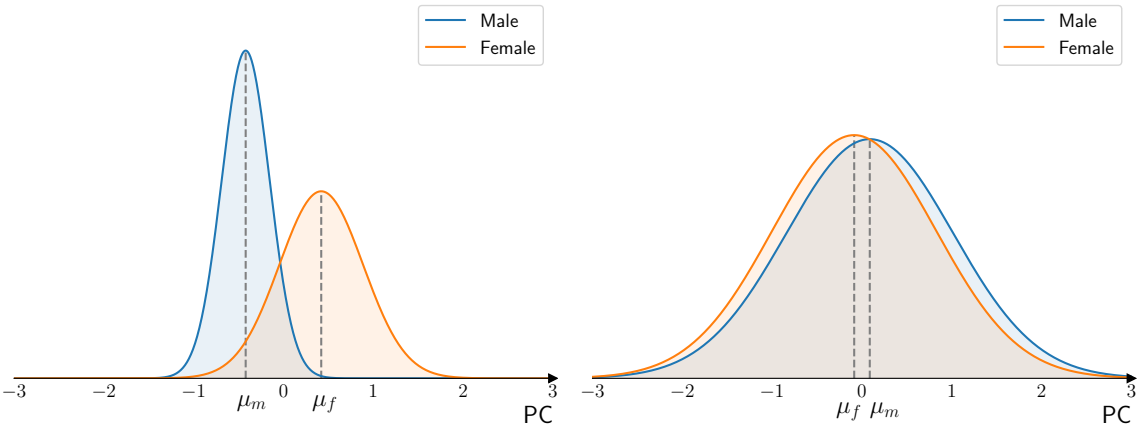
- have a strong enough correlation with the target task;
- be an *easier* pattern to learn that the target features.

Malignant biases are more harmful for the generalization capabilities of a model, as they can prevent the true discriminative features from being learned. For this reason, debiasing algorithms usually focus on this kind of biases. To illustrate this concept with an example, let us consider a scenario where we are dealing with medical data. In this context, $\mathbb{S}_i$ could represent a patient's blood pressure, while $\mathbb{S}_j$ could represent their cholesterol levels. A correlation shift would occur if the relationship between blood pressure and cholesterol levels differs between the source domain S and the target domain T. For instance, in the source domain, high blood pressure might be strongly correlated with high cholesterol levels, while in the target domain, this correlation might be weaker or even reversed.
This change in the relationship between different sources of information can have a significant impact on the performance of a machine learning model, especially if the model relies on these correlations to make accurate predictions or classifications.
Not only malign biases but also benign biases could represent an issue, as their collateral information can be easily learned by the model and impact its behavior. Furthermore, this could be exploited post-training, in order to retrieve some characteristics of the training set, leading to a leakage of potentially private information.

1.2.2 Site-effect in medical imaging

Site-effect in medical imaging is an instance of collateral learning and is essentially an issue of domain shift. However, given its relevance in the field of machine learning for medical imaging (Chen et al., 2022; Glocker et al., 2019; Howard et al., 2021; Nguyen et al., 2018; Wachinger et al., 2021), it is worth dedicating a separate discussion to this topic.

Site-effect refers to a phenomenon where the performance or behavior of machine learning model is influenced by the specific origin or source of the data. Each medical center may have its own imaging equipment, protocols, and practices for acquiring these images. Due to these variations, the images from different centers may have subtle differences in terms of image quality, resolution, and even factors like lighting or positioning of the patient. When training a machine learning model to analyze



(a) (b)

Figure 1.2.2: Principal Component (PC) of the learned representations for the IMDB Face dataset using a vanilla model (a) and a regularized model (b). We indicate with μ_f and μ_m the mean of the female and male samples' representations, respectively. Collateral gender features are learned by the model even if training for a different task (age prediction) if no explicit care is taken. More details in Chapter 8.

these images, the model may inadvertently learn patterns that are specific to the site or medical center where the images originated. For example, it might learn to recognize certain image artifacts or characteristics that are unique to a particular imaging device. This becomes problematic because the model's performance may degrade when applied to images from a different medical center. It may struggle to accurately interpret the data or make accurate diagnoses just because it has overfitted the collateral information of the images from the original training site. This thesis tackles the site-effect problem in Part III, in the context of neuroimaging for brain age prediction and Chest X-ray Images for Covid-19 prediction.

1.2.3 Leakage of private information

Privacy is another central matter of any kind of information processing system. The problem of data privacy in AI-based algorithms deepens its roots before the uprising of DL. However, with the advent of DL, it has become hard to provide guarantees on what type of information is learned by the models, due to their black-box nature. For example, how can we ensure that no collateral private information is learned by a DL model deployed on users' data? Or, for example, in a federated learning context, how can ensure that the shared information does not contain any identifying feature? These are relevant questions one needs to take into consideration when deploying DL-based solutions, as it has been shown that it is possible to disclose private information from such systems, for example with inversion attacks as shown by Fredrikson et al. (2015). Allowing information not relevant to the learning task to be stored inside the network is a known phenomenon. For example, Song et al. (2017) empirically show how accurately some side information can be recovered, resulting in a potential lack of privacy.

An example of how potentially sensitive information can be retrieved from the

learned representations is presented in Figure 1.2.2, where a model is trained on the IMDB Face dataset (Rothe et al., 2018) for age prediction from facial images. Looking at the learned representations, it is very easy to tell apart male individuals from female ones, even if such (collateral) information was not directly used during training. By using regularization methods such as the ones proposed in this thesis it is possible to prevent the model from learning such information.

1.3 Dealing with Collateral Learning

We have seen how collateral learning affects deep learning in different ways. It is clear that devising methods for dealing with this issue is necessary in order to make deep learning more reliable and trustworthy. This thesis proposes different approaches for this purpose; some of them are based on regularization techniques that constrain the learning process, while others more generally derive an optimization objective which can be less prone to collateral learning itself. Additionally, we also provide hints on how collateral learning can be mitigated by careful choice of pre-training tasks and transfer learning strategies.

Representation Learning A more throughout understanding of how deep models can learn powerful representations can certainly be helpful in mitigating the collateral learning issue. In this thesis, we adopt a metric learning point of view to formalize the learning process, which allows us to derive a set of contrastive loss functions, suited for both classification and regression. In Chapter 4 we will derive a contrastive loss for classification on natural images, while in Section 6.1 we will propose a contrastive loss for regression in neuroimaging for estimating brain age. In the latter case, the proposed loss will also exhibit a certain degree of invariance to site effect, which is a collateral learning issue specific to medical imaging.

Regularization Regularization, which will be explained in Section 2.1, is a well-known technique for improving the generalization capabilities of a model. In Section 3 we will present the first proposed regularization method, EnD, to tackle biases in the data. An improved regularization technique, FairKL, will be then presented in Section 4.2, which will be shown to be more effective in mitigating collateral learning.

Robust pre-training and transfer learning Robust representations can be achieved not only through regularization or derivation of novel loss functions, but also by carefully choosing the pre-training task and the transfer learning strategy. In Chapter 7 we will show how pre-training on a large dataset of Chest X-ray images can be used to learn robust representations, useful for Covid-19 prediction.

1.4 Organization of this thesis

This work introduces the concept of Collateral Learning in Deep Learning, which refers to all those instances in which DL models learn "more" information than we expect. This concept aims at bridging the gap among different fields of research such

as robustness, debiasing, generalization in medical imaging, and privacy preservation. For this reason, this thesis is organized into different parts.

In Part I, we provide the background of this work. Chapter 1 defines the aim of the work, and provides a definition of Collateral Learning, along with relevant examples and practical applications. Chapter 2 provides an overview of the relevant related literature. It illustrates the basis for deep representation learning, debiasing and medical imaging.

Part II and III represent the main contribution of this work. In Part II we focus on developing methods for robust representation learning on natural images. In Chapters 3 and 4 we propose different supervised learning approaches for representation learning and debiasing. In Chapter 5 we present our ongoing work for extending the previous methods to the unsupervised scenario.
Part III focuses on Collateral Learning in medical images. We present, in Chapter 6, our efforts to build a robust brain age prediction model leading to the diagnosis of brain conditions. After that, Chapter 7 will present our contributions for Covid-19 diagnosis from chest X-ray imaging, highlighting the threats posed by Collateral Learning.

In Part IV, we will briefly touch upon another instance of Collateral Learning, specifically the issue of privacy preservation in DL-based systems in Chapter 8.

Finally, in Part V, we will draw the conclusions of this work, summarizing the contributions and findings during this PhD. We will also illustrate some of ongoing the related work, and provide insights about future developments in the field of Collateral Learning and robust representation learning.

1.5 Publications

The work in this thesis has led to the publication of several papers, both in conference and journals. Below is a list of the publications achieved.

Journal articles

(J1) **Unsupervised learning of unbiased visual representations.** C. A. Barbano, E. Tartaglione, and M. Grangetto. Submitted to *Journal of Machine Learning Research*.

(J4) **Detection of subclinical atherosclerosis by image-based deep learning on chest x-ray: a retrospective model development and validation study**. G. Gallone, A. Presta, F. Iodice, D. Tore, O. D. Filippo, M. Visciano, C. A. Barbano, A. Serafini, W. G. Marra, J. Hughes, M. Iannacone, P. Fonio, A. Fiandrotti, A. Depaoli, M. Grangetto, G. M. D. Ferrari, F. D'Ascenzo. Submitted to *Radiology*, 2023.

(J2) **Simplify: A python library for optimizing pruned neural networks.** A. Bragagnolo and C. A. Barbano. *SoftwareX*, 2022.

(J3) **Unveiling covid-19 from chest x-ray with deep learning: A hurdles race with small data.** E: Tartaglione, C. A. Barbano, C. Berzovini, M. Calandri, and M. Grangetto. *International Journal of Environmental Research and Public Health*, 2020.

Conference articles

(C1) **Unbiased supervised contrastive learning.** C. A. Barbano, B. Dufumier, E. Tartaglione, M. Grangetto, and P. Gori. *ICLR*, 2023.

(C2) **Integrating Prior Knowledge in Contrastive Learning with Kernel** B. Dufumier, C. A. Barbano, R. Louiset, E. Duchesnay, and P. Gori. *ICML*, 2023

(C3) **Contrastive learning for regression in multi-site brain age prediction.** C. A. Barbano, B. Dufumier, E. Duchesnay, M. Grangetto, and P. Gori. *ISBI*, 2023.

(C4) **A two-step radiologist-like approach for Covid-19 computer-aided diagnosis from chest X-ray images.** C. A. Barbano, E. Tartaglione, C. Berzovini, M. Calandri, and M. Grangetto. *ICIAP*, 2022.

(C5) **End: Entangling and disentangling deep representations for bias correction.** E. Tartaglione, C. A. Barbano, and M. Grangetto. *CVPR*, 2021.

(C6) **Bridging the gap between debiasing and privacy for deep learning.** C. A. Barbano, E. Tartaglione, and M. Grangetto. *ICCV (Workshop)*, 2021.

(C7) **Unitopatho, a labeled histopathological dataset for colorectal polyps classification and adenoma dysplasia grading.** C. A. Barbano, Daniele Perlo, E. Tartaglione, A. Fiandrotti, L. Bertero, P. Cassoni, and M. Grangetto. *ICIP*, 2021.

Chapter 2

Related Works

This Chapter provides an overview of the relevant literature in the fields of deep representation learning, debiasing, and medical imaging. First, we introduce the basics of deep learning, starting from fully connected neural networks up to convolutional networks and the different existing methods for training them. Then, we focus on existing debiasing techniques, by providing a high-level categorization (supervised, prior guided, unsupervised) of the different state-of-the-art methods. Finally, we introduce the topic of medical imaging and provide an overview of the relevant deep learning methods in neuroimaging (for brain age prediction) and chest X-ray (for Covid-19 detection).

2.1 Deep Representation Learning

Deep Learning (DL) is a subfield of Machine Learning (ML), which has become predominant in almost all ML applications. Compared to traditional ML, which requires a lot more manual crafting of input features and transformations (e.g. dimensionality reduction), and, in general, relies more on human domain knowledge, DL models aim at learning a desired representation starting directly from the raw data, by applying a number of parametric non-linear transformations. The parameters are optimized by minimizing an error function defined over the output of the model and some ground truth labels (i.e. supervised learning) or, for example, over some other metric computed on the output alone (i.e. self-supervised / unsupervised). Compared to previous ML models (e.g. support vector machines, trees, etc.) DL models are more computationally expensive due to the higher number of parameters.

Deep Learning has shown unprecedented performance, especially when dealing with large quantities of data. This was made possible by multiple technological advances, such as widespread access to the Internet, which allowed the creation of large datasets such as ImageNet (Deng et al., 2009), and the advances in computational power, made possible by the rapid development of Graphics Processing Units (GPU).

In this Section, we provide an overview of DL models, from simple fully-connected networks to the state-of-the-art architectures and optimization techniques that are commonly used nowadays.

2.1.1 Multi-Layer Perceptrons

Multilayer perceptrons are a type of ANNs. Strictly following the definition, they are composed of multiple layers of perceptrons (with thresholding function), but the term is often also used when referring to feedforward artificial networks. ANNs are composed of layers of more general computing units (*neurons*) with different activation functions and affine transformations (i.e. dot product as in the previous examples, or convolution as we will later see). From now on we will use the more generic term *neural network*.

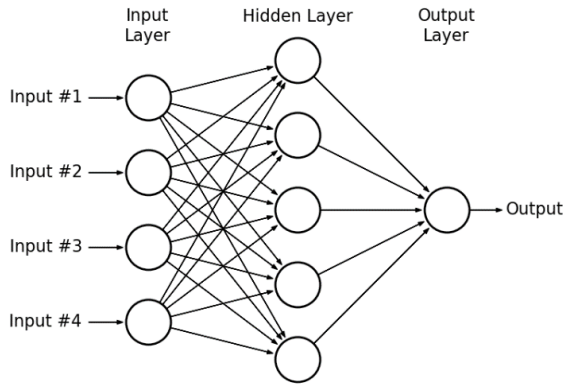


Figure 2.1.1: Sample architecture of a multilayer perceptron / ANN.

Figure 2.1.1 shows a sample architecture for a multilayer neural network. Neural networks are typically acyclic graphs, organized in layers. In a fully-connected network, such as the one shown in Figure 2.1.1, each neuron in a layer is connected to every neuron in the previous layer. Every connection is weighted by a learnable *weight*. The fully-connected layers are also called *dense* layers. The *depth* of the network is given by the number of layers, which is usually at least three: the input layer, one or more hidden layers and the output layer. Each neuron performs an affine transformation (dot product in fully-connected networks), followed by a non-linear activation function. Activation functions are a fundamental component in any neural network. Their non-linearity allows the network to approximate complex functions. Without non-linear activation functions, a network of any given depth could be replaced by a single layer performing a linear transformation. Moreover, to allow for the gradient-based optimization techniques (that will be explained in Section 2.1.1), the activation functions should be differentiable (i.e. tanh or sigmoid). By far, the most common activation function is ReLU, defined as:

$$ReLU(x) = \max(0, x) \tag{2.1.1}$$

Even though formally ReLU is non-differentiable at zero (the value of the derivative at zero is arbitrarily chosen to be either 1 or 0), it provides a number of advantages when compared to standard sigmoidal functions: it has a more efficient computation, promotes sparsity in the network and reduces the *vanishing gradient* problem Glorot et al. (2011).

Most of the operations in a neural network can be expressed in terms of matrix multiplication, which has the advantage of being highly parallelizable. Let's now

assume a network with N layers. Given the n-th layer, with $0 \leq n \leq N$, containing j neurons, we can express its set of parameters θ_n:

$$\theta_n = \begin{bmatrix} \overline{w}_n^0 \\ \overline{w}_n^1 \\ \vdots \\ \overline{w}_n^j \end{bmatrix} = \begin{bmatrix} w_{n,0}^0 & w_{n,1}^0 & \dots & w_{n,k}^0 & b_n^0 \\ w_{n,0}^1 & w_{n,1}^1 & \dots & w_{n,k}^1 & b_n^1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{n,0}^j & w_{n,1}^j & \dots & w_{n,k}^j & b_n^j \end{bmatrix} \tag{2.1.2}$$

where $w_{n,i}^j$ is the i-th weight of the j-th neuron in the n-th layer, b_n^j is a bias term of the j-th neuron in the n-th layer, and k is the number of neurons in the previous layer (for $n = 0$, k represents the number of dimensions of the input vector). Each row of the matrix contains the weights of a neuron in that layer.
To obtain a prediction from the neural network, it is sufficient to perform a *feed-forward* pass, recursively applying the transformation for each layer in the network. Denoting with $\hat{y}_n$ the output of the n-th layer, the forward pass of a neural network can be summarized as:

$$\hat{y}_n = f(\hat{y}_{n-1} \cdot \theta_n^T) \tag{2.1.3}$$

where f is the non-linear activation function, and $\hat{y}_0$ is the input layer[1].
The outputs of the final layer of a neural network (called *logits*) are then usually normalized to obtain a probability distribution, where each value represents the probability of the sample belonging to a certain class. Commonly used normalization functions include *Sigmoid* or *Softmax*.

Gradient Descent Training a neural network is usually achieved by defining an error function with respect to some desired output. The error is also usually called *loss*. This can be accomplished by computing the gradient of the loss with respect to each parameter in the network and using it to make adjustments by moving in the direction of the steepest decrease in the error. Gradient-based optimization requires the loss and activation functions to be differentiable. This technique is known as *Gradient Descent* (GD). We can define the update for a set of parameter w as:

$$w^{(t+1)} = w^{(t)} + \eta \nabla L(y, \hat{y}) \tag{2.1.4}$$

where L is the loss function, y is the desired output, $\hat{y}$ is the network prediction, and η is a hyperparameter which determines convergence speed called *learning rate*. The standard process for training a neural network using GD consists in performing a *forward* pass on the entire dataset, computing the global error (i.e. sum or average of error for each sample), and then obtaining the gradients for each layer's parameters. This last step is usually called *backward* pass: the gradients of a layer will depend on the output of successive layers and can be computed with the *backpropagation* algorithm (Rumelhart et al., 1986), which makes use of the *chain rule*[2].
Using the entire dataset just to perform a single update step is often very time consuming, when dealing with large datasets (or even too computationally expensive, if

[1] We won't be explicitly showing the bias term for each set of weights, as it is irrelevant to the understanding of the proposed concepts. To account for it, it is sufficient to consider an additional dimension with the corresponding input value set to 1.

[2] Most of the modern deep learning frameworks such as TensorFlow and PyTorch makes this step trivial, by constructing a computation graph, which allows to easily compute chained derivatives.

storing the computational graph for each sample is required). This is why Stochastic Gradient Descent (SGD) is usually employed. In SGD, a forward-backward pass is performed on each sample (randomly drawn from the dataset), making small adjustments at every iteration. This procedure, however, could lead to a noisy learning process, especially in the presence of outliers. Hence, it is much more common to employ a variant of SGD called *mini-batch stochastic gradient descent*, where more than one sample are drawn at every iteration. The error is then computed on the entire mini-batch and averaged among the samples, before propagating the gradients. In this work, when talking about SGD, we will be referring to its mini-batch implementation. Having employed a matrix notation in the previous equations, it is now very easy to understand this approach: instead of an input vector $\overline{x}$, a matrix x of size $N \times K$ will be used, where N is the size of the mini-batch and K is the number of dimensions of a single sample. Each row of the matrix will represent a single data point.

Loss functions Common loss functions include Mean Squared Error (MSE), Binary Cross Entropy (BCE) and Cross Entropy (CE). For classification tasks, BCE and CE are usually employed. BCE is generally used on binary classification tasks, while CE is commonly employed for *multiclass* classification (problems with more than one class, but in which samples are assigned a single label). Denoting with $\tilde{y}$ the normalized output of the network, the general form of the CE loss, for a given sample, is:

$$L_{\text{CE}} = -\sum_{n=0}^{N-1} y_n \log \tilde{y}_n \tag{2.1.5}$$

where N is the total number of classes, y_n is a binary value (0 or 1) indicating whether the sample belongs to n-th class and $\tilde{y}_n$ is the model prediction for the n-th class. For a binary classification task, Equation 2.1.5 can be reduced to the BCE formula:

$$L_{\text{BCE}} = -\left[y \log \tilde{y} + (1-y) \log (1-\tilde{y})\right] \tag{2.1.6}$$

where y is the label for the given sample. BCE is also used in *multilabel* classification problems, where more than one label can be assigned to a single sample. In a multilabel problem with N classes, each n-th output component is treated independently from the others (this is effectively the same as having N different binary classifiers). Hence, for every sample, a different loss is computed on each single class. Denoting with L^n the loss on a given sample for the n-th class, the total loss can be obtained, for example, by summation:

$$L_{BCE} = \sum_{n=0}^{N-1} L^n_{\text{BCE}} = -\sum_{n=0}^{N-1} \left[y_n \log \tilde{y}_n + (1-y_n) \log(1-\tilde{y}_n)\right] \tag{2.1.7}$$

where y_n and $\hat{y}_n$ are respectively the ground truth and the prediction for the n-th class. An example of a multilabel classification problem will be presented in Section 7.2.1 for classifying different lung pathologies.

Regularization One of the most common problems in deep learning and machine learning algorithms is *overfitting*. This happens when a model cannot generalize to unseen data because the training data was just memorized. To address this issue, the most common approach is adding a *regularization* term to the loss function, which prevents the models from memorizing the training data. The general form of a regularized loss function can be denoted as:

$$L_{\text{reg}} = L + \lambda R \tag{2.1.8}$$

where L denotes the loss function, R is the regularization term, and λ is a hyper-parameter determining the amount of regularization to apply.
By far, the most common technique is the $L2$ regularization, which adds a penalty for larger parameters and drives them towards zero. An L2-regularized loss function is formulated as follows:

$$L_{\text{L2}} = L + \lambda \|w\|_2^2 \tag{2.1.9}$$

where $\|.\|_2^2$ is the squared $L2$-norm. When deriving the regularized loss, we obtain the following update rule (dropping the constant 2):

$$w \leftarrow (1 - \eta\lambda)w + \eta\Delta w \tag{2.1.10}$$

The L2-regularization is often called *weight decay* as the weight is multiplied for a factor smaller than one before applying the update. This makes it harder for the model to overfit, as noise in the data will have less of an impact on the network weights (Krogh and Hertz, 1992). Other regularization techniques exist besides L2, such as L1 or, for example, the ones proposed in this thesis in Chapters 3 and 4.

2.1.2 Neural networks for image processing

When dealing with images, another kind of layer is preferred instead of dense layers: the *convolutional* layer. Neural networks built using this kind of layer are commonly called *convolutional network*. As the name implies, these models adopt convolution as their primary affine transformation function. The major advantage of convolutional networks is that they do not require as much pre-processing compared to traditional image processing algorithms, eliminating the need to hand-craft filters, as they are learned through gradient descent. In 1989, Yann LeCunn proposed a system to recognize hand-written ZIP code numbers, where the convolution filters were learned through backpropagation, which became the foundation of convolutional networks for image processing (LeCun et al., 1989). Convolutional networks are composed of a set of convolution filters that are successively applied in order to extract meaningful features from the image, while also reducing the input dimensionality.

A typical architecture of a convolutional network is shown in Figure 2.1.2. We can identify two main building blocks: the *encoder*, made by the convolutional layers, which has the goal of extracting features from the input data, and the *classifier*, which is usually a fully-connected network taking as input the extracted features. As we can notice from Figure 2.1.2, convolution is usually applied together with

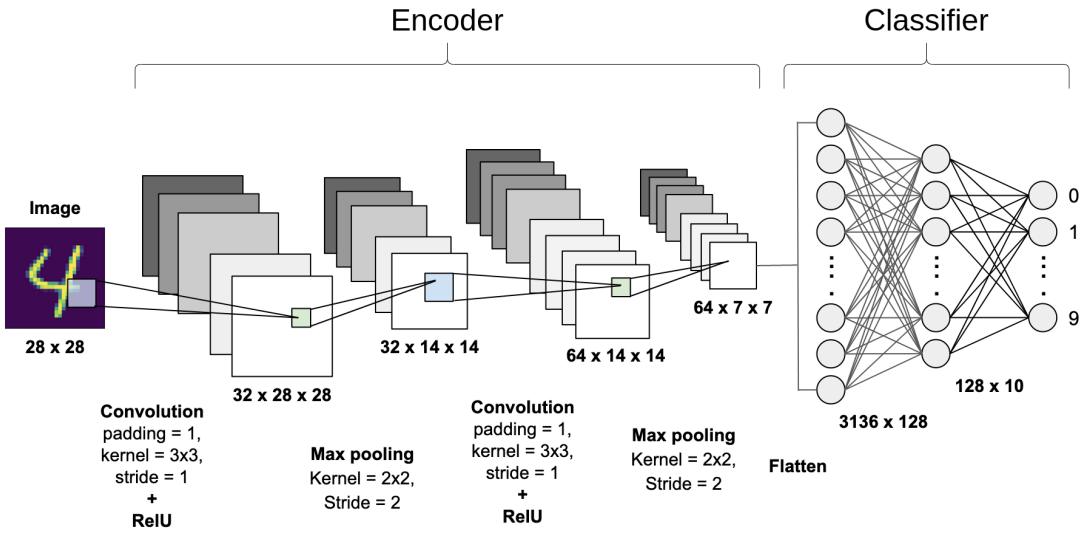


Figure 2.1.2: A convolutional network. Credits to Shreyak (2020)

another transformation called *pooling*.

We are now going to briefly describe the main types of layers we can find in a standard convolutional network. For both of the layers presented below, we can control a number of hyperparameters: the *size* of the filter, the *stride* (which determines the amount by which the filters shift on the input tensor), and the amount of *zero padding* applied on the input tensor.

Convolutional layer We can denote a convolutional layer by its parameter tensor θ of shape $C_{out} \times C_{in} \times W_k \times H_k$, where C_{out} is the number of channels in the resulting tensor, C_{in} the number of input channels (i.e. 3 for RGB images) and W_k and H_k are the width and height of the convolution filters. Given an input tensor x of shape $C_{in} \times W \times H$, where W and H are the width and height, the affine transformation performed by the layer (here denoted by $\diamond$) for the c-th output channel can be described with:

$$x \diamond \theta_c = \sum_{i=0}^{C_{in}-1} x_i * \theta_{c,i} \tag{2.1.11}$$

where $*$ is the 2D *cross-correlation* operation.[3] Each output channel θ_c can be obtained by summing across the input channel i the result of the cross-correlation between the i-th channel of the input tensor x_i and the corresponding convolution filter $\theta_{c,i}$. The output of a convolutional layer is called *feature map*. Similarly to dense layers, convolutional layers are followed by an activation function, which is commonly ReLU.

Pooling layer Pooling layers perform a fixed transformation, which helps in reducing data dimensionality and the computational complexity of the network. The pooling operation consists of a sliding window that computes a fixed function of their

[3]Similarly to the fully-connected network, the bias term can be taken into account with this formulation by considering an additional input channel, filled with the constant value 1.

input and does not require any learnable parameter. They are usually employed after a number of convolutional layers, also helping in achieving better translation invariance. The two most common types of pooling are *max pooling*, where for each windowed region only the maximum value is retained, and *average pooling* in which each region is substituted with the average value.

State-of-the-art network architectures

Throughout the years, many architectures of convolutional networks were proposed, ranging from the older but foundational LeNet-5 (LeCun et al., 1998) to more recent networks such as VGG (Simonyan and Zisserman, 2014), ResNet (He et al., 2016) and DenseNet (Huang et al., 2017), just to name a few. This section concludes the introduction to deep learning and neural networks, by presenting the state-of-the-art convolutional architectures that will be later used in this work.

ResNet ResNet was originally introduced in 2015 by the Microsoft Research team (He et al., 2016). It won first place in the *ImageNet Large Scale Visual Recognition Competition* (Deng et al., 2009) in 2015 with a top-5 error rate of 3,57%. ResNet allows for much deeper networks compared to previous architectures like VGG, thanks to the introduction of residual connections.
Figure 2.1.3 shows an example of residual connection. $\mathcal{F}(x)$ denotes the output of a *block* of layers on a certain input x. The residual connection consists in adding a shortcut from x to the block output (also called *skip connection*), which allows the gradients to flow in two directions. The reason for skip connections is that, when dealing with deep networks, gradients can become increasingly small after each layer and this could prevent the network from learning. This problem is known as *gradient degradation*. Residual connections help in attenuating this problem, by providing an alternative path for gradient flow, without experiencing the degradation problem. A block of layers with a residual connection is called *residual block*. Residual blocks are repeated multiple times along the depth of the network. The original paper provides different ResNet variants, containing up to 152 layers. Resnet-18 and ResNet-50 were used in this work, with 18 and 50 layers respectively.

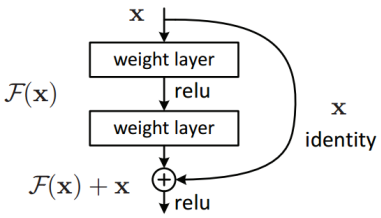


Figure 2.1.3: A residual block. Credits to He et al. (2016)

DenseNet DenseNet (Huang et al., 2017) is a widely used network architecture, which elaborates on the residual connection idea proposed by ResNet, taking it a step further. Figure 2.1.4 shows an example of DenseNet architecture. Every convolutional layer is connected to all of the following convolutional layers with skip connections. The major difference between DenseNet and ResNet is that skip-connections are implemented by concatenation rather than addition. This allows for a better *feature reuse* as each feature map will be used as additional input by the subsequent layers. DenseNet architectures achieve better performance than ResNet on *ImageNet* and a number of other datasets. In this work, the popular DenseNet-121 (containing 121 layers) was used.

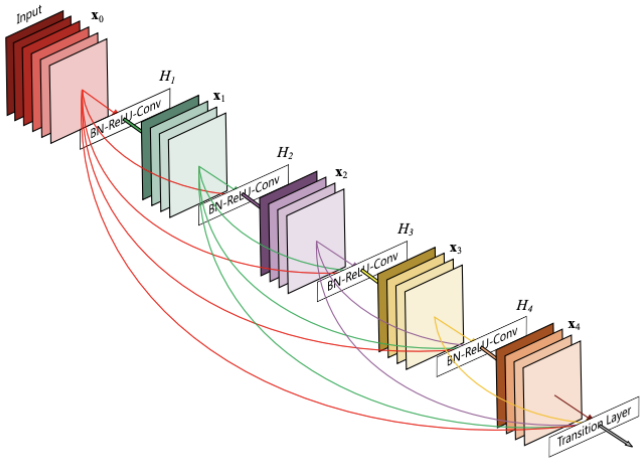


Figure 2.1.4: DenseNet architecture. Credits to Huang et al. (2017).

Transfer Learning

Some of the networks used in this work have been pre-trained on *ImageNet* (Deng et al., 2009) and later on different datasets. This pre-training step is commonly employed in deep learning tasks, as it provides a good initialization for the model encoder. Many of the features that the network has learned to extract can in fact be re-used successfully on different tasks (Pan and Yang, 2009). This approach is commonly known as *transfer learning* and has been exploited multiple times in this work. When using transfer learning, the "knowledge" that the network gained could prove to be useful on a different (but related) problem. Usually, transfer learning is applied to the network encoder, as the fully-connected classifier is replaced with one suited for the new task. Also, depending on the task at hand and the size of the training dataset, different choices on how to implement transfer learning can be taken: when switching to another problem, the entire network could be re-trained on the new data (also known as *finetuning*), or, conversely, the encoder could be *frozen*, meaning that only the new fully-connected classifier will be trained.

2.1.3 Contrastive Learning

In recent years, the topic of deep representation learning has increasingly gained traction in the machine learning community. Contrastive Learning (CL) has become the most widespread approach for this purpose, and many losses and frameworks have been proposed (Chen et al., 2020; Khosla et al., 2020; Oord et al., 2019; Poole et al., 2019). In short, Contrastive Learning approaches aim at pulling positive samples' representations (e.g. of the same class) closer together while repelling representations of negative ones (e.g. different classes) apart from each other. Contrasting positive pairs against negative ones is an idea that dates back to previous research (Hadsell et al., 2006; Oord et al., 2019; Tian et al., 2020) and has seen various applications in different tasks, such as face recognition (Schroff et al., 2015). Within the different proposed Contrastive Learning methods, we can identify two

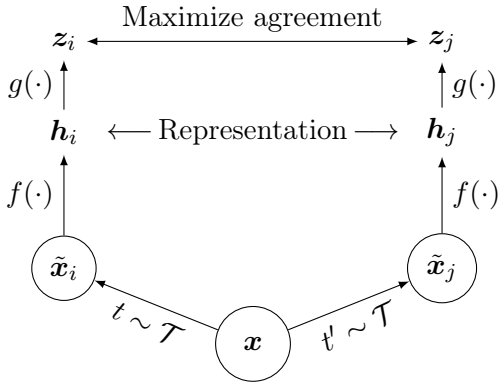


Figure 2.1.5: SimCLR framework. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlated views. b A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. Credits to Chen et al. (2020).

prominent approaches, self-supervised with SimCLR (Chen et al., 2020) and supervised with SupCon (Khosla et al., 2020).

Self-supervised CL SimCLR is designed for learning powerful representations from unlabeled data. It employs strong data augmentation techniques, creating multiple augmented versions of an input image. These augmented samples are then passed through a deep neural network, typically based on architectures like ResNet. The contrastive loss function encourages the model to bring representations of similar data points closer together while pushing apart representations of dissimilar data points. The idea of aligning representations of samples through small transformations actually dates back to 1992 (Becker and Hinton, 1992). Figure 2.1.5 provides an overview of the SimCLR framework: an image x is transformed by applying a set of augmentations T, obtaining two different views $\tilde{\boldsymbol{x}}_i$ and $\tilde{\boldsymbol{x}}_j$ (positive samples). The agreement between the latent representation of the positive samples is then maximized with the InfoNCE loss, also known as NT-Xent:

$$\mathcal{L}_{i,j}^{InfoNCE} = -\log \frac{\exp(sim(z_i, z_j)/\tau)}{\sum_{k \neq i} \exp(sim(z_i, z_k)/\tau)} \tag{2.1.12}$$

where z_i and z_j are the normalized latent representation of the positive pair, $sim(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T\boldsymbol{v}/\|\boldsymbol{u}\|\|\boldsymbol{v}\|$ is the cosine similarity function, and τ is a temperature parameter. In this framework, all the samples $k \neq i$ in the minibatch are considered negatives, thus their agreement is minimized by the loss function. Indeed, Eq. 2.1.12 can be decomposed into two separate terms:

$$\mathcal{L}_{i,j}^{InfoNCE} = \underbrace{-\log\left(\exp(sim(z_i, z_j)/\tau\right)}_{alignment} + \underbrace{\log \sum_{k \neq i} \exp(sim(z_i, z_k)/\tau)}_{uniformity} \tag{2.1.13}$$

known as alignment and uniformity (Dufumier et al., 2021a; Wang and Isola, 2020). Alignment pushes the encoder to encode similar samples with similar features, while uniformity favors a distribution on the hypersphere that preserves maximal information, such as a uniform distribution. The result of this process is that samples

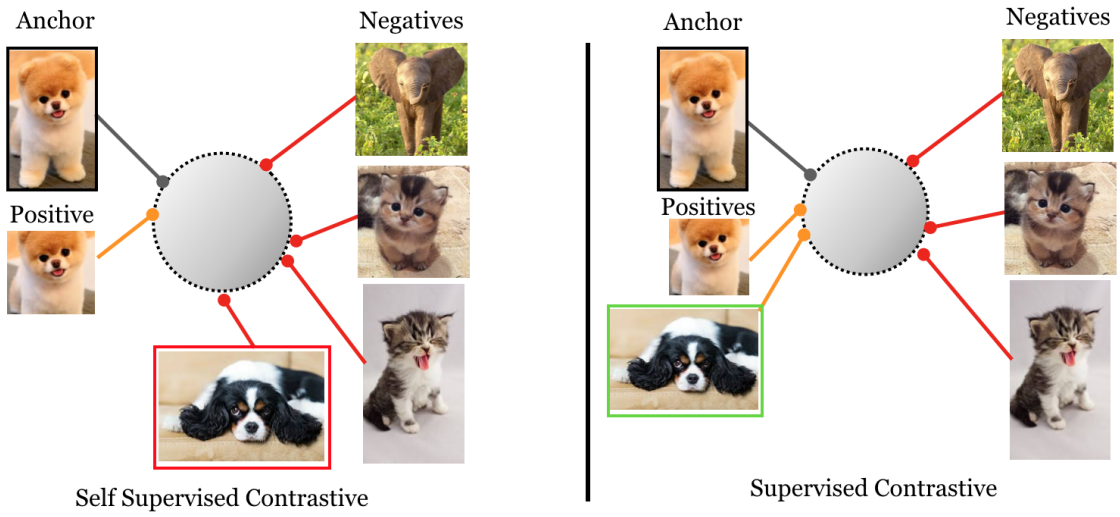


Figure 2.1.6: Difference between self-supervised and supervised contrastive learning. With SupCon, samples from the same class are closer in the latent space than with SimCLR. Credits to Khosla et al. (2020).

are distributed in the latent space according to their similarity in the input space, which, for self-supervised learning is strongly dependent on the chosen augmentation scheme. Picking the best kind of augmentations or finding alternative solutions when applying data augmentation is not trivial (e.g. medical images) and is an active area of research (Dufumier, 2022; Dufumier et al., 2023). Chapter 9 will briefly mention this topic. In the main parts of this thesis, however, we will not deal with this issue, as we will focus on supervised learning. Self-supervised contrastive approaches like SimCLR are usually employed for pre-training on large datasets, with subsequent finetuning on a downstream task such as classification or regression.

Supervised CL SupCon (which stands for Supervised Contrastive) incorporates labels from the original task (e.g., classification labels) to guide the learning process. While data augmentation can still be applied, SupCon mainly leverages the labels for defining positive and negative samples in the contrastive loss computation. The SupCon loss is proposed as an extension of the InfoNCE loss employed by SimCLR:

$$\mathcal{L}_i^{SupCon} = -\frac{1}{|P(i)|} \sum_{j \in P(i)} \log \frac{\exp(sim(z_i, z_j)/\tau)}{\sum_{k \in A(i)} \exp(sim(z_i, z_k)/\tau)} \tag{2.1.14}$$

where $P(i)$ is the set of indices of all positive samples of i, and $A(i)$ is the set of indices of all the other samples in the minibatch, excluding the positive sample generated by augmentation. Including all positive samples based on the actual label in the alignment term helps in creating a better-organized representation space, as illustrated in Figure 2.1.6. Furthermore, it has been shown that SupCon can outperform standard optimization using cross-entropy (Khosla et al., 2020), and is also more robust against label corruption (Graf et al., 2021) which could perhaps be seen as an instance of collateral learning.

CL for regression Few works have tackled regression tasks with contrastive learning approaches, as the definition of positive and negative samples is rooted in the CL framework. For regression problems, however, no such hard distinction can be made as the target variable is continuous. A first approach that tries to avoid this issue can be found in Xue et al. (2022). In this work, a threshold is used to determine positive and negative samples based on the difference of target labels, then the SupCon loss is used. The main shortcoming of this approach is that the threshold has to be chosen manually. A relevant approach that avoids thresholding is y-Aware (Dufumier et al., 2021a,b). Here, a contrastive loss is weighted by continuous meta-data (in a weakly supervised setting); however, the final aim of this method is to obtain a latent space suitable for downstream classification tasks, and not for regression. A detailed comparison with this method will be provided in Chapter 6. A similar approach is also proposed by Wang et al. (2022) for eye gaze estimation, where the alignment term is conditioned by the similarity between gaze directions. Another relevant approach can be found in Zha et al. (2022), where the distance of the labels of two samples is used to condition the uniformity term.

Why CL is relevant The contrastive learning approach is especially relevant for this thesis as it can be analyzed analytically, thus providing a more in-depth understanding of the optimization process. For this purpose, in this thesis, we will adopt a metric learning point of view, allowing us to precisely define the goal of the learning process using simple, yet powerful, metric constraints. From there, we will be able to derive both InfoNCE and SupCon losses, characterize their behavior, and, most importantly, propose novel contrastive losses that can help mitigate the Collateral Learning problem.

2.2 Debiasing

As we have seen in the Introduction (1), one of the most important instances of Collateral Learning is represented by biased data. Addressing the issue of biased data and how it affects neural network generalization has been the subject of numerous works. Back in 2011, Torralba et al. (2011) showed that many of the most commonly used datasets are affected by biases. In their work, they evaluate the cross-dataset generalization capabilities based on different criteria, showing how data collection could be improved. With a similar goal, Tommasi et al. (2017) propose different benchmarks for cross-dataset analysis, aimed at verifying how different debiasing methods affect the final performances. Data collection should be carried out with great care, in order not to include unwanted biases. Leveraging data already publicly available could be another way of tackling the issue. Gupta et al. (2018) explore the possibility of reducing biases by exploiting different data sources, in the practical context of sensors-collected data. They propose a strategy to minimize the effects of imperfect execution and calibration errors, showing improvements in the generalization capability of the final model.

Khosla et al. (2012) employ max-margin learning (SVM) to explicitly model dataset bias for different vision datasets. Beutel et al. (2019) provide insights on algorithmic fairness in a production setting, and propose a metric named *conditional equality*. They also propose a method, absolute correlation regularization, for optimizing this

metric during training. Another possibility of addressing these issues on a data level is to employ generative models, such as GANs (Goodfellow et al., 2014), to clean up the dataset with the aim of providing fairness (Sattigeri et al., 2018; Xu et al., 2018). Madras et al. (2018) also employ a GAN to obtain fair representations.

All the above-mentioned approaches generally deal directly at the data level and provide useful insights for designing more effective debiasing techniques. In the related literature, we can most often find debiasing approaches based on ensembling methods, adversarial setups, or regularization terms which aim at obtaining an *unbiased* model using *biased* data. We distinguish three different classes of approaches, in order of complexity: those that need full explicit supervision on the bias features (e.g. using bias labels), those that do not need explicit bias labels but leverage some prior knowledge of the bias features, those which no dot need neither supervision nor prior-knowledge.

2.2.1 Supervised approaches

Among the relevant related works, debiasing techniques which are supervised, meaning that they require explicit bias knowledge in the form of labels, can be most commonly found. The most widespread approach is to use an additional bias-capturing model, with the task of specifically capturing bias features. This bias-capturing model is then leveraged, either in an adversarial or collaborative fashion, to enforce the selection of unbiased features on the main model. The typical multi-model approach for debiasing can be formally described as follows: given a shared encoder $f(\cdot)$, a target classifier $g(\cdot)$, and a bias classifier $d(\cdot)$, the goal is to optimize the following objectives:

$$\begin{aligned} \mathcal{L}_{primary} &= \mathcal{L}_{CE}(y, g(z)) + (1 - \mathcal{L}_{CE}(b, d(z))) \\ \mathcal{L}_{bias} &= \mathcal{L}_{CE}(b, d(z)) \end{aligned} \tag{2.2.1}$$

where $z = f(x)$, and y and b are the target and bias labels respectively. By alternating the optimization of the two objectives, this formulation forces the encoder to encode samples into latent representations z that do not contain bias features. The idea behind this kind of optimization is shared by different works, some of which focus more on adversarial learning whereas others on making the predictions of the target and bias classifier independent from each other (e.g. orthogonal).

We can find the typical supervised adversarial approach in the work by Alvi *et al.*: BlindEye (Alvi et al., 2018). They employ an explicit bias classifier, which is trained on the same representation space as the target classifier, using a min-max optimization approach. In this way, the shared encoder is forced to extract unbiased representations. Similarly, Kim et al. (2019) propose Learning Not to Learn (LNL), which leverages adversarial learning and gradient inversion to reach the same goal. Adversarial approaches can be found in many other works, for example in Wang et al. (2019b), where they show that biases can be learned even when using balanced datasets, and they adopt an adversarial approach to remove unwanted features from intermediate representations of a neural network. Also, Xie et al. (2017) propose an adversarial framework for learning invariant representations with respect to some attribute in the data, similarly to Alvi et al. (2018). Moving away from adversarial

approaches, Wang et al. (2020b) perform a thorough review of the related literature, and propose a technique based on an ensemble of classifiers trained on a shared feature space. A similar approach is followed by Clark *et al.* with LearnedMixin (Clark et al., 2019). They train a biased model with explicit supervision on the bias labels, and then they build a robust model forcing its prediction to be made on different features.

Another possibility is represented by the application of adjusted loss functions or regularization terms. For example, Sagawa *et al.* propose Group-DRO (Sagawa et al., 2019), which aims at improving the model performance on the *worst-group* in the training set, defined based on prior knowledge of the bias distribution. Generally, in this context, the objective function has a form similar to the following:

$$\mathcal{L} = \mathcal{L}_{CE}(y, g(z)) + R(z, y, b) \tag{2.2.2}$$

where R is a regularization term that tries to force the invariance to bias features in the latent space. The debiasing methods that will be proposed in this thesis fall into this latter category.

2.2.2 Prior-guided approaches

In many real-world cases, explicit bias labels are not available. However, it might still be possible to make some assumptions or have some prior knowledge about the nature of the bias. Bahng et al. (2020) propose ReBias, an ensembling-based technique. Similarly to the work presented earlier, they build a bias-capturing model (an ensemble in this case). The prior knowledge about the bias is used in designing the bias-capturing architecture (e.g. by using smaller receptive fields for texture and color biases). The optimization process consists in solving a min-max problem with the aim of promoting independence between the biased representations and the unbiased ones. A similar assumption for building the bias-capturing model is made by Cadene et al. (2019) with RUBi. In this work, logits re-weighting is used to promote independence of the predictions on the bias features. Borrowing from domain generalization techniques, another kind of approach aiming at learning robust representation is proposed by with HEX (Wang et al., 2019a). They propose a differentiable neural-network-based gray-level co-occurrence matrix (Haralick et al., 1973; Lam, 1996), to extract biased textural information, which is then employed for learning invariant representations. A different context is presented by Hendricks et al. (2018). They propose an Equalizer model and a loss formulation that explicitly takes into account gender bias in image captioning models. In this work, the prior is given by annotation masks indicating which features in an image are appropriate for determining gender. Related to this approach, another possibility is to constrain the model prediction to match some prior annotation of the input, as done in the work of Ross et al. (2017), where gradients re-weighting is used to encourage the model to focus on the right input regions. Similarly, Selvaraju et al. (2019) propose HINT, which optimizes the alignment between manual visual annotation and gradient-based importance masks, such as Grad-CAM (Selvaraju et al., 2017).

2.2.3 Unsupervised approaches

Increasing in complexity, we consider as unsupervised approaches those methods that do not

- require explicit bias information,
- use prior knowledge to design specific architectures.

In this setting, building a bias-capturing model is a more difficult task, as it should rely on more general assumptions. For example, Nam et al. (2020) propose a technique named Learning from Failure (LfF). They exploit the training dynamics: a bias-capturing model is trained with a focus on *easier* samples, using the Generalized Cross-Entropy (Zhang and Sabuncu, 2018) (GCE) loss, which are assumed to be aligned with the bias, while a debiased network is trained emphasizing samples which the bias-capturing model struggles to learn. Similar assumptions are also made by Luo et al. (2022) where GCE is also used for dealing with biases in a medical setting using Chest X-ray images. Ji et al. (2019) propose an unsupervised clustering method that is able to learn representations invariant to some unknown or "distractor" classes in the data, by employing over-clustering. Although not strictly for debiasing purposes, another clustering-based technique is proposed by Van Gansbeke et al. (2020): they employ a two-step approach for unsupervised learning of representations, where they mine the dataset to obtain pseudo-labels based on neighbor clusters. A recently proposed approach can be found in Nam et al. (2022) with SSA. Here, the authors propose to assign pseudo-labels based on biased clusters, similarly to the method proposed in Section 5.1. However, in order to do so, they still require a small set with bias annotations.

2.3 Medical Imaging

In this section we provide an overview of the related works in the field of medical imaging, focusing on Neuroimaging and Chest X-Ray (CXR) images. The main focus of this thesis is dealing with collateral learning, which, in medical imaging, is often represented by site-effect and domain generalization issues, as already presented in Section 1.2.2. In the context of Neuroimaging, we will focus on structural brain Magnetic Resonance Images (MRIs) with the aim of building robust brain age prediction models. This task has gained relevance in the field, as accurate age estimatation can enable the detection of cognitive decline and neurodegeneration. For what concerns CXR images, we will deal with the detection of Covid-19. This has of course a great relevance, given the recent pandemic.

2.3.1 Neuroimaging

Neuroimaging is a branch of medical imaging that focuses on the brain. It is an important tool to diagnose diseases and brain health, and for studying and analyzing how the brain works and responds to different activities. There are two main categories of imaging techniques:

- Structural imaging, which is used to analyze the brain structure (e.g. sMRI);

- Functional imaging, which is used to analyze the brain function (e.g. fMRI, PET, MEG).

In this thesis, we will focus on structural brain Magnetic Resonance Images (MRI). The structural information of the brain is the basis for providing a diagnosis or a prediction such as brain age, so we give here a small overview of the human brain anatomy[4]. The brain can be divided into three high-level parts: the the brainstem, the cerebellum, and the cerebrum. The latter is the largest part and comprises gray matter (also called the cerebral cortex) and white matter (at the center). As shown in Figure 2.3.1, the cerebral cortex is divided into four sections, called lobes. Each lobe is responsible for specific functions:

- **Fontal lobe.** This is the largest lobe, and is involved in the determination of the personality, decision-making and movement. It also contains Broca's area, which is associated with speech ability.
- **Parietal lobe.** This lobe is responsible for visual object identification, understanding spatial relationships, and interpreting stimuli such as pain and touch. It also contains Wernicke's Area, which is associated with speech understanding.
- **Occipital lobe.** This lobe is involved with vision.
- **Temporal lobe.** This lobe is involved in short-term memory, speech processing, and other skills such as musical rhythm.

MRI uses magnetic fields and radio waves to produce three-dimensional images of the brain structures, without employing ionizing radiation (X-rays). The resolution of the image is determined by the strength of the magnetic field. It is able to measure gray matter structure (cerebral cortex). The output of an MRI scan is a 3D volume composed of voxels, from which different measurements can be derived, such as cortical thickness, grey matter density, and others. Such information can be relevant for diagnosing brain disorders such as Alzheimer's Disease, Schizophrenia, Bipolar Disorder, etc. Furthermore, from MRI scans, it is possible to predict the *BrainAGE* (Franke and Gaser, 2019; Franke et al., 2010) which has become a very important indicator of brain health in neuroimaging.

Brain Age Prediction

During a healthy aging process, the brain changes due to progressive and regressive neuronal changes, following a specific pattern: for example, gray matter shows an increase in volume from birth to the age of four, and then progressively decreases until around 70; white matter increases until around 20 years, from which it remains constant; cerebrospinal fluid increases steadily from after 20 years (Pfefferbaum et al., 1994). On the other hand, neurodegenerative diseases and brain conditions show an altered aging pattern. For this reason, accurately modeling the healthy aging of the brain is important. BrainAGE (Franke et al., 2010) was proposed as an automatic method for estimating the age of healthy subjects in T1-weighted MRI

[4] https://www.hopkinsmedicine.org/health/conditions-and-diseases/anatomy-of-the-brain

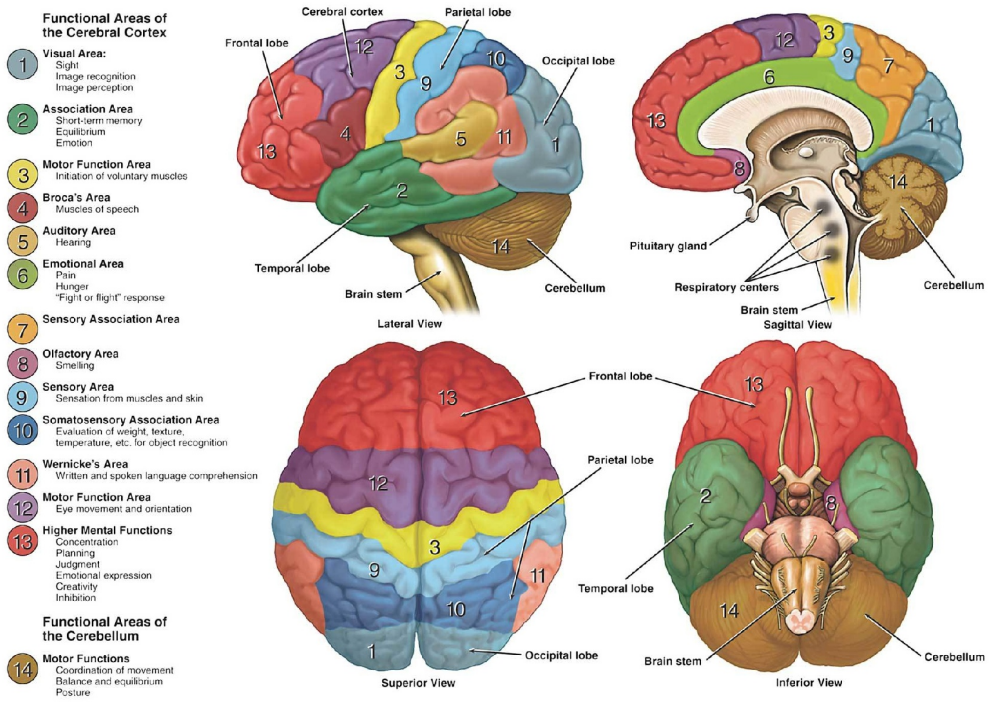


Figure 2.3.1: Brain anatomy. Credits to Sukel (2019).

scans, using a kernel method for regression. It consists of a preprocessing step of the images, dimensionality reduction with Principal Component Analysis (PCA), and age prediction with a Relevance Vector Machine (RVM) (Tipping and Bishop, 2000; Tipping, 2001). It achieved a Mean Absolute Error (MAE) of 5 years on healthy subjects in aged 19-86 years. BrainAGE has been subsequently validated and employed by many works in the field (Franke and Gaser, 2019), and it now represents a relevant marker for assessing healthy aging of the brain. With the advent of Deep Learning, arguably, the process can be reduced to training a feed-forward neural network minimizing a simple supervised loss such as L1. In this thesis, we will adopt this approach, as it also removes the need for explicit dimensionality reduction in the input data.

Accelerated aging and Alzheimer's Disease detection

Neurodegenerative disorders and brain conditions such as Alzheimer's Disease (AD) exhibit an altered (e.g. accelerated) brain aging process. Comparing the chronological age of a patient with their predicted brain age can be a good indicator of whether the aging process is following a healthy path, or shows possible alterations. Franke and Gaser (2019) show that AD patients exhibit on average a brain age delta (difference between brain age and chronological age) of around 6 to 10 years, while healthy patients show no significant gap, during both preliminary and follow-up acquisitions. Additionally, patients showing progressive Mild Cognitive Impairment (MCI) showed an increase in age gap across follow-ups, depending on whether their condition was stable (sMCI) or progressive (pMCI), leading to AD in the latter case. This makes building robust models for accurate modeling the brain aging a very relevant topic in the field.

Of course, with the aim of detective conditions such as AD or MCI, other approaches not tied to brain aging were proposed. Wen et al. (2020) provides a comprehensive review of the recent state-of-the-art on deep learning for the classification of such diseases. Most of these works leverage convolutional neural networks, with the aim of directly predicting a final diagnosis form the input MRI scan (either using the whole 3D volume or 2D slices). The most predominant classification task found is Healthy Cases (HC) vs AD, followed by the differentiation of MCI cases from HC. A less frequent, but clinically relevant task, is distinguishing pMCI subjects from sMCI. In this thesis, we will focus on the first two tasks, leaving the last as future work.

Site-effect in neuroimaging data

In Franke et al. (2010), authors show that the BrainAGE method exhibits some robustness to the influence of different scanners in the data. This is indeed a relevant problem in the neuroimaging field (Chen et al., 2022; Fortin et al., 2016; Glocker et al., 2019; Nguyen et al., 2018), as different scanners can influence the resulting image and thus have an effect on the model prediction. We explained the issue of site effect in the Introduction section (1.2.2). One of the most common methods for dealing with this issue in neuroimaging is ComBat (Fortin et al., 2017), a data harmonization method that was originally developed for genomics data (Johnson et al., 2007). In this thesis, we will compare to the ComBat baseline, in order to assess the robustness of our methods towards Collateral Learning.

2.3.2 Chest X-ray and Covid-19

In this Section, we provide a brief overview of the main works on the topic of deep learning diagnosis from CXR images, specifically on Covid-19 detection.
Previous to the Covid-19 pandemic, the topic of DL diagnosis from CXRs was already of interest in the scientific community. For example, in Shin et al. (2016) CNNs are investigated for the classification of interstitial lung disease (ILD). Other works also showed that deep learning can be used to detect and classify ILD tissue (Anthimopoulos et al., 2016; Bondfale and Bhagwat, 2018). Anthimopoulos et al. (2016) focus on designing a CNN tailored to match the ILD CT texture features, e.g. small filters and no pooling to guarantee spatial locality. Other contributions focus on the classification of CXRs for SARS diagnosis (Xiaoou Tang et al., 2004; Xie Xuanyang et al., 2005).

The issue of Collateral Learning for Covid-19 was particularly evident in the early phases of the pandemic, as the community rushed to develop DL-based diagnostic systems. Some of the proposed approaches leveraged transfer learning and publicly available data (Apostolopoulos and Mpesiana, 2020; Narin et al., 2020; Sethy and Behera, 2020) to achieve reasonable performance on Covid-19 diagnosis. Wang and Wong (2020) represented one of the most relevant approaches, proposing a novel neural network architecture named COVID-Net.
However, the main issue of all these approaches was represented by the scarcity of available data. They typically employed the COVID-ChestXRay dataset (Cohen et al., 2020), consisting of, at the time, approximately 100 CXR Covid-19 cases.

Furthermore, in order to build Covid-19 negative cases, data were sampled from other datasets, such as the Kermany dataset (Kermany, 2017). However, as we will discuss in detail in Chapter 7, this introduces a number of issues related to Collateral Learning. Some of these were related to transfer learning, as the choice of the pretraining task plays a relevant role in the final accuracy. Also, the most widely used datasets did not contain exhaustive metadata about the population (e.g. gender or age) and this could lead to models exploiting hidden biases.

Following research, including our own contributions, helped raise awareness on this issue, by providing recommendations for employing DL to detect Covid-19 from CXR images (Roberts et al., 2021). In this thesis, we will retrace the development of this field, which represents a real-world example of how Collateral Learning should be taken into account when developing DL tools.

Part II

Collateral Learning in Natural Images

Chapter 3

Debiasing Through Disentanglement

3.1 Introduction

In this Chapter, we present the first approach that we propose for dealing with the issue of collateral learning in biased data. Specifically, we deal with the issue of correlation shift, as explained in the Introduction (Section 1.2). The basic intuition behind this approach is that representations of biased samples tend to be naturally clustered together based on the common collateral features, which in this case represent the bias. The method we present in this section is a supervised method, thus it assumes that prior knowledge about the bias is available, in the form of labels. Although having this prior knowledge may seem unrealistic at first, some realistic scenarios of such occurrence can be encountered (they will be presented later). Furthermore, in Chapter 5, we will also present how to extend this approach to the unsupervised case.

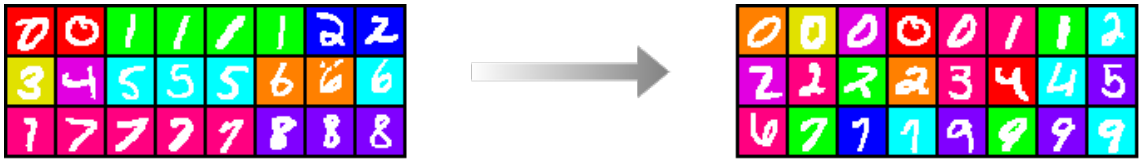


Figure 3.1.1: Biased-MNIST by Bahng et al. (2020). The bias is given by the correlation between digit and background color. This dataset is an example of correlation shift (1.2), as the color distribution C in the training set (S) and in the test set (T) is different, and is determined by the sample label y: $p_S(C|y) \neq p_T(C|y)$.

To visualize an example, we now introduce the Biased-MNIST dataset (Bahng et al., 2020) which will be used throughout the rest of this work as a first benchmark for the proposed methods. An example of Biased-MNIST is presented in Figure 3.1.1: in this dataset, the collateral information is represented by the color. This dataset is built upon the well-known MNIST dataset (Deng, 2012), by injecting color into the background of the images. The color is injected in such a way that there is a high degree of correlation with the different digits. Ten predefined colors are associated with the ten different classes. Given an image, the background is colored with the predefined color for that class with a probability ρ, and with any one of the other colors with a probability $(1 - \rho)$. Higher values of ρ will lead to more biased

data. In this work, we will experiment with different degrees of correlation in the training dataset. An *unbiased* test set is built with $\rho = 0.1$, meaning that, for any given digit, a random color is selected. Given the absence of correlation between color and digit class in the unbiased test set, a model must learn to classify shapes instead of colors, in order to reach a high accuracy on the unbiased test set. This is a very simple yet effective benchmark for assessing whether the model is learning collateral features, which should in fact be discarded or ignored in order to obtain robust representations.

3.2 Preliminary analysis

To assess the efficacy of this benchmark, and to show the natural tendency of neural networks to prefer simpler patterns, we train a vanilla model without any debiasing method, using the setup presented in Section 3.4.1. By analyzing the training process with different values of ρ, we can identify when the color bias shifts from being benign to malignant (as defined in Section 1.2). Figure 3.2.1 shows the training accuracy of a vanilla model trained with different values of ρ. Given that, in this case, the number of target classes and the number of different colors (bias classes) is the same, we are able to compute a bias *pseudo-accuracy* by finding the permutation of the predicted labels which maximizes the accuracy with respect to the ground truth bias labels: this value provides an indication of how the final predictions of the model are aligned with the bias. From Figure 3.2.1a we observe that the target accuracy on the training set is, as expected, close to 100%, while the bias accuracy is exactly the value of ρ, meaning that the models learned to recognize the digit. This holds true also for the unbiased test set (Fig 3.2.1b), where the value $\rho = 0.1$. However, if we focus on the higher end of ρ values (most difficult settings) as shown in Figure 3.2.1c, we observe a rapid inversion in the trend: the target accuracy decreases, dropping to 10% for $\rho = 0.999$, while the bias accuracy becomes higher, close to 100% towards the end of the ρ range. In these settings, given the strong correlation between target and bias classes, it is clear that the bias has become easier for the model to learn, and thus malignant.

These results can be viewed as further confirmation that neural networks tend to prefer and prioritize the learning of simpler patterns first, as noted by Nam et al. (2020); Shah et al. (2020) and especially by Arpit et al. (2017).

3.3 The EnD regularization

The first debiasing approach that we propose consists of a regularization term that aims at removing bias features from the learned representations, through means of entanglement and disentanglement across different samples. To give a basic idea of the intuition behind this approach, let us take Biased-MNIST as an example. If we consider two "8" with two different background colors (e.g. purple and orange), our goal is to force the entanglement between their latent representations in such a way that the common features (i.e. the digit) will be predominant. We name this technique *EnD* (from Entangling and Disentangling).

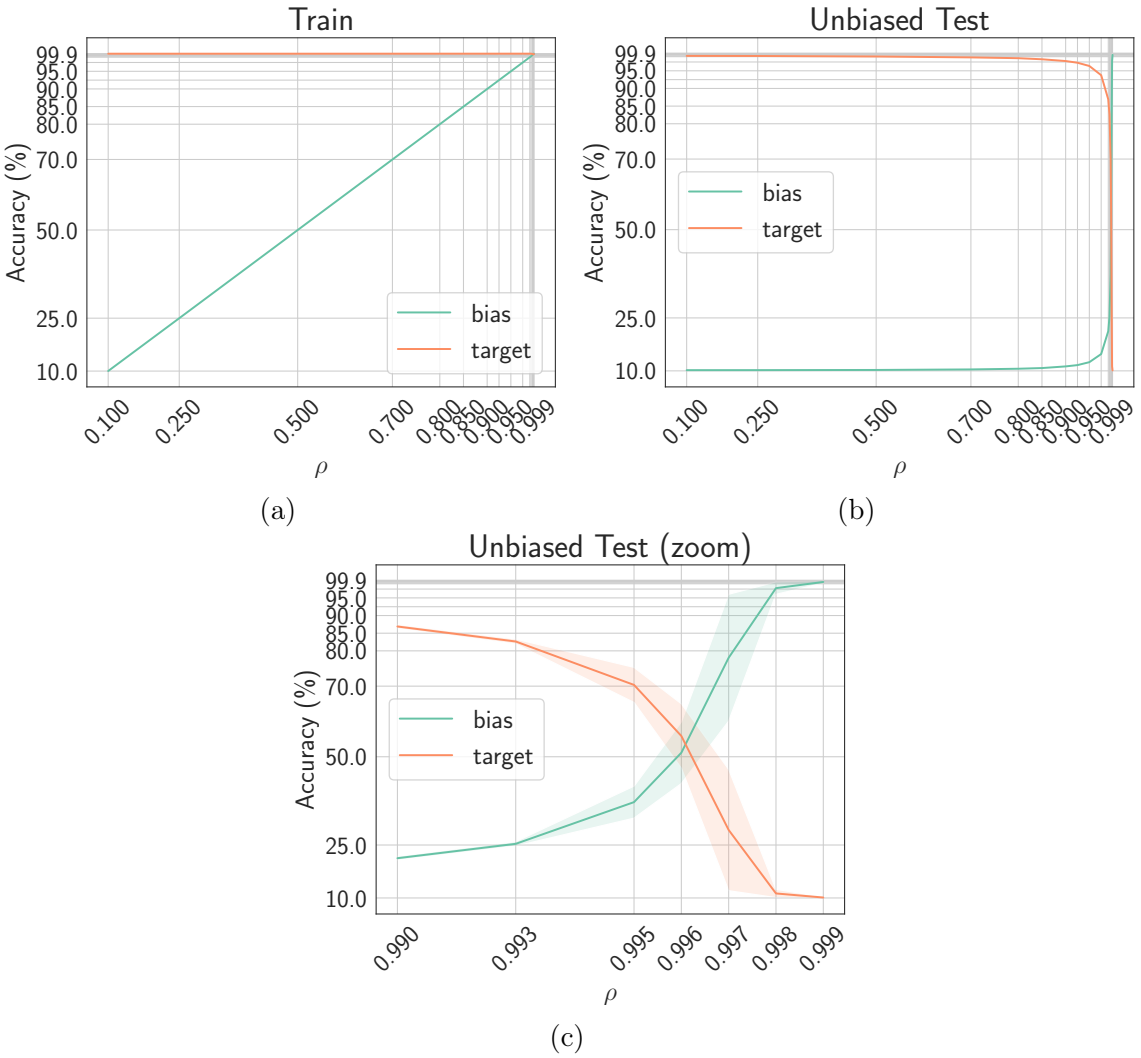


Figure 3.2.1: Effect of varying bias strength (ρ) on the model, on the training set (a), and on the unbiased test set (b) and (c). Results are reported in terms of mean and std across three different runs for every value of ρ. Given that the number of bias classes (colors) and target classes (digits) is the same, we can compute the bias accuracy by finding the permutation of predicted labels which maximizes the overlap with the ground truth bias labels. From (c) we can observe when the color bias really starts affecting the classification performance of the model, turning into a malignant bias. From around $\rho = 0.99$, models start making their predictions based on the color.

Our goal is to train a model to correctly classify the data into the T possible classes but at the same time prevent the use of the bias features contained in the data. Toward this end, we are going to build our regularization strategy, which consists of two terms:

- a *disentangling* term, whose task is to try to de-correlate as much as possible the representations of all the samples belonging to the same bias class b;
- an *entangling* term, which attempts to force correlations between the representation of samples from different bias classes but having the same target class t.

3.3.1 Method

Given a neural network *encoder* $f : \mathcal{X} \to R^N$ which extracts feature vectors of size N and a *classifier* $g : R^N \to \mathbb{N}$ which provides the final prediction, we consider the neural network $(g \circ \gamma \circ f)(\cdot)$ where $\gamma : \mathbb{R}^N \to \mathbb{R}^N$ is a normalization function to obtain $z = \gamma(x) = x/\|x\|_2$. The EnD regularization term $\mathcal{R}^{end}$ is applied jointly with the loss function $\mathcal{L}$ (e.g. cross-entropy), forcing $(\gamma \circ f)(\cdot)$ to filter out biased features from the extracted representation z. Hence, the overall objective function we aim to minimize is

$$\mathcal{J} = \mathcal{L} + \mathcal{R}^{end}, \tag{3.3.1}$$

where $\mathcal{R}^{end}$ is the sum of the disentangling and entangling terms, weighted by two hyper-parameters $\alpha \geq 0$ and $\beta \geq 0$:

$$\mathcal{R}^{end} = \alpha\mathcal{R}^{\perp} + \beta\mathcal{R}^{\parallel} \tag{3.3.2}$$

Within a mini-batch, let $i \in I \equiv \{1 \dots M\}$ be the index of an arbitrary sample x_i. We define y_i, t_i and b_i as the predicted, ground truth target and bias label for the i-th sample, respectively. The disentangling term $R^{\perp}$ is defined, for the i-th sample, as:

$$\mathcal{R}_i^{\perp} = \frac{1}{|B(i)|} \sum_{a \in B(i)} |z_i \cdot z_a| \tag{3.3.3}$$

where $B(i) := \{j \in I \mid b_j = b_i\} \setminus \{i\}$ is the set of all samples sharing the same bias class of x_i, which are commonly named as *bias-aligned* in the related literature. The goal of this term is to suppress the common features among bias-aligned samples. The entangling term $R^{\parallel}$ is defined, for the i-th sample, as:

$$\mathcal{R}_i^{\parallel} = -\frac{1}{|J(i)|} \sum_{j \in J(i)} z_i \cdot z_j \tag{3.3.4}$$

where $J(i) := \{j \in I \mid t_j = t_i\} \setminus B(i)$ is the set of all samples sharing the same target class of x_i but with different biases, also known as *bias-conflicting*. Complementarily to the disentangling term, the goal of this term is to encourage correlation bias-conflicting samples of the same target class, in order to introduce invariance with

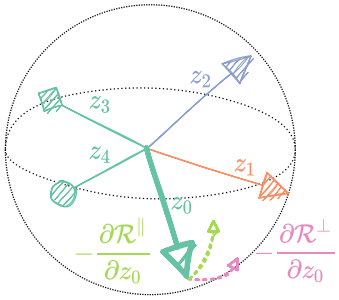


Figure 3.3.1: Effect of EnD: representations of bias-conflicting samples of the same target class (represented by the arrow shape), with respect to z_0, are entangled through the $R^{\parallel}$ term, bias-aligned samples (represented by the arrow color) are disentangled through the $R^{\perp}$ term.

respect to the biased features. So, for the i-th sample, the entire EnD regularization term $\mathcal{R}_i^{end}$ can be written as:

$$\mathcal{R}_i^{end} = \alpha \frac{1}{|B(i)|} \sum_{a \in B(i)} |z_i \cdot z_a| - \beta \frac{1}{|J(i)|} \sum_{j \in J(i)} z_i \cdot z_j \,. \tag{3.3.5}$$

The final $\mathcal{R}^{end}$ of Eq. 3.3.2 is then just computed as the average over the mini-batch:

$$\mathcal{R}^{end} = \frac{1}{M} \sum_i \mathcal{R}_i^{end} \tag{3.3.6}$$

To visualize the effect of $\mathcal{R}^{end}$ as expressed in Eq. 3.3.5, consider a simple classification problem with three target classes and three different bias as illustrated in Figure 3.3.1. Training a model without explicitly addressing the presence of biases in the data, will most likely results in representations aligned by the bias attributes rather then the actual target class (Figure 3.3.1). The goal of $\mathcal{R}^{end}$ is to encourage the alignment of representations based on the correct features by *i.)* disentangling representations of the same bias ($\mathcal{R}^{\perp}$) and *ii.)* entangling representations of the same target in order to introduce invariance to the bias features ($\mathcal{R}^{\parallel}$).

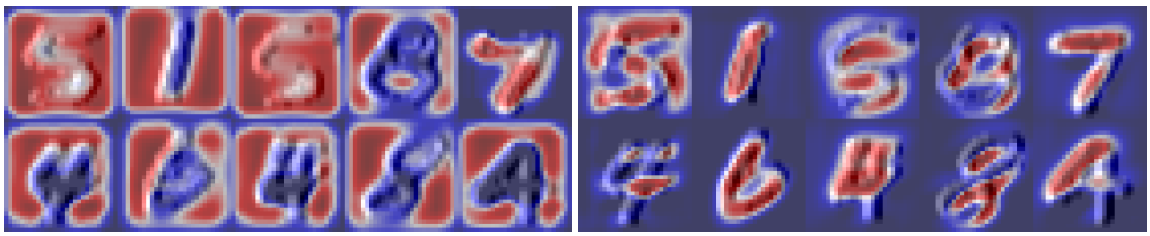
3.4 Experiments

In the experiments we present in this section, we aim to remove different types of biases such as color, age, gender which can have a high impact on classification performance when recognizing, for example, attributes such as hair color and presence of makeup on facial images. In all the results tables, the best results are denoted as boldface, the second best results are underlined. "Vanilla" denotes the baseline model performance for the learning problem, with no debiasing technique applied. All the EnD's results are averaged over three different runs. In our experiments, EnD is always applied after the network's encoder ($\gamma \circ f$), which is typically a bottleneck: this is a reasonable choice in order to exploit the whole encoder to extract unbiased features[1].

[1]The source code for the EnD techinque, including the Biased MNIST example, is publicly available and can be found at https://github.com/EIDOSlab/entangling-disentangling-bias.

Method	ρ values			
	0.999	0.997	0.995	0.990
Vanilla (Bahng et al., 2020)	10.40±0.50	33.40±12.21	72.10±1.90	<u>89.10</u>±0.10
LearnedMixin (Clark et al., 2019)	12.10±0.80	50.20±4.50	78.20±0.70	88.30±0.70
HEX (Wang et al., 2019a)	10.80±0.40	16.60±0.80	19.70±1.90	24.70±1.60
RUBi (Cadene et al., 2019)	13.70±0.70	43.00±1.10	<u>90.40</u>±0.40	<u>93.60</u>±0.40
ReBias (Bahng et al., 2020)	<u>22.70</u>±0.40	<u>64.20</u>±0.80	76.00±0.60	88.10±0.60
EnD	**52.30**±2.39	**83.70**±1.03	**93.92**±0.35	**96.02**±0.08

Table 3.4.1: Biased-MNIST accuracy on the unbiased test set. Reference results from Bahng et al. (2020). The best results are highlighted in bold, the second best results are underlined.



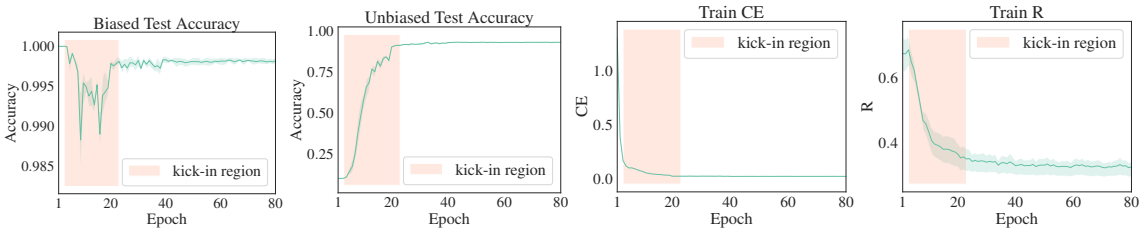
(a) (b)

Figure 3.4.1: Grad-CAM (Selvaraju et al., 2017) on Colored MNIST: vanilla model (a) and EnD-regularized model (b). Images were processed with an edge detection filter in order to improve the readability of the activation map.

3.4.1 Controlled experiments

In this section we describe the controlled experiments that we performed in order to assess the performance of EnD. Full control over the amount and type of bias allows to correctly analyze EnD's behavior, excluding noise and uncertainty given by real-world data.

We test our method on the Biased-MNIST dataset, where we can control the bias in the training data. To vary the level of difficulty in the dataset, we select $\rho \in \{0.990, 0.995, 0.997, 0.999\}$, as done in Bahng et al. (2020).



(a) (b) (c) (d)

Figure 3.4.2: EnD learning curves on Colored MNIST for $\boldsymbol{\rho}$**= 0.995**. Biased accuracy (a), unbiased accuracy (b), L value on the training set (c) and R value on the training set (d).

Experimental Setup

We use the network architecture proposed by Bahng et al. (2020), consisting of four convolutional layers with 7×7 kernels. The EnD regularization term is applied on the average pooling layer, before the fully connected classifier of the network. Following Bahng et al. (2020), we use the Adam optimizer with a learning rate of 0.0001, a weight decay of 10^{-4} and a batch size of 256. We train for 80 epochs. We do not use any data augmentation scheme. We use 30% of the training set as validation set, and we colorize it using a ρ value of 0.1. The EnD hyperparameters α and β are searched using the Bayesian optimization (Snoek et al., 2012) implementation provided by *Weights and Biases* (Biewald, 2020) on the validation set. For $\rho \in \{0.990, 0.995, 0.997\}$, α and β are searched in the interval $[0; 1]$, for $\rho = 0.999$ in $[0; 50]$. To provide a mean performance along with the standard deviation, we select the top 3 models based on the best validation accuracy obtained, and we report the average accuracy on the final test set.

Results

Results are shown in Table 3.4.1. EnD's results are averaged across three different runs for each value of ρ. For all values of ρ we report the accuracy obtained by EnD on the unbiased evaluation set, compared with other debiasing algorithms.

EnD successfully mitigates bias propagation. The improvement obtained with EnD with respect to the baseline model is noticeable, especially in the higher levels of difficulty. We observe an increase of accuracy across all values of ρ. Notably, for $\rho = 0.999$ the vanilla model reaches 10.4% accuracy, meaning that the background color is used as the only cue for classifying the digits, whereas employing EnD yields an accuracy of 52.30%. Figure 3.4.1 shows the effect of EnD, using Grad-CAM (Selvaraju et al., 2017) to highlight the important regions of the input image for the model prediction. We observe that the vanilla model (Figure 3.4.1a) focuses on the background, while the EnD-regularized model (Figure 3.4.1b) correctly learns to focus on the digit shape.

Comparison with other techniques. We observe that EnD yields the highest results among all of the compared debiasing algorithms. Such gap is especially higher in the most difficult settings for $\rho \in \{0.999, 0.997\}$ where many algorithms are unable to generalize to the unbiased set, especially HEX (Wang et al., 2019a) and LearnedMixin (Clark et al., 2019). Some of the compared algorithms even show a collapse in accuracy compared to the vanilla baseline in certain cases (HEX for most values of ρ, LearnedMixin and ReBias for $\rho = 0.990$).

Ablation study. We also perform an ablation study of EnD to analyze how each of the EnD's terms affect the performance of the trained model. For a fixed $\rho = 0.997$, we evaluate only the contribution of the disentangling term $R_\perp$ and disable the entangling term $R_\parallel$ by setting $\beta = 0$. We then perform the opposite evaluation by setting $\alpha = 0$, to only take into account the entangling term. The results are shown in Table 3.4.2. We observe that both the regularization terms contribute to boost the model's generalization capability. As expected, the best results are achieved when both of them are jointly applied. The entangling term yields a higher increase in

Setting	α	β	Unbiased accuracy
Vanilla	0	0	33.4
Disentangling only	$[0; 1]$	0	$45.67_{\pm 0.67}$
Entangling only	0	$[0; 1]$	$75.36_{\pm 0.94}$
EnD	$[0; 1]$	$[0; 1]$	$\mathbf{83.70}_{\pm 1.03}$

Table 3.4.2: Ablation study of EnD on the Biased MNIST dataset, $\rho = 0.997$.

performance compared to the disentangling one, however it is in general not always applicable. Given some i-th sample in a mini-batch, the entangling term can be applied if and only if:

$$\exists j, j \neq i \,|\, t_i = t_j \wedge b_i \neq b_j d \tag{3.4.1}$$

The bias's distribution over the training set and the batch size play an important role in the possibility of applying the entangling term on every update step. If there are dominant biases for specific target classes, this can be accounted for by clever batching (i.e. applying a weighted sampler). This would maximize the chances of satisfying the condition in equation 3.4.1. In our experiments, we applied the entangling term when the condition is satisfied. The disentangling term provides a smaller benefit in this case, but, on the other hand, it can always be applied. We find that the ideal case for EnD is when both of the terms can be used in the learning process, leading to better generalization capabilities. Furthermore, we observe a similar pattern in the learning process when employing the full EnD regularization for different values of ρ. Figure 3.4.2 shows the learning curves for $\rho = 0.995$. We notice how models tend to quickly learn the color bias in the first few epochs, as the accuracy on the biased test set is close to 100% (Figure 3.4.2a). However, once the value of the loss (in this case, we have used the cross-entropy loss, Figure 3.4.2c) falls below a certain threshold, the contribution R of the EnD term becomes predominant (Figure 3.4.2d). In this phase, which we call *kick-in region*, the optimization process begin to rapidly minimize R, stopping the model from relying on the bias-related features. This can be observed in the rapid increase of the accuracy on the unbiased test set (Figure 3.4.2b), whereas the biased accuracy momentarily drops as the models shift their focus from the background color to the digit shape.

3.4.2 Real world datasets

After benchmarking EnD in a controlled scenario on synthetic data, we move to real world datasets where biases might be subtle and harder to handle. In this section we aim at removing age and gender bias in different datasets. We also apply EnD on a computer-aided diagnosis task, where hidden biases might lead to sub-optimal generalization of the model.

Target	Method	Unbiased	Bias-conflicting
Hair Color	Vanilla	70.25±0.35	52.52±0.19
	Group DRO (Sagawa et al., 2019)	<u>85.43</u>±0.53	<u>83.40</u>±0.67
	LfF (Nam et al., 2020)	84.24±0.37	81.24±1.38
	EnD	**91.21**±0.22	**87.45**±1.06
Heavy Makeup	Vanilla	62.00±0.02	33.75±0.28
	Group DRO (Sagawa et al., 2019)	64.88±0.42	<u>50.24</u>±0.68
	LfF (Nam et al., 2020)	<u>66.20</u>±1.21	45.48±4.33
	EnD	**75.93**±1.31	**53.70**±5.24

Table 3.4.3: Performance on CelebA. Reference results from Nam et al. (2020). The best results are highlighted in bold, the second best results are underlined.

CelebA

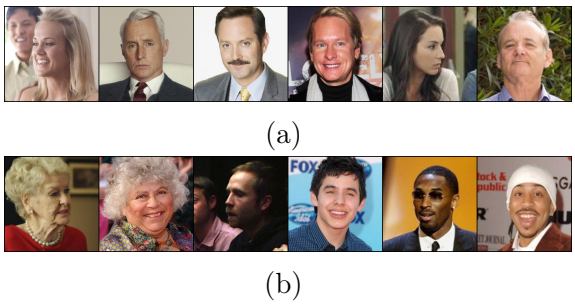
CelebA (Liu et al., 2015) is a dataset of for face-recognition tasks, providing 40 attributes for every image. Following Nam et al. (2020), we select *BlondHair* and *HeavyMakeup* as target attributes t and *Male* as bias attribute b. This choice is dictated by the fact that there is a high correlation between these attributes (i.e. most women have blond hair or wear heavy makeup in this dataset). The dataset contains a total of 202,599 images, and following the official train-validation split we obtain 162,770 images for training and 19,867 images for testing our models. Nam et al. (2020) build two types of testing dataset: *unbiased*, by selecting the same number of samples for every possible value of the pair (t, b), and *bias-conflicting*, by removing from the unbiased set all of the samples where b and t are equal.

Experimental Setup Following Nam et al. (2020), we use the Adam optimizer with a learning rate of 0.001, a batch size of 256, and a weight decay of 10^{-4}. We train for 50 epochs. Images are resized to 224×224 and augmented with random horizontal flip. To construct the validation set, we sample N images from each pair (t, b) of the training set, where N is 20% the size of the least populated group (t, b). The EnD hyperparameters α and β are searched using the Bayesian optimization (Snoek et al., 2012) implementation provided by *Weights and Biases* (Biewald, 2020) on the validation set, in the interval $[0; 50]$. To provide a mean performance along with the standard deviation, we select the top 3 models based on the best validation accuracy obtained, and we report the average accuracy on the final test sets.

Results. As in Nam et al. (2020), the accuracy is computed as average accuracy over all the (t, b) pairs. Table 3.4.3 shows the results obtained on the CelebA dataset. We observe how the vanilla model heavily relies on the bias attribute, scoring a low accuracy especially on the bias-conflicting sets. EnD, on the other hand, outperforms the baseline in both the tasks. We report reference results (Nam et al., 2020) of other debiasing algorithms, specifically Group DRO (Sagawa et al., 2019) and LfF (Nam et al., 2020), for comparison with EnD. The results we obtain are significantly higher across most of the evaluation sets, and comparable with Group DRO and LfF on the bias-conflicting set when the target attribute is HeavyMakeup.

IMDB Face

The IMDB Face dataset (Rothe et al., 2018) contains 460,723 face images annotated with age and gender information. To filter out the misannotated labels of this dataset (Rothe et al., 2018; Torralba et al., 2011), Kim et al. (2019) use a model trained on the Audience benchmark (Eidinger et al., 2014), keeping the images where the prediction matches the provided label. Following Kim *et al.*'s proposed data split, 20% of the IMDB is used as test set, containing samples with age 0-29 or 40+. The remaining data is then split into two extreme-bias subset: *EB1* contains women in the age range 0-29 and men with age 40+, while *EB2* contains men aged 0-29 and women 40+. Thus, when learning to predict the gender attribute, the bias is given by the age and vice-versa. An example of the EB1 and EB2 training sets is shown in Figure 3.4.3.



(a)

(b)

Figure 3.4.3: IMDB train splits: EB1 (a) and EB2 (b).

Experimental Setup We use the Adam optimizer with a learning rate of 0.001, a batch size of 256 and a weight decay of 10^{-4}. We train for 50 epochs. As with CelebA, images are resized to 224×224 and randomly flipped at training time for augmentation. In this case, it is not possible to construct a validation set including samples from both EB1 and EB2, without altering the test set composition. Hence, we perform a 4-fold cross validation for every experiment. For example, when training on EB1, we use one fold of EB2 as validation set and the remaining three folds as EB2 test set. We repeat this process until each EB2 fold is used both as validation and as test set. The same process is repeated when training on EB2, by splitting EB1 in validation and test folds. When training for age prediction, we follow Kim et al. (2019), by binning the age values in the intervals 0-19, 20-24, 25-29, 30-34, 34-39, 40-44, 45-49, 50-54, 55-59, 60-64, 65-69, 70-100, proposed by Alvi et al. (2018). For every fold, the EnD hyperparameters α and β are searched using the Bayesian optimization (Snoek et al., 2012) implementation provided by *Weights and Biases* (Biewald, 2020) on the validation set, in the interval $[0; 50]$, as in the previous experiments. To provide a mean performance along with the standard deviation, we select the top model for each fold, based on the best validation accuracy obtained. We report the accuracy obtained on the final test sets, as average accuracy among the different folds.

Results. Table 3.4.4 shows the results obtained on the IMDB Face dataset. We performed two main experiments: gender and age prediction. Besides the perfomance evaluation on the test set, when training on EB1 we also tested the model's performance on EB2, and viceversa. This allows us to better evaluate the bias features' influence on the model prediction. We notice how the baseline model is heavily biased towards age when predicting gender, and towards gender when predicting age. This can be observed on the performance achieved on the EB2 and EB1 sets, both for gender and age prediction. When employing our regularization

Target	Method	Trained on EB1		Trained on EB2	
		EB2	Test	EB1	Test
Gender	Vanilla	59.86	84.42	57.84	69.75
	BlindEye (Alvi et al., 2018)	63.74	85.56	57.33	69.90
	LfF (Kim et al., 2019)	**68.00**	<u>86.66</u>	<u>64.18</u>	<u>74.50</u>
	EnD	<u>65.49</u>±0.81	**87.15**±0.31	**69.40**±2.01	**78.19**±1.18
Age	Vanilla	54.30	77.17	48.91	61.97
	BlindEye (Alvi et al., 2018)	<u>66.80</u>	75.13	<u>64.16</u>	62.40
	LfF (Kim et al., 2019)	65.27	<u>77.43</u>	62.18	<u>63.04</u>
	EnD	**76.04**±0.25	**80.15**±0.96	**74.25**±2.26	**78.80**±1.48

Table 3.4.4: Performance on IMDB Face. When gender is learned, age is the bias, and when age is learned the gender is the bias. Reference results from Kim et al. (2019). The best results are highlighted in bold, the second best results are underlined.

term, we observe an increase across all of the obtained results: in particular, when training on EB2 for age prediction, we notice an increase from 48.91% to 74.25% on the EB1 set. We also report reference results of other debiasing algorithms, specifically BlindEye (Alvi et al., 2018) and the adversarial approach proposed by Kim et al. (2019). In general, EnD obtains the best results among all the other debiasing algorithms we compared to.

3.5 Conclusions and Limitations

In this Chapter, we aimed to discourage the selection of biased features in deep models trained on biased datasets. We proposed the EnD regularization, whose task is to both disentangle representations of bias-aligned samples and to entangle representations of positive bias-conflicting ones. Differently from other debiasing techniques, we do not introduce any additional parameters to be learned and we do not modify the input data: the model is naturally driven into choosing unbiased deep features, without introducing additional priors to the data. Our experiments show the effectiveness of EnD when compared to other state-of-the-art techniques, excelling in the cases of heavily biased data.

The results shown so far by EnD seem promising and have represented state-of-the-art performance for some time. However, subsequent works (Hong and Yang, 2021; Lee et al., 2021; Zhao et al., 2021) achieved better results, and highlighted some of the limitations of EnD:

- Ideally, on datasets such as Biased-MNIST, it should be possible to achieve higher test accuracy (e.g. in the upper range of 90%), as the task is quite easy once the bias is removed;
- Being a supervised debiasing technique, it requires complete annotation of the bias labels, which sometimes is not trivial to achieve;
- A major disadvantage of EnD is the hyperparameters tuning, and the requirement of an unbiased validation set. Fulfilling this requirement is not always

possible, for example with benchmarks such as Corrupted-CIFAR10 (Hendrycks and Dietterich, 2019), bFFHQ (Lee et al., 2021) and ImageNet-A (Hendrycks et al., 2021). The absence of such tuning may lead to suboptimal results.

Chapter 4

Unbiased Representation Learning with FairKL

In this Chapter, we present a unified framework to analyze and compare existing formulations of contrastive losses[1] such as the InfoNCE loss (Chen et al., 2020; Oord et al., 2019), the InfoL1O loss (Poole et al., 2019) and the SupCon loss (Khosla et al., 2020). Furthermore, we also propose a new supervised contrastive loss that can be seen as the simplest extension of the InfoNCE loss (Chen et al., 2020; Oord et al., 2019) to a supervised setting with multiple positives.

Using the proposed metric learning approach, we can reformulate each loss as a set of contrastive, and surprisingly sometimes even non-contrastive, conditions. We show that the widely used SupCon loss is not a "straightforward" extension of the InfoNCE loss since it actually contains a set of "latent" non-contrastive constraints. Our analysis results in an in-depth understanding of the different loss functions, fully explaining their behavior from a metric point of view. Furthermore, by leveraging the proposed metric learning approach, we explore the issue of biased learning. We outline the limitations of the studied contrastive loss functions when dealing with biased data, even if the loss on the training set is apparently minimized. By analyzing such cases, we provide a more formal characterization of bias. This eventually allows us to derive a new set of regularization constraints for debiasing that is general and can be added to any contrastive or non-contrastive loss. Our contributions are summarized below:

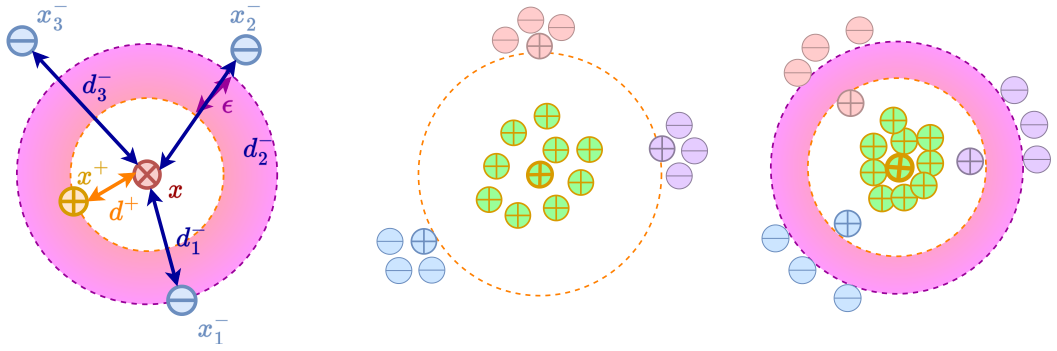
1. We introduce a simple but powerful theoretical framework for supervised representation learning, from which we derive different contrastive loss functions. We show how existing contrastive losses can be expressed within our framework, providing a uniform understanding of the different formulations. We derive a generalized form of the SupCon loss (ϵ-SupCon), propose a novel loss ϵ-*SupInfoNCE*, and demonstrate empirically its effectiveness;

2. We provide a more formal definition of bias, thanks to the proposed metric learning approach, which is based on the distances among representations. This allows us to derive a new set of effective debiasing regularization constraints, which we call *FairKL*. We also analyze, theoretically and empiri-

[1] We refer to any contrastive loss and not necessarily to losses based on pairs of samples as in (Sohn, 2016).

cally, the debiasing power of the different contrastive losses, comparing ϵ-SupInfoNCE and SupCon.

4.1 A metric framework for contrastive learning



(a) (b) (c)

Figure 4.1.1: With ϵ-SupInfoNCE (a) we aim at increasing the minimal margin ϵ, between the distance d^+ of a positive sample x^+ (+ symbol inside) from an anchor x and the distance d^- of the closest negative sample x^- (− symbol inside). By increasing the margin, we can achieve a better separation between positive and negative samples. We show two different scenarios without margin (b) and with margin (c). Filling colors of datapoints represent different biases. We observe that, without imposing a margin, biased clusters might appear containing both positive and negative samples (b). This issue can be mitigated by increasing the ϵ margin (c).

Let $x \in \mathcal{X}$ be an original sample (i.e., anchor), x_i^+ a similar (positive) sample, x_j^- a dissimilar (negative) sample and P and N the number of positive and negative samples respectively. Contrastive learning methods look for a parametric mapping function $f : \mathcal{X} \to \mathbb{S}^{d-1}$ that maps "semantically" similar samples close together in the representation space (a (d-1)-sphere) and dissimilar samples far away from each other. Once pre-trained, f is fixed and its representation is evaluated on a downstream task, such as classification, through linear evaluation on a test set. In general, positive samples x_i^+ can be defined in different ways depending on the problem: using transformations of x (unsupervised setting), samples belonging to the same class as x (supervised) or with similar image attributes of x (weakly-supervised). The definition of negative samples x_j^- varies accordingly. Here, we focus on the supervised case, thus samples belonging to the same/different class, but the proposed framework could be easily applied to the other cases. We define $s(f(a), f(b))$ as a similarity measure (e.g., cosine similarity) between the representation of two samples a and b. Please note that since $||f(a)||_2 = ||f(b)||_2 = 1$, using a cosine similarity is equivalent to using a L2-distance ($d(f(a), f(b)) = ||f(a) - f(b)||_2^2$).

Similarly to Chopra et al. (2005); Hadsell et al. (2006); Schroff et al. (2015); Sohn (2016); Wang et al. (2014, 2019c); Weinberger et al. (2006); Yu and Tao (2019), we propose to use a metric learning approach which allows us to better formalize recent contrastive losses, such as InfoNCE (Chen et al., 2020; Oord et al., 2019),

InfoL1O (Poole et al., 2019) and SupCon (Khosla et al., 2020), and derive new losses that better approximate the mutual information and can take into account data biases.

Using an ϵ-margin metric learning point of view, probably the simplest contrastive learning formulation is looking for a mapping function f such that the following ϵ-condition is always satisfied:

$$\underbrace{d(f(x), f(x^+))}_{d^+} - \underbrace{d(f(x), f(x_j^-))}_{d_j^-} < -\epsilon \iff \underbrace{s(f(x), f(x_j^-))}_{s_j^-} - \underbrace{s(f(x), f(x^+)}_{s^+} \leq -\epsilon \quad \forall j \tag{4.1.1}$$

where $\epsilon \geq 0$ is a margin between positive and negative samples and we consider, for now, a single positive sample.

4.1.1 Derivation of InfoNCE

The constraint of Eq. 4.1.1 can be transformed in an optimization problem using, as it is common in contrastive learning, the max operator and its smooth approximation *LogSumExp* (full derivation in the Appendix A.1.1):

$$\begin{gathered} s_j^- - s^+ \leq -\epsilon \quad \forall j \\ \arg\min_f \max(-\epsilon, \{s_j^- - s^+\}_{j=1,\dots,N}) \approx \arg\min_f \underbrace{-\log\left(\frac{\exp(s^+)}{\exp(s^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)}_{\epsilon - InfoNCE} \end{gathered} \tag{4.1.2}$$

Here, we can notice that when $\epsilon = 0$, we retrieve the InfoNCE loss, also known as N-Pair loss (Sohn, 2016), whereas when $\epsilon \to \infty$ we obtain the InfoL1O loss. It has been shown in Poole et al. (2019) that these two losses are lower and upper bound of the Mutual Information $I(X^+, X)$ respectively:

$$\mathop{\mathbb{E}}_{\substack{(x,x^+)\sim p(x,x^+) \\ x_j^-\sim p(x^-)}} \left[\underbrace{\log \frac{\exp s^+}{\exp s^+ + \sum_j \exp s_j^-}}_{InfoNCE}\right] \leq I(X^+, X) \leq \mathop{\mathbb{E}}_{\substack{(x,x^+)\sim p(x,x^+) \\ x_j^-\sim p(x^-)}} \left[\underbrace{\log \frac{\exp s^+}{\sum_j \exp s_j^-}}_{InfoL1O}\right] \tag{4.1.3}$$

where $p(x, x^+)$ is the joint (positive) distribution and $p(x^-)$ is the marginal (negative) distribution. By using a value of $\epsilon \in [0, \infty)$, one might find a tighter approximation of $I(X^+, X)$ since the exponential function at the denominator $\exp(-\epsilon)$ monotonically decreases as ϵ increases.

4.1.2 Proposed supervised loss (ϵ-SupInfoNCE)

The inclusion of multiple positive samples (s_i^+) can lead to different formulations. Some of them can be found in the Appendix A.1.2. Here, considering a supervised setting, we propose to use the following one, that we call ϵ-SupInfoNCE:

$$s_j^- - s_i^+ \leq -\epsilon \quad \forall i, j$$

$$\sum_i \max(-\epsilon, \{s_j^- - s_i^+\}_{j=1,\dots,N}) \approx \underbrace{-\sum_i \log\left(\frac{\exp(s_i^+)}{\exp(s_i^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)}_{\epsilon - SupInfoNCE} \tag{4.1.4}$$

Please note that this loss could also be used in other settings, like in an unsupervised one, where positive samples could be defined as transformations of the anchor. Furthermore, even here, the ϵ value can be adjusted in the loss function, in order to increase the ϵ-margin. This time, contrarily to what happens with Eq. 4.1.2 and InfoNCE, if we consider $\epsilon = 0$, we do not obtain the SupCon loss.

4.1.3 Derivation of ϵ-SupCon (generalized SupCon)

It's interesting to notice that Eq. 4.1.4 is similar to $\mathcal{L}_{out}^{sup}$, which is one of the two SupCon losses proposed in Khosla et al. (2020), but they differ for a sum over the positive samples at the denominator. The $\mathcal{L}_{out}^{sup}$ loss, presented as the "most straightforward way to generalize" the InfoNCE loss, actually contains another non-contrastive constraint on the positive samples: $s_t^+ - s_i^+ \leq 0 \quad \forall i, t$. Fulfilling this condition alone would force all positive samples to collapse to a single point in the representation space. However, it does not take into account negative samples. That is why we define it as a non-contrastive condition. Considering both contrastive and non-contrastive conditions, we obtain:

$$s_j^- - s_i^+ \leq -\epsilon \quad \forall i, j \quad \text{and} \quad s_t^+ - s_i^+ \leq 0 \quad \forall i, t \neq i$$

$$\frac{1}{P}\sum_i \max(0, \{s_j^- - s_i^+ + \epsilon\}_j, \{s_t^+ - s_i^+\}_{t \neq i}) \approx \epsilon \underbrace{-\frac{1}{P}\sum_i \log\left(\frac{\exp(s_i^+)}{\sum_t \exp(s_t^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)}_{\epsilon - SupCon} \tag{4.1.5}$$

when $\epsilon = 0$ we retrieve exactly $\mathcal{L}_{out}^{sup}$. The second loss proposed in Khosla et al. (2020), called $\mathcal{L}_{in}^{sup}$, minimizes a different contrastive problem, which is a less strict condition and probably explains the fact that this loss did not work well in practice (Khosla et al., 2020):

$$\max(s_j^-) < \max(s_i^+) \approx \log(\sum_j \exp(s_j^-)) - \log(\sum_i \exp(s_i^+)) < 0 \tag{4.1.6}$$

$$\arg\min_f \max(0, \max(s_j^-) - \max(s_i^+)) \approx \underbrace{-\log\left(\sum_i \frac{\exp(s_i^+)}{\sum_t \exp(s_t^+) + \sum_j \exp(s_j^-)}\right)}_{\mathcal{L}_{in}^{sup}} \tag{4.1.7}$$

It's easy to see that, differently from Eq. 4.1.4 and $\mathcal{L}_{out}^{sup}$, this condition is fulfilled when just *one* positive sample is more similar to the anchor than all negative samples. Similarly, another contrastive condition that should be avoided is $\sum_j s(f(x), f(x_j^-)) - \sum_i s(f(x), f(x_i^+)) < -\epsilon$ since one would need only *one* (or few) negative samples far away from the anchor in the representation space (i.e., orthogonal) to fulfill the condition.

4.2 Failure case of InfoNCE: the issue of biases

Satisfying the ϵ-condition (4.1.1) can generally guarantee good downstream performance, however, it does not take into account the presence of biases (e.g. selection biases). A model could therefore take its decision based on certain visual features, i.e. the bias, that are correlated with the target downstream task but don't actually characterize it. This means that the same bias features would probably have a worse performance if transferred to a different dataset (e.g. different acquisition settings or image quality). Specifically, in contrastive learning, this can lead to settings where we are still able to minimize any InfoNCE-based loss (e.g. SupCon or ϵ-SupInfoNCE), but with degraded classification performance (Figure 4.1.1b). To tackle this issue, in this work, we propose the FairKL regularization technique, a set of debiasing constraints that prevent the use of the bias features within the proposed metric learning approach. In order to give a more in-depth explanation of the ϵ-InfoNCE failure case, we employ the notion of *bias-aligned* and *bias-conflicting* samples as in Nam et al. (2020). In our context, a bias-aligned sample shares the same bias attribute of the anchor, while a bias-conflicting sample does not. In this work, we assume that the bias attributes are either known *a priori* or that they can be estimated using a bias-capturing model, such as in Hong and Yang (2021).

4.2.1 Characterization of bias

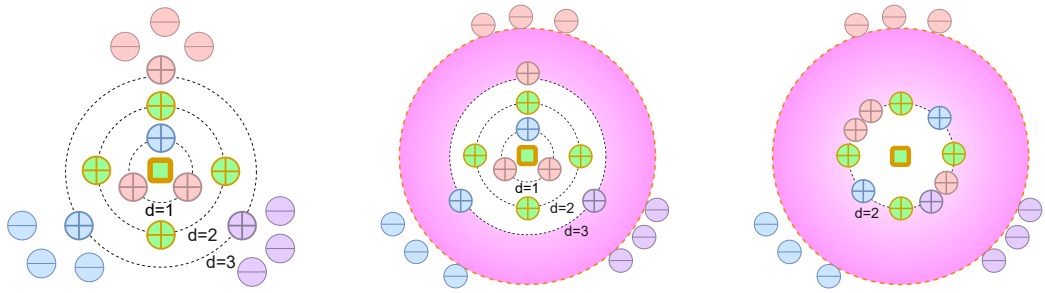
We denote bias-aligned samples with $x^{\cdot,b}$ and bias-conflicting samples with $x^{\cdot,b'}$. Given an anchor x, if the bias is "strong" and easy-to-learn, a *positive bias-aligned* sample $x^{+,b}$ will probably be closer to the anchor x in the representation space than a *positive bias-conflicting* sample (of course, the same reasoning can be applied for the negative samples). This is why even in the case in which the ϵ-condition is satisfied and the ϵ-SupInfoNCE is minimized, we could still be able to distinguish between bias-aligned and bias-conflicting samples. Hence, we say that there is a bias if we can identify an ordering on the learned representations, such as:

$$\underbrace{d(f(x), f(x_i^{+,b}))}_{d_i^{+,b}} < \underbrace{d(f(x), f(x_k^{+,b'})}_{d_k^{+,b'}} \leq \underbrace{d(f(x), f(x_t^{-,b}))}_{d_t^{-,b}} - \epsilon < \underbrace{d(f(x), f(x_j^{-,b'}))}_{d_j^{-,b'}} - \epsilon \quad \forall i, k, t, j \tag{4.2.1}$$

This represents the worst-case scenario, where the ordering is total (i.e., $\forall i, k, t, j$). Of course, there can also be cases in which the bias is not as strong, and the ordering may be partial.

4.2.2 FairKL regularization for debiasing

Ideally, we would enforce the conditions $d_k^{+,b'} - d_i^{+,b} = 0 \quad \forall i, k$ and $d_t^{-,b'} - d_j^{-,b} = 0 \quad \forall t, j$, meaning that every positive (resp. negative) bias-conflicting sample should have the same distance from the anchor as any other positive (resp. negative) bias-aligned sample. However, in practice, this condition is very strict, as it would enforce uniform distance among all positive (resp. negative) samples. A more relaxed



(a) (b) (c)

Figure 4.2.1: When considering only Eq. 4.2.2 (average of distances or similarities)), we may obtain a sub-optimal configuration such as (a), where we can still (partially) order the distances of positive samples from the anchor based on the bias features. We can see that the conditions in Eq. 4.2.2 are fulfilled, namely the average of the distances of bias-aligned and bias-conflicting samples from the anchor are the same ($\mu_{+,b} = \mu_{+,b'} = 2$). This is only partially mitigated when using a margin $\epsilon > 0$ (b). However, the standard deviations of the distances of bias-aligned and bias-conflicting samples in (a) and (b) are different ($\sigma_{+,b} = 0$, while $\sigma_{+,b'} = 1$). This can be computed using the distances d reported in the figure. If we also consider the conditions on the standard deviations of the distances, as proposed in FairKL (Eq. 4.2.3), the ordering is removed and thus also the effect of the bias (c). In (c), we show the case in which both mean and standard deviation of the distributions match (in a simplified case with σ=0). A simulated example is shown in Figure 4.2.2.

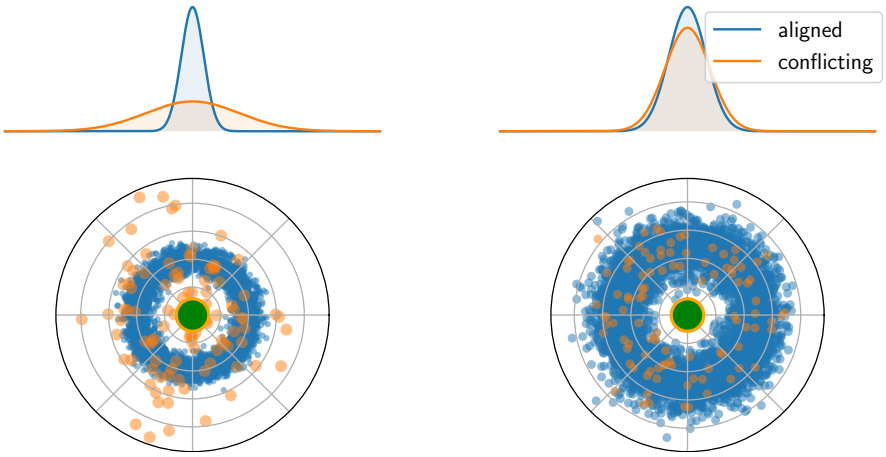


Figure 4.2.2: Toy example with simulated data to better explain the suboptimal solution of Figure 4.2.1. We make the hypothesis that the distributions of the distances do follow a Gaussian distribution. In blue and in orange are shown the bias-aligned and the bias-conflicting samples respectively. The green sample represents the anchor. On the left, data points are sampled from two normal distributions with the same mean but *different* std. We can see that the two distributions do not match. This shows that, even if the first order constraints of Eq. 4.2.2 are fulfilled, there might still be an effect of the bias. On the contrary, on the right, the two distributions have almost the same statistics (both average and std) and the KL divergence is almost 0. In that case, the bias effect is basically removed.

condition would instead force the distributions of distances, $\{d_k^{\cdot,b'}\}$ and $\{d_i^{\cdot,b}\}$, to be similar. Here, we propose two new debiasing constraints for both positive and negative samples using either the first moment (mean) of the distributions or the first two moments (mean and variance). Using only the average of the distributions, we obtain:

$$\frac{1}{P_a}\sum_i d_i^{+,b} - \frac{1}{P_c}\sum_k d_k^{+,b'} = 0 \iff \frac{1}{P_c}\sum_k s_k^{+,b'} - \frac{1}{P_a}\sum_i s_i^{+,b} = 0 \tag{4.2.2}$$

where P_a and P_c are the numbers of positive bias-aligned and bias-conflicting samples, respectively[2]. Doing so, on average, the bias-aligned and bias-conflicting samples would have the same distance (or similarity) to the anchor. However, even if this constraint is fulfilled there might still be an effect of the bias features on the ordering of the positive samples, due to difference in the second moments of the distributions, as illustrated visually in Figure 4.2.1. From the figure, we can also see how increasing the *epsilon* margin can help in mitigate this issue but does not solve it completely. In order to avoid this sub-optimal case, we extend the constraint of Eq. 4.2.2 to also include the second moments of the distance/similarity distributions.

Denoting the first moments with $\mu_{+,b} = \frac{1}{P_a}\sum_i d_i^{+,b}$, $\mu_{+,b'} = \frac{1}{P_c}\sum_k d_k^{+,b'}$, and the second moments of the distance distributions with $\sigma^2_{+,b} = \frac{1}{P_a}\sum_i (d_i^{+,b} - \mu_{+,b})^2$, $\sigma^2_{+,b'} = \frac{1}{P_c}\sum_k (d_k^{+,b'} - \mu_{+,b-})^2$, and making the hypothesis that the distance distributions follow a normal distribution, we propose a new regularization term $\mathcal{R}^{FairKL}$ which employs the Kullback–Leibler divergence:

$$\mathcal{R}^{FairKL} = D_{KL}(B_{+,b}||B_{+,b'}) = \frac{1}{2}\left(\frac{\sigma^2_{+,b} + (\mu_{+,b} - \mu_{+,b'})^2}{\sigma^2_{+,b'}} - \log\frac{\sigma^2_{+,b}}{\sigma^2_{+,b'}} - 1\right) \tag{4.2.3}$$

where $B_{+,b} \sim \mathcal{N}(\mu_{+,b}, \sigma^2_{+,b})$ and $B_{+,b'} \sim \mathcal{N}(\mu_{+,b'}, \sigma^2_{+,b'})$ are the positive bias-aligned and positive bias-conflicting distance distributions respectively. In practice, one could also use another distribution such as the log-normal, the Jeffreys divergence $(D_{KL}(B_{+,b}||B_{+,b'}) + D_{KL}(B_{+,b'}||B_{+,b}))$, or a simplified version, such as the difference of the two statistics (e.g., $(\mu_{+,b} - \mu_{+,b'})^2 + (\sigma_{+,b} - \sigma_{+,b'})^2$).

The proposed debiasing constraints can be easily added to any contrastive (or non-contrastive) as a regularization term $\mathcal{R}^{FairKL}$. In this work, the final loss function that we propose to minimize is the combination of ϵ-SupInfoNCE and FairKL:

$$\mathcal{L} = \underbrace{-\alpha\sum_i \log\left(\frac{\exp(s_i^+)}{\exp(s_i^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)}_{\epsilon-SupInfoNCE} + \lambda\mathcal{R}^{FairKL} \tag{4.2.4}$$

where α and λ are positive two hyperparameters weighting the contribution of each term.

[2]The same reasoning can be applied to negative samples (omitted for brevity.)

Analysis of other losses and debiasing methods

Leveraging the metric framework we described in the previous sections, we are able to study more in-depth some related methods and provide an interpretable explanation of their behavior.

SupCon It is interesting to notice that the non-contrastive conditions in Eq. 4.1.5: $s_t^+ - s_i^+ \leq 0 \quad \forall i, t \neq i$ are actually all fulfilled only when $s_i^+ = s_t^+ \quad \forall i, t \neq i$. This means that one tries to align all positive samples, regardless of their bias b, to a single point in the representation space. In other terms, at the optimal solution, one would also fulfill the following conditions:

$$s_i^{+,b} = s_t^{+,b} \ , \ s_i^{+,b'} = s_t^{+,b'} \ , \ s_i^{+,b} = s_t^{+,b'} \ , \ s_i^{+,b'} = s_t^{+,b} \quad \forall i, t \neq i \tag{4.2.5}$$

Realistically, this could lead to suboptimal solutions: we argue that the optimization process would mainly focus on the easier task, namely aligning bias-aligned samples, and neglecting the bias-conflicting ones. In highly biased settings, this could lead to worse performance than ϵ-SupInfoNCE. More empirical results supporting this hypothesis are presented in Appendix C.2.

EnD The constraint in Eq. 4.2.2 is very similar to the EnD method that we presented in Section 3.3. In fact, EnD lacks the additional constraint on the standard deviation of the distances, which is given by Eq. 4.2.3. We can show analytically that the EnD regularization term can be, under certain conditions, equivalent to Eq. 4.2.2. Using the notation introduced in this chapter, we can rewrite EnD (Eq. 3.3.5) as:

$$\mathcal{R}^{end} = \alpha \frac{1}{P_a + N_a} \sum_{a \in B(i)} |s_a^{\cdot,b}| - \beta \frac{1}{P_c} \sum_k s_k^{+,b'} \tag{4.2.6}$$

where N_a is the number of negative bias-aligned samples. Assuming, for simplicity, $\alpha = \beta = 1$, we can split the EnD orthogonal term $\mathcal{R}^{\perp}$ into a positive ($\mathcal{R}^{\perp}_{pos}$) and a negative ($\mathcal{R}^{\perp}_{neg}$) term:

$$\mathcal{R}^{end} = \underbrace{\frac{1}{P_a + N_a} \sum_i |s_i^{+,b}|}_{\mathcal{R}^{\perp}_{pos}} + \underbrace{\frac{1}{P_a + N_a} \sum_n |s_n^{-,b}|}_{\mathcal{R}^{\perp}_{neg}} - \underbrace{\frac{1}{P_c} \sum_k s_k^{+,b'}}_{\mathcal{R}^{\parallel}} \quad . \tag{4.2.7}$$

In order to reach the equivalence between EnD and Eq. 4.2.2, we can make a few realistic assumptions:

- The term $\mathcal{R}^{\perp}_{neg}$ can be safely ignored, as long as the target loss function (e.g. CE or ϵ-SupInfoNCE) seeks to maximize the similarity between positive samples and minimize it for negative samples (this is, of course, trivial);
- We can assume a non-negative similarity for positive and bias-aligned samples, such that the absolute value can be dropped from $\mathcal{R}^{\perp}_{pos}$ (which is also reasonable if the previous assumption holds);
- In terms of minimization, $\min \frac{1}{P_a + N_a}(..) = \min \frac{1}{P_a}(..)$, given that $P_a, N_a > 0$.

Thus, we finally obtain:

$$\mathcal{R}^{end} = \frac{1}{P_a} \sum_i s_i^{+,b} - \frac{1}{P_c} \sum_k s_k^{+,b'} \tag{4.2.8}$$

which can be obtained by turning the condition of Eq. 4.2.2 into a minimization term $\mathcal{R}^{mean}$, using the method of Lagrange multipliers:

$$\mathcal{R}^{mean} = -\lambda \left(\frac{1}{P_c} \sum_k s_k^{+,b'} - \frac{1}{P_a} \sum_i s_i^{+,b} \right) \tag{4.2.9}$$

with $\lambda = 1$. Of course, in practice, some differences between the formulations remain as, for example, the terms are weighted differently.

BiasCon In Hong and Yang (2021), authors propose a BiasCon loss, which is similar to SupCon but only aligns positive bias-conflicting samples. It looks for an encoder f that fulfills:

$$s_j^- - s_i^{+,b'} \leq -\epsilon \quad \forall i, j \quad \text{and} \quad s_p^{+,b} - s_i^{+,b'} \leq 0 \quad \forall i, p \quad \text{and} \quad s_t^{+,b'} - s_i^{+,b'} \leq 0 \quad \forall i, t \neq i \tag{4.2.10}$$

The problem here is that we try to separate the negative samples from only the positive bias-conflicting samples, ignoring the positive bias-aligned samples. This is probably why the authors proposed to combine this loss with a standard Cross Entropy.

4.3 Experiments

In this section, we describe the experiments we perform to validate our proposed losses. We perform two sets of experiments. First, we benchmark our framework, presented in Section 4.1, on standard vision datasets such as: CIFAR-10 (Krizhevsky et al., a), CIFAR-100 (Krizhevsky et al., b) and ImageNet-100 (Deng et al., 2009). Then, we analyze biased settings with FairKL, employing BiasedMNIST (Bahng et al., 2020), Corrupted-CIFAR10 (Hendrycks and Dietterich, 2019) and bFFHQ (Lee et al., 2021).

Experiments on generic vision datasets

We conduct an empirical analysis of the ϵ-SupCon and ϵ-SupInfoNCE losses on standard vision datasets to evaluate the different formulations and to assess the impact of the ϵ parameter. We compare our results with baseline implementations including Cross Entropy (CE) and SupCon.

Experimental details We use the original setup from SupCon (Khosla et al., 2020), employing a ResNet-50, a large batch size (1024), a learning rate of 0.5, a temperature of 0.1, and multiview augmentation, for CIFAR-10 and CIFAR-100. Additional experimental details (including ImageNet-100) and the different hyper-parameters configurations are provided in Section B of the Appendix.

Loss	Acc@1
ϵ-SupInfoNCE	**83.3**$\pm$0.06
ϵ-SupCon	82.83$\pm$0.11

Table 4.3.1: Comparison of ϵ-SupInfoNCE and ϵ-SupCon on ImageNet-100.

Results First, we compare our proposed ϵ-SupInfoNCE loss with the ϵ-SupCon loss derived in Section 4.1. As reported in Table 4.3.1, ϵ-SupInfoNCE performs better than ϵ-SupCon: we conjecture that the lack of the non-contrastive term of Eq. 4.1.5 leads to increased robustness, as it will also be shown in Section 4.3. For this reason, we focus on ϵ-SupInfoNCE. Further comparison with different values of ϵ can be found in Section C.1, showing that $SupCon \leq \epsilon\text{-}SupCon \leq \epsilon\text{-}SupInfoNCE$ in terms of accuracy.
Results on general computer vision datasets are presented in Table 4.3.2, in terms of top-1 accuracy. We report the performance for the best value of ϵ; the complete results can be found in Section C.1. The results are averaged across 3 independent trials for every configuration, and we also report the standard deviation. We obtain significant improvement with respect to all baselines and, most importantly, SupCon, on all benchmarks: on CIFAR-10 (+0.5%), on CIFAR-100 (+0.63%), and on ImageNet-100 (+1.31%).

Dataset	Network	SimCLR	Max-Margin	SimCLR*	CE*	SupCon*	ϵ-SupInfoNCE*
CIFAR-10	ResNet-50	93.6	92.4	91.74$\pm$0.05	94.73$\pm$0.18	95.64$\pm$0.02	**96.14**$\pm$0.01
CIFAR-100	ResNet-50	70.7	70.5	68.94$\pm$0.12	73.43$\pm$0.08	75.41$\pm$0.19	**76.04**$\pm$0.01
ImageNet-100	ResNet-50	-	-	66.14$\pm$0.08	82.1$\pm$0.59	81.99$\pm$0.08	**83.3**$\pm$0.06

Table 4.3.2: Accuracy on standard vision datasets. SimCLR and Max-Margin results from Khosla et al. (2020). Results denoted with * were (re)implemented with mixed precision due to memory constraints. The best results are highlighted in bold, the second best results are underlined.

Experiments on biased datasets

Next, we move on to analyzing how our proposed loss performs on biased learning settings. We employ five datasets, ranging from synthetic data to real facial images: Biased-MNIST, Corrupted-CIFAR10, and bFFHQ. The detailed setup and experimental details are provided in the Appendix B.

Biased-MNIST

We compare with cross entropy baseline and with other debiasing techniques, namely EnD, LNL (Nam et al., 2020) and BiasCon (BC) and BiasBal (BB) (Hong and Yang, 2021).

Analysis of ϵ-SupInfoNCE and ϵ-SupCon First, we perform an evaluation of the ϵ-SupCon and ϵ-SupInfoNCE losses alone, without our debiasing regularization

term. Figure 4.3.1 shows the accuracy on the unbiased test set, with the different values of ρ. Baseline results of a cross-entropy model (CE) are reported in Table 4.3.3. Both losses result in higher accuracy compared to the cross entropy. The generally higher robustness of contrastive-based formulations is also confirmed by the related literature (Khosla et al., 2020). Interestingly, in the most biased setting ($\rho = 0.999$), we observe that ϵ-SupInfoNCE obtains higher accuracy than ϵ-SupCon. Our conjecture is that the non-contrastive term of SupCon in Eq. 4.1.5 ($s_t^+ - s_i^+ \leq 0 \quad \forall i, t$) can lead, in highly biased settings, to more biased representations as the bias-aligned samples will be especially predominant among the positives. For this reason, we focus on ϵ-SupInfoNCE in the remaining of this work.

Debiasing with FairKL Next, we apply our regularization technique FairKL jointly with ϵ-SupInfoNCE, and compare it with the other debiasing methods. The results are shown in Table 4.3.3. Our technique achieves the best results in all experiments, with high gaps in accuracy, especially in the most difficult settings (lower ρ). For completeness, we also evaluate the debiasing power of FairKL with different losses, i.e. CE and ϵ-SupCon. With FairKL we obtain better results than most of the other baselines with either CE, ϵ-SupCon or ϵ-SupInfoNCE; the latter achieves the best performance, confirming the results observed in Sec 4.3. For this reason, in the rest of the work, we focus on ϵ-SupInfoNCE.

Method	0.999	0.997	0.995	0.99
CE (Hong and Yang, 2021)	11.8±0.7	62.5±2.9	79.5±0.1	90.8±0.3
LNL (Kim et al., 2019)	18.2±1.2	57.2±2.2	72.5±0.9	86.0±0.2
ϵ-SupCon	24.36±3.23	74.35±0.09	84.13±1.31	91.12±0.35
ϵ-SupInfoNCE	33.16±3.57	73.86±0.81	83.65±0.36	91.18±0.49
EnD (Section 3.3)	59.5±2.3	82.70±0.3	94.0±0.6	94.8±0.3
BiasCon+BiasBal* (Hong and Yang, 2021)	30.26±11.08	82.83±4.17	88.20±2.27	95.04±0.86
BiasBal (Hong and Yang, 2021)	76.8±1.6	91.2±0.2	93.9±0.1	96.3±0.2
BiasCon+CE* (Hong and Yang, 2021)	15.06±2.22	90.48±5.26	95.95±0.11	<u>97.67</u>±0.09
CE + FairKL	79.9±4.29	93.86±1.13	94.85±0.55	95.92±0.17
ϵ-SupCon + FairKL	<u>89.45</u>±1.82	<u>95.75</u>±0.16	<u>96.31</u>±0.81	96.72±0.2
ϵ-SupInfoNCE + FairKL	**90.51**±1.55	**96.19**±0.23	**97.00**±0.06	**97.86**±0.02

Table 4.3.3: Top-1 accuracy (%) on Biased-MNIST. Reference results from Hong and Yang (2021). Results denoted with * are re-implemented without color-jittering and bias-conflicting oversampling, for fairness of comparison. The best results are highlighted in bold, the second best results are underlined.

Corrupted CIFAR-10

Corrupted CIFAR-10 is built from the CIFAR-10 dataset, by correlating each class with a certain texture (brightness, frost, etc.) following the protocol proposed in Hendrycks and Dietterich (2019). Similarly to Biased-MNIST, the dataset is provided with five different levels of ratio between bias-conflicting and bias-aligned samples, where lower values indicate more biased versions of the dataset. The results are shown in Table 4.3.4. Notably, we obtain the best results in the most difficult

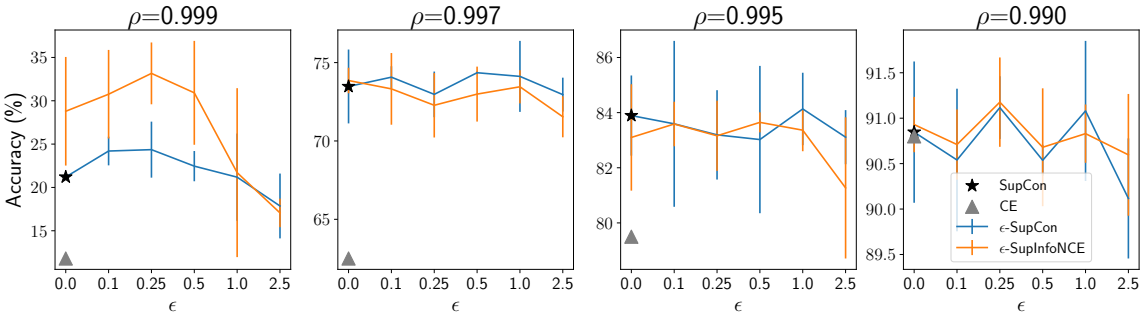


Figure 4.3.1: Comparison of ϵ-SupCon and ϵ-SupInfoNCE on Biased-MNIST. It is noticeable that for $\rho \leq 0.997$, ϵ-SupInfoNCE and ϵ-SupCon are comparable, while for $\rho = 0.999$ the gap is significantly larger: this could be due to the additional non-contrastive condition of SupCon.

scenario, when the amount of bias-conflicting samples is the lowest. Again, for the other settings, we obtain comparable results with the state of the art.

bFFHQ

bFFHQ is a dataset proposed by Lee et al. (2021), and contains facial images. They construct the dataset in such a way that most of the females are young (age range 10-29), while most of the males are older (age range 40-59). The ratio between bias-conflicting and bias-aligned provided for this dataset is 0.5. The results are shown in Table 4.3.4, where our technique outperforms all other methods.

	Corrupted CIFAR-10 Ratio				bFFHQ Ratio
Method	0.5	1.0	2.0	5.0	0.5
Vanilla (Lee et al., 2021)	23.08±1.25	25.82±0.33	30.06±0.71	39.42±0.64	56.87±2.69
EnD (Tartaglione et al., 2021)	19.38±1.36	23.12±1.07	34.07±4.81	36.57±3.98	56.87±1.42
HEX (Wang et al., 2019a)	13.87±0.06	14.81±0.42	15.20±0.54	16.04±0.63	52.83±0.90
ReBias (Bahng et al., 2020)	22.27±0.41	25.72±0.20	31.66±0.43	43.43±0.41	59.46±0.64
LfF (Nam et al., 2020)	28.57±1.30	33.07±0.77	39.91±0.30	50.27±1.56	62.2±1.0
DFA (Lee et al., 2021)	29.95±0.71	36.49±1.79	**41.78**±2.29	**51.13**±1.28	<u>63.87</u>±0.31
ϵ-SupInfoNCE + FairKL	**33.33**±0.38	**36.53**±0.38	<u>41.45</u>±0.42	<u>50.73</u>±0.90	**64.8**±0.43

Table 4.3.4: Top-1 accuracy (%) on Corrupted CIFAR-10 with different corruption ratio (%) and on bFFHQ. Reference results are taken from Lee et al. (2021). The best results are highlighted in bold, the second best results are underlined.

4.4 Conclusions

In this chapter, we introduced a novel contrastive loss ϵ-SupInfoNCE which is able to achieve state-of-the-art performance on standard vision datasets, compared to previously existing losses such as SupCon. Furthermore, the loss is formally derived thanks to the metric learning framework we described, which makes it very easy to formalize what the different loss formulations aim at optimizing, thanks to simple metric conditions. Notably, the representations learning by ϵ-SupInfoNCE seem to

be partially more robust to the collateral learning issue, for example on biased data, when compared with SupCon. This is probably due to the difference in the starting metric conditions of the two losses.

One limitation of ϵ-SupInfoNCE, with respect to SupCon, is that a new hyperparameter ϵ is introduced, which has to be manually chosen, slightly adding to the complexity of the training. Future works may focus on proposing an automatic way for optimizing ϵ during training. Also, in this work, we did not perform an analysis on ϵ-SupInfoNCE when used for self-supervised learning (i.e. defining positives based on data augmentation). We leave this as future work.

Focusing on the issue of biases, with our framework, we were able to analyze the failure case of contrastive learning losses when dealing with biased data (the reasoning can be applied to InfoNCE-based losses). This has prompted us to formulate the FairKL regularization term, which aims at avoiding the ordering of the representations based on bias. We have shown that, with FairKL, it is possible to successfully mitigate this issue, achieving the best results in the most biased settings and improving our previously proposed method EnD. However, our proposed method FairKL is still affected by some limitations:

- Like EnD, it still requires bias annotation in the data, preventing its usage in certain applications and datasets such as ImageNet-A (Hendrycks et al., 2021);
- On the harder dataset Corrupted CIFAR-10, the best results are achieved in the most biased settings (e.g. ratio of 0.5 and 1.0). While still a notable result, one may argue that in realistic scenarios biases might be more subtle. From the point of view of Collateral Learning, those cases are especially relevant for fighting hidden and potentially more harmful biases;
- Additionally, as FairKL is based on computing the distribution of different groups, the "goodness" of the statistics is heavily dependent on the sample size (i.e. mini-batch size). Although we did not perform such analysis in this work, it is possible that FairKL performance might degrade significantly with smaller batch sizes.

In the rest of this work, we will attempt to resolve some of the highlighted limitations of FairKL and EnD, for example by focusing on unsupervised debiasing techniques.

Chapter 5

Extending To The Unknowns

In this section, we present the work we are carrying on for the development of unsupervised debiasing methods. Based on the limitations of the previous methods, discussed in Section 3.5 and 4.4, we formulate some different approaches for adapting such techniques to the unsupervised case. The methods that we propose are based on the assumption that the strength of the bias is such that, if no precaution is taken, a vanilla model will be heavily affected. From this assumption, we show that it is possible to leverage a biased model to obtain either *pseudo*-labels or a bias score that can be employed in supervised methods such as EnD or FairKL to make up for the missing ground-truth annotations.

5.1 Unsupervised debiasing via subgroup discovery

In this section, we present our proposed unsupervised end-to-end debiasing approach, showing how an explicitly supervised technique such as EnD[1] can be extended to the unsupervised case, where the bias labels are unavailable. We do this by showing how the bias information can be partially, and sometimes fully, recovered in a completely unsupervised manner.

To achieve that, our proposed algorithm consists of three sequential steps, as illustrated in Figure 5.1.1. First, we train a bias-capturing classifier, employing standard optimization techniques (e.g. SGD or Adam); then, we recover bias-related information from the latent space of the biased classifier via clustering, in order to obtain a bias predictor, which we employ to categorize all of the training samples into different bias classes. Lastly, we apply the EnD debiasing technique using the predicted bias labels, in order to obtain a debiased classifier. A general scheme of the entire pipeline can be found in Algorithm 1. Throughout this section, we make the assumption that an *unbiased* validation set is available: this is needed for searching the optimal EnD hyper-parameters.

5.1.1 Training a bias-capturing model

The first step of our proposed algorithm is to train a bias-capturing model, which in our case is represented by a *biased* encoder. To achieve this, we perform a vanilla

[1] In this section, we mainly focus on EnD. Extending the proposed method also to FairKL, and also to existing supervised techniques, is the subject of ongoing and future research.

Algorithm 1: General scheme of U-EnD

Input: Training and validation data $X^t = \{(x_i, y_i)\}$, $X^v = \{(\hat{x}_i, \hat{y}_i)\}$; Randomly initialized parameters $\theta_B = \{\theta_f, \theta_g\}$ and $\theta_D = \{\theta_f^D, \theta_g^D\}$ of the biased and unbiased classifiers.

Output: Trained parameters θ_D of the unbiased classifier.

Train bias-capturing model

Train the biased classifier using vanilla SGD: $\theta_B \leftarrow \text{SGD}(\theta_B, X^t)$

Compute the biased representations: $Z^t = \{f(x; \theta_f)\} \quad \forall x \in X^t$ and $Z^v = \{f(x; \theta_f)\} \quad \forall x \in X^v$

end

Train bias predictor

Compute the PCA projections P^t, P^v of Z^t, Z^v

Fit k clusters on P^v choosing the optimal k based on silhoutte and compute the cluster centroids: $\{\mu_1, ..., \mu_k\} \leftarrow \text{KMeans}(P^v, k)$

Assign the pseudo-labels $\hat{b}_i \leftarrow \underset{b \leq k}{\text{argmin}} \, (P_i^t, \, \mu_b)$

Update the training set $X^t \leftarrow \{(x_i, y_i, \hat{b}_i)\}$

end

Train unbiased classifier

Learn the parameters θ_D on X^t searching the optimal α and β on X^v: $\theta_D \leftarrow \text{SGD}(\theta_D, X^t) + R(\theta_f^D, X^t, \alpha, \beta)$

end

training of a CNN classifier on the available training data. Here, we do not employ any technique aimed at dealing with the presence of biases in the data. The intuition of this approach is that if bias features are easier to learn than the desired target attributed, then the resulting model will also be biased, as shown in the beginning of this section.

Figure 5.1.1 shows a visualization of the embeddings obtained with a biased encoder on the Biased-MNIST dataset, where the background color correlates very well with the target digit class, as shown in Figure 3.1.1. It is clear how the different clusters emerging in the latent space correspond to the different background color, rather than to the actual digit. This first step is summarized in Algorithm 1, and we now provide a more formal description. Let $\theta_B = \{\theta_f, \theta_g\}$ be the set of parameters of the bias-capturing model $p(x; \theta) = g(f(x; \theta_f); \theta_g)$ where f and g are the encoder and the classifier, respectively. The objective function we aim to minimize is the cross-entropy loss (CE):

$$\mathcal{L}_{\text{CE}}(p(x; \theta_B), q(x)) = -\sum_{t \in T} q(t|x) \log p(t|x; \theta_B) \tag{5.1.1}$$

where $q(x)$ represents the ground truth class distribution. We say that there is a benign bias in the dataset, if we can identify some distribution $r(x)$, related to some other confounding factor in the data, such that there exists a set of parameters θ' which is a local minimizer of equation 5.1.1 and $\theta' = \text{argmin}_{\theta_B} \mathcal{L}_{\text{CE}}(p(x; \theta_B), r(x))$. If, additionally, $r(x)$ is also easier to approximate than $q(x)$, then the bias is malig-

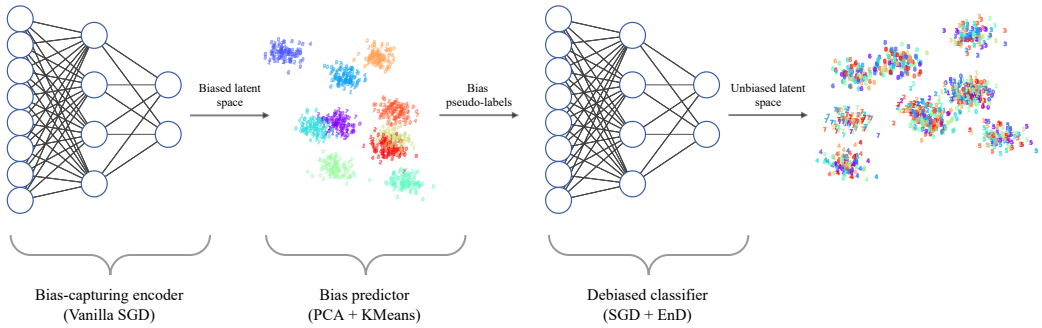


Figure 5.1.1: Overview of our unsupervised debiasing approach: first we train a bias-capturing encoder, then we determine bias pseudo-labels with a bias predictor. Finally, we employ the predicted labels for training a final debiased classifier. In this figure we use Biased-MNIST (Bahng et al., 2020) as example.

nant and by applying the optimization process we obtain a bias-capturing model. Once the biased model is trained, we only consider the encoder $f(x;\theta_f)$, as we are interested in analyzing its latent space in order to retrieve bias-related information.

5.1.2 Fitting a bias predictor

The second step consists in obtaining a predictor which can identify the bias in the data. Based on the observations made in Section 5.1.1, we employ a clustering algorithm to categorize the extracted representations into different classes. As shown in Figure 5.1.1, the identified clusters correspond to the biases in the dataset. In this work, we choose KMeans (Lloyd, 1982) as it is one of the most well-known clustering algorithms. Given a set of representation $z = \{z_1, z_2, \ldots, z_n\}$ extracted by $f(x;\theta_B)$ we aim to partition z into k sets $C = \{C_1, C_2, \ldots, C_k\}$ in order to minimize the within-clusters sum of squares (WCSS), which can be interpreted as the distance of each sample from its corresponding cluster centroid, by finding:

$$\underset{C}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{z \in C_i} ||z - \mu_i||^2 \tag{5.1.2}$$

where μ_i is the centroid (average) of C_i. Furthermore, once the clusters have been determined, it is very easy to use the determined centroids for classifying a new sample $\hat{z}$ based on its distance, simply by finding

$$b_i = \underset{j \leq k}{\operatorname{argmin}} ||z_i - \mu_j||^2 \tag{5.1.3}$$

where $\hat{b}$ denotes the resulting pseudo-label. The KMeans algorithm requires a pre-specified number of clusters k: in this work, we automatically tune this parameter based on the best silhouette score (Rousseeuw, 1987), obtained by performing a grid search in the range $[2, 15]$. Considering that the representations obtained on the training set might be over-fitted, we choose to minimize Eq. 5.1.3 on the validation set. Then, once the centroids of the clusters have been found, we use them for pseudo-labelling the training set. Additionally, as KMeans is based on euclidean distance, which can yield poor results in highly dimensional spaces, we perform a

PCA projection of the latent space before solving Eq. 5.1.2 and Eq. 5.1.3. For the same reasons as above, the PCA transformation matrix is also computed on the validation set. We refer to the ensemble of the PCA+KMeans as *bias predictor* model. The cluster information is then used as a bias pseudo-label, as explained in Section 5.1.3.

5.1.3 Training an unbiased classifier

The third and final step of our proposed framework consists in training an unbiased classifier. For this purpose, we use the clusters discovered in the previous phase as pseudo-labels for the bias classes, as shown in Figure 5.1.1. This allows us to employ the fully supervised EnD regularization term for debiasing. Here we follow the approach described in Section 3.3. Denoting with $\theta_D = \{\theta_f^D, \theta_g^D\}$ the parameters of the encoder and the classifier of the debiased model $p'(x;\theta_D) = g(\gamma(f(x;\theta_f^D));\theta_g^D)$. The objective function that we aim to minimize in this phase is:

$$\mathcal{L}_{\text{CE}}(p'(x;\theta_D), q(x)) + R(\gamma(f(x;\theta_f^D)), q(x), b(x)) \tag{5.1.4}$$

where $b(x)$ is the distribution corresponding to the pseudo-labels computed in the clustering step of Section 5.1.2. The closer $b(x)$ is to the real distribution $r(x)$, the more minimizing equation 5.1.4 will lead to minimizing R with respect to the unknown ground-truth bias labels.

5.1.4 Experiments

For testing the unsupervised extension U-EnD, we perform the same experiments described in Section 3.4.1 and 3.4.2. We also run some preliminary tests on FairKL, which we indicate with U-FairKL; this is the subject of ongoing and future work.

Biased-MNIST

The results for Biased-MNIST are presented in Table 5.1.1. We report the accuracy on the unbiased test set, obtained with the supervised EnD technique and the unsupervised extension. We also report reference results (Bahng et al., 2020) of other debiasing algorithms, both supervised and unsupervised. For U-EnD, we evaluate the results employing pseudo-labels computed at different training iterations (T) of the biased encoder: at an early stage after 10 epochs, and at a late stage at the end of training (80 epochs). In this section, when possible, we perform the experiments with different values of T. Using the unsupervised method we are able to match the original performance of EnD and FairKL with the ground-truth bias labels in most settings: this is true especially when the bias is stronger (higher ρ values). This is because in these cases, the bias-capturing models will produce representations strongly biased towards the color, and the pseudo-labels obtained with the bias predictor model will be accurate. On the other hand, a slightly larger gap is observed when there is less correlation between target and bias features. This is the most difficult setting for the unsupervised clustering of the bias features: however, a significant improvement with respect to the baseline is always achieved.

Method	ρ values			
	0.999	0.997	0.995	0.990
Vanilla (Bahng et al., 2020)	10.40±0.50	33.40±12.21	72.10±1.90	89.10±0.10
LearnedMixin (Clark et al., 2019)	12.10±0.80	50.20±4.50	78.20±0.70	88.30±0.70
BiasCon+BiasBal* (Hong and Yang, 2021)	30.26±11.08	82.83±4.17	88.20±2.27	95.04±0.86
BiasCon+CE* (Hong and Yang, 2021)	15.06±2.22	<u>90.48</u>±5.26	**95.95**±0.11	**97.67**±0.09
EnD (Chapter 3)	<u>52.30</u>±2.39	83.70±1.03	93.92±0.35	<u>96.02</u>±0.08
FairKL+CE (Section 4.3)	**79.9**±4.29	**93.86**±1.13	<u>94.85</u>±0.55	95.92±0.17
HEX† (Wang et al., 2019a)	10.80±0.40	16.60±0.80	19.70±1.90	24.70±1.60
RUBi† (Cadene et al., 2019)	13.70±0.70	43.00±1.10	90.40±0.40	<u>93.60</u>±0.40
ReBias† (Bahng et al., 2020)	22.70±0.40	64.20±0.80	76.00±0.60	88.10±0.60
BiasBal† (Hong and Yang, 2021)	<u>76.8</u>±1.6	<u>91.2</u>±0.2	<u>93.9</u>±0.1	**96.3**±0.2
U-EnD† (T=80)	53.90±4.03	82.16±0.63	74.39±0.43	88.05±0.16
U-EnD† (T=10)	55.29±1.27	85.94±0.33	92.92±0.35	93.48±0.06
U-FairKL+CE† (T=10)	**79.85**±1.14	**94.11**±0.76	**95.36**±1.04	89.24±0.05

Table 5.1.1: Biased-MNIST accuracy on the unbiased test set, with unsupervised extension. Techniques which can be used in an unsupervised way are denoted with †. The best results are highlighted in bold, the second best results are underlined.

It may be argued that in such cases of weaker bias (or even absence of it), the representations extracted by the biased encoder will be more aligned with the target class rather the the bias features. In this case, the resulting pseudo-labels will be less representative of the actual bias, leading to the disentangling, instead, of the target labels. We identify two worst-case scenarios that might lead to inaccurate pseudo-labels: *i.)* the training set is already unbiased, *ii.)* the pseudo-labels we identify correspond to the target rather than to the bias labels. In these cases, applying a debiasing technique might lead to worse performance with respect to the baseline, however, we are able to avoid this issue thanks to the hyper-parameters optimization policy that we employ. A more detailed analysis of the worst-case settings can be found in Appendix D.

Quantifying the model bias We can quantify how much bias has been learned by the bias-capturing model, by computing the conditional distribution of the prediction and the biases over an unbiased set. We call this quantity *unfairness* and indicate it with ϕ:

$$\phi = \frac{1}{|B|} \sum_{b \in B} [p_T(Y = y|b) + (1 - p_T(Y \neq y|b))] \tag{5.1.5}$$

where B is the set of different bias labels and y in this case is the target label that correlates with b in the training set. If a model is perfectly unbiased, ϕ will be at its minimum, while higher values of ϕ indicate that the model is more affected by the bias. In fact, for an unbiased model $p_T(Y = y|b) = 1/T$ and $1 - p_T(Y \neq y|b) = 1 - (T-1)/T$ thus $\phi = 1/T - 1 - (1 - 1/T) = 2/T$. For a biased model, if we quantify with $\hat{\rho}$ the actual conditional probability of the model $p_T(y|b) = \hat{\rho}$, we obtain $\phi = 2\hat{\rho}$. If we consider a completely biased model, that is $\hat{\rho} = 1$, then we have $\phi = 2$. We can use this unfairness quantity to study the trained bias-capturing models.

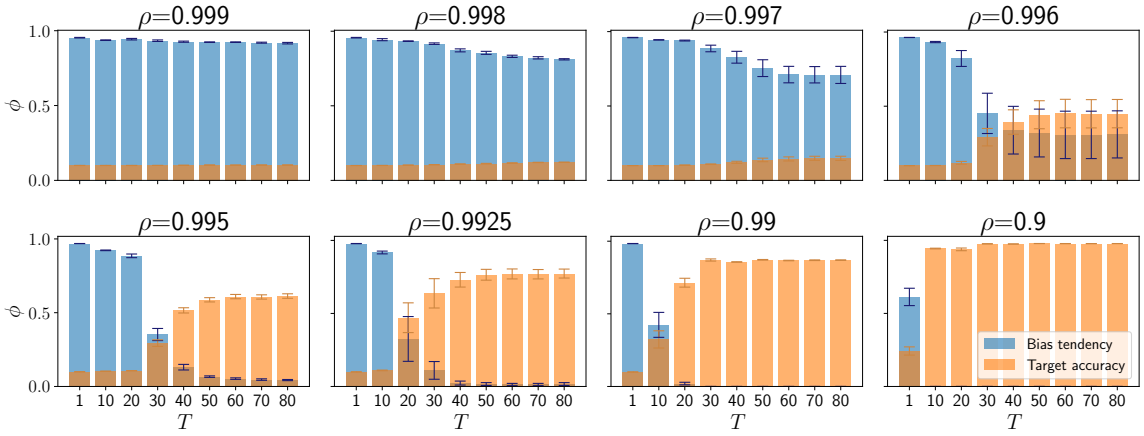


Figure 5.1.2: Unfairness (ϕ), or tendency towards learning bias features, as a function of the training epoch (T), in terms of mean and std computed across three independent runs for different values of ρ, normalized in the range $[0; 1]$ for comparison with the target accuracy.

Easier patterns are learned first Besides being easier to learn than the target task, as explained in Section 1.2.1, we also find that biases tend to be learned in the first epochs. This is also evident when looking at the results in Table 3.4.1, with $T = 10$: using an early bias predictor results in more precise pseudo-labels, especially when ρ is lower. In Figure 5.1.2 we show the value of the unfairness ϕ measured at different training iterations. As expected, the models tend to show stronger tendency towards bias when ρ is higher. Interestingly, looking at the dynamics it is also clear that this behavior is exhibited predominantly in earlier epochs. Under certain conditions, i.e. when the correlation between target and bias is not as strong, it is possible for the optimization process to escape the local minimum corresponding to a biased model. These findings are also confirmed by the related literature (Arpit et al., 2017; Nam et al., 2020). Especially in Arpit et al. (2017), the authors suggest that "the networks learn gradually more complex hypotheses during training for all the datasets" that they used. Of course, this phenomenon is clearly evident on simpler datasets. On more difficult and realistic datasets, measuring the unfairness and determining at which stage of the training the bias is most predominant would probably be less trivial (especially in the unsupervised case).

CelebA We report the results in Table 5.1.2. Results are reported for both the target attributes hair color and makeup. Techniques which can be used in an unsupervised manner are denoted with †. We report baseline results (vanilla) and we observe how vanilla models suffer significantly from the presence of the bias, scoring a quite low accuracy (especially since this is a binary task). This is evident on the bias-conflicting set, where the performance is close random-guess on hair color prediction, and even lower on the makeup detection. We report reference results (Nam et al., 2020) of other debiasing algorithms, specifically Group DRO (Sagawa et al., 2019), LfF (Nam et al., 2020) and EnD. Focusing on supervised techniques (Group DRO and EnD) we observe a significant increase in performance, in both the tasks and test sets combinations. For the unsupervised methods, we report results of our U-EnD at different T of the biased encoder, as done in Table 3.4.1, and compare to LfF. We achieve better performance than the vanilla baseline in all settings, even

Target	Method	Unbiased	Bias-conflicting
Hair Color	Vanilla (Nam et al., 2020)	70.25±0.35	52.52±0.19
	Group DRO (Sagawa et al., 2019)	<u>85.43</u>±0.53	<u>83.40</u> ±0.67
	EnD (Chapter 3)	**91.21**±0.22	**87.45**±1.06
	LfF† (Nam et al., 2020)	<u>84.24</u>±0.37	**81.24**±1.38
	U-EnD† (T=50)	83.97±2.90	<u>74.18</u>±6.07
	U-EnD† (T=30)	**84.39**±2.38	72.53±4.47
Heavy Makeup	Vanilla (Nam et al., 2020)	62.00±0.02	33.75±0.28
	Group DRO (Sagawa et al., 2019)	<u>64.88</u>±0.42	<u>50.24</u>±0.68
	EnD (Chapter 3)	**75.93**±1.31	**53.70**±5.24
	LfF† (Nam et al., 2020)	66.20±1.21	**45.48**±4.33
	U-EnD† (T=50)	**72.22**±0.00	<u>44.44</u>±0.00
	U-EnD† (T=30)	<u>67.59</u>±3.46	35.19±6.93

Table 5.1.2: Performance on CelebA. with the unsupervised extension. Techniques which can be used in an unsupervised way are denoted with †. The best results are highlighted in bold, the second best results are underlined.

though we still observe a gap with respect to the fully supervised techniques. The same observation can be made for LfF, which in general performs better on the harder cases in the bias-conflicting set, while U-EnD provides better performance in the more general case of the unbiased test set. The observed results are similar to the lower ρ settings of BiasedMNIST: the amount of biased information is sufficient for it to be considered as a malignant bias, although it becomes slightly harder to perform pseudo-labeling in the biased encoder latent space. However, the assumptions we make in Section 5.1.3 about the pseudo-labeling accuracy hold, resulting in better results with respect to the baseline models.

IMDB Face We report the results on the IMDB Face dataset in Table 5.1.3, with regards to both gender and age prediction. Besides the test set, every model is also tested on the opposite EB set, to better evaluate the debiasing performance. As in the previous experiments, we use † to denote the techniques which can be used in an unsupervised way. Focusing on the supervised techniques, we observe a significant improvement with respect to the baselines, especially with EnD and LNL, across the different combinations of test sets and task. Interestingly, in this case we are able to achieve even better results when employing the U-EnD approach, contrarily to the CelebA results. Especially for learning gender, we notice the the performance are noticeable higher than the best supervised results. This might be due to the noisy age labels in the dataset, and even if the described cleaning procedure is applied some labels could still be incorrect. With pseudo-labeling, on the other hand, we do not make use of the provided labels. This might be confirmed by the performances obtained when training for age prediction. As the gender label is of course far less noisy than the age, the performance gap between EnD and U-EnD is far less noticeable. We believe these results are very important, as they show that it is sometimes possible to achieve better results with unsupervised approaches.

Target	Method	Trained on EB1		Trained on EB2	
		EB2	Test	EB1	Test
Gender	Vanilla (Kim et al., 2019)	59.86	84.42	57.84	69.75
	BlindEye (Alvi et al., 2018)	63.74	85.56	57.33	69.90
	LNL (Kim et al., 2019)	<u>68.00</u>	86.66	64.18	74.50
	EnD (Chapter 3)	65.49±0.81	<u>87.15</u>±0.31	<u>69.40</u>±2.01	<u>78.19</u>±1.18
	U-EnD† (T=50)	**81.32**±2.17	**90.98**±0.46	**78.10**±0.70	**83.03**±0.45
Age	Vanilla (Kim et al., 2019)	54.30	77.17	48.91	61.97
	BlindEye (Alvi et al., 2018)	66.80	75.13	64.16	62.40
	LNL (Kim et al., 2019)	65.27	77.43	62.18	63.04
	EnD (Chapter 3)	<u>76.04</u>±0.25	<u>80.15</u>±0.96	**74.25**±2.26	**78.80**±1.48
	U-EnD† (T=50)	**80.41**±2.96	**83.43**±2.49	<u>70.82</u>±1.04	<u>76.09</u>±0.91

Table 5.1.3: Performance on IMDB Face with the unsupervised extension. When gender is learned, age is the bias, and when age is learned the gender is the bias. Techniques which can be used in an unsupervised way are denoted with †. The best results are highlighted in bold, the second best results are underlined

5.2 Debiasing without clusters: auxiliary models as prior

We have seen that clustering the biased latent space is a suitable approach for obtaining bias pseudo-labels. In this section, we show, with some preliminary experiments, that the clustering step might be avoided and that the biased encoder can be directly used as a bias similarity function, similarly to other works (Hong and Yang, 2021; Nam et al., 2020).

Avoiding the pseudo-labeling step can significantly reduce the training complexity, by removing the choice of a clustering algorithm and the related hyperparameters. Furthermore, employing a continuous score rather than a hard label might help in exploiting richer information and in being more robust against clustering errors.

5.2.1 FairKL with bias-capturing model

To use a continuous score, rather than a discrete bias label, we compute the similarity of the bias features $\tilde{b}_i^+ = s(g(x), g(x_i^+))$, where $g(\cdot)$ is the bias-capturing model. The bias similarity $\tilde{b}_i$ is used to obtain a weighted sample similarity: $\tilde{s}_i^{+,b} = s_i^+ \tilde{b}_i^+$ for bias-aligned samples, and $\hat{s}_i^{+,b'} = s_i^+(1-\tilde{b}_i^+)$ for bias-conflicting. By doing so, for example, the terms $\mu_{+,b} = \frac{1}{P_a}\sum_i d_i^{+,b}$ and $\mu_{+,b'} = \frac{1}{P_c}\sum_k d_k^{+,b'}$ become $\hat{\mu}_{+,b} = \frac{1}{N}\sum_i d_i^+ \hat{b}_i^+$ and $\hat{\mu}_{+,b'} = \frac{1}{N}\sum_i d_i^+(1-\hat{b}_i^+)$, where N is the batch size. By plugging these new definitions into Eq. 4.2.3, we obtain a new regularization term that can work with a continuous score[2].

[2]As in Section 5.1, the work presented in this section is the subject of ongoing research. We plan to include additional experiments and loss formulations in the future.

5.2.2 Experiments

9-Class ImageNet and ImageNet-A

We test our method on the more complex and realistic 9-Class ImageNet (Ilyas et al., 2019) dataset. This dataset is a subset of ImageNet, which is known to contain textural biases (Geirhos et al., 2019). It aggregates 42 of the original classes into 9 macro categories. Following Hong and Yang (2021), we train a BagNet18 (Brendel and Bethge, 2019) as the bias-capturing model, which we then use to compute a bias score for the training samples, to apply within our regularization term.

Setup We pretrain the bias-capturing model BagNet18 (Brendel and Bethge, 2019) for 120 epochs. For the main model ResNet18, we use the Adam optimizer, with learning rate 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$, weight decay of 0.0001 and a cosine decay of the learning rate. We use a batch size of 256 and train for 200 epochs. We employ as augmentation: random resized crop, random flip, and, as done in Hong and Yang (2021) random color jitter and random gray scale ($p = 0.2$). We use $\epsilon = 0.5$ and $\lambda = 1$. Given the higher complexity of this dataset, we employ $\alpha = 0.5$.

We evaluate the accuracy on the test set (biased) along with the unbiased accuracy (UNB), computed with the texture labels assigned in Brendel and Bethge (2019). We also report accuracy results on ImageNet-A (IN-A) dataset, which contains bias-conflicting samples (Hendrycks et al., 2021). Results are shown in Table 5.2.1. On the biased test set, the results are comparable with SoftCon, while on the harder sets unbiased and ImageNet-A we achieve SOTA results.

	Vanilla	SIN	LM	RUBi	ReBias	LfF	SoftCon	ϵ-SupInfoNCE + FairKL
Biased	94.0±0.1	88.4±0.9	79.2±1.1	93.9±0.2	94.0±0.2	91.2±0.1	**95.3**±0.2	<u>95.1</u>±0.1
UNB	92.7±0.2	86.6±1.0	76.6±1.2	92.5±0.2	92.7±0.2	89.6±0.3	<u>94.1</u>±0.3	**94.8**±0.3
IN-A	30.5±0.5	24.6±2.4	19.0±1.2	31.0±0.2	30.5±0.2	29.4±0.8	<u>34.1</u>±0.6	**35.7**±0.5

Table 5.2.1: Top-1 accuracy (%) on 9-Class ImageNet biased and unbiased (UNB) sets, and ImageNet-A (IN-A). Reference results from Hong and Yang (2021). The best results are highlighted in bold, the second best results are underlined.

5.3 Conclusions

In this Chapter, we have proposed a method for extending supervised debiasing techniques to unsupervised debiasing. We did that by leveraging our findings that

- neural networks tend to prefer simpler patterns (e.g. bias)
- biases tend to be learned early in the training.

Based on these observations, we proposed a way to recover the unknown bias labels by clustering the latent space of a biased model. The cluster labels were then used

as bias pseudo-labels for employing techniques such as EnD and FairKL. We have also shown that, instead of clustering, it is possible to directly integrate information from the bias-capturing model into the regularization term, thus reducing the overall training complexity.

We aim to improve this approach by:

- Using specific loss functions for training the bias-capturing model, such as generalized cross-entropy (GCE) (Zhang and Sabuncu, 2018) as also done in Nam et al. (2020). With GCE, more weight can be given to bias-aligned samples, achieving a stronger bias-capturing model;
- Training the bias-capturing model with self-supervised methods. In fact, it may be possible that training using CE on the target classes put some unnecessary constraints in the bias-capturing latent space, especially with regard to the number of clusters.

Part III

Collateral Learning in Medical Imaging

Chapter 6

Neuroimaging

In this Part, we turn our attention towards collateral learning in medical imaging. In Part II, we have developed methods aimed at fighting collateral learning in the general case of natural images. In this chapter, we focus on one specific instance of collateral learning in medical images: brain age prediction from multi-site imaging datasets.

As illustrated in the Introduction (1), medical datasets are often affected by the site-effect problem. Dealing with multi-center medical datasets has become one of the most predominant issues in the machine learning community (Dewey et al., 2019; Fortin et al., 2017; Glocker et al., 2019). Towards this aim, in this section, we leverage the results and methods obtained in the previous sections, building upon them to propose novel techniques, such as contrastive learning regression for brain age prediction. Having an accurate estimate of brain age has proved to be highly beneficial for detecting abnormal acceleration with respect to the chronological age, a phenomenon which is usually linked with cognitive decline and neurodegeneration (Cumplido-Mayoral et al., 2023; Elliott et al., 2021; Franke et al., 2010; Gaser et al., 2013; Koutsouleris et al., 2014; Millar et al., 2023).

Brain aging involves complex biological processes, such as cortical thinning, that are highly heterogeneous across individuals, suggesting that people do not age in the same manner. Accurately modeling brain aging at the subject level is a long-standing goal in neuroscience as it could enhance our understanding of age-related diseases such as neurodegenerative disorders. To this end, brain-age predictors linking neuroanatomy to chronological age have been proposed using Deep Learning (DL) (Peng et al., 2021).

Brain age is a relatively novel measure, originated from neuroimaging, and is usually obtained by training machine learning algorithms on structural magnetic resonance images (MRI) with the aim of predicting the patient age (Elliott et al., 2021). The difference between the predicted value based on a patient's MRI and their true chronological age is referred to as brain age delta.

In order to build accurate biomarkers of aging, DL models need large-scale neuroimaging datasets for training, which often involves multi-site studies, partly because of the high cost per patient in each study.

6.1 Building a robust brain age prediction model

Recent works have shown that DL models, and in particular Deep Neural Networks (DNN), largely over-fit site-related noise when trained on such multi-site datasets, notably due to the difference in acquisition protocols, scanner constructors, physical properties such as permanent magnetic field (Glocker et al., 2019; Wachinger et al., 2021). This also implies poor generalization performance on data from new incoming sites, highly limiting the applicability of these models to real-life scenarios. In order to build more robust and accurate brain age models insensitive to site, the OpenBHB challenge (Dufumier et al., 2022) has been recently released.

While most DNN used to derive brain age gap are usually trained as standard regressors with the optimization of mean absolute error (Cole et al., 2017; Jonsson et al.), Ridge or cross-entropy loss (Peng et al., 2021) (if age is binarized), these frameworks do not pay particular care about site-related information during training to produce robust representations of brain imaging data. On the other hand, contrastive learning paradigms for DNN training have been recently proposed in various contexts such as supervised (Khosla et al., 2020), weakly-supervised (Dufumier et al., 2021a, 2023; Tsai et al., 2022) and unsupervised representation learning (Chen et al., 2020). More importantly, as we have demonstrated in Chapter 4, contrastive learning has been shown to be more robust than traditional end-to-end approaches, such as cross-entropy, against noise in the data or the labels, resulting in better generalizing models (Graf et al., 2021; Khosla et al., 2020). For this reason, in this work, we propose a novel contrastive learning loss for regression in the context of the OpenBHB challenge, where chronological age must be learned without being affected by site-related noise. With our method, we obtain the best results in the official leaderboard.

Our contributions are twofold:

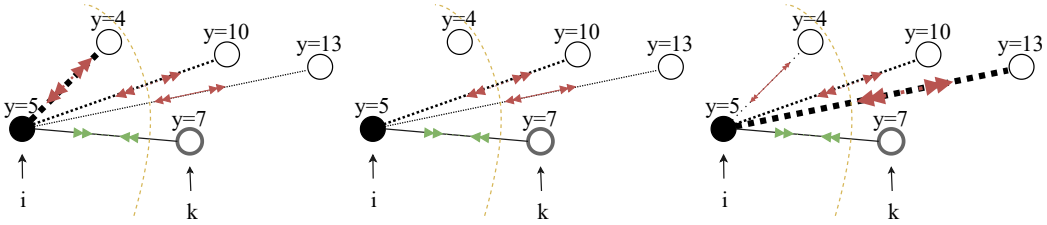
- We propose a novel contrastive learning regression loss for brain age prediction;
- We achieve state-of-the-art performance in brain age prediction on the OpenBHB challenge.

It is worth noting that, at the time of writing and to the best of our knowledge, the loss that we propose is one of the first attempts at employing contrastive learning for regression tasks.

6.1.1 The OpenBHB challenge

The OpenBHB challenge was launched with the goal of building robust brain-age prediction models. The core of the challenge lies in accurately predicting the age of patients from different acquisition sources, and in being robust to the problem of the site-effect, which affects many similar multi-site neuroimaging datasets.
The challenge is hosted on the RAMP platform[1], and provides a ranking for the submitted algorithms that take into account both the prediction accuracy and the robustness to site-effect. To achieve this, models are tested on a private internal

[1] https://ramp.studio/events/brain_age_with_site_removal_open_2022



(a) y-aware (b) threshold (c) exp

Figure 6.1.1: Comparison between different contrastive learning regression losses and their effect on the representations. Samples are aligned ($\gg \ll$) and repelled ($\ll\gg$) with varying strength (line thickness) based on the continuous label y, through the application of a kernel function w. Compared to the other losses, the behavior of our proposed loss $\mathcal{L}^{exp}$ is more desirable for regression, as the alignment in the representation space more closely reflects the one in the kernel space.

test set, which contains the same acquisition sites as the public training set, and on a private external test set, which contains independent sites from the training ones. The performance is measured in terms of Mean Absolute Error (MAE) for age prediction; furthermore, a Balanced Accuracy (BAcc) for classifying the different acquisition sites is computed on the challenge platform, with a logistic regression on the output representation. This metric helps in quantifying how much the learned representations are affected by the site noise (the lower the better). To provide a final ranking for the submitted algorithms, a challenge score $\mathcal{L}_c$ is computed as:

$$\mathcal{L}_c = \text{BAcc}^{0.3} \cdot \text{MAE}_{ext} \tag{6.1.1}$$

where MAE_{ext} is the MAE computed on the external private test set.

The OpenBHB challenge provides a large dataset, which is a comprehensive collection that aggregates 10 publicly available datasets: ABIDE 1 (Di Martino et al., 2013), ABIDE 2 (Di Martino et al., 2017), CoRR (Zuo et al., 2014), GSP (Holmes et al., 2015), LOCALIZER (Orfanos et al., 2017), MPI-Leipzig (**?**), NAR (Nastase et al., 2021), NPC (Sunavsky and Poppenk, 2020) and RBP (Follmer et al., 2018).

The dataset focuses on healthy control (HC) cases to model normal brain development and build a robust brain age predictor. It includes 5330 3D T1 brain MRI scans from HC acquired on 71 different acquisition sites with multiple acquisition protocols per site. The subjects come from European-American, European, and Asian genetic backgrounds, achieving demographic diversity in the dataset. The dataset provides participants' phenotype information, including age, sex, acquisition site, diagnosis, MRI scanner magnetic field, and MRI scanner settings identifier. Some common confounds are also included, such as Total Intracranial Volume (TIV), CerebroSpinal Fluid Volume (CSFV), Gray Matter Volume (GMV), and White Matter Volume (WMV). OpenBHB shows a well-balanced sex distribution for all age bins, with two main modes centered around 10 years old and 25 years old, and a long tail above 40 until 88 years.

6.1.2 A novel contrastive loss for regression

Supervised contrastive learning, as presented in Section 2.1.3 and Chapter 4, leverages discrete labels (i.e., classes) to define positive and negative samples. Starting from a sample x_i, called the *anchor*, and its latent representation $z_i = f(x_i)$, contrastive losses such as SupCon or ϵ-SupInfoNCE align the representations of all positive samples (i.e. sharing the same class as x_i) to z_i, while repelling the representations of the negative ones (i.e., different class). The notion of "negative" (dissimilar from the anchor) and "positive" (similar to the anchor) samples is thus rooted in the contrastive learning framework.

These losses are thus not adapted for regression, where the target is a continuous variable, as it is not possible to determine a hard boundary between positive and negative samples. All samples are somehow positive and negative at the same time. Given the continuous label y_i for the anchor and y_k for a sample k, one could threshold the distance d between y_i and y_k at a certain value τ in order to create positive and negative samples (i.e., k is positive if $d(y_i, y_k) < \tau$), as done in Xue et al. (2022). The problem would then be how to choose τ.

Differently, we propose to define a degree of "positiveness" between samples using a kernel function $w_k = K(y_i - y_k)$, where $0 \leq w_k \leq 1$, for example a Gaussian kernel or a Radial Basis Function (RBF) kernel. Our goal is thus to learn a parametric function $f : \mathcal{X} \to \mathbb{S}^d$ that maps samples with a high degree of positiveness ($w_k \sim 1$) close in the latent space and samples with a low degree ($w_k \sim 0$) far away from each other. To derive our proposed loss, we employ the same metric learning approach that we presented in Chapter 4, which allows us to easily add conditioning and regularisation. Thanks to it, we are able to develop multiple formulations, which are illustrated in Figure 6.1.1.

Recalling the basics of our metric framework, we aim at satisfying the following condition on the representation space (Eq. 4.1.4):

$$s_t^- - s_k^+ \leq 0 \quad \forall t, k \tag{6.1.2}$$

where $s_j^+ = sim(f(x_i), f(x_j^+))$ and, for simplicity, we impose $\epsilon = 0$. In the regression case, however, we no longer distinguish between positive and negative samples. A first possible approach would be to consider as "positive" only the samples y_k that have a degree of positiveness w_k greater than 0, and align them with a strength proportional to the degree, namely:

$$\frac{w_k}{\sum_j w_j}(s_t - s_k) \leq 0 \quad \forall j, k, t \neq k \in A(i) \tag{6.1.3}$$

where we have normalized the kernel so that the sum over all samples is equal to 1 and we denote with $A(i)$ the indices of samples in the minibatch distinct from x_i. Following the same steps that we showed in Section 4.1, the starting metric condition in Eq. 6.1.3 can be transformed in an optimization problem using, the max operator and its smooth approximation *LogSumExp*:

$$\begin{aligned}
&\arg\min_f \sum_k \max(0, \frac{w_k}{\sum_t w_t}\{s_t - s_k\}_{\substack{t=1,\dots,N \\ t\neq k}}) = \\
&\arg\min_f \sum_k \frac{w_k}{\sum_t w_t} \max(0, \{s_t - s_k\}_{\substack{t=1,\dots,N \\ t\neq k}}) \\
&\approx \mathcal{L}^{y-aware} = -\sum_k \frac{w_k}{\sum_t w_t} \log\left(\frac{\exp(s_k)}{\sum_{t=1}^{N}\exp(s_t)}\right)
\end{aligned} \tag{6.1.4}$$

Interestingly, this is exactly the *y-aware* loss proposed in Dufumier et al. (2021b) for classification with weak continuous attributes. Due to the non-hard boundary between positive and negative samples, both s_t and s_k are defined over the entire minibatch. The kernel w_k is used to avoid aligning samples not similar to the anchor (i.e. $w_k \approx 0$). It can be noted that, while the numerator aligns x_k, in the denominator, the uniformity term (as defined in Wang and Isola (2020) and in Eq. 2.1.13) focuses more on the closest samples in the representation space, due to the exponential term (which gives more importance to higher s_t values). This could be undesirable, as the samples for which s_t is higher, might have a greater degree of positiveness than the considered x_k. This phenomenon is illustrated in Figure 6.1.1a. Of course, this goes against the goal of obtaining a semantic mapping between the kernel space and the learned representation space.

To avoid that, we formulate a first extension ($\mathcal{L}^{thr}$) of equation 6.1.3, which limits the uniformity term (i.e., denominator) to the samples that are at least more distant from the anchor than the considered x_k in the kernel space (omitting the normalization in the starting condition):

$$w_k(s_t - s_k) \leq 0 \quad \text{if } w_t - w_k \leq 0 \quad \forall k, t \neq k \in A(i) \tag{6.1.5}$$

Using an indicator function $\delta_{w_t<w_k}$ to express the "if" condition, we can obtain the following loss function:

$$\mathcal{L}^{thr} = -\sum_k \frac{w_k}{\sum_t \delta_{w_t<w_k} w_t} \log\left(\frac{\exp(s_k)}{\sum_{t\neq k}\delta_{w_t<w_k}\exp(s_t)}\right) \tag{6.1.6}$$

Ideally, $\mathcal{L}^{thr}$ avoids repelling samples more similar than x_k, by using the value w_k as threshold (hence the name). However, it still focuses more on the closest sample "less positive" than x_k, i.e. x_t s.t $w_t > w_x$ and $w_t \leq w_j \ \forall j \neq k$. This is shown in Figure 6.1.1b, in which the highest repulsion strength is focused on $y = 10$. The idea of limiting the uniformity term based on the label distance can be found also in Zha et al. (2022) (in this work, however, the alignment term is not weighted). Compared to y-aware, which repels more $y = 4$, $\mathcal{L}^{thr}$ is better, however, it is still not optimal. As noted in Section 4.1 and in Khosla et al. (2020), increasing the margin with respect to the closest "negative" sample works well for classification, however, it might not be best suited for regression.

For this reason, we propose a second formulation ($\mathcal{L}^{exp}$) that takes an opposite approach. Instead of focusing on repelling the closest "less positive" sample, we increase the repulsion strength for samples proportionally to their distance from the

Loss	Behavior
$\mathcal{L}^{y-aware}$	Align samples based on the kernel distance repel all
$\mathcal{L}^{thr}$	Align samples based on kernel distance , repel only samples more distant than k
$\mathcal{L}^{exp}$	Align samples based on kernel distance, repel more samples with greater distance

Table 6.1.1: Summary of the proposed contrastive losses with kernel weighting.

anchor in the kernel space. This is achieved by weighting not only the alignment term, but also the uniformity term:

$$w_k[s_t(1-w_t)-s_k] \leq 0 \quad \forall k, t \neq k \in A(i) \tag{6.1.7}$$

which yields the following loss function:

$$\mathcal{L}^{exp} = -\frac{1}{\sum_t w_t} \sum_k w_k \log \frac{\exp(s_k)}{\sum_{t \neq k} \exp(s_t(1-w_t))} \quad . \tag{6.1.8}$$

In the resulting $\mathcal{L}^{exp}$ formulation, the weighting factor $1-w_t$ acts like a (varying) temperature value, by giving more weight to the samples that are farther away from the anchor in the kernel space. Figure 6.1.1c shows the behavior of $\mathcal{L}^{exp}$: the repulsion strength is proportional to the difference of the y values. Taking a closer look, we might see that also closer samples are repelled; however, for a proper kernel choice, samples closer than x_k will be repelled with very low strength ($\sim$0). We argue that this approach is more suited for continuous attributes (i.e., regression task), as it enforces that samples close in the kernel space will be close in the representation space, thus achieving the semantic mapping between the kernel and the representation space.

All of the losses that we introduced are summarized in Table 6.1.1, with a short description characterizing each one of them.

6.1.3 Experiments and Results

As network architecture, we employ the 3D implementations of ResNet-18 (33.2M parameters), AlexNet (2.5M parameters), and DenseNet-121 (11.3M parameters). For comparison with Dufumier et al. (2022), we use the Adam optimizer, with an initial learning rate of 10^{-4} decayed by a factor of 0.9 every 10 epochs, and a weight decay of $5 * 10^{-5}$. We use a batch size of 32, and train for a total of 300 epochs. Our trainings are implemented in PyTorch, and run on the Jean Zay cluster[2] and on a cluster of 8 NVIDIA A40 GPU, with a single training taking 24h. For every model, we report the mean absolute error (MAE) on both the internal and external test sets, along with the balanced accuracy for site classification (BAcc). We also report the final challenge score $\mathcal{L}_c$ (the lower the better).

[2]http://www.idris.fr/jean-zay/

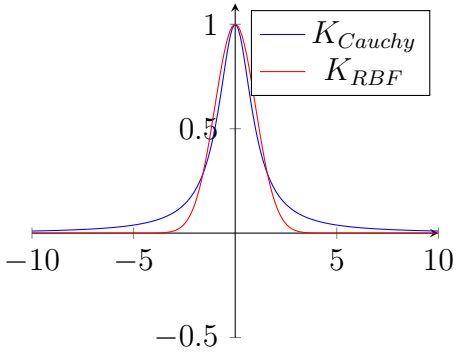


Figure 6.1.2: Employed kernel functions.

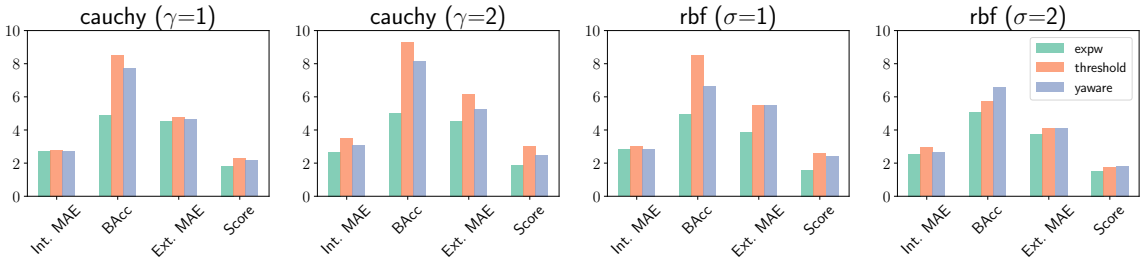


Figure 6.1.3: Ablation study of the kernel functions. The Gaussian kernel (rbf) with $\sigma = 2$ yields the best generalization results (Ext. MAE) and final score across all three loss functions. We also notice an overall slight improvement in the site balanced accuracy.

Experimental data We conduct our experiments on the OpenBHB dataset, described in Section 6.1.1. We focus this study on gray matter volumes (VBM).

Kernel function ablation study We test two different kernels: a Gaussian kernel $K_g(u) = \exp\left(-||u||^2/2\sigma^2\right)$ and Cauchy kernel $K_c(u) = 1/(\gamma||u||^2+1)$, illustrated in Figure 6.1.2. We perform an ablation study for the two different kernels and hyperparameters, employing a ResNet-18 model. Figure 6.1.3 shows the ablation results. For each kernel choice and value, we report the metrics on the test set along with the final challenge score, for the three loss functions. Focusing on the final score, it's easy to see that a Gaussian kernel with $\sigma = 2$ produces the best results for all losses (for readability, the final score is also reported in Table 6.1.2). This can be attributed to the overall lower error on the external set (Ext. MAE), showing that, with this setting, the models can generalize better. Furthermore, we also notice an overall lower balanced accuracy for site prediction, showing that this configuration is somewhat more robust to site noise.

Comparison of contrastive regression losses In Table 6.1.3 we compare the results obtained with the different losses. Focusing on the aggregate score, the best results are obtained with $\mathcal{L}^{exp}$ (1.54). Furthermore, $\mathcal{L}^{exp}$ also outperforms the other losses in every evaluated metric. Most significantly, it shows the best generalization capability in the external test set, which, undoubtedly, is the most relevant result from a practical clinical perspective. On the internal test, we score a MAE of 2.55, which is also slightly better than the related literature on a similarly sized dataset

Kernel	σ / γ	$\mathcal{L}^{y-aware}$	$\mathcal{L}^{threshold}$	$\mathcal{L}^{exp}$
Cauchy	1	2.15	2.28	1.82
	2	2.48	3.03	1.83
RBF	1	2.43	2.63	1.58
	2	**1.82**	**1.74**	**1.54**

Table 6.1.2: Ablation study of kernel functions, in terms of challenge's score. Best results are highlighted in bold.

Method	Int. MAE	BAcc	Ext. MAE	$\mathcal{L}_c$
$\mathcal{L}^{y-aware}$	<u>2.66</u>±0.00	6.60±0.17	4.10±0.01	1.82
$\mathcal{L}^{thr}$	2.95±0.01	<u>5.73</u>±0.15	4.10±0.01	1.74
$\mathcal{L}^{exp}$	**2.55**±0.00	**5.1**±0.1	**3.76**±0.01	**1.54**

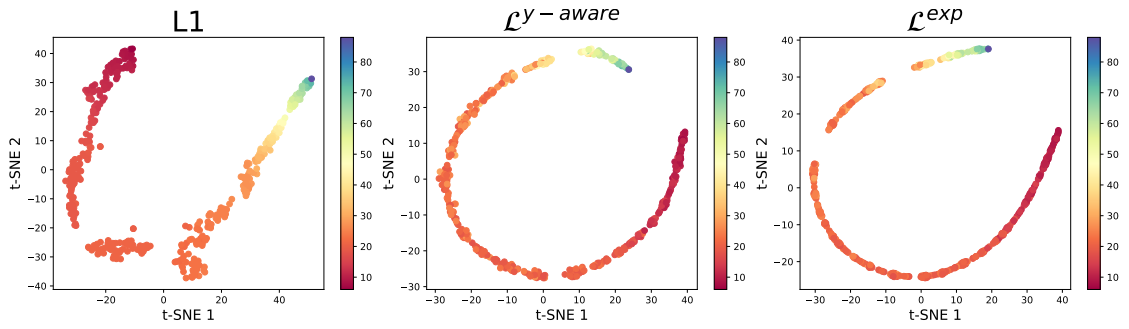
Table 6.1.3: Comparison of contrastive losses on the OpenBHB challenge dataset. The best results are highlighted in bold.

with UKB (Peng et al., 2021). Interestingly, $\mathcal{L}^{exp}$ also shows the best robustness to site-related noise (with a BAcc of 5.1), which indicates that the learned space preserves the neuroanatomical features very well while also removing site noise.

Final results on the OpenBHB challenge Finally, we report the ranking of $\mathcal{L}^{exp}$ of the OpenBHB leaderboard, testing also AlexNet and DenseNet-121. Table 6.1.4 shows the results. We compare with baseline models (Dufumier et al., 2022) trained with the L1 loss, and with ComBat (Fortin et al., 2017), a site harmonization algorithm developed for MRIs. Our proposed $\mathcal{L}^{exp}$ achieves state-of-the-art performance on the final leaderboard, scoring the best final score and metrics on both the internal and external test set, with ResNet-18. The improvement observed in the external test is also reflected for both AlexNet and DenseNet compared to all baselines. For these models, the internal MAE reached by the L1 baseline is slightly

Method	Model	Int. MAE	BAcc	Ext. MAE	$\mathcal{L}_c$
Baseline (ℓ_1)	DenseNet	2.55±0.01	8.0±0.9	7.13±0.05	3.34
	ResNet-18	2.67±0.05	6.7±0.1	4.18±0.01	1.86
	AlexNet	2.72±0.01	8.3±0.2	4.66±0.05	2.21
ComBat	DenseNet	5.92±0.01	2.23±0.06	10.48±0.17	3.38
	ResNet-18	4.15±0.01	**4.5**±0.0	4.76±0.03	1.88
	AlexNet	3.37±0.01	6.8±0.3	5.23±0.12	2.33
$\mathcal{L}^{exp}$	DenseNet	2.85±0.00	5.34±0.06	4.43±0.00	1.84
	ResNet-18	**2.55**±0.00	5.1±0.1	**3.76**±0.01	**1.54**
	AlexNet	2.77±0.01	5.8±0.1	4.01±0.01	1.71

Table 6.1.4: Final scores on the OpenBHB leaderboard. Reference results from Dufumier et al. (2022). The best results are highlighted in bold.



(a) (b) (c)

Figure 6.1.4: Visualization of the learned representation space by a baseline L1 model (a), a model trained with y-aware (b) and one trained with $\mathcal{L}^{exp}$ (c). The color represents the age. Compared to the L1 baseline, it is easy to see how the learned representations are more smoothly aligned. Furthermore, going from left to right, it appears that progressively less additional information (e.g. site noise) is encoded in the latent space: for L1, we observe that the latent space, compared to the contrastive losses, tends to form larger and disconnected clusters; while for $\mathcal{L}^{y-aware}$ it is possible to observe that samples with a similar age more loosely arranged compared to $\mathcal{L}^{exp}$. The t-SNE was run for 5000 iterations in order to guarantee convergence, with a perplexity value of 15 in order to balance local and global similarities in the data.

lower than $\mathcal{L}^{exp}$. However, when looking at the other metrics, it is easy to see that this is due to more overfitting on the internal sites for the baseline. Lastly, regarding the balanced accuracy, we observe a significant improvement with respect to the L1 baseline, showing that $\mathcal{L}^{exp}$ possesses some debiasing capability towards site noise. Besides AlexNet, however, ComBat still achieves a lower accuracy, showing that there is room for improvement.

Why $\mathcal{L}^{exp}$ is more invariant to site-effect As we observed from the results, $\mathcal{L}^{exp}$ exhibits the lowest balance accuracy among all losses, meaning that it is somehow more robust towards the site effect of the data. We hypothesize that this is due to the difference in the formulation of $\mathcal{L}^{exp}$ with respect to the other losses, such as y-aware. As explained in Section 6.1.2, the presence of the exponential in the uniformity term of y-aware, may force some samples to be pushed apart from the anchor more than they should (i.e. when $s_t > s_k$). This could result in a less "compact" latent space, that might capture more variance in the input data such as noise from the site effect. On the other hand, $\mathcal{L}^{exp}$ may avoid this phenomenon as the samples are repelled according to the kernel value, as shown in Figure 6.1.1, achieving latent representations that are more invariant to such noise. To illustrate this concept, in Figure 6.1.4 we visualize the learned latent space of the models with t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm (Van der Maaten and Hinton, 2008), obtained on the local validation set of the OpenBHB dataset. Altough the visual difference between $\mathcal{L}^{y-aware}$ and $\mathcal{L}^{exp}$ is marginal, it appears that the result of the visualization agrees with our guess. This explanation is however still just a conjecture and a more in-depth analysis of such behavior should be the focus of future work.

6.1.4 Addressing site effect with regularization

As we have seen in the previous section, in the context of the OpenBHB challenge, in addition to learning meaningful representation for age prediction, biases related to the acquisition sites must be taken into account. We showed how $\mathcal{L}^{exp}$ improves in this regard with respect to the other losses, however its debiasing performance is still limited when compared to specialized method such as ComBat.

For this purpose, we propose to extend the FairKL regularization presented in Section 4.2 and extend it to the regression case.
As a brief reminder, FairKL aims at minimizing the Kullback-Leibler divergence of the distance distributions of positive bias-aligned $B_{+,b} \sim \mathcal{N}(\mu_{+,b}, \sigma^2_{+,b})$ and positive bias-conflicting and $B_{+,b'} \sim \mathcal{N}(\mu_{+,b'}, \sigma^2_{+,b'})$:

$$\mathcal{R}^{FairKL} = \frac{1}{2}\left(\frac{\sigma^2_{+,b} + (\mu_{+,b} - \mu_{+,b'})^2}{\sigma^2_{+,b'}} - \log\frac{\sigma^2_{+,b}}{\sigma^2_{+,b'}} - 1\right) \tag{6.1.9}$$

where $\mu_{+,b} = \frac{1}{P_a}\sum_i s_i^{+,b}$ and $\sigma^2_{+,b} = \frac{1}{P_a}\sum_i (s_i^{+,b} - \mu_{+,b})^2$ are the first and second moments of the distance distribution for the positive bias-aligned samples ($\mu_{+,b'}$ and $\sigma_{+,b'}$ are defined in the same way on positive bias-conflicting), and P_a is the number of positive bias-aligned samples. Similarly to Section 6.1.2, given that we cannot discretely identify positive samples, we propose to extend FairKL to the continuous target label by employing a kernel, obtaining:

$$\begin{aligned}
\mu_{+,b} &= \frac{1}{\sum_{b\in B(i)} w_b}\sum_{b\in B(i)} w_b s_b \\
\sigma^2_{+,b} &= \frac{1}{\sum_{b\in B(i)} w_b}\sum_{b\in B(i)} (w_b s_b - \mu_{+,b})^2
\end{aligned} \tag{6.1.10}$$

where, denoting with b_j the bias of the j-th sample, $B(i) \equiv \{k \in A(i) \mid b_j = b_i\}$ is the set of indices of all bias-aligned samples. The same reasoning can be applied for positive bias-conflicting samples. With this approach, we expect to be able to achieve better results towards mitigating the site-effect[3].

6.2 Detecting Alzheimer's Disease and Cognitive Impairment

In this section, we turn our attention towards detecting neurodegeneration from brain MRIs. Linking brain age with the insurgence of neurodegenerative diseases, such as Alzheimer's Disease, and cognitive decline has been proposed by related literature (Elliott et al., 2021; Gaser et al., 2013) and shows promising results in this direction. Leveraging the methods and models proposed in the previous section, we aim to improve the detection of such conditions. For this purpose, we perform a set of experiments to assess whether our proposed contrastive learning framework for brain age regression might help in achieving more accurate results[4].

[3] Ongoing work.
[4] This section is part of ongoing work. The results are preliminary.

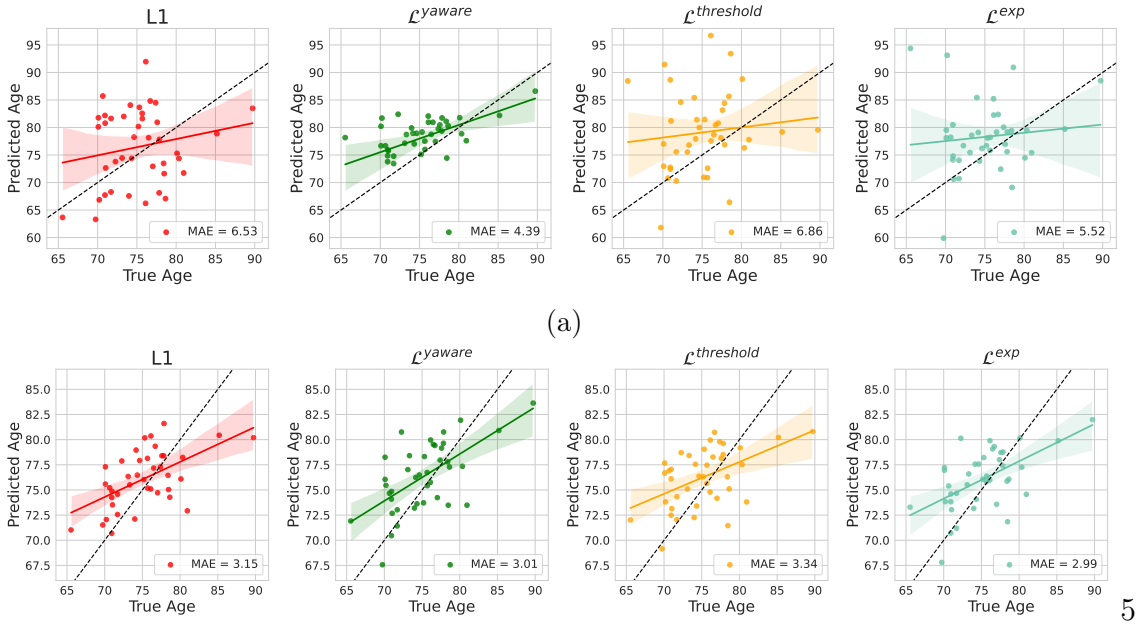


Figure 6.2.1: Performance of age regression on ADNI HC group: (a) raw adjusted predictions (b) finetuned predictions. Age predicted by each model (y axis) is plotted against true age (x axis). Colored lines and shaded areas represent regression lines and 95% confidence regions. Dashed black lines represent perfect prediction.

Experimental data We leverage the data from the Alzheimer's Disease Neuroimaging Initiative (Petersen et al., 2010) (from the ADNI-GO collection)[5]. We use 716 co-registered T1-weighted MRI images divided in 199 Healthy Control (HC) cases, 329 Mild Cognitive Impairment cases (MCI) and 188 Alzheimer's Disease (AD) patients. We only included one scan per patient at the first session (baseline). All images have been pre-processed in the same way with a non-linear registration to the MNI template and a gray matter extraction step. The final spatial resolution is 1.5mm isotropic and the images are of size $121 \times 145 \times 121$. To conduct our experiments, we adopt a 80%-20% train-test split at the patient level, obtaining 572 patients for training and 144 for testing.

6.2.1 Finetuning age prediction

First of all, we assess the generalization capability of the age prediction models trained on OpenBHB on the ADNI dataset. For this purpose, we only consider the HC group, as the brain age should (approximately) match the chronological one. In order to evaluate the prediction, we consider two approaches:

- using the raw predictions from the models, adjusted just by a bias corrective term δ_{age} in order to take into account the possible dataset shift;
- finetuning a linear regression layer on top of the frozen encoder.

For the first approach, we compute δ_{age} as $\mathbb{E}[y_{HC}^{train}] - \mathbb{E}[\hat{y}_{HC}^{train}]$, where y is the ground truth label and $\hat{y}$ is the model prediction. For the second approach, for compar-

[5] http://adni.loni.usc.edu/about/adni-go

Method	OpenBHB BAcc	OpenBHB Int. MAE	OpenBHB Ext. MAE	ADNI HC MAE (ft)	Avg. MAE
L1	6.7±0.1	2.67±0.05	4.18±0.01	3.15	3.34
$\mathcal{L}^{y-aware}$	6.60±0.17	<u>2.66</u>±0.00	4.10±0.01	<u>3.01</u>	<u>3.26</u>
$\mathcal{L}^{thr}$	<u>5.73</u>±0.15	2.95±0.01	4.10±0.01	3.34	3.46
$\mathcal{L}^{exp}$	**5.1**±0.1	**2.55**±0.00	**3.76**±0.01	**2.99**	**3.10**

Table 6.2.1: Summary of age regression results on OpenBHB and ADNI HC (ft stands for finetuned). The best results are highlighted in bold, the second best results are underlined.

ison with how the results are computed on the OpenBHB leaderboard (Dufumier et al., 2022), we employ the Ridge regression (Hoerl and Kennard, 2000) implementation provided by sklearn (Pedregosa et al., 2011). Figure 6.2.1 shows the result of age regression on ADNI. Using raw-adjusted predictions, we achieve a minimum MAE of 4.39 with $\mathcal{L}^{y-aware}$, which is better than related literature on similarly-sized datasets (Cumplido-Mayoral et al., 2023; Millar et al., 2023). Finetuning the prediction layers on the ADNI HC group improves the results notably, lowering the MAE to 2.99 for $\mathcal{L}^{exp}$. It is important to keep in mind that the whole encoder is frozen, thus the representation space is the same as learned on the OpenBHB dataset. The only purpose of finetuning the regression layer is to minimize the dataset shift. The results of age regression for both OpenBHB and ADNI are summarized in Table 6.2.1. Our proposed loss $\mathcal{L}^{exp}$ consistently outperforms the other losses in each test set; this is probably due to the increased robustness to the site-effect (lower BAcc).

6.2.2 Using brain-age delta for detecting neurodegeneration

Based on the age prediction models that we obtained in the previous section, we now compute the age prediction for the rest of the ADNI dataset, including MCI and AD cases. We are interested in computing the brain-age delta, given by the difference between the predicted brain age value and the true chronological age of the patient. As shown in Cumplido-Mayoral et al. (2023); Elliott et al. (2021); Gaser et al. (2013); Millar et al. (2023), brain-age delta can be a useful proxy for detecting neurodegeneration.

In Figure 6.2.2 we report the accuracy for detecting MCI and AD against HC cases at different values of brain-age delta. To compute the accuracy, we consider the patient as AD (or MCI) if the brain-age delta for that patient is higher than a certain threshold (x axis). We compare the different contrastive losses and the baseline model (L1). As expected, we found the lowest accuracy (∼50%) with very small age deltas (e.g. 1 or 2) in most cases, meaning that with such a small delta it is impossible to discriminate MCI or AD cases from healthy ones. For all models, except for $\mathcal{L}^{exp}$, we can clearly see that the peak accuracy for AD is found at a higher age delta than for MCI. Focusing on $\mathcal{L}^{exp}$, the peak for AD is less noticeable, however, the accuracy achieved is the highest. We do not have a clear explanation on the reason for such behavior. The highest accuracy of $\mathcal{L}^{exp}$ is also confirmed by the ROC curve,

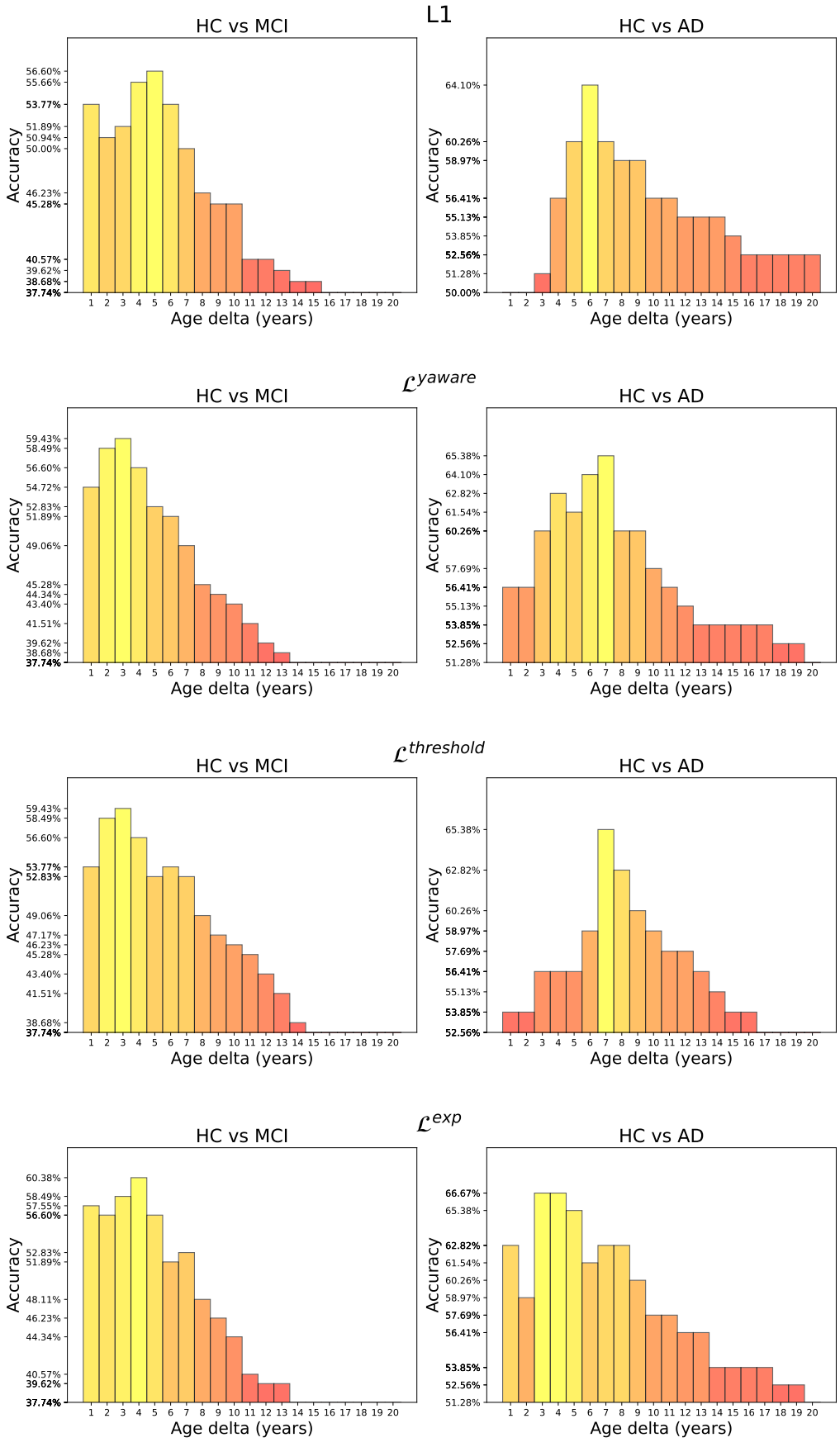


Figure 6.2.2: Accuracy of MCI and AD detection at increasing brain-age delta thresholds.

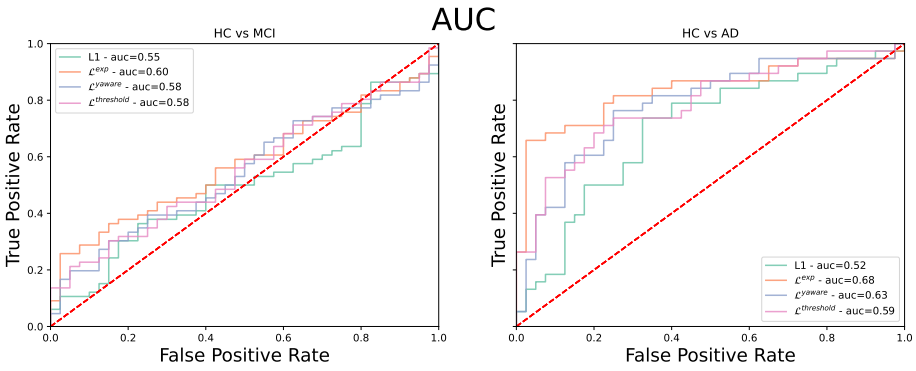


Figure 6.2.3: ROC curve for AD and MCI detection with brain-age delta.

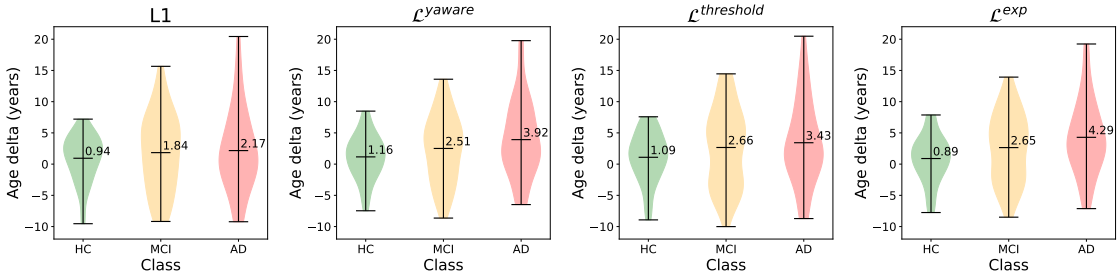


Figure 6.2.4: Age delta distribution for HC, MCI and AD classes. The shaded areas represent the age delta distribution (histogram) for each class. The horizontal black line in the middle highlights the mean value (explicitly annotated for readability).

shown in Figure 6.2.3, which reaches the highest AUC value for both classification tasks.

To conclude the analysis on the correlation between brain-age delta and neurodegeneration, in Figure 6.2.4 shows the mean brain-age delta for each of the three classes. Again, $\mathcal{L}^{exp}$ seems to work better as there is a larger gap between the different groups, and the average age delta on HC patients is the lowest. From these results, we observe that we most of the healthy patients exhibit a difference in chronological and brain age of 0.89 years, while patients affected by MCI and AD show a delta on average of 2.65 years and 4.29 years respectively. Of course, these are the results of a preliminary analysis, and they should be improved in future works, for example by including sex stratification, which was not taken into account in this work.

6.2.3 Transfer learning for AD and MCI detection

Finally, similarly to what we did in Section 6.2.1 in order to achieve better results, we trained a classification layer on top of the frozen encoders in order to discriminate among HC, MCI and AD classes. For this purpose, we employed a Logistic Regression classifier with a 5-fold cross-validation on the training set in order to find the best regularization hyperparameters. The results are reported in Table 6.2.2 in terms of accuracy. For each class combination, we train a separate classifier and test it on the corresponding test set. We report the accuracy of a model trained from scratch on the ADNI dataset with "target". Also in this case, by employing the

Method	All	HC vs AD	HC vs MCI	MCI vs AD	Avg.
Baseline (L1)	0.60	0.83	0.68	0.76	0.72
$\mathcal{L}^{y-aware}$	0.61	0.88	0.69	0.76	0.73
$\mathcal{L}^{thr}$	**0.64**	0.86	**0.73**	0.73	<u>0.74</u>
$\mathcal{L}^{exp}$	<u>0.62</u>	**0.90**	<u>0.71</u>	<u>0.80</u>	**0.76**
Target	<u>0.62</u>	**0.90**	<u>0.64</u>	**0.82**	<u>0.74</u>
Bäckström et al. (2018)	-	0.90	-	-	
Cheng and Liu (2017)	-	0.85	-	-	
Korolev et al. (2017)	-	0.80	-	-	
Li et al. (2017)	-	0.90	0.74	-	
Senanayake et al. (2018)	-	0.76	0.75	0.76	
Valliani and Soni (2017)†	0.57	0.81	-	-	

Table 6.2.2: Transfer learning results on ADNI in terms of accuracy. "Target" means a model trained on ADNI from scratch. † the results in Valliani and Soni (2017) are on 2D slices rather then 3D volumes; however it was included for comparison with the "all" classification. The best results are highlighted in bold, the second best results are underlined

Method	Backbone	HC vs AD
$\mathcal{L}^{exp}$	ResNet-18	0.95
Dufumier et al. (2021b)	DenseNet-121	0.96

Table 6.2.3: Transfer-learning results for ADNI HC vs AD in terms of AUC.

contrastive losses we are able to consistently improve upon the L1 baseline. $\mathcal{L}^{exp}$ achieves the best results on average and obtains an improvement with respect to the target performance. This means that the pre-training has provided a better initialization for the transfer-learning task. We also report some reference results from the literature, taken from Wen et al. (2020). The results we achieve are competitive with the other works, however it should be noted that many of these works use larger datasets, which in neuroimaging has been shown to correlate with a decrease in prediction accuracy (Varoquaux and Cheplygina, 2021). Finally, for additional comparison on the same dataset, we report in Figure 6.2.3 the AUC score for the HC vs AD task, compared with Dufumier et al. (2021b) where they finetune the whole model on the ADNI dataset. Notably, we achieve a competitive result by using a smaller backbone, compared to the DenseNet-121 employed in their study.

6.3 Limitations and Conclusions

In this chapter, we have made the following contributions:

1. We have proposed a novel loss for contrastive learning suited for regression, which leverages a "degree of positiveness" by employing a kernel function to measure the similarity of the target variable;

2. We have shown that our proposed loss achieves state-of-the-art performance for brain age prediction on the OpenBHB challenge, and is inherently more robust to collateral learning (e.g. site effect) than other losses

3. We have performed some preliminary experiments hinting the proposed method is promising with regard to a more accurate detection of neurodegenerative conditions.

The results that we presented show how a formal analysis of representation learning through a theoretical framework, such as the one we proposed, can help in deriving more effective loss functions for a specific task. Our simple, yet effective, metric approach, has allowed us to propose tackle the issues of existing contrastive losses. We have also shown how collateral learning can effectively cripple the model prediction accuracy, making it less reliable and subject to issues such as site-effect. This is an important step towards building robust and reliable deep-learning models and is especially relevant in the neuroimaging area. Our results further confirm the link between brain age acceleration and cognitive decline, and we aim to improve our results in future work.

Nonetheless, our analysis still presents some limitations. The most important ones are:

- Even if less than the other methods, our brain age prediction approach is still affected by the site noise to some extent. This can be observed when comparing the debiasing effect with methods such as ComBat. There is still room for improvement, and one possible direction is represented by the proposed extension of the FairKL regularization to regression tasks;

- Similarly, while promising, the results for MCI and AD detection should be improved. Also, in our analysis we did not make distinctions between different classes of MCIs such as sMCI (patients who will remain stable) and pMCI (those who will progress to AD). This is a very relevant task from a clinical perspective;

- Some factors such as the patient's sex have not been taken into account in this study; however it has been shown in the related literature that anatomical differences in the brain aging process exist between males and females. A detailed analysis on how to better include this information in the models should be performed (e.g. using multiple models based on sex);

- The considered sample size for MCI and AD detection is not very large, and the population sample should be extended to include the latest ADNI iterations and other publicly available datasets. In fact, as highlighted in Varoquaux and Cheplygina (2021), using smaller datasets can result in more optimistic performance estimates;

- Related to the latter point, the analysis should also be extended to include other kinds of conditions such as Schizophrenia, Bipolar disorder, and Autism spectrum disorder. Furthermore, it could be relevant to assess whether the models are focusing on

- Also, longitudinal studies should be included in the analysis, as done e.g. in Franke and Gaser (2019), as they provide a way to assess an individual's aging process over time, and can provide useful information about the different conditions.

We aim to focus on these limitations in our future research.

Chapter 7

The COVID-19 Experience

Among all the events that recently affected the world, the most remarkable can be probably identified with the Covid-19 pandemic. During the beginning of 2020, Covid-19 virus has rapidly spread in China and out into multiple countriess worldwide (Zu et al., 2020). As of September 2023, there were more than 770 million confirmed cases worldwide, with almost 7 million deaths[1].

As the pandemic spread and lockdown measures were adopted by most countries, the attention of the deep learning community turned towards aiding the detection of early symptoms of Covid-19 infections. To this end, Chest X-Ray (CXR) imaging was regarded as the most favorable imaging methodology, as it is quicker, and, most importantly, cheaper to perform than other kinds of exams, such as Computer Tomography (CT), easier to sanitize after each usage, and it can be deployed directly on patient's bed if needed, limiting possible exposure in health care workers and other patients. All of these reasons make CXR a useful tool in emergency settings: even if less sensitive compared to CT, it can allow a first rough evaluation of the extent of lungs involvement. Furthermore, CXR can be repeated over time to monitor the evolution of lung disease.

In order to train machine learning models for detecting Covid-19, datasets needed to be built. However, this was not an easy challenge, and, especially at the beginning of the pandemic, datasets were scarce and built from different sources, including pre-existing datasets collected from publicly available sources. However, in doing so, the resulting dataset were heavily impacted by the diversity of the gathered data as they contained biases and confounding information from the different sites. For this reason, many of the published work scored good performance only apparently, and were in fact affected by the Collateral Learning problem.

In this Chapter, we describe the research work that we have been carrying out in the last three years, starting from early efforts to provide a methodological contribution for correctly assessing the models' performance, to the ongoing efforts that have resulted in Co.R.S.A[2], a funded project for assessing the impact and usefulness of AI-based Covid-19 detection tools in everyday clinical practice. As with Neuroimaging in Chapter 6, in this Chapter we will deal with the Collateral Learning

[1] https://covid19.who.int/
[2] https://corsa.di.unito.it/

issue. Differently from the previous Chapters, however, we will show that Collateral Learning can be mitigated not only with novel losses or regularization techniques but also with other methods such as transfer learning. This Chapter is meant to be a practical example of how Collateral Learning can especially affect the medical field.

Thanks to the collaboration with the radiology units of Città della Salute e della Scienza di Torino (CDSS) and San Luigi Hospital in Turin (SLG) in the last days of March 2020 (at the peak of epidemic in Italy), we managed to start the collection of the COvid Radiographic images DAta-set for AI (CORDA). The collection has now extended to other hospitals and institutions; this will be explained in Section 7.4.

Our contributions include two different approaches for COVID detection: a first deep learning pipeline targeting direct diagnosis from the CXR images (as typically done by most of the deep learning-based works) and a second method comprising an intermediate step, in which first radiological findings are highlighted and then diagnosis is formulated. We will show that the latter approach is the most effective; in particular, attempting to directly elaborate a diagnosis from CXRs is prone biases or site effects. Mimicking the radiologist decision process turns out to be more robust to such issues since it focuses on detecting objective radiological findings (as defined by Hansell et al. (2008)), which help in building a more robust representation space.

7.1 Collateral Learning in Chest X-Ray datasets

In this section, we describe the first attempt that we pursued for Covid-19 detection. The experiments presented in this section are based on the first iteration of the CORDA dataset, coming from a single institution (CDSS) and containing a total of 447 CXRs (297 Covid-19 positive images and 150 negatives). This dataset is referred to as CORDA-CDSS. Due to the imbalance in the dataset, we sampled data from publicly available CXRs datasets, in order to increase the number of COVID-negative samples in CORDA-CDSS. Table 7.1.1 shows the different combinations of datasets we benchmarked. For this purpose, we employed two popular datasets, *Kermany/Guangzhou* (Kermany, 2017) and *RSNA Pneumonia* (Stein, 2018). The Kermany dataset contains 5857 CXR images of normal cases (1583), bacterial pneumonia (2780) and viral pneumonia (1493). The RSNA dataset contains 20,672 normal CXR scans and 6012 pneumonia cases, for a total of 26,684 images. We also performed experiments on COVID datasets only, CORDA-CDSS and COVID-ChestXRay (Cohen et al., 2020)[3]. In this case the positive class has been undersampled in order to obtain a more balanced training set.

The first method that we studied consists of a binary classifier based on a deep convolutional neural network architecture. We used the popular ResNet-18 architecture as backbone for the classifier. We also experimented with pretraining, using both Kermany and RSNA datasets. Each resulting model has been tested on all of the different combinations of test sets, to provide more meaningful insights on the obtained results and determine which configuration is less likely to suffer from

[3]The dataset size is referred to the version as of April, 2020

potential biases and issues we previously discussed.

COMPOSED DATASET		ORIGINAL DATASETS A +	A -	C +	C -	B +	B -	D +	D -	TOTAL +	TOTAL -
A	train	126	105	-	-	-	-	-	-	126	105
	test	90	45	-	-	-	-	-	-	90	45
AB	train	207	105	-	-	-	102	-	-	207	207
	test	90	45	-	-	-	45	-	-	90	90
AC	train	207	105	-	102	-	-	-	-	207	207
	test	90	45	-	45	-	-	-	-	90	90
AD	train	116	105	-	-	-	-	49	24	165	129
	test	90	45	-	-	-	-	10	5	100	50
D	train	-	-	-	-	-	-	98	24	98	24
	test	-	-	-	-	-	-	10	5	10	5

Table 7.1.1: Datasets composition. The datasets used at training and test time are in the rows, and the total size is in the last two columns. For easier readability, each dataset has been assigned to a letter: CORDA-CDSS (A), Kermany (B), RSNA (C) and COVID-ChestXRay (D). The COVID-positive samples are indicated as "+" while the negative ones with "-".

7.1.1 Experiments

For all of the experiments we adopted a 70%-30% train-test split. We used SGD as optimization technique, with a starting learning rate of 0.01 and a weight decay of 10^{-5}. Part of the training set (20%) was then used as validation set, to tune hyper-parameters such as learning rate. We adopted the same learning rate decay policy (*on plateau*), across all of the experiments, with a patience of 15 epochs and a decay factor of 0.1: whenever the loss on the validation set reached a plateau lasting for at least 15 epochs, the learning rate was multiplied by 0.1. The training was stopped when the learning rate dropped to 10^{-5}. All of the experiments were run on NVIDIA Tesla T4 GPUs using PyTorch 1.4[4].

Results

First, we evaluate different choices of pre-training and network architectures, then we discuss the different options for augmenting COVID data. We also provide a comparison with the results obtained by similar works, specifically with COVID-Net (Wang and Wong, 2020). In the following subsections, CORDA will be used for brevity when referring to CORDA-CDSS. We also experimented with lung segmentation, which can help in removing bias sources, such as the presence of medical devices (typically correlated to sick patients), and various text that might be embedded in the scan (like annotations for the detected pathology). However, no significant difference was observed with respect to using full images, probably due to the dataset size and the strength of the site effect.

[4]Source code available at https://github.com/EIDOSlab/unveiling-covid19-from-cxr

Pre-training	Architecture	Sensitivity	Specificity	B. Accuracy	AUC
-	ResNet-18	0.56	0.58	0.57	0.59
Kermany	ResNet-18	0.54	0.58	0.56	0.67
RSNA	ResNet-18	0.54	0.80	**0.67**	**0.72**
Kermany	ResNet-50	0.64	0.56	0.60	0.65
RSNA	ResNet-50	0.61	0.71	0.66	0.67
Kermany	DenseNet-121	0.63	0.52	0.63	0.70
RSNA	DenseNet-121	0.77	0.38	0.57	0.63

Table 7.1.2: Comparison of different pretraining and architectures for finetuning on CORDA-CDSS. The best results are highlighted in bold, in terms of balanced accuracy and AUC score.

Comparisons of different pre-trainings and network architectures The results are summarized in Table 7.1.2. To evaluate the impact of pretraining compared to training from scratch, we focus on ResNet-18. It is clear that the choice of pre-training dataset is very important for the final accuracy. Even though the Kermany dataset also contains information about the type of pneumonia (bacterial or viral) and so, at first glance, it might seem a better fit for the pre-training, we observe a clear improvement when employing RSNA rather than Kermany. This is probably due to the larger size of the dataset. Moreover, Kermany contains CXRs coming from child patients. On the other hand, RSNA is closer to the CORDA dataset (where the average age is 61 years). In fact, the higher specificity obtained with RSNA seems to suggest that the pretraining is able to better capture discriminative features of control cases, compared to Kermany. From the results we achieved, we conclude that pretraining is a favorable choice.
Focusing on deeper architecture (ResNet-50 and DenseNet-121) we observe that the overall results are lower than with ResNet-18. This is probably due to overfitting, as these models may be too large for the dataset sizes. For this reason, we also tested an opposite approach, by training a smaller neural network made of 8 convolutional layers and a final fully-connected layer, which takes inspiration from the ALL-CNN-C architecture (Springenberg et al., 2014). However, with this smaller architecture, we did not achieve higher results than ResNet-18, obtaining a balanced accuracy of 0.61. Even with the best results achieved, however, we did not obtain satisfactory performance, considering that the goal is a binary classification task. In the next sections, we will discuss whether augmenting the CORDA-CDSS dataset with public datasets can lead to better results.

Augmenting Covid-19 data with public datasets: biases and site-effect As stated in the introduction of the Chapter, augmenting the training set by leveraging public datasets was a common practice, especially for gathering negative patients. Hence, instead of achieving class balance in CORDA-CDSS by undersampling the positive class, we did that by including negative samples from the Kermany and RSNA datasets. This allowed us to leverage the full potential of our CORDA-CDSS dataset. The explored dataset combinations are summarized in Table 7.1.1. To obtain insights on possible biases, for every dataset combination, we tested the models

Training dataset	Test Dataset	Sensitivity	Specificity	B. Accuracy	AUC
CORDA + RSNA	CORDA	0.68	0.44	**0.56**	**0.61**
	CORDA + Kermany	0.68	0.22	0.45	0.49
	CORDA + RSNA	0.68	0.90	0.79	0.90
CORDA + Kermany	CORDA	0.82	0.38	**0.60**	**0.63**
	CORDA + Kermany	0.82	0.95	0.89	0.97
	CORDA + RSNA	0.82	0.30	0.56	0.59

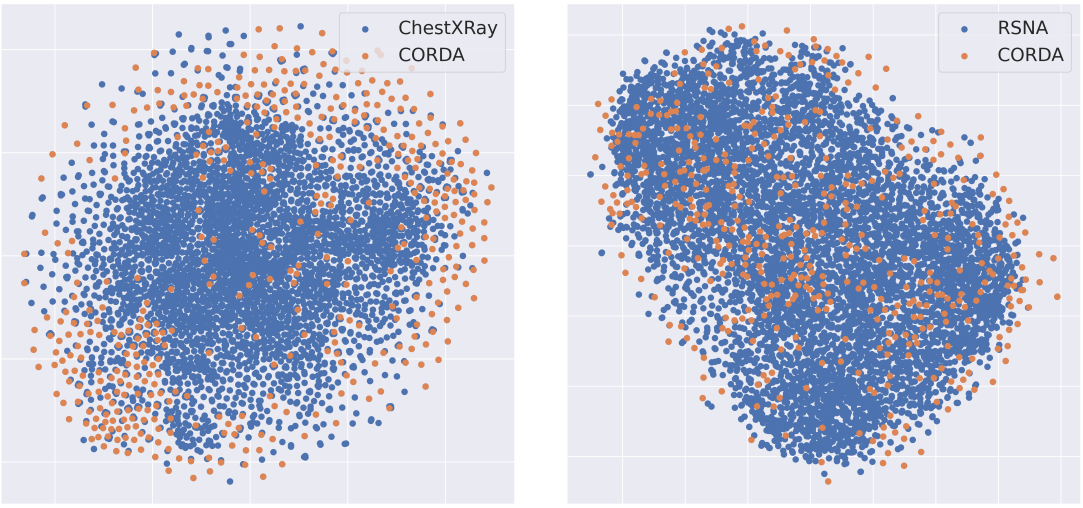
Table 7.1.3: Comparison of different training dataset combinations. The model is a ResNet-18 pretrained on RSNA. The highlighted rows represent the apparently good results, while the values in bold are the true Covid-19 detection accuracy. The complete results for all training and test combinations can be found in Appendix E.1.

on both the merged and on the CORDA-CDSS test set separately. Additionally we also tested on the opposite combination of datasets. Table 7.1.3 summarizes the best results that we achieved with combined training sets. The complete results for all datasets, pretraining and architectures can be found in Appendix E.1.

If we consider the results obtained on the same combinations of test and training sets (highlighted in the table), we apparently achieve very good results (0.79 and 0.89 balanced accuracy for using RSNA and Kermany respectively, with a highest AUC of 0.97). However there results are just the effect of Collateral Learning. In fact, when testing on only CORDA and the opposite combination, the specificity drops noticeably while the sensitivity remains exactly the same. What this essentially means is that the models is discriminating the data sources (i.e. RSNA vs others or Kermany vs others). In Figure 7.1.1, we visualize the extracted features from an encoder trained on CORDA+Kermany using t-SNE (blue and orange dots represent Kermany/RSNA and CORDA data samples respectively, regardless of the COVID label). We can notice how CORDA samples are clearly distinguished from Kermany data (7.1.1a), but not from RSNA (7.1.1b) even if they are all negative. This observation is confirmed by the fact that the apparently best performance is achieved on the CORDA+Kermany combination: as Kermany exhibits the largest difference with CORDA, it is easier to discriminate than RSNA.

A more reliable classification accuracy can be found by looking at the result on the CORDA test set (marked as bold in the table). However, we find that the accuracy is worse that Table 7.1.2, meaning that merging different datasets is actually harming the performance. This prime example of Collateral Learning is actually determined by a combination of factors:

- Strong selection biases in the datasets, with respect to either the illness and population sample (e.g. age), such as for Kermany;
- Almost a certain presence of site effect in the images, which makes the domain gap between datasets even larger;
- Strong correlation between dataset and target label.



(a) (b)

Figure 7.1.1: Visualization of features extracted from a model trained on CORDA+Kermany on different test set combinations: (a) Kermany vs CORDA (b) RSNA vs CORDA. Due to the large gap between the training datasets, the model has learned to classify based on the data source (Kermany vs other). A true Covid-19 classifier would be able to also correctly separate RSNA.

In the context of this thesis, after all the previously presented work on Collateral Learning, these results might seem trivial. However, this issue affected most of the published works during the peak of the pandemic (Apostolopoulos and Mpesiana, 2020; Narin et al., 2020; Sethy and Behera, 2020; Wang and Wong, 2020), and our contribution was one of the first highlighting it. This was also later confirmed by a number of other works (Garcia Santa Cruz et al., 2021; López-Cabrera et al., 2021; Roberts et al., 2021).

Comparison of networks trained on COVID-ChestXRay One very promising approach for Covid-19 detection, was represented by COVID-Net (Wang and Wong, 2020). The authors report very high performance on the COVID-ChestXRay dataset, achieving 0.90 accuracy. They also share the source code and the trained model[5], which allowed us to validate the generalization performance of their method, and compare it to our work.
Table 7.1.4 shows the classification metrics obtained with COVID-Net and our ResNet-18 model: both models were trained on the COVID-ChestXRay dataset, and tested on both CORDA (A) and COVID-ChestXRay (D). We also provide a comparison with DenseNet-121 as its architecture is very similar to COVID-Net.
In line with the discussion above we can notice that both COVID-Net and ResNet-18 yield surprising results when the same dataset is used for training and testing. The performance of COVID-Net on the COVID-ChestXRay (D) test set is very but it drops significantly when tested on CORDA. This drop can be explained by looking at sensitivity and specificity values: it is clear that the model classifies as COVID- almost all of the unseen data coming from CORDA. We observed similar behaviors with ResNet-18 and DenseNet-121: the results we obtained on COVID-ChestXRay

[5] https://github.com/lindawangg/COVID-Net

Architecture	Test dataset	Sensitivity	Specificity	BA	AUC
COVID-Net	A	0.12	0.98	0.55	0.55
COVID-Net	D	0.90	0.80	0.85	0.85
ResNet-18	A	0.91	0.20	0.56	0.61
ResNet-18	D	1.00	1.00	1.00	1.00
DenseNet-121	A	0.99	0.07	0.53	0.61
DenseNet-121	D	1.00	0.80	0.90	0.98

Table 7.1.4: Comparison of COVID-Net, Resnet-18 and Densenet-121 trained on COVID-ChestXRay. Dataset naming follows Table. 7.1.1

are comparable to COVID-Net, and, in fact, similar numbers are also claimed in other works on ResNet-like architectures (Apostolopoulos and Mpesiana, 2020; Narin et al., 2020; Sethy and Behera, 2020). However, testing on CORDA revealed that the models have likely learned some hidden biases in COVID-ChestXRay and hence misclassified COVID- samples as COVID+ (given that the specificity is here 0.20).

7.2 Transfer Learning avoids Collateral Learning

In this section, we describe the second method that we developed for Covid-19 classification, aiming at mitigating some of the issues related to Collateral Learning highlighted in the previous section. Differently from Chapters 3, 4, 5 and 6, here we do not propose a regularization technique or a loss function, but rather, we consider how transfer learning can, in some instances, help in fighting Collateral Learning.

The method that we propose consists of two steps: first, a deep model is trained to detect different types of objective radiological findings (including, most importantly, non-specific radiological findings), then a classifier is trained to predict the target disease (COVID-19 in our case) from the extracted features.
Detecting and classifying these kinds of objective findings, without taking into consideration the clinical diagnosis, can help in reducing biases given by hidden stratification in medical data, and provide an optimal initialization for transfer-learning tasks which can then shift the focus on predicting specific diseases, such as COVID-19, on smaller datasets. In fact, hidden stratification is a very important issue as recently highlighted by Oakden-Rayner et al. (2020). Another important factor to keep in mind is that the same non-specific radiological findings can be the results of different diseases, including previously new ones like COVID-19, and thus this method might also adapt well to different or future diseases. Finally, this approach has the advantage of mimicking the radiologists' workflow, in which the detected lung anomalies are employed when making a final diagnosis.

7.2.1 Detection of objective radiological findings

For this task, we leveraged a large scale dataset, *CheXpert*, which contains annotation for different kinds of common radiological findings that can be observed in CXR images (like opacity, pleural effusion, cardiomegaly, etc.). This large dataset is

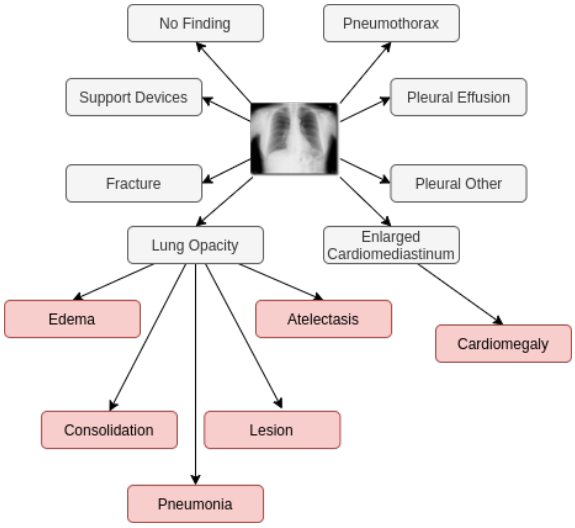


Figure 7.2.1: Hierarchy of *CheXpert* labels: levels are identified by color coding.

Method	Atelectasis	Cardiomegaly	Consolidation	Edema	Pleural Eff.
Baseline (Irvin et al., 2019)	0.79	**0.81**	0.90	0.91	0.92
Ours	**0.83**	0.79	**0.93**	**0.93**	**0.93**

Table 7.2.1: Performance (AUC) for a DenseNet-121 trained on CheXpert.

well suited for multi-label classification tasks; in fact more than one finding can be commonly observed in ill patients' lungs at the same time. CheXpert provides 14 different types of observations for each image in the dataset. For each class, the labels were generated from radiology reports associated with the studies with NLP techniques, conforming to the Fleischner Society's recommended glossary Hansell et al. (2008), and marked as: *negative*, *positive*, *uncertain* or *blank* (when not mentioned in the report). Following the relationship among labels illustrated in Figure 7.2.1, as proposed by Irvin et al. (2019), we can identify 8 top-level pathologies and 6 child ones.

We test two neural network architectures as backbone for our models, ResNet-18 and DenseNet-121. The complete experimental detail and results on CheXpert can be found in Appendix E.2. Given the scale of the dataset, we obtain the best results with DenseNet-121, which are summarized in Table 7.2.1 in terms of AUC for the CheXpert selected test classes[6].

7.2.2 COVID diagnosis

The second step of the proposed approach training the final Covid-19 classifier. We perform a transfer learning step, freezing the model obtained in the previous section and using it to train a new binary classifier on the CORDA dataset.
We tested two different types of classifiers, namely: *i)* a decision tree trained on the output probability for each radiological finding *ii)* a fully connected classifier trained on the extracted features. As it will be later discussed these two possible

[6]Defined in https://stanfordmlgroup.github.io/competitions/chexpert/

choices represent a trade-off between easiness of interpretability and discriminative power.

7.2.3 Experiments

We report the final results on Covid-19 detection on the CORDA-CDSS dataset in Table 7.2.2, comparing with the simple direct method of Section 7.1. The best score using the direct approach is obtained by pre-training the encoder on the *RSNA* dataset.

Method	Backbone	Pretrain dataset	Sensitivity	Specificity	BA	AUC
Direct	ResNet-18	-	0.56	0.58	0.57	0.59
	ResNet-18	RSNA	0.54	0.80	0.67	0.72
	ResNet-18	Kermany	0.54	0.58	0.56	0.67
Two-step	ResNet-18	CheXpert	0.69	0.73	0.71	0.76
	DenseNet-121	CheXpert	0.72	**0.78**	**0.75**	**0.81**
	DenseNet-121†	CheXpert	**0.77**	0.60	0.68	0.70

Table 7.2.2: Comparison of direct diagnosis method and with two-step on the CORDA-CDSS dataset. We denote the classification tree with †.

Transfer learning can help mitigate Collateral Learning We can clearly see how the two-step method outperforms the direct diagnosis: using the same network architecture (ResNet-18 as backbone and a fully-connected classifier), we obtain a significant increase in all of the assessed metrics. Even better results are achieved by using a DenseNet-121 as backbone. For the decision tree (denoted by the † symbol in the Table), in our experiments, we found that a maximum depth of 4 gave the best results in terms of model complexity and generalization ability. From Table 7.2.2 we can see that the performance of the tree model is not far from radiologists expectation (Wong et al., 2020), and compared to the fully-connected classifier provides better explainability. More details about this can be found in Appendix E.3.
The fully-connected classifier was instead trained on the raw features extracted by the encoder: this choice was justified by the fact that training this classifier on the output probabilities yields results similar to the decision tree, with the added downside of loosing a lot of its interpretability. Using the raw extracted features, on the other hand, also provided a boost in generalizing Covid-19 classification.

Link with Concept Learning The particular kind of transfer learning approach that we employed is resemblant to Concept Learning, in particular to Concept Bottleneck Models (Koh et al., 2020). In Concept Bottleneck Models (CBM), the training happens in two separate phases: first the model is trained to map the input image to a set of predefined concept, such as radiological findings, as in our case, then these concepts are used to predict the final label. Compared to traditional end-to-end approaches where models are trained to directly predict the label from the image, CBMs can achieve a higher degree of interpretability while maintaining

Method	Backbone	Dataset	BA
Two-Step	DenseNet-121	CORDA-CDSS	0.75
Two-Step	DenseNet-121	CORDA-SLG	0.81
Two-Step	DenseNet-121	All	0.69

Table 7.2.3: The accuracy drops when combining the datasets, probably due to site-effect

competitive performance. Our results show that a particular choice of training concepts can help mitigate the Collateral Learning problem, by providing a more robust representation space.

More data might mean more site effect During the time of these experiments, we were able to collect more data within the CORDA dataset, coming from another institution (SLG). This new dataset, CORDA-SLG, contains 451 CXR images, with 129 Covid-19 positive cases and 322 negative ones. Hence, we performed additional benchmarks of the best-performing architecture (DenseNet121 with a fully connected classifier). First, we applied the whole pipeline using only CORDA-SLG in the second step, then we merged both datasets. The results are presented in Table 7.2.3. We observe that, while we achieve quite high balanced accuracy of 0.81 on CORDA-SLG alone, the result noticeably decreases when using both datasets. Despite the robust pretraining employed, it is possible that the introduction of additional data still results in issues such as site effect, thus lowering the final performance. This once again confirms that adding more data does not always improve the model accuracy, as also explained in Varoquaux and Cheplygina (2021).

7.3 Limiting site-effect with regularization

In order to exploit the full dataset at our disposal, and still achieve competitive results, we employed our proposed debiasing method EnD (explained in Chapter 3) using the acquisition site as bias label.

Results are shown in Table 7.3.1. Looking at the results obtained without regularization, we can notice that the predictions are skewed towards the negative class, given the lower sensitivity and higher specificity. This can be the effect of various differences between the collecting institution such as acquisition techniques, the composition of the dataset (i.e. building the COVID- class by collecting older CXRs of previous patients) and other unknown reasons which can lead to hidden stratification and biases in the data. When applied, EnD shows in fact a sensible improvement in the achieved performance. The balanced accuracy (BA) increased across all test sets, notably merged (All) set from 0.69 to 0.76. Also the sensitivity and specificity show more balanced values on the two CORDA subsets.

At the time of writing of this manuscript, the CORDA dataset has grown to include more institutions, as will be explained in Section 7.4. To exploit the full dataset, we leveraged our most recent debiasing technique FairKL (explained in Chapter 4). The

	CORDA-CDSS			CORDA-SLG			ALL		
	Sens	**Spec**	**BA**	**Sens**	**Spec**	**BA**	**Sens**	**Spec**	**BA**
Baseline	0.44	0.89	0.67	0.38	0.98	0.68	0.43	0.95	0.69
EnD	0.79	0.62	**0.71**	0.62	0.87	**0.74**	0.74	0.79	**0.76**

Table 7.3.1: CORDA results with EnD applied on the DenseNet-121 classifier.

results are reported in Table 7.3.2. By employing FairKL we achieve a consistent improvement with respect to the baseline. The research on the topic is still ongoing, and we expect to further improve the results.

	Inst. 1 (CDSS)	Inst. 2 (SLG)	Inst. 3	Inst. 4	Avg.
Baseline	0.68	0.83	0.74	0.87	0.78
FairKL	**0.70**	**0.85**	**0.77**	**0.88**	**0.80**

Table 7.3.2: Results of site effect debiasing with FairKL on the up-to-date CORDA dataset, in terms of balanced accuracy.

7.4 The CORDA data collection

As explained in the previous section, being able to leverage a greater amount of data turned out to be crucial, as other institutions have joined the CORDA data collection, which now comprises four different Italian hospitals:

1. A.O.U Città della Salute e delle Scienza (Molinette), Torino (previously labelled as CDSS);
2. A.O.U San Luigi Gonzata, Orbassano, Torino (previously labelled as SLG);
3. A.O. Mauriziano, Torino
4. Centro Cardiologico Monzino, Milano

CORDA contains a total of 3852 images of different modalities, with 1604 CXR and 2242 CT images. The dataset was made publicly available in January 2023[7]. The aim of this dataset is to provide a multi-center collection of radiographic images for Covid-19 detection, in order to build more robust machine learning algorithms and models. The curation of the dataset is part of the ongoing project Co.R.S.A[8], which received funding from the Piedmont Region, and aims at deploying and validating computer-aided Covid-19 diagnostic tools in a clinical setting. Although the emergency setting of the Covid-19 pandemic peak is fortunately over, the groundwork built by this project should serve as a solid foundation for quick-starting future responses to epidemics, should the need arise.

[7] https://zenodo.org/record/7821611
[8] https://corsa.di.unito.it/

Part IV

Other Instances of Collateral Learning

Chapter 8

A Few Hints About Collateral Learning and Privacy

In this Chapter, we briefly analyze another threat posed by Collateral Learning, which is related to learning potentially sensitive information of the data. As explained in the Introduction (Section 1.2), neural networks can learn more features than intended. For example, a model trained for age prediction on facial images might additionally learn gender features. This could also happen in medical images. As we will show in this chapter, it might be trivial to retrieve this information from the model's output or latent space. This can of course represent a real-world issue when deploying DL-based systems to production, as they might cause leakage of private or sensitive information. In this Chapter, we analyze whether the techniques that we proposed for debiasing can also help prevent this from happening. The rationale behind this approach is that we might be able to treat private information in the same way that we treat biases.

Privacy-preserving approaches ideally aim at hiding some information, making it un-recoverable (or difficult to recover) from a potential attacker (Gentry, 2009; Lindell, 2005; Sweeney, 2002). The concept of privacy-aware learning is not novel in machine learning and deep learning (Abadi et al., 2016; Iyengar et al., 2019; Kanagavelu et al., 2020; Phan et al., 2016; Shokri and Shmatikov, 2015; Xie et al., 2018; Yu et al., 2019). One of the very first works in such an area can be found in Warner (1965). Specifically, this work suggested privacy-preserving methods for survey sampling. Following this path, in the 70s many works were proposed in different areas, like census taking and analysis of tabular data by Fellegi (1972).

Overall, we can say that while privacy-preserving approaches erase or hide some information to prevent an attacker from recovering it, debiasing approaches do not necessarily do so. For example, just re-weighting the bias features could be enough for debiasing purposes. The question we aim to answer in this Chapter is whether there are there debiasing techniques that can be used to completely remove private information. To assess this, we have selected four of the most common debiasing techniques, which we employed in Chapter 3: LearnedMixin (Clark et al., 2019), RUBi (Cadene et al., 2019), ReBias (Bahng et al., 2020), and our proposed method EnD. Given the similarities between EnD and FairKL (Section 4.2), in this Chapter we only report results for the former method; the same conclusions can be reached

for both techniques.

8.1 Background

Recently, thanks to the increase in computational capabilities, many works have been proposed on privacy-preserving in computational frameworks. We can divide these into the following categories.

Data anonymization. These approaches address the problem of collecting data from many different sources making it impossible to back-trace their source. To these approaches, the most common approach, especially in the medical domain, we find vanilla data anonymization: this simple approach consists in simply hiding the sensitive metadata information containing the information to be kept private, according to the most recent GDPR[1]. Such data cleaning procedure is standard for releasing medical imaging, where the original DICOM file format, by standard, contains sensitive information for the patients, like name, birth date and gender of the patient. However, this is certainly not sufficient to prevent back-tracing information: Narayanan and Shmatikov (2008), for example, were able to recover sensitive anonymized information from the Netflix prize dataset.

K-anonymization. More advanced and safe data aggregation approaches consist, for example, in guaranteeing the so-called k-anonimity. Sweeney proposed a framework for which anonymity of data is guaranteed when compared to $k-1$ others, and it is mainly thought to fight re-identification, guaranteeing the redundancy of similar features (Sweeney, 2002). An important limitation of this technique, however, is determined in its low performance on high-dimensional data, which is a common setup in DL scenarios.

Homomorphic encryption. This is a special category of encryption that allows users to perform computation directly on encrypted data, without the need of decrypting it (Gentry, 2009). Despite this approach, by definition, being able to discourage the mining of private information from the attackers, its computational complexity is also very high, limiting its deployment in real-life scenarios (Riazi et al., 2019).

Multi-party computation. A current challenge for deep learning, especially in the medical field, lies in the impossibility of publicly sharing data. Due to physical, ethical, and legal constraints, it might happen that data is not allowed to be shared outside the infrastructure where it was acquired (Lindell, 2005). Towards this end, federated learning-based approaches are uprising: they consist of having the dataset distributed across many infrastructures. Each of these peers trains independently a DL model and occasionally exchanges information about the trained model, which might involve quantities like the model's parameters or the gradients (Kanagavelu et al., 2020). This approach, if achieved in a round-robin fashion, averages naturally the information related to the private features. A major drawback of this approach,

[1] https://eur-lex.europa.eu/eli/reg/2016/679/oj

however, lies in the need for intensive communication between the infrastructures, which significantly slows the training process (Makri et al., 2019).

Differential privacy. Differential privacy is a very general approach to withholding private information from individuals in a set of data. In general, we can say that if data belonging to different individuals (or in our context, belonging to different private classes) are sufficiently close to be each other indistinguishable, the private information can not be retrieved. Behind this very simple yet effective idea, a number of approaches have been proposed and can be categorized in four groups.

- *Perturbing the input data.* Introducing a proper perturbation to the data themselves can hide the private target information. To this class belong centralized approaches (Dwork and Lei, 2009) and recently, decentralized alternatives have been proposed as well (Erlingsson et al., 2014; Kairouz et al., 2014).

- *Perturbing the output of the trained model.* This approach consists of applying a sufficiently large noise to the output of the model such that the samples belonging to different private classes are each other indistinguishable (Iyengar et al., 2019). However, efficiently computing the noise to be applied in a high-dimensional scenario is not straightforward due to the non-convexity of the objective function: to this end, convex proxies have been recently proposed to overcome this obstacle (Phan et al., 2016, 2017).

- *Perturbing the gradient update.* Applying a specific noise to the update signal for the model is possible to enhance differential privacy in the model. Towards this approach, many proposals, ranging from the deployment of a distributed framework (Shokri and Shmatikov, 2015) to the design of momentum-based optimizers accounting for the private class membership (Abadi et al., 2016) have been proposed. The main drawback of these strategies lies in the low convergence and high computation complexity required.

- *Perturbing the target labels.* Finally, deploying noise to the target labels for the learning task can also be deployed to hide the private information, despite such an approach is mainly meant to boost gradient perturbation approaches (Xie et al., 2018; Yu et al., 2019).

8.2 Testing framework

In order to assess the presence/absence of private information on the trained DL models, we design a model inversion-like and membership inference strategy. In such a frame, the attacker attempts to infer some attributes or private class membership from the output of a DL model or to reconstruct the input (Wu et al., 2016). We indicate with $\mathcal{P}(x_i)$ the private class label associated with x_i. This can represent any private attribute (e.g. identity, gender, etc.). Our general framework consists of two main steps.

1. *Train the model.* In this step, we train the DL model (Figure 8.2.1a). In this phase, standard learning strategy is used, and eventually a debiasing strategy

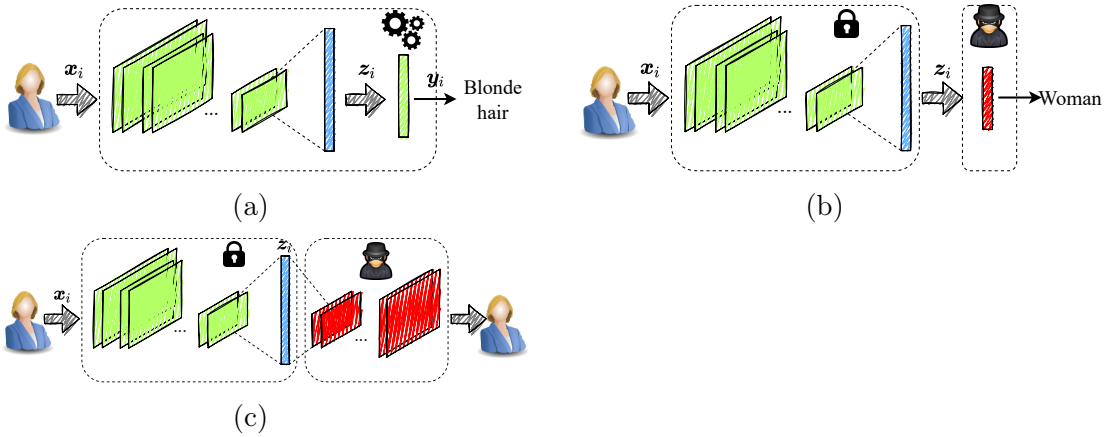


Figure 8.2.1: Standard training on some target features, like hair color recognition (a), gender membership recovery (b), and input reconstruction (c). In the image, in green are the layers deployed at training time (where parallelograms are convolutional layers while the rectangular box is a fully-connected layer), in red the layer trained by the attacker to obtain the private information from the bottleneck layer and in blue a plain reshaping layer.

can be deployed besides training, attempting to hide the information related to $\mathcal{P}(x_i)$. In this work, we name the accuracy measured on the target classes *Target Accuracy*.

2. *Attack.* After train is completed, an attacker attempts to recover the information of $\mathcal{P}(x_i)$ from the extracted representation of x_i. For this purpose, we train a classifier to retrieve $\mathcal{P}(x_i)$ (Figure 8.2.1b). We indicate the accuracy measured on the private classes with *Private Class Accuracy*. Besides the private class membership, we can also attempt to recover the original input x_i itself, using a decoder network, with a similar result as in Fredrikson et al. (2015) (Figure 8.2.1c).

We perform our experiments on four datasets: Biased-MNIST, CelebA, IMDB Face dataset, SIIM-FISABIO-RSNA[2].

The experimental setup is similar to Section 3.4. For SIIM-FISABIO-RSNA we split the dataset in a training set comprising the 85% of the scans, a validation set of 5% scans and a test set of the remaining 10%. We train a DenseNet-121 model (Huang et al., 2017) to classify between two classes: "Negative for Pneumonia" and "Typical Appearance". The training has been performed using SGD, with an initial learning rate of 0.1, decayed by a factor 10 after no improvement over the validation set loss has been detected for 5 consecutive epochs. The training stops when the learning rate drops below 10^{-3}. We use batch size 16 with momentum of 0.9 and weight decay of 10^{-4}.

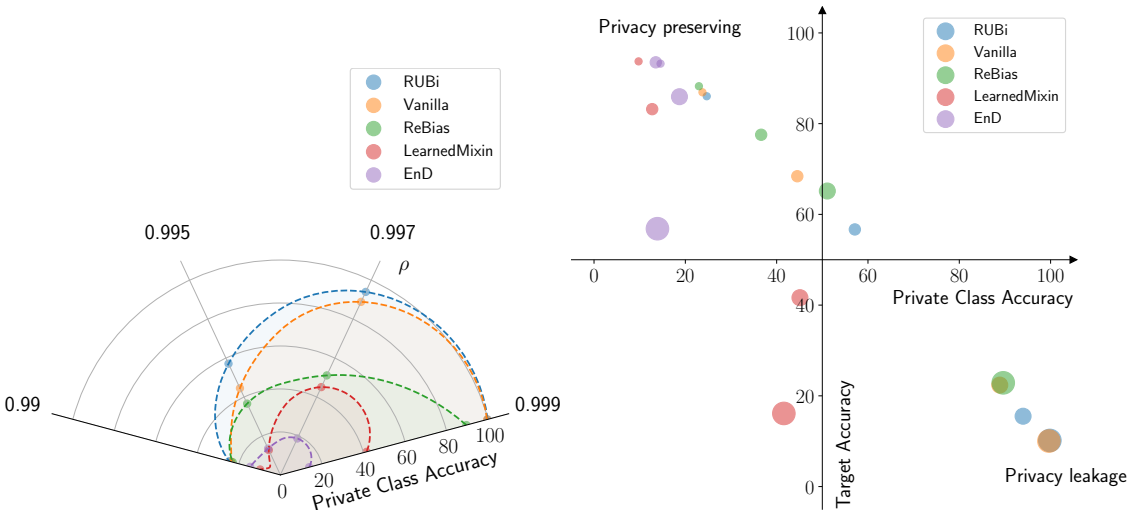


Figure 8.3.1: (a) Biased-MNIST private class accuracy. The closer a curve is to the origin of the polar plot, the better the corresponding technique is at preventing private information leakage. (b) Private class accuracy vs target accuracy. Larger markers indicate higher values of ρ.

8.3 Color leakage in Biased-MNIST

As a base benchmark, we employ the synthetic dataset Biased-MNIST. Looking at the results presented in Table 3.4.1 we expect an attack to be trivial on a vanilla model, and we also hypothesize that it could be prevented by some of the debiasing techniques. Figure 8.3.1a shows the private class accuracy obtained by the linear classifier at the different values of ρ. The vanilla model shows in fact a significant leakage of color-related information, as the attack reaches almost 100% accuracy in the higher range of ρ. Surprisingly, not even RUBi manages to prevent an attack, obtaining performances even worse than vanilla. Considering all of the difficulty settings, the techniques that better prevent privacy leakages are LearnedMixin and EnD. In order to rank the different techniques, in Figure 8.3.1b we compare the private class accuracy and the target accuracy. Debiasing algorithms that are able to avoid leakages while retaining (or improving) the target accuracy are found in the top left portion of the plot. From this analysis, we find EnD to be the best-performing technique, followed by LearnedMixin and Rebias.

We now concentrate on the best technique (EnD), and we further assess the absence of a privacy leakage by conducting a model inversion attack, as pictured in Figure 8.2.1c. Figure 8.3.2 shows the reconstructed images. When using a vanilla encoder, the color is fully preserved (and the digit is transformed into the corresponding training class). On the other hand, with an EnD-regularized encoder, the digit information is preserved while the color information is almost completely removed, as it seems to be randomly guessed by the decoder.

[2] https://www.kaggle.com/competitions/siim-covid19-detection/data

(a) (b) (c)

Figure 8.3.2: attack on Biased-MNIST: (a) ground truth images (b) decoder trained from a privacy leaking encoder (c) decoder trained with an EnD-regularized encoder. From (c) it can be clearly seen that the decoder has to guess a random color.

Task	Method	Target	Private Class
Hair Color	Vanilla	70.25	59.20
	EnD	**91.21**	**50.00**
Makeup	Vanilla	62.00	80.56
	EnD	**75.93**	**63.89**

Table 8.4.1: CelebA target accuracy (higher is better) and private class accuracy (lower is better).

8.4 Gender leakage from face images

Next, we focus on gender information leakage on real facial images on two different tasks: face attribute classification and age prediction. For the first task, we employ CelebA. For age prediction, instead, we use the IMDB Face dataset.

Results for the CelebA dataset are presented in Table 8.4.1. As for the Biased-MNIST experiments, we observe an increase in the target accuracy when employing EnD for both of the classification tasks. Compared to the baseline, we also observed a significant decrease in the accuracy of the attack. Considering that the provided gender attribute is binary, an accuracy of 50% represents a random guess by the attacker, meaning that there is no private information leakage. The same considerations apply to the age regression task on the IMDB dataset. Table 8.4.2 shows the results. Here, we obtain an accuracy of around 50% on both training sets.

We further investigate the effect of the debiasing technique on the model, by analyzing the distribution of the latent space of a vanilla model compared to a regularized model. We fit a gaussian distribution on the principal component of the embeddings computed on the IMDB dataset. Figure 1.2.2 shows the distributions. We observe that, while in the vanilla model, the two distributions $\mathcal{N}_m\,(-0.42, 0.27)$, $\mathcal{N}_f\,(0.42, 0.47)$ are clearly separate, they are almost overlapping in the regularized model ($\mathcal{N}_m\,(-0.09, 0.93)$, $\mathcal{N}_f\,(-0.09, 0.91)$).

Split	Method	Target	Private Class
EB1	Vanilla	77.17	82.36
	EnD	**80.15**	**49.95**
EB2	Vanilla	61.97	63.74
	EnD	**78.80**	**50.05**

Table 8.4.2: IMDB target accuracy (higher is better) and private class accuracy (lower is better). On age detection, gender is guessed correctly 50% of the time, which is equal to random guessing.

Method	Target	Private Class
Vanilla	78.12	87.1
EnD (low)	**78.21**	63.4
EnD (high)	78.02	**55.3**

Table 8.5.1: SIIM-FISABIO-RSNA target accuracy (higher is better) and private class accuracy (lower is better).

8.5 Gender leakage in medical data

We also test the capability of removing sensitive information on a medical dataset. SIIM-FISABIO-RSNA is a dataset comprising more than 6k chest X-ray (CXR) scans in DICOM format, anonymized according to the current GDPR guidelines. For study purposes, however, the metadata associated with these scans comprises information about the gender, which will be used as private class. The scans are converted using the meta-information contained in the DICOM files, and rescaled to 448×448 resolution.

Results are provided in Table 8.5.1. In this case, for EnD, we provide two different results: one is achieved with a small weight for the regularization (specifically, it weights over the 1% on the total objective function minimized - low) while another has a higher weight (10% - high). Also in this case we observe that from a vanilla approach, we are able to recover the information about the gender with a good accuracy (above 87%) while the effect of EnD drops as the weight of the regularization term increases. Differently from the previous scenarios, the performance, in this case, is not significantly affected: this is explained by the natural disentanglement between gender and the given medical task (presence of pneumonia and typical COVID presence). However, the gender information is still naturally forwarded to the bottleneck layer, which is postulated as plausible by some works in the literature (Arpit et al., 2017; Shmatikov and Song).

8.6 Conclusions

In this Chapter, we have shed some light on the possibility of bridging debiasing and privacy-preserving approaches for deep learning. To address our investigation, we have considered the special case in which debiasing algorithms consider private

information as the bias for the learning problem. We have conducted some empirical evaluations from which we evidenced that, under our constraint, there *exists* a non-empty class of debiasing algorithms that can be deployed for both purposes. In particular, if the given debiasing algorithm is also able to hide private information rather than simply re-weighting it, then it can be successfully deployed for privacy preservation. The investigation on whether the sufficient condition also holds is left as future work.

Part V

Closing Remarks

Chapter 9

Additional Works

In this Chapter, we present some additional works that are connected to the topic of learning robust representations. We did not include them in the main part of this thesis, as they do not directly deal with Collateral Learning; however, the methodologies developed could be also useful in that regard.

9.1 Leveraging prior knowledge for better representations

In some parts of this thesis, we have already dealt with the idea of leveraging prior knowledge for training a model. For example, in Chapter 5, we used the information from a bias-capturing model in order to guide the debiasing process. Also, in Chapter 6 we were able to integrate age information into the loss function with the use of a kernel. While, in this context, age itself is the target we are interested in predicting, this approach can also be used in weakly-supervised contexts, where we only have access to some metadata about our samples (Dufumier et al., 2021b).

In these instances, being able to include additional or prior information enabled us to achieve better results. In this Chapter, we present some other approaches that we propose for achieving better representations by leveraging available prior knowledge.

9.1.1 Integrating prior knowledge in CL

As mentioned in Section 2.1.3, one of the key elements of self-supervised contrastive learning is represented by the data augmentation scheme. Data augmentation determines how positive samples are defined and, ultimately, the quality of the learned representation. The most-used augmentations for visual representations involve aggressive crop and color distortion. Cropping induces representations with high occlusion invariance (Purushwalkam and Gupta, 2020) whereas color distortion may avoid the encoder f from taking a shortcut (Chen et al., 2020) while aligning positive samples and therefore fall into the simplicity bias (Shah et al., 2020) (e.g. color distortion would efficiently mitigate the bias in Biased-MNIST).

Nevertheless, learning a representation that mainly relies on augmentations comes at a cost: both crop and color distortion induce strong biases in the final representation (Purushwalkam and Gupta, 2020). Specifically, dominant objects inside

images can prevent the model from learning features of smaller objects (Chen et al., 2021) (which is not apparent in object-centric datasets such as ImageNet) and few, irrelevant and easy-to-learn features, that are shared among views, are sufficient to collapse the representation (Chen et al., 2021) (a.k.a feature suppression). Finding the right augmentations in other visual domains, such as medical imaging, remains an open challenge Dufumier et al. (2021b) since we need to find transformations that preserve semantic anatomical structures (e.g. discriminative between pathological and healthy) while removing unwanted noise. If the augmentations are too weak or inadequate to remove irrelevant signal w.r.t. a discrimination task, then how can we define positive and negative samples?

In Dufumier et al. (2023), we propose to integrate *prior information*, learnt from generative models (viewed as features extractor or prior representation) or given as auxiliary weak attributes (e.g. phenotypes of participants for medical images), into contrastive learning, to make it less dependent on data augmentation. Using the theoretical understanding of CL through the augmentation graph, we make the connection with kernel theory and introduce a novel loss with theoretical guarantees on downstream performance. This loss additionally solves the Negative-Positive Coupling (NPC) problem that affects InfoNCE-based frameworks (Yeh et al., 2021). Prior information is integrated into the proposed decoupled contrastive loss using a kernel. In the unsupervised setting, we leverage pre-trained generative models, such as GAN (Goodfellow et al., 2014) and VAE (Kingma and Welling, 2013), to learn *a prior representation* of the data. We provide a solution to the feature suppression issue in CL (Chen et al., 2021) and also demonstrate SOTA results with weaker augmentations on visual benchmarks (both on natural and medical images). In the weakly supervised setting, we use instead auxiliary image attributes as prior knowledge (e.g. birds color or size) and we show better performance than previous conditional formulations based on these attributes (Tsai et al., 2022).

In summary, we make the following contributions:

1. We propose a new decoupled contrastive loss that allows the integration of prior information, given as auxiliary attributes or learned from generative models, into the positive and negative sampling.

2. We derive general guarantees, relying on weaker assumptions than existing theories, on the downstream classification task, especially in the finite-samples case.

3. We empirically show that our framework performs competitively with small batch sizes and benefits from the latest advances in generative models to learn a better representation than existing CL methods.

4. We show that we achieve SOTA results in the unsupervised and weakly supervised setting.

In this work, we show that we can integrate prior information into CL to improve the final representation. Empirically, we show that generative models provide a good prior when augmentations are too weak or insufficient to remove easy-to-learn noisy features. We also show applications in medical imaging in both unsupervised and

weakly supervised settings where our method outperforms all other models. Thanks to our theoretical framework, we hope that CL will benefit from the future progress in generative modeling and it will widen its field of application to challenging tasks, such as computer-aided-diagnosis.

9.1.2 Synthetic data augmentation in histopathology

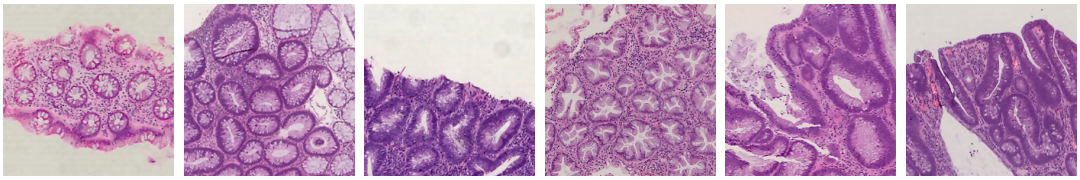
In this Section, we present our work in the field of histopathology for the detection and characterization of colorectal polyps. First, we introduce our contributions in the area, then we present how we aim to include prior medical knowledge into generative models for the creation of synthetic data. The goal of this approach is to increase the accuracy in detecting high-risk adenomas.

Histopathological characterization of colorectal polyps allows the tailoring of patients' management and follow-up, with the ultimate aim of avoiding or promptly detecting an invasive carcinoma. Colorectal polyps characterization relies on the histological analysis of tissue samples to determine the polyp's malignancy and dysplasia grade.

Background

The demand for gastrointestinal histopathology is on the rise (Gonzalez, 2020), fostered by the wide-spreading of cancer screening programs. Gastrointestinal histopathologists inspect tissue samples, collected during colonoscopies, looking for hints that can predict the insurgence of invasive carcinoma (Bevan and Rutter, 2018). Colorectal polyps are pre-malignant lesions found in the intestinal mucosa that pathologists analyze to *i)* ascertain the polyp type (hyperplastic, adenoma) and *ii)* assess the dysplasia grade in the case of adenomas. Examination of colorectal polyps represents a large share of histopathologists' workload, thus methods for automating these tasks are highly sought. Despite such clinical relevance, the concordance rate even among expert pathologists, in the diagnostic assessment of colorectal polyps, is far from optimal (Denis et al., 2009; Mollasharifi et al., 2020). Although the distinction between non-adenomatous and adenomatous tissue is usually reliable, the inter-observer agreement between different histological types and dysplasia grades is sub-optimal. For instance, the concordance in assessing a tubulo-villous polyp or low-grade dysplasia ranged around 70% (Denis et al., 2009).

Deep learning-based methods have shown promising results in assisting the pathologists' work (Janowczyk A, 2016). Korbar et al. (2017) present a patch-based framework using ResNet, to classify different types of colorectal polyps from whole-slide images. Their work provides empirical suggestions that residual architectures are better suited to this task. Wei et al. (2020) propose an analysis model for annotated tissue samples and perform a study on the generalization of neural models with external medical institutions. Their work describes a hierarchical evaluation mechanism to extend the classification of tissue fragments to the entire slide. Song et al. (2020) propose a patch-based fully convolutional approach for the classification and grading of adenomas, with a strong focus on model interpretability. They also highlight how different patch sizes should be used for adenoma classification and grading.



(a) NORM (b) TA.LG (c) TA.HG (d) HP (e) TVA.LG (f) TVA.HG

Figure 9.1.1: Example of 800×800µm patches for the six UniToPatho colorectal polyps classes.

However, the scarcity of datasets large enough and suitably labeled represents a major hurdle for training deep-learning-based algorithms to predict polyp type and adenoma dysplasia grade.

Our contributions

During this PhD, we made a number of contributions (Barbano et al., 2021) towards automatic colorectal polyps characterization, in the context of the DeepHealth (2019) project:

- First, we make available UniToPatho[1], a high-resolution annotated dataset of Hematoxylin and Eosin (H&E)-stained colorectal images. UniToPatho enables training deep neural networks to classify different colorectal polyps types and adenomas grading. We make available our annotated dataset as a collection of high-resolution patches extracted at different scales.
- Second, we show that the direct application of a deep neural network fails to classify both the tissue type and adenoma dysplasia grade.
- Lastly, we propose a multi-resolution deep learning approach solving the previous issues, that achieves promising accuracy in the characterization of colorectal polyps and in the dysplasia grading.

UniToPatho comprises different histological samples of colorectal polyps, collected from patients undergoing cancer screening. The dataset is a collection of the most relevant patch images extracted from 292 Whole-Slide Images (WSI), in accordance with UniTo pathologists' evaluation. The slides are acquired through a Hamamatsu Nanozoomer S210 scanner at 20x magnification (0.4415μm/px), as exemplified in Fig. 9.1.1. Each slide belongs to a different patient and is annotated by expert UniTo pathologists, according to six classes as follows:

NORM - Normal tissue
HP - Hyperplastic Polyp
TA.HG - Tubular Adenoma, High-Grade dysplasia
TA.LG - Tubular Adenoma, Low-Grade dysplasia
TVA.HG - Tubulo-Villous Adenoma, High-Grade dysplasia
TVA.LG - Tubulo-Villous Adenoma, Low-Grade dysplasia

[1] https://ieee-dataport.org/open-access/unitopatho

Hyperplastic polyps usually exhibit no malignant potential (Tseung, 2005), while adenomas are more likely to progress into invasive carcinomas. Tubular and tubulovillous are common colorectal adenomas, with villous adenomas generally presenting higher malignant potential given the larger surface (Tseung, 2005). Adenomas are associated with a grade of dysplasia, low grade (LG) or high grade (HG), which measures the abnormality in cellular growth and differentiation[2]. Higher grade dysplasia indicates higher malignant potential. Arguably, correctly recognizing the dysplasia grade is the most relevant task from a clinical point of view. However, HG samples are also harder to acquire and label.

Improving grading with prior-based augmentation

To tackle this issue, we are exploiting generative models to augment the collected datasets and improve the final classification accuracy. The aim of this research is to include medical prior knowledge related to the morphological structure of HG tissue into the generative process, in order to address the lack of HG data. At first, the prior knowledge will be provided in the form of hand-crafted tissue mask by the pathologists, to overlay onto normal or LG tissue. Future research will focus on removing the need for hand-crafted features, by exploiting diagnostic guidelines (Gibson and Odze, 2016).

Also in this case, leveraging prior knowledge can finally lead to obtaining more robust models for the detection of relevant conditions.

9.2 Conclusions

By incorporating prior knowledge into DNNs, especially in the medical field, we can guide the learning process towards the selection of relevant input features, and constraining it to align with established medical principles. Merging explicit knowledge (e.g. using symbolic reasoning) and machine learning is referred to as Hybrid AI, and has recently received interest in the deep learning community (Mao et al., 2019). The potential impact of this proposal extend to different important issue in the field:

- *Interpretability and Explainability:* DNNs are black-box models and the existing techniques which try to explain their behaviour often provides limited insights. The integration of prior knowledge into DNNs can introduce better interpretability, enabling healthcare professionals to trust and validate the model's output (Delmonte et al., 2019)

- *Data Efficiency:* acquiring labeled data can be time-consuming and expensive. Human prior knowledge can help overcome data limitations by guiding the learning process with domain-specific insights. By incorporating this knowledge, DNNs can make more accurate predictions even with smaller datasets, leading to improved diagnostics and treatment strategies. Furthermore, human

[2] https://www.cancer.gov/publications/dictionaries/cancer-terms/def/dysplasia

expertise can aid in the identification of crucial data features and provide valuable insights for data augmentation techniques, further enhancing the model's generalizability

- *Dealing with Collateral Learning* Integrating human prior knowledge into DNNs also allows for the incorporation of ethical considerations. Bias and fairness are critical concerns in healthcare, as decisions based on biased models can lead to disparities in patient outcomes. Human expertise can help identify potential biases in the data, guide the model's training process to account for diverse patient populations, and ensure that the predictions align with established ethical guidelines.

Chapter 10

Conclusions

This thesis introduced the concept of Collateral Learning in deep learning, which refers to the instances where DL models learn more information than expected. Collateral Learning is a fundamental aspect to be considered in the area of deep learning, and it has significant implications across various research domains, including robustness, debiasing, generalization in medical imaging, and privacy preservation. By proposing this concept, we seek to unify and advance these often distinct research fields. In Part I we introduced Collateral Learning, and presented some of its most common instances.

In Part II we have laid the foundations of our work, by developing techniques for robust representation learning on natural images. We focused on one of the most important instances of Collateral Learning, namely biased data. Our first contribution, presented in Chapter 3, consists of a regularization term named EnD, that aims at reducing the impact of spurious correlations in the input data through feature entanglement and disentanglement, by leveraging bias annotations.
To better study this problem, in Chapter 4, we presented a metric framework for representation learning. This framework allowed us to derive a novel supervised contrastive loss function (ϵ-SupInfoNCE), which obtained superior performance than the current state-of-the-art. Most importantly, thanks to this framework, we were able to provide a formal characterization of the effect of the bias in the network's latent space. This enabled us to formulate a novel debiasing regularization term, FairKL, which obtained the best performance compared to both EnD and other existing methods, on a number of benchmarks.
In Chapter 5, we proposed a way to overcome one of the major limitations of our debiasing methods, which is requiring bias labels. We showed that it is possible to leverage the natural tendency of the networks to prefer simpler features in the latent space, in order to retrieve the unknown biases with clustering. This approach enabled us to apply both EnD and FairKL in unsupervised debiasing settings. Furthermore, we also showed that the clustering step may be avoided and that it is possible to leverage a biased model to obtain an unbiased one, by integrating a kernel function into the regularization term.

We then dealt with the Collateral Learning problem in real-world cases, focusing on medical imaging, in Part III.
In Chapter 6, we proposed a novel contrastive learning method for the regression of

brain age. This method aims at obtaining a semantic mapping between the learned latent space and the target regression variable (age), by employing a kernel. To the best of our knowledge, our method is one of the first attempts at solving regression problems with contrastive learning (while some works exist, they generally leverage existing contrastive loss functions for pretraining). Our approach showed state-of-the-art performance for brain age prediction on OpenBHB, a large multi-site brain MRI dataset. Besides reaching lower test error than other existing methods such as BrainAGE, our approach also showed increased robustness to the site noise, a common Collateral Learning phenomenon in multi-site medical datasets. We also proposed a possible extension of the FairKL regularization for regression; with it we expect to achieve even higher robustness. Based on our brain age model, we were also able to obtain promising preliminary results for the detection and classification of neurodegenerative diseases and brain conditions, such as Mild Cognitive Impairment (MCI) and Alzheimer's Disease (AD). With our method, it is possible to observe a separation in brain age gap (the difference between an individual's brain age and their chronological age) across healthy, MCI and AD patients; confirming that accurate brain age prediction can be an invaluable tool for detecting unhealthy aging patterns in the brain.

In Chapter 7 we showed another practical and relevant instance of Collateral Learning in medical imaging, related to the Covid-19 detection from chest X-ray images. This task posed a great challenge, especially in the early phases of the pandemics, as the available dataset were limited and highly biased. Our methodological contribution highlighted the issues in some of the most commonly used datasets at the time, as their use could introduce strong biases and spurious information such as site noise. Based on these findings, we then proposed a two-step transfer learning approach for mitigating such Collateral Learning issues. This approach, based on the detection of objective radiological findings, which shares some similarities with Concept Learning, has achieved better results for Covid-19 prediction. Additionally, we presented our latest (and ongoing) efforts, within the CoR.S.A. project, for building CORDA, a publicly available multi-site CXRs dataset for Covid-19, and we have obtain promising results towards removing the site effect with FairKL.

Finally, in Part IV we focused on another instance of Collateral Learning, related to privacy preservation in deep learning applications. After showing that even features not related to the primary learning task can be picked up by DL models, we have empirically demonstrated that this issue can be mitigated by also employing debiasing techniques such as EnD. In fact, by doing so, it becomes almost impossible to recover potentially sensitive information from the model's output or latent space (e.g. gender). We have provided practical examples of such occurrence on facial images and CXR images.

In summary, this thesis dealt with the more general and relevant topic of reliability in deep learning. In fact, collateral-free learning could be seen as one of the requirements for achieving trustworthy, and perhaps more explainable, deep learning models. Overall, we argue that proposing a common context for referring to different instances of the same core phenomenon (e.g. biases, site noise, etc.) can help advance the scientific progress in this regards; as Collateral Learning still presents

many challenges to study and overcome in the future (e.g. subtler biases, multiple biases at once, privacy concerns in federated learning, etc.). We will present some of these aspects in the following chapter.

Chapter 11

Future Perspectives

In this Chapter, we present some potentially relevant future research directions, on the topic of Collateral Learning. First of all, while in this thesis we proposed a first definition of this phenomenon, there may be other instances of Collateral Learning that have not been identified or explored. Future research should focus on discovering and characterizing these instances and developing strategies for addressing them. Here is a brief list of relevant topics that could be the focus of future research:

- Developing more effective debiasing methods: While the debiasing methods proposed in this thesis have shown promising results, there is still room for improvement. Future research could focus on developing more effective debiasing methods that can handle a wider range of Collateral Learning scenarios, especially when biases are subtler and not so easy to detect.

- Addressing Collateral Learning in other domains: The impact of Collateral Learning is not limited to medical imaging and natural images. Future research could explore how Collateral Learning manifests in other domains, such as text, speech, or robotics, and develop strategies for addressing it in these domains.

- Studying the transfer learning aspect of Collateral Learning: Transfer learning is a common technique used in deep learning that involves using pre-trained models as a starting point for new tasks. Future research could explore how transfer learning affects Collateral Learning and develop strategies for mitigating its impact.

- Exploring the relationship between Collateral Learning and Adversarial attacks: Adversarial attacks are designed to manipulate the predictions made by deep learning models. Future research could explore the relationship between Collateral Learning and adversarial attacks and develop strategies for mitigating their impact.

- Studying more in-depth the relationship between Collateral Learning and privacy preservation: in this thesis, we have shown how Collateral Learning may lead to the leakage of private information, and we have proposed some solutions. Future work should focus also on this area, in relevant contexts such as federated learning.

In the following sections, we will present some of these aspects in more detail.

11.1 Collateral Learning and PCA

The framework that we presented in Chapter 4 is developed for supervised learning. We aim to extend it to the self-supervised case, where learning unbiased and robust representation is still an open issue to tackle. In fact, dealing with Collateral Learning in a self-supervised scenario is a topic of relevance and, currently, there are very few works dealing with this issue.

As we defined in Section 1.2.1, we can view a sample x as the composition of different components coming from some signal sources $\mathbb{S}$. Based on this formulation, we gave a definition of diversity or correlation shift. Looking at this formulation with another goal, e.g. classification, we might say that some of the components $\mathbb{S}_c \subset \mathbb{S}$ will be more important than others for determining the target class. For the sake of simplicity, we can assume that the target class can be determined by one single component $\mathbb{S}_i \in \mathbb{S}$, while all of the other components encode additional information that is not necessary for correct classification. For example, these additional components might encode collateral information (bias, noise, etc.). Being able to discriminate the principal components in the data is at the core of methods such as Principal Component Analysis (PCA). However, such analysis can be difficult in instances of strong correlation among the components, such as in biased data. Furthermore, in complex data, it can become hard to provide an interpretable meaning to the principal components. For this reason, methods to determine and disentangle the contribution of each component independently can be highly relevant. The idea of determining disentangled representations of data is often found in the generative modeling literature (Karras et al., 2019; Kingma and Welling, 2013; Tran et al., 2017), and perhaps it could be leveraged to also deal with Collateral Learning.

Some approaches based on contrastive learning could be useful for this purpose. For example, Hinton (2022) proposes a forward learning method that aims at maximizing the model activation on positive samples and minimizing it on negative ones. It can be shown that the proposed objective function is similar to PCA. Another work (Abid et al., 2017) proposes Contrastive PCA with the goal of discovering common and distinctive patterns across different datasets. Generally speaking, being able to separate the different components in the data may allow us to better mitigate the Collateral Learning problem.

11.2 Robust self-supervised learning

In all of this thesis, we dealt with supervised learning. While it is relevant in several contexts, self-supervised approaches are also useful for obtaining good representations of the data when labels are not available. In fact, although much of the debiasing and fairness literature deals with supervised learning, in many real-world cases we want to leverage large unlabelled datasets. Dealing with biases and noise in such cases, however, is inherently more difficult, and very few works have proposed methods for tackling this scenario. A possible approach is proposed in the work by Chai and Wang (2022), where authors aim to study how to achieve fair classification without bias labels and target labels. Toward that aim, they employ a reweighing-based contrastive learning method. However, their approach still needs

a small set annotated with labels.

Developing ways to achieve robustness to biases, or in general to Collateral Learning, in self-supervised scenarios can potentially have a high relevance in several applications. The direction suggested in the previous Section could be relevant to this goal.

11.3 Removing multiple biases

Another relevant research direction on the topic of debiasing is dealing with multiple biases in the data. All of the methods presented in this thesis, and also almost all related literature (Kim et al., 2023; Wu et al., 2020), deal with the case in which only one spurious correlation exists in the data (e.g. color, gender, age, etc.). In real-world scenarios, however, it may happen that the data present multiple spurious correlations, each one having a contribution in the way the model is biased.

A simple example, which we already encountered in this thesis in Chapter 3, is given by the CelebA dataset. In the experiments that we presented, we considered separately the bias caused by the correlation of gender and hair color, or gender and facial makeup. Considering the contribution of both attributes towards the bias, and also the other attributes present in the dataset should be the focus of future research. Doing so could be as simple as applying the same regularization to all the possible bias attributes, e.g.:

$$\mathcal{L} = \mathcal{L}_{CE} + \mathbb{E}_{b \in \{hair, makeup, \dots\}} \left[\mathcal{R}_b^{debias} \right] \tag{11.3.1}$$

However, this trivial extension might result in suboptimal results, as different biases may have different strengths (Wu et al., 2020). For this reason, developing novel debiasing methods specifically for dealing with multiple biases should be the focus of future research, for example by leveraging the metric framework of Chapter 4 and extending the characterization of bias (4.2.1) to this case. Dealing with multiple biases can also be relevant for medical imaging, where usually data can be stratified with respect to multiple factors.

11.4 Federated learning and privacy concerns

Federated Learning (FL) is an area of research that has gained significant attention in recent years due to its ability to enable distributed machine learning training on private data. FL is an approach that enables different participants, each one holding some private data, to train machine learning models in a collaborative way, without having to centralize the data. FL is highly relevant for various applications, including healthcare, finance, and industry. In the context of FL, there are several instances of Collateral Learning that are relevant to the theme of this thesis, specifically privacy of the data and bias and fairness.

In the context of FL, we can identify a few relevant instances of Collateral Learning:

- privacy of the data

- bias and fairness

While FL is already developed for protecting data privacy, it would be interesting to study whether methods such as the ones proposed in this thesis could also be applied in FL for privacy preservation purposes, similarly to what we presented in Chapter 8. In fact, research about potential issues, attacks, and defensive strategies in FL is highly relevant to the field (Enthoven and Al-Ars, 2021; Liu et al., 2022; Lyu et al., 2022).

Regarding biases and fairness in FL context, recent works have highlighted this issue (Abay et al., 2020; Chang and Shokri, 2023; Djebrouni, 2022) and some solutions was proposed (Chu et al., 2021; Du et al., 2021; Zeng et al., 2021; Zhang et al., 2020). An interesting research direction would be extending our proposed debiasing methods to the FL context. Focusing on medical data, also issues such as site effects should be carefully considered. Toward this end, recently proposed FL datasets and benchmarks such as FLamby (Terrail et al., 2022) could be leveraged. Specifically, the FLamby dataset is built to simulate a realistic scenario of cross-silo FL, with up to 50 participating clients.

In summary, FL presents an excellent opportunity to apply the concepts of Collateral Learning discussed in this thesis. Investigating the application of these methods in FL would not only contribute to the development of more robust and reliable machine learning models but also help address critical challenges related to data privacy and bias in distributed learning settings.

Bibliography

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016. 113, 115

Annie Abay, Yi Zhou, Nathalie Baracaldo, Shashank Rajamoni, Ebube Chuba, and Heiko Ludwig. Mitigating bias in federated learning. *arXiv preprint arXiv:2012.02447*, 2020. 136

Abubakar Abid, Martin J Zhang, Vivek K Bagaria, and James Zou. Contrastive principal component analysis. *arXiv preprint arXiv:1709.06716*, 2017. 134

AI HLEG. *Ethics guidelines for trustworthy AI*. High-Level Expert Group on Artificial Intelligence, 2019. URL https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai. 13

Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 34, 52, 53, 76

Marios Anthimopoulos, Stergios Christodoulidis, Lukas Ebner, Andreas Christe, and Stavroula Mougiakakou. Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. *IEEE transactions on medical imaging*, 35(5):1207–1216, 2016. 39

Ioannis D Apostolopoulos and Tzani A Mpesiana. Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*, page 1, 2020. 15, 39, 104, 105

Devansh Arpit, Stanisław Jastrzundefinedbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 233–242. JMLR.org, 2017. 17, 44, 74, 119

Joshua Attenberg, Panos Ipeirotis, and Foster Provost. Beat the machine: Challenging humans to find a predictive model's "unknown unknowns". *Journal of Data and Information Quality (JDIQ)*, 6(1):1–17, 2015. 15

Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning (ICML)*, 2020. 35, 43, 48, 49, 63, 66, 71, 72, 73, 113, 168

Carlo Alberto Barbano, Daniele Perlo, Enzo Tartaglione, Attilio Fiandrotti, Luca Bertero, Paola Cassoni, and Marco Grangetto. Unitopatho, a labeled histopathological dataset for colorectal polyps classification and adenoma dysplasia grading. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 76–80. IEEE, 2021. URL https://ieeexplore.ieee.org/abstract/document/9506198. 126

Suzanna Becker and Geoffrey Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161–3, 02 1992. doi: 10.1038/355161a0. 31

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 472–489, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01270-0. 15

Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Allison Woodruff, Christine Luu, Pierre Kreitmann, Jonathan Bischof, and Ed H Chi. Putting fairness principles into practice: Challenges, metrics, and improvements. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 453–459, 2019. 33

Roisin Bevan and Matthew D Rutter. Colorectal cancer screening—who, how, and when? *Clinical endoscopy*, 51(1):37, 2018. 125

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com. 49, 51, 52

N. Bondfale and D. S. Bhagwat. Convolutional neural network for categorization of lung tissue patterns in interstitial lung diseases. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 1150–1154, 2018. 39

Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *7th International Conference on Learning Representations, ICLR 2019*, 3 2019. doi: 10.48550/arxiv.1904.00760. URL https://arxiv.org/abs/1904.00760v1. 77

Karl Bäckström, Mahmood Nazari, Irene Yu-Hua Gu, and Asgeir Store Jakola. An efficient 3d deep convolutional network for alzheimer's disease diagnosis using mr images. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 149–153, 2018. doi: 10.1109/ISBI.2018.8363543. 95

Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. In *Advances in neural information processing systems*, pages 841–852, 2019. 35, 48, 73, 113

Junyi Chai and Xiaoqian Wang. Self-supervised fair representation learning without demographics. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=7TGpLKADODE. 134

Hongyan Chang and Reza Shokri. Bias propagation in federated learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=V7CYzdruWdm. 136

Andrew A. Chen, Joanne C. Beer, Nicholas J. Tustison, Philip A. Cook, Russell T. Shinohara, and Haochang Shou. Mitigating site effects in covariance for machine learning in neuroimaging data. *Human Brain Mapping*, 43:1179–1195, 3 2022. ISSN 10970193. doi: 10.1002/HBM.25688. 15, 17, 39

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, November 2020. URL http://proceedings.mlr.press/v119/chen20j.html. ISSN: 2640-3498. 14, 30, 31, 55, 56, 82, 123, 167, 168

Ting Chen, Calvin Luo, and Lala Li. Intriguing Properties of Contrastive Losses. *arXiv:2011.02803 [cs, stat]*, June 2021. URL http://arxiv.org/abs/2011.02803. arXiv: 2011.02803. 124

Danni Cheng and Manhua Liu. Cnns based multi-modality classification for ad diagnosis. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–5, 2017. doi: 10.1109/CISP-BMEI.2017.8302281. 95

S. Chopra, R. Hadsell, and Y. LeCun. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In *CVPR*, volume 1, pages 539–546. IEEE, 2005. ISBN 978-0-7695-2372-9. doi: 10.1109/CVPR.2005.202. URL http://ieeexplore.ieee.org/document/1467314/. 56

Lingyang Chu, Lanjun Wang, Yanjie Dong, Jian Pei, Zirui Zhou, and Yong Zhang. Fedfair: Training fair models in cross-silo federated learning. *arXiv preprint arXiv:2109.05662*, 2021. 136

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4067–4080. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1418. URL https://doi.org/10.18653/v1/D19-1418. 35, 48, 49, 73, 113

Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 image data collection. *arXiv 2003.11597*, 2020. URL https://github.com/ieee8023/covid-chestxray-dataset. 39, 100

James H. Cole, Rudra P.K. Poudel, Dimosthenis Tsagkrasoulis, Matthan W.A. Caan, Claire Steves, Tim D. Spector, and Giovanni Montana. Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker. *NeuroImage*, 163:115–124, 2017. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2017.07.059. URL https://www.sciencedirect.com/science/article/pii/S1053811917306407. 82

Irene Cumplido-Mayoral, Marina García-Prat, Grégory Operto, Carles Falcon, Mahnaz Shekari, Raffaele Cacciaglia, Marta Milà-Alomà, Luigi Lorenzini, Silvia Ingala, Alle Meije Wink, Henk JMM Mutsaerts, Carolina Minguillón, Karine Fauria, José Luis Molinuevo, Sven Haller, Gael Chetelat, Adam Waldman, Adam J Schwarz, Frederik Barkhof, Ivonne Suridjan, Gwendlyn Kollmorgen, Anna Bayfield, Henrik Zetterberg, Kaj Blennow, Marc Suárez-Calvet, Verónica Vilaplana, and Juan Domingo Gispert. Biological brain age prediction using machine learning on structural neuroimaging data: Multi-cohort validation against biomarkers of alzheimer's disease and neurodegeneration stratified by sex. *eLife*, 12, 4 2023. ISSN 2050-084X. doi: 10.7554/eLife.81067. URL https://elifesciences.org/articles/81067. 81, 92

DeepHealth. Deep-learning and hpc to boost biomedical applications for health. 2019. URL https://deephealth-project.eu/. 126

A. Delmonte, C. Mercier, J. Pallud, I. Bloch, and P. Gori. White matter multiresolution segmentation using fuzzy set theory. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 459–462, 2019. doi: 10.1109/ISBI.2019.8759506. 127

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 14, 17, 23, 29, 30, 63

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 43

Bernard Denis, Carol Peters, Catherine Chapelain, Isabelle Kleinclaus, Anne Fricker, Richard Wild, Bernard Auge, Isabelle Gendre, Philippe Perrin, Denis Chatelain, et al. Diagnostic accuracy of community pathologists in the interpretation of colorectal polyps. *European journal of gastroenterology & hepatology*, 21 (10):1153–1160, 2009. 125

Blake E Dewey, Can Zhao, Jacob C Reinhold, Aaron Carass, Kathryn C Fitzgerald, Elias S Sotirchos, Shiv Saidha, Jiwon Oh, Dzung L Pham, Peter A Calabresi, et al. Deepharmony: A deep learning approach to contrast harmonization across scanner changes. *Magnetic resonance imaging*, 64:160–170, 2019. 81

J. Dewey. *Experience And Education*. Free Press, 1997. ISBN 9780684838281. URL https://books.google.fr/books?id=UWbuAAAAMAAJ. 14

Adriana Di Martino, Chao-Gan Yan, Qingyang Li, Erin Denio, Francisco Castellanos, Kaat Alaerts, Jeffrey Anderson, Michal Assaf, Susan Bookheimer, Mirella Dapretto, Ben Deen, Sonja Delmonte, Ilan Dinstein, Ertl-Wagner Birgit, Damien

Fair, Louise Gallagher, Daniel Kennedy, Christopher Keown, Christian Keysers, and Michael Milham. The autism brain imaging data exchange: Towards large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19, 06 2013. doi: 10.1038/mp.2013.78. 83

Adriana Di Martino, David O'Connor, Bosi Chen, Kaat Alaerts, Jeffrey Anderson, Michal Assaf, Joshua Balsters, Leslie Baxter, Anita Beggiato, Sylvie Bernaerts, Laura Blanken, Susan Bookheimer, B. Braden, Lisa Byrge, Francisco Castellanos, Mirella Dapretto, Richard Delorme, Damien Fair, Inna Fishman, and Michael Milham. Enhancing studies of the connectome in autism using the autism brain imaging data exchange ii. *Scientific Data*, 4:170010, 03 2017. doi: 10.1038/sdata.2017.10. 83

Yasmine Djebrouni. Towards bias mitigation in federated learning. *16th EuroSys Doctoral Workshop*, 2022. 136

Nathan Drenkow, Numair Sani, Ilya Shpitser, and Mathias Unberath. A systematic review of robustness in deep learning for computer vision: Mind the gap? *arXiv preprint arXiv:2112.00639*, 2021. 13

Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. Fairness-aware agnostic federated learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 181–189. SIAM, 2021. 136

Benoit Dufumier. *Representation learning in neuroimaging: transferring from big healthy data to small clinical cohorts*. PhD thesis, Université Paris-Saclay, 2022. 32

Benoit Dufumier, Pietro Gori, Julie Victor, Antoine Grigis, and Edouard Duchesnay. Conditional alignment and uniformity for contrastive learning with continuous proxy labels. *NeurIPS Workshop on Medical Imaging Meets NeurIPS*, 2021a. 31, 33, 82

Benoit Dufumier, Pietro Gori, Julie Victor, Antoine Grigis, Michele Wessa, Paolo Brambilla, Pauline Favre, Mircea Polosan, Colm McDonald, Camille Marie Piguet, et al. Contrastive learning with continuous proxy meta-data for 3d mri classification. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part II 24*, pages 58–68. Springer, 2021b. 33, 85, 95, 123, 124

Benoit Dufumier, Antoine Grigis, Julie Victor, Corentin Ambroise, Vincent Frouin, and Edouard Duchesnay. Openbhb: a large-scale multi-site brain mri data-set for age prediction and debiasing. *NeuroImage*, page 119637, 2022. 15, 82, 86, 88, 92

Benoit Dufumier, Carlo Alberto Barbano, Robin Louiset, Edouard Duchesnay, and Pietro Gori. Integrating prior knowledge in contrastive learning with kernel. *Fortieth International Conference on Machine Learning (ICML)*, 2023. 32, 82, 124

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380, 2009. 115

Eran Eidinger, Roee Enbar, and Tal Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12): 2170–2179, 2014. 52

Maxwell Elliott, Daniel Belsky, Annchen Knodt, David Ireland, Tracy Melzer, Richie Poulton, Sandhya Ramrakha, Avshalom Caspi, Terrie Moffitt, and Ahmad Hariri. Brain-age in midlife is associated with accelerated biological aging and cognitive decline in a longitudinal birth cohort. *Molecular Psychiatry*, 26:1–10, 08 2021. doi: 10.1038/s41380-019-0626-7. 81, 90, 92

David Enthoven and Zaid Al-Ars. An overview of federated deep learning privacy attacks and defensive strategies. *Federated Learning Systems: Towards Next-Generation AI*, pages 173–196, 2021. 136

Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014. 115

Ivan P Fellegi. On the question of statistical confidentiality. *Journal of the American Statistical Association*, 67(337):7–18, 1972. 113

D. Jake Follmer, Shin-Yi Fang, Roy Clariana, Bonnie Meyer, and Ping Li. What predicts adult readers' understanding of stem texts? *Reading and Writing*, 31, 01 2018. doi: 10.1007/s11145-017-9781-x. 83

Jean Philippe Fortin, Elizabeth M. Sweeney, John Muschelli, Ciprian M. Crainiceanu, and Russell T. Shinohara. Removing inter-subject technical variability in magnetic resonance imaging studies. *NeuroImage*, 132:198–212, 5 2016. ISSN 10959572. doi: 10.1016/J.NEUROIMAGE.2016.02.036. 39

Jean-Philippe Fortin, Nicholas Cullen, Yvette I. Sheline, Warren D. Taylor, Irem Aselcioglu, Phil Adams, Crystal Cooper, Maurizio Fava, Patrick J. McGrath, Melvin McInnis, Ramin V. Parsey, Mary L. Phillips, Madhukar H. Trivedi, Myrna M. Weissman, and Russell T. Shinohara. Harmonization of cortical thickness measurements across scanners and sites. *bioRxiv*, 2017. doi: 10.1101/148502. URL https://www.biorxiv.org/content/early/2017/06/10/148502. 39, 81, 88

Katja Franke and Christian Gaser. Ten years of brainage as a neuroimaging biomarker of brain aging: What insights have we gained? *Frontiers in Neurology*, 10, 08 2019. doi: 10.3389/fneur.2019.00789. 37, 38, 97

Katja Franke, Gabriel Ziegler, Stefan Klöppel, Christian Gaser, Alzheimer's Disease Neuroimaging Initiative, et al. Estimating the age of healthy subjects from t1-weighted mri scans using kernel methods: exploring the influence of various parameters. *Neuroimage*, 50(3):883–892, 2010. 37, 39, 81

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA, 2015. Association for Computing

Machinery. ISBN 9781450338325. doi: 10.1145/2810103.2813677. URL https://doi.org/10.1145/2810103.2813677. 18, 116

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ganin15.html. 16

Beatriz Garcia Santa Cruz, Matías Nicolás Bossa, Jan Sölter, and Andreas Dominik Husch. Public covid-19 x-ray datasets and their impact on model bias – a systematic review of a significant problem. *Medical Image Analysis*, 74:102225, 2021. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2021.102225. URL https://www.sciencedirect.com/science/article/pii/S136184152100270X. 104

Christian Gaser, Katja Franke, Stefan Klöppel, Nikolaos Koutsouleris, Heinrich Sauer, and ADNI. Brainage in mild cognitive impaired patients: Predicting the conversion to alzheimer's disease. *PLoS ONE*, 8:e67346, 06 2013. doi: 10.1371/journal.pone.0067346. 81, 90, 92

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bygh9j09KX. 16, 17, 77

Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009. 113, 114

Joanna A Gibson and Robert D. Odze. Pathology of premalignant colorectal neoplasia. *Digestive Endoscopy*, 28:312 – 323, 2016. 127

Afina S Glas, Jeroen G Lijmer, Martin H Prins, Gouke J Bonsel, and Patrick MM Bossuyt. The diagnostic odds ratio: a single indicator of test performance. *Journal of clinical epidemiology*, 56(11):1129–1135, 2003. 181

Ben Glocker, Robert Robinson, Daniel C. Castro, Qi Dou, and Ender Konukoglu. Machine learning with multi-site imaging data: An empirical study on the impact of scanner effects. 10 2019. URL http://arxiv.org/abs/1910.04597. 15, 17, 39, 81, 82

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011. 24

Raul S Gonzalez. Updates and challenges in gastrointestinal pathology. *Surgical Pathology Clinics*, 13(3):ix, 2020. 125

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets.

In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf. 34, 124

Florian Graf, Christoph D Hofer, Marc Niethammer, and Roland Kwitt. Dissecting supervised constrastive learning. 2021. 32, 82, 166

Wei-jie Guan, Zheng-yi Ni, Yu Hu, Wen-hua Liang, Chun-quan Ou, Jian-xing He, Lei Liu, Hong Shan, Chun-liang Lei, David SC Hui, et al. Clinical characteristics of coronavirus disease 2019 in china. *New England journal of medicine*, 382(18): 1708–1720, 2020. 189

Abhinav Gupta, Adithyavairavan Murali, Dhiraj Prakashchand Gandhi, and Lerrel Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. In *Advances in Neural Information Processing Systems*, pages 9094–9104, 2018. 33

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality Reduction by Learning an Invariant Mapping. In *CVPR*, volume 2, pages 1735–1742. IEEE, 2006. 30, 56

David M Hansell, Alexander A Bankier, Heber MacMahon, Theresa C McLoud, Nestor L Muller, and Jacques Remy. Fleischner society: glossary of terms for thoracic imaging. *Radiology*, 246(3):697–722, 2008. 100, 106

Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973. doi: 10.1109/TSMC.1973.4309314. 35

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 29, 167, 186

Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *European Conference on Computer Vision*, pages 793–811. Springer, 2018. 35

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *7th International Conference on Learning Representations, ICLR 2019*, 3 2019. doi: 10.48550/arxiv.1903.12261. URL https://arxiv.org/abs/1903.12261v1. 54, 63, 65

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. 54, 67, 77

Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2022. 134

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, 2000. ISSN 00401706. URL http://www.jstor.org/stable/1271436. 92

Avram Holmes, Marisa Hollinshead, Timothy O'Keefe, Victor Petrov, Gabriele Fariello, Lawrence Wald, Bruce Fischl, Bruce Rosen, Ross Mair, Joshua Roffman, Jordan Smoller, and Randy Buckner. Brain genomics superstruct project initial data release with structural, functional, and behavioral measures. *Scientific data*, 2:150031, 07 2015. doi: 10.1038/sdata.2015.31. 83

Youngkyu Hong and Eunho Yang. Unbiased classification through bias-contrastive and bias-balanced learning. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=2OqZZAqxnn. 53, 59, 63, 64, 65, 73, 76, 77, 167

Frederick M. Howard, James Dolezal, Sara Kochanny, Jefree Schulte, Heather Chen, Lara Heij, Dezheng Huo, Rita Nanda, Olufunmilayo I. Olopade, Jakob N. Kather, Nicole Cipriani, Robert L. Grossman, and Alexander T. Pearson. The impact of site-specific digital histology signatures on deep learning model accuracy and bias. *Nature Communications 2021 12:1*, 12:1–13, 7 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-24698-1. URL https://www.nature.com/articles/s41467-021-24698-1. 15, 17

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 29, 30, 116, 186

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in Neural Information Processing Systems*, 32, 5 2019. ISSN 10495258. doi: 10.48550/arxiv.1905.02175. URL https://arxiv.org/abs/1905.02175v4. 77

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 590–597, 2019. 106, 187

Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 299–316. IEEE, 2019. 113, 115

Madabhushi A Janowczyk A. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of pathology informatics*, pages 7–29, 2016. 125

Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019. 36

Alistair E. W. Johnson, Mohammad M. Ghassemi, Shamim Nemati, Katherine E. Niehaus, David Clifton, and Gari D. Clifford. Machine learning and decision support in critical care. *Proceedings of the IEEE*, 104(2):444 – 466, 2016. doi: 10.1109/JPROC.2015.2501978. 13

W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007. 39

B A Jonsson, G Bjornsdottir, T E Thorgeirsson, L M Ellingsen, G Bragi Walters, D F Gudbjartsson, H Stefansson, K Stefansson, and M O Ulfarsson. Brain age prediction using deep learning uncovers associated sequence variants. *Nature Communications*. doi: 10.1038/s41467-019-13163-9. URL https://doi.org/10.1038/s41467-019-13163-9. 82

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. *Advances in neural information processing systems*, 27: 2879–2887, 2014. 115

Renuga Kanagavelu, Zengxiang Li, Juniarto Samsudin, Yechao Yang, Feng Yang, Rick Siow Mong Goh, Mervyn Cheah, Praewpiraya Wiwatphonthana, Khajonpong Akkarajitsakul, and Shangguang Wang. Two-phase multi-party computation enabled privacy-preserving federated learning. In *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, pages 410–419. IEEE, 2020. 113, 114

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 134

Kang; Goldbaum Michael Kermany, Daniel; Zhang. Labeled optical coherence tomography (oct) and chest x-ray images for classification. https://data.mendeley.com/datasets/rscbjbr9sj/3, 2017. 40, 100

A. Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012. 33

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. volume 33, pages 18661–18673. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf. 14, 30, 31, 32, 55, 57, 58, 63, 64, 65, 82, 85, 167, 168

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 34, 52, 53, 65, 76

Nayeong Kim, Juwo Kang, Sungsoo Ahn, Jungseul Ok, and Suha Kwak. Removing multiple biases through the lens of multi-task learning. *ICML Workshop on Spurious Correlations, Invariance and Stability (SCIS)*, 2023. 135

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 124, 134

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/koh20a.html. 107

Bruno Korbar, Andrea M Olofson, Allen P Miraflor, Catherine M Nicka, Matthew A Suriawinata, Lorenzo Torresani, Arief A Suriawinata, and Saeed Hassanpour. Deep learning for classification of colorectal polyps on whole-slide images. *Journal of pathology informatics*, 8, 2017. 125

Sergey Korolev, Amir Safiullin, Mikhail Belyaev, and Yulia Dodonova. Residual and plain convolutional neural networks for 3d brain mri classification. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 835–838, 2017. doi: 10.1109/ISBI.2017.7950647. 95

Nikolaos Koutsouleris, Christos Davatzikos, Stefan Borgwardt, Christian Gaser, Ronald Bottlender, Thomas Frodl, Peter Falkai, Anita Riecher-Rössler, Hans-Jürgen Möller, Maximilian Reiser, et al. Accelerated brain aging in schizophrenia and beyond: a neuroanatomical marker of psychiatric disorders. *Schizophrenia bulletin*, 40(5):1140–1153, 2014. 81

Mathias Kraus and Stefan Feuerriegel. Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems*, 104: 38 – 48, 2017. doi: 10.1016/j.dss.2017.10.001. 13

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). a. URL http://www.cs.toronto.edu/~kriz/cifar.html. 63

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). b. URL http://www.cs.toronto.edu/~kriz/cifar.html. 63

Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957, 1992. 27

S.W.-C. Lam. Texture feature extraction using gray level gradient based co-occurence matrices. In *1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No.96CH35929)*, volume 1, pages 267–271 vol.1, 1996. doi: 10.1109/ICSMC.1996.569778. 35

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 27

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 29

Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=-oUhJJILWHb. 53, 54, 63, 66, 168

Fan Li, Danni Cheng, and Manhua Liu. Alzheimer's disease classification based on combination of multi-model convolutional networks. In *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–5, 2017. doi: 10.1109/IST.2017.8261566. 95

Yingwei Li, Qihang Yu, Mingxing Tan, Jieru Mei, Peng Tang, Wei Shen, Alan Yuille, and cihang xie. Shape-texture debiased neural network training. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Db4yerZTYkz. 16

Yehida Lindell. Secure multiparty computation for privacy preserving data mining. In *Encyclopedia of Data Warehousing and Mining*, pages 1005–1009. IGI global, 2005. 113, 114

Pengrui Liu, Xiangrui Xu, and Wei Wang. Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. *Cybersecurity*, 5(1):1–19, 2022. 136

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 51

S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. doi: 10.1109/TIT.1982.1056489. 71

Luyang Luo, Dunyuan Xu, Hao Chen, Tien-Tsin Wong, and Pheng-Ann Heng. Pseudo bias-balanced learning for debiased chest x-ray classification. 3 2022. doi: 10.48550/arxiv.2203.09860. URL https://arxiv.org/abs/2203.09860v1. 36

Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2507–2516, 2019. 16

Lingjuan Lyu, Han Yu, Xingjun Ma, Chen Chen, Lichao Sun, Jun Zhao, Qiang Yang, and S Yu Philip. Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems*, 2022. 136

José López-Cabrera, Jorge Portal Diaz, Ruben Orozco, Orlando Lovelle, and Marlen Perez-Diaz. Current limitations to identify covid-19 using artificial intelligence with chest x-ray imaging (part ii). the shortcut learning problem. *Health and Technology*, 11, 10 2021. doi: 10.1007/s12553-021-00609-8. 104

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. *arXiv preprint arXiv:1802.06309*, 2018. 34

Eleftheria Makri, Dragos Rotaru, Nigel P Smart, and Frederik Vercauteren. Epic: efficient private image classification (or: Learning from the masters). In *Cryptographers' Track at the RSA Conference*, pages 473–492. Springer, 2019. 115

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJgMlhRctm. 127

Peter R Millar, Brian A Gordon, Patrick H Luckett, Tammie LS Benzinger, Carlos Cruchaga, Anne M Fagan, Jason J Hassenstab, Richard J Perrin, Suzanne E Schindler, Ricardo F Allegri, Gregory S Day, Martin R Farlow, Hiroshi Mori, Georg Nübling, Randall J Bateman, John C Morris, and Beau M Ances. Multimodal brain age estimates relate to alzheimer disease biomarkers and cognition in early stages: a cross-sectional observational study. *eLife*, 12, 1 2023. ISSN 2050-084X. doi: 10.7554/eLife.81869. URL https://elifesciences.org/articles/81869. 81, 92

Tahmineh Mollasharifi, Mahsa Ahadi, Elena Jamali, Afshin Moradi, Parisa Asghari, Saman Maroufizadeh, and Behrang Kazeminezhad. Interobserver agreement in assessing dysplasia in colorectal adenomatous polyps: A multicentric iranian study. *Iranian Journal of Pathology*, pages 167–174, 2020. 125

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. In *Advances in Neural Information Processing Systems*, 2020. 17, 36, 44, 51, 59, 64, 66, 74, 75, 76, 78

Junhyun Nam, Jaehyung Kim, Jaeho Lee, and Jinwoo Shin. Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. 4 2022. doi: 10.48550/arxiv.2204.02070. URL https://arxiv.org/abs/2204.02070v1. 36

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008. 114

Ali Narin, Ceren Kaya, and Ziynet Pamuk. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *arXiv preprint arXiv:2003.10849*, 2020. 39, 104, 105

Samuel A Nastase, Yun-Fei Liu, Hanna Hillman, Asieh Zadbood, Liat Hasenfratz, Neggin Keshavarzian, Janice Chen, Christopher J Honey, Yaara Yeshurun, Mor Regev, et al. The "narratives" fmri dataset for evaluating models of naturalistic language comprehension. *Scientific data*, 8(1):250, 2021. 83

Harrison Nguyen, Richard W. Morris, Anthony W. Harris, Mayuresh S. Korgoankar, and Fabio Ramos. Correcting differences in multi-site neuroimaging data using generative adversarial networks. 3 2018. URL http://arxiv.org/abs/1803.09375. 15, 17, 39

Luke Oakden-Rayner, Jared Dunnmon, Gustavo Carneiro, and Christopher Re. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, CHIL '20, page 151–159, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370462. doi: 10.1145/3368555.3384468. URL https://doi.org/10.1145/3368555.3384468. 105

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv:1807.03748 [cs, stat]*, January 2019. URL http://arxiv.org/abs/1807.03748. arXiv: 1807.03748. 30, 55, 56

Dimitri Papadopoulos Orfanos, Vincent Michel, Yannick Schwartz, Philippe Pinel, Antonio Moreno, Denis Le Bihan, and Vincent Frouin. The brainomics/localizer database. *NeuroImage*, 144:309–314, 2017. 83

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. 30

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011. 92

Han Peng, Weikang Gong, Christian F Beckmann, Andrea Vedaldi, and Stephen M Smith. Accurate brain age prediction with lightweight deep neural networks. *Medical Image Analysis*, 68:101871, 2021. doi: 10.1016/j.media.2020.101871. URL https://doi.org/10.1016/j.media.2020.101871. 81, 82, 88

R. C. Petersen, P. S. Aisen, L. A. Beckett, M. C. Donohue, A. C. Gamst, D. J. Harvey, Jr C. R. Jack, W. J. Jagust, L. M. Shaw, A. W. Toga, J. Q. Trojanowski, and M. W. Weiner. Alzheimer's disease neuroimaging initiative (adni). *Neurology*, 74(3):201–209, 2010. ISSN 0028-3878. doi: 10.1212/WNL.0b013e3181cb3e25. URL https://n.neurology.org/content/74/3/201. 91

Adolf Pfefferbaum, Daniel H. Mathalon, Edith V. Sullivan, Jody M. Rawles, Robert B. Zipursky, and Kelvin O. Lim. A Quantitative Magnetic Resonance Imaging Study of Changes in Brain Morphology From Infancy to Late Adulthood. *Archives of Neurology*, 51(9):874–887, 09 1994. ISSN 0003-9942. doi: 10.1001/archneur.1994.00540210046012. URL https://doi.org/10.1001/archneur.1994.00540210046012. 37

NhatHai Phan, Yue Wang, Xintao Wu, and Dejing Dou. Differential privacy preservation for deep auto-encoders: an application of human behavior prediction. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 113, 115

NhatHai Phan, Xintao Wu, and Dejing Dou. Preserving differential privacy in convolutional deep belief networks. *Machine learning*, 106(9):1681–1704, 2017. 115

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A. Alemi, and George Tucker. On Variational Bounds of Mutual Information. In *ICML*, 2019. 30, 55, 57

Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *Advances in Neural Information Processing Systems*, 33:3407–3418, 2020. 123

Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008. 16

M Sadegh Riazi, Mohammad Samragh, Hao Chen, Kim Laine, Kristin Lauter, and Farinaz Koushanfar. {XONN}: Xnor-based oblivious deep neural network inference. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1501–1518, 2019. 114

Michael Roberts, Derek Driggs, Matthew Thorpe, Julian Gilbey, Michael Yeung, Stephan Ursprung, Angelica I Aviles-Rivero, Christian Etmann, Cathal McCague, Lucian Beer, et al. Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans. *Nature Machine Intelligence*, 3(3):199–217, 2021. 40, 104

A. Ross, M. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *IJCAI*, 2017. 35

Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126(2-4):144–157, 2018. 19, 52

Peter Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, 1987. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/0377-0427(87)90125-7. URL http://portal.acm.org/citation.cfm?id=38772. 71

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. 25

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019. 35, 51, 74, 75

Prasanna Sattigeri, Samuel C Hoffman, Vijil Chenthamarakshan, and Kush R Varshney. Fairness gan. *arXiv preprint arXiv:1805.09910*, 2018. 34

Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020. 13

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, June 2015. doi: 10.1109/CVPR.2015.7298682. URL http://arxiv.org/abs/1503.03832. arXiv: 1503.03832. 30, 56

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. 35, 48, 49, 191

Ramprasaath R. Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 35

Upul Senanayake, Arcot Sowmya, and Laughlin Dawes. Deep fusion pipeline for mild cognitive impairment diagnosis. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 1394–1997, 2018. doi: 10.1109/ISBI.2018.8363832. 95

Prabira Kumar Sethy and Santi Kumari Behera. Detection of coronavirus disease (covid-19) based on deep features. *Preprints*, 2020030300:2020, 2020. 15, 39, 104, 105

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020. 44, 123

Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016. 39

Vitaly Shmatikov and Congzheng Song. What are machine learning models hiding? 119

Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015. 113, 115

Shreyak. Building a convolutional neural network (cnn) model for image classification., Jun 2020. URL https://becominghuman.ai/building-a-convolutional-neural-network-cnn-model-for-image-classification-116f77a7a236. 28

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 29

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012. 49, 51, 52

Kihyuk Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://papers.nips.cc/paper/2016/hash/6b180037abbebea991d8b1232f8a8ca9-Abstract.html. 55, 56, 57

Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 587–601. ACM, 2017. 18

Zhigang Song, Chunkai Yu, Shuangmei Zou, Wenmiao Wang, Yong Huang, Xiaohui Ding, Jinhong Liu, Liwei Shao, Jing Yuan, Xiangnan Gou, et al. Automatic deep learning-based colorectal adenoma detection system and its similarities with pathologists. *BMJ open*, 10(9):e036423, 2020. 125

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 102

Carol; Carr Chris; Shih George; Dulkowski Jamie; kalpathy; Chen Leon; Prevedello Luciano; Kohli Marc; McDonald Mark; Phil Culliton Peter; Halabi Safwan; Xia Tian Stein, Anouk; Wu. Rsna pneumonia detection challenge. https://kaggle.com/competitions/rsna-pneumonia-detection-challenge, 2018. 100

Pierre Stock and Moustapha Cissé. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VI*, volume 11210 of *Lecture Notes in Computer Science*, pages 504–519. Springer, 2018. doi: 10.1007/978-3-030-01231-1_31. URL https://doi.org/10.1007/978-3-030-01231-1_31. 13

Kayt Sukel. Neuroanatomy: The basics. https://www.brainfacts.org/brain-anatomy-and-function/anatomy/2019/neuroanatomy-the-basics-022819https://www.dana.org/article/neuroanatomy-the-basics/, 2019. 38

Adam Sunavsky and Jordan Poppenk. Neuroimaging predictors of creativity in healthy adults. *NeuroImage*, 206:116292, 2020. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2019.116292. URL https://www.sciencedirect.com/science/article/pii/S1053811919308833. 83

Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002. 113, 114

Enzo Tartaglione, Carlo Alberto Barbano, Claudio Berzovini, Marco Calandri, and Marco Grangetto. Unveiling covid-19 from chest x-ray with deep learning: A hurdles race with small data. *International Journal of Environmental Research and Public Health 2020, Vol. 17, Page 6933*, 17(18):6933, 9 2020. ISSN 1660-4601. doi: 10.3390/IJERPH17186933. URL https://www.mdpi.com/1660-4601/17/18/6933/htmhttps://www.mdpi.com/1660-4601/17/18/6933. 15

Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. End: Entangling and disentangling deep representations for bias correction. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 13503–13512, June 2021. ISSN 10636919.

doi: 10.1109/CVPR46437.2021.01330. URL https://ieeexplore.ieee.org/abstract/document/9577751. 66

Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, et al. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. *arXiv preprint arXiv:2210.04620*, 2022. 136

Stefano Teso and Kristian Kersting. Explanatory interactive machine learning. In *Proceedings of the 2nd AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2019. 13

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Multiview Coding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pages 776–794, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58621-8. doi: 10.1007/978-3-030-58621-8_45. tex.ids= tian_contrastive_2020 arXiv: 1906.05849. 30

ME Tipping and CM Bishop. Advances in neural information processing systems. 2000. 38

Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001. 38

Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. In *Domain adaptation in computer vision applications*, pages 37–55. Springer, 2017. 17, 33

Antonio Torralba, Alexei A Efros, et al. Unbiased look at dataset bias. In *CVPR*, page 7. Citeseer, 2011. 17, 33, 52

Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1283–1292, 2017. doi: 10.1109/CVPR.2017.141. 134

Yao-Hung Hubert Tsai, Tianqin Li, Martin Q. Ma, Han Zhao, Kun Zhang, Louis-Philippe Morency, and Ruslan Salakhutdinov. Conditional contrastive learning with kernel. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=AAJLBoGt0XM. 82, 124

J. Tseung. Robbins and cotran pathologic basis of disease: 7th edition. *Pathology*, 37:190, 2005. 127

Aly Valliani and Ameet Soni. Deep residual nets for improved alzheimer's diagnosis. pages 615–615, 08 2017. doi: 10.1145/3107411.3108224. 95

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 89

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pages 268–285. Springer, 2020. 36

Gaël Varoquaux and V. Cheplygina. Machine learning for medical imaging: methodological failures and recommendations for the future. *NPJ Digital Medicine*, 5, 2021. URL https://api.semanticscholar.org/CorpusID:232269760. 95, 96, 108

Christian Wachinger, Anna Rieckmann, Sebastian Pölsterl, Alzheimer's Disease Neuroimaging Initiative, et al. Detect and correct bias in multi-site neuroimaging datasets. *Medical Image Analysis*, 67:101879, 2021. 17, 82

Angelina Wang, Arvind Narayanan, and Olga Russakovsky. REVISE: A tool for measuring and mitigating bias in visual datasets. *European Conference on Computer Vision (ECCV)*, 2020a. 13

Haohan Wang, Zexue He, Zachary L. Lipton, and Eric P. Xing. Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=rJEjjoR9K7. 35, 48, 49, 66, 73

Jiang Wang, Yang song, Thomas Leung, Chuck Rosenberg, Jinbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning Fine-grained Image Similarity with Deep Ranking. In *CVPR*, 2014. 56

Linda Wang and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest radiography images. *arXiv preprint arXiv:2003.09871*, 2020. 39, 101, 104

Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *International Conference on Computer Vision (ICCV)*, October 2019b. 34

Tongzhou Wang and Phillip Isola. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. *arXiv:2005.10242 [cs, stat]*, November 2020. URL http://arxiv.org/abs/2005.10242. arXiv: 2005.10242. 31, 85

Xinshao Wang, Yang Hua, Elyor Kodirov, and Neil M. Robertson. Ranked List Loss for Deep Metric Learning. In *CVPR*, 2019c. 56

Yaoming Wang, Yangzhou Jiang, Jin Li, Bingbing Ni, Wenrui Dai, Chenglin Li, Hongkai Xiong, and Teng Li. Contrastive regression for domain adaptation on gaze estimation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19354–19363, 2022. doi: 10.1109/CVPR52688.2022.01877. 33

Zeyu Wang, Klint Qinami, Ioannis Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020b. 35

Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965. 113

Jason W Wei, Arief A Suriawinata, Louis J Vaickus, Bing Ren, Xiaoying Liu, Mikhail Lisovsky, Naofumi Tomita, Behnaz Abdollahi, Adam S Kim, Dale C Snover, et al. Evaluation of a deep neural network for automated classification of colorectal polyps on histopathologic slides. *JAMA Network Open*, 3(4):e203398–e203398, 2020. 125

Kilian Q Weinberger, John Blitzer, and Lawrence Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2005/hash/a7f592cef8b130a6967a90617db5681b-Abstract.html. 56

Junhao Wen, Elina Thibeau-Sutre, Mauricio Diaz-Melo, Jorge Samper-González, Alexandre Routier, Simona Bottani, Didier Dormont, Stanley Durrleman, Ninon Burgos, and Olivier Colliot. Convolutional neural networks for classification of alzheimer's disease: Overview and reproducible evaluation. *Medical Image Analysis*, 63:101694, 2020. doi: 10.1016/j.media.2020.101694. URL https://doi.org/10.1016/j.media.2020.101694. 39, 95

Ho Yuen Frank Wong, Hiu Yin Sonia Lam, Ambrose Ho-Tung Fong, Siu Ting Leung, Thomas Wing-Yan Chin, Christine Shing Yen Lo, Macy Mei-Sze Lui, Jonan Chun Yin Lee, Keith Wan-Hang Chiu, Tom Chung, et al. Frequency and distribution of chest radiographic findings in covid-19 positive patients. *Radiology*, page 201160, 2020. 107, 189

Mingzhu Wu, Nafise Sadat Moosavi, Andreas Rücklé, and Iryna Gurevych. Improving qa generalization by concurrent modeling of multiple biases. *arXiv preprint arXiv:2010.03338*, 2020. 135

Xi Wu, Matthew Fredrikson, Somesh Jha, and Jeffrey F Naughton. A methodology for formalizing model-inversion attacks. In *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, pages 355–370. IEEE, 2016. 115

Xiaoou Tang, Dacheng Tao, and G. E. Antonio. Texture classification of sars infected region in radiographic image. In *2004 International Conference on Image Processing, 2004. ICIP '04.*, volume 5, pages 2941–2944 Vol. 5, 2004. 39

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018. 113, 115

Qizhe Xie, Zihang Dai, Yulun Du, E. Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. In *NIPS*, 2017. 34

Xie Xuanyang, Gong Yuchang, Wan Shouhong, and Li Xi. Computer aided detection of sars based on radiographs data mining. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 7459–7462, 2005. 39

Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575. IEEE, 2018. 34

Tengfei Xue, Fan Zhang, Leo R Zekelman, Chaoyi Zhang, Yuqian Chen, Suheyla Cetin-Karayumak, Steve Pieper, William M Wells, Yogesh Rathi, Nikos Makris, et al. A novel supervised contrastive regression framework for prediction of neurocognitive measures using multi-site harmonized diffusion mri tractography. *arXiv preprint arXiv:2210.07411*, 2022. 33, 84

Nanyang Ye, Kaican Li, Haoyue Bai, Runpeng Yu, Lanqing Hong, Fengwei Zhou, Zhenguo Li, and Jun Zhu. Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7947–7958, 2022. 15, 16

Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. *arXiv preprint arXiv:2110.06848*, 2021. 124

Baosheng Yu and Dacheng Tao. Deep Metric Learning With Tuplet Margin Loss. In *IEEE ICCV*, pages 6489–6498, 2019. 56

Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. Differentially private model publishing for deep learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 332–349. IEEE, 2019. 113, 115

Yuchen Zeng, Hongxu Chen, and Kangwook Lee. Improving fairness via federated learning. *arXiv preprint arXiv:2110.15545*, 2021. 136

Kaiwen Zha, Peng Cao, Yuzhe Yang, and Dina Katabi. Supervised contrastive regression. *arXiv preprint arXiv:2210.01189*, 2022. 33, 85

Baobao Zhang and Allan Dafoe. Artificial intelligence: American attitudes and trends. *Available at SSRN 3312874*, 2019. 13

Daniel Yue Zhang, Ziyi Kou, and Dong Wang. Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1051–1060, 2020. doi: 10.1109/BigData50022.2020.9378043. 136

Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in Neural Information Processing Systems*, 2018-December:8778–8788, 5 2018. ISSN 10495258. doi: 10.48550/arxiv.1805.07836. URL https://arxiv.org/abs/1805.07836v4. 36, 78

Bowen Zhao, Chen Chen, Qi Ju, and Shutao Xia. Learning debiased models with dynamic gradient alignment and bias-conflicting sample mining. 11 2021. doi: 10.48550/arxiv.2111.13108. URL https://arxiv.org/abs/2111.13108v1. 53

Zi Yue Zu, Meng Di Jiang, Peng Peng Xu, Wen Chen, Qian Qian Ni, Guang Ming Lu, and Long Jiang Zhang. Coronavirus disease 2019 (covid-19): A perspective from china. *Radiology*, page 200490, 2020. 99

Xi-Nian Zuo, Jeffrey S Anderson, Pierre Bellec, Rasmus M Birn, Bharat B Biswal, Janusch Blautzik, John Breitner, Randy L Buckner, Vince D Calhoun, F Xavier Castellanos, et al. An open science resource for establishing reliability and reproducibility in functional connectomics. *Scientific data*, 1(1):1–13, 2014. 83

Part VI

Appendices

Appendix A

Additional Theoretical Results for Chapter 4

A.1 Complete derivations for Section 4.1

In this section, we present the complete analytical derivation for the equations found in Sec. 4.1. All of the presented derivations are based on the smooth max approximation with the LogSumExp (LSE) operator:

$$\max(x_1, x_2, ..., x_3) \approx \log(\sum_i \exp(x_i)) \tag{A.1.1}$$

A.1.1 Full derivation of ϵ-InfoNCE (4.1.2)

We consider Eq. 4.1.2 and we obtain:

$$\arg\min_f \max(-\epsilon, \{s_j^- - s^+\}_{j=1,...,N}) \approx \arg\min_f \underbrace{\left[-\log\left(\frac{\exp(s^+)}{\exp(s^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)\right]}_{\epsilon - InfoNCE} \tag{A.1.2}$$

Starting from the left-hand side, we have:

$$
\begin{aligned}
\max(-\epsilon, \{s_j^- - s^+\}_{j=1,\dots,N}) &\approx \log\left(\exp(-\epsilon) + \sum_j \exp(s_j^- - s^+)\right) \\
&= \log\left(\exp(-\epsilon) + \exp(-s^+)\sum_j \exp(s_j^-)\right) \\
&= \log\left(\exp(-s^+)\left(\exp(s^+ - \epsilon) + \sum_j \exp(s_j^-)\right)\right) \\
&= \log\exp(-s^+) + \log\left(\exp(s^+ - \epsilon) + \sum_j \exp(s_j^-)\right) \\
&= \underbrace{-\log\left(\frac{\exp(s^+)}{\exp(s^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)}_{\epsilon - InfoNCE}
\end{aligned}
\tag{A.1.3}
$$

A.1.2 Multiple positive extension

Extending Eq. 4.1.2 to multiple positives can be done in different ways. Here, we list four possible choices. Empirically, we found that solution c) gave the best results and is the most convenient to implement for efficiency reasons.

$$
\begin{aligned}
&a)\ \max(-\epsilon, \{s_j^- - s_i^+\}_{\substack{i=1,\dots,P \\ j=1,\dots,N}}) = -\log\left(\frac{\exp(\sum_i s_i^+)}{\exp(\sum_i s_i^+ - \epsilon) + (\sum_j \exp(s_j^-))(\sum_i \exp(\sum_{t\neq i} s_t^+))}\right) \\
&b)\ \sum_j \max(-\epsilon, \{s_j^- - s_i^+\}_{i=1,\dots,P}) = -\sum_j \log\left(\frac{\exp(\sum_i s_i^+)}{\exp(\sum_i s_i^+ - \epsilon) + \exp(s_j^-)(\sum_i \exp(\sum_{t\neq i} s_t^+))}\right) \\
&c)\ \sum_i \max(-\epsilon, \{s_j^- - s_i^+\}_{j=1,\dots,N}) = -\sum_i \log\left(\frac{\exp(s^+)}{\exp(s^+ - \epsilon) + \sum_j \exp(s_j^-)}\right) \\
&d)\ \sum_i \sum_j \max(-\epsilon, s_j^- - s_i^+) = -\sum_i \sum_j \log\left(\frac{\exp(s^+)}{\exp(s^+ - \epsilon) + \exp(s_j^-)}\right)
\end{aligned}
\tag{A.1.4}
$$

A.1.3 Full derivation of ϵ-SupInfoNCE (4.1.4)

The computations are very similar to Eq. A.1.3. We obtain:

$$
\arg\min_f \sum_i \max(-\epsilon, \{s_j^- - s_i^+\}_{j=1,\dots,N}) \approx \arg\min_f \left[\sum_i \log\left(\exp(-\epsilon) + \sum_j \exp(s_j^- - s_i^+)\right)\right]
\tag{A.1.5}
$$

Starting from the left-hand side, we have:

$$
\begin{aligned}
\sum_i \max(-\epsilon, \{s_j^- - s_i^+\}_{j=1,\dots,N}) &\approx \sum_i \log\left(\exp(-\epsilon) + \sum_j \exp(s_j^- - s_i^+)\right) \\
&= \sum_i \log\left(\exp(-\epsilon) + \frac{\sum_j \exp(s_j^-)}{\exp(s_i^+)}\right) \\
&= \sum_i \log\left(\frac{\exp(s_i^+ - \epsilon) + \sum_j \exp(s_j^-)}{\exp(s_i^+)}\right) \\
&= \underbrace{-\sum_i \log\left(\frac{\exp(s_i^+)}{\exp(s_i^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)}_{\epsilon - SupInfoNCE}
\end{aligned}
\tag{A.1.6}
$$

A.1.4 Full derivation of ϵ-SupCon (4.1.5)

We extend Eq. 4.1.4 by adding the non contrastive conditions:

$$
s_j^- - s_i^+ \leq -\epsilon \quad \forall i, j \quad \text{and} \quad s_t^+ - s_i^+ \leq 0 \quad \forall i, t \neq i \tag{A.1.7}
$$

and we show

$$
\frac{1}{P}\sum_i \max(0, \{s_j^- - s_i^+ + \epsilon\}_j, \{s_t^+ - s_i^+\}_{t \neq i}) \approx \epsilon \underbrace{-\frac{1}{P}\sum_i \log\left(\frac{\exp(s_i^+)}{\sum_t \exp(s_i^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)}_{\epsilon - SupCon} \tag{A.1.8}
$$

Starting from the left-hand side, we have:

$$
\begin{aligned}
&\frac{1}{P}\sum_i \max(0, \{s_j^- - s_i^+ + \epsilon\}_{j=1,\dots,N}, \{s_t^+ - s_i^+\}_{t\neq i}) \\
&\qquad \approx \frac{1}{P}\sum_i \log\left(1 + \sum_j \exp(s_j^- - s_i^+ + \epsilon) + \sum_{t\neq i} \exp(s_t^+ - s_i^+)\right) \\
&\qquad = \frac{1}{P}\sum_i \log\left(1 + \frac{\sum_j \exp(s_j^-)}{\exp(s_i^+ - \epsilon)} + \frac{\sum_{t\neq i} \exp(s_t^+)}{\exp(s_i^+)}\right) \\
&\qquad = \frac{1}{P}\sum_i \log\left(\frac{\exp(s_i^+ - \epsilon) + \sum_j \exp(s_j^-) + \sum_{t\neq i} \exp(s_t^+ - \epsilon)}{\exp(s_i^+ - \epsilon)}\right) \\
&\qquad = -\frac{1}{P}\sum_i \log\left(\frac{\exp(s_i^+ - \epsilon)}{\sum_t \exp(s_t^+ - \epsilon) + \sum_j \exp(s_j^-)}\right) \\
&\qquad = \epsilon \underbrace{-\frac{1}{P}\sum_i \log\left(\frac{\exp(s_i^+)}{\sum_t \exp(s_t^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)}_{\epsilon - SupCon}
\end{aligned}
\tag{A.1.9}
$$

A.1.5 Full derivation of $\mathcal{L}_{in}^{sup}$ (4.1.7)

Here we show that:

$$
\max(s_j^-) < \max(s_i^+) \approx \underbrace{-\log\left(\sum_i \frac{\exp(s_i^+)}{\sum_t \exp(s_t^+) + \sum_j \exp(s_j^-))}\right)}_{\mathcal{L}_{in}^{sup}} \tag{A.1.10}
$$

Starting from the left-hand side, and given that:

$$
\max(s_j^-) < \max(s_i^+) \approx \log(\sum_j \exp(s_j^-)) - \log(\sum_i \exp(s_i^+)) < 0
$$

we have:

$$
\begin{aligned}
&\max(0, \log(\sum_j \exp(s_j^-)) - \log(\sum_i \exp(s_i^+))) \\
&\qquad \approx \log\left(1 + \exp\left(\log(\sum_j \exp(s_j^-)) - \log(\sum_i \exp(s_i^+))\right)\right) \\
&\qquad = \log\left(1 + \exp\left(\log\left(\frac{\sum_j \exp(s_j^-)}{\sum_i \exp(s_i^+)}\right)\right)\right) \\
&\qquad = \log\left(1 + \frac{\sum_j \exp(s_j^-))}{\sum_i \exp(s_i^+)}\right) \\
&\qquad = \log\left(\frac{\sum_i \exp(s_i^+) + \sum_j \exp(s_j^-))}{\sum_i \exp(s_i^+)}\right) \\
&\qquad = \underbrace{-\log\left(\sum_i \frac{\exp(s_i^+)}{\sum_t \exp(s_t^+) + \sum_j \exp(s_j^-))}\right)}_{\mathcal{L}_{in}^{sup}}
\end{aligned}
\tag{A.1.11}
$$

A.1.6 Full derivation of Eq.A.1.4-a

$$
\arg\min_f \max(-\epsilon, \{s_j^- - s_i^+\}_{\substack{i=1,..,P \\ j=1,..,N}}) \approx \arg\min_f \log\left(\exp(-\epsilon) + \sum_i \sum_j \exp(s_j^- - s_i^+)\right)
\tag{A.1.12}
$$

We have:

$$
\begin{aligned}
&\mathcal{L} = \log\left(\exp(-\epsilon) + \sum_i^P \left(\frac{\sum_j \exp(s_j^-)}{\exp(s_i^+))}\right)\right) = \log\left(\sum_i^P \left(\frac{\exp(-\epsilon)}{P} + \frac{\sum_j \exp(s_j^-)}{\exp(s_i^+))}\right)\right) \\
&= \log\left(\sum_i^P \left(\frac{\exp(s_i^+ - \epsilon) + P(\sum_j \exp(s_j^-))}{P(\exp(s_i^+)))}\right)\right) = \log\left(\sum_i \left(\frac{\frac{1}{P}\exp(s_i^+ - \epsilon)) + \sum_j \exp(s_j^-)}{\exp(s_i^+)}\right)\right) \\
&= \log\left(\frac{\sum_i \left[\frac{1}{P}\exp(s_i^+ - \epsilon)(\prod_{t\neq i}\exp(s_t^+)) + (\sum_j \exp(s_j^-))(\prod_{t\neq i}\exp(s_t^+))\right]}{\prod_i \exp(s_i^+)}\right) \\
&= \log\left(\frac{\exp(-\epsilon)\prod_i \exp(s_i^+) + (\sum_j \exp(s_j^-))(\sum_i \prod_{t\neq i}\exp(s_t^+))}{\prod_i \exp(s_i^+)}\right) \\
&= -\log\left(\frac{\exp(\sum_i s_i^+)}{\exp(\sum_i s_i^+ - \epsilon) + (\sum_j \exp(s_j^-))(\sum_i \exp(\sum_{t\neq i} s_t^+))}\right)
\end{aligned}
\tag{A.1.13}
$$

A.2 Boundness of the ϵ-margin

In this section, we give insights on how an optimal value of ϵ can be estimated. First of all, it is easy to show that ϵ is bounded and cannot grow to infinity. This is the case in which the two samples are aligned at opposite poles of the hypersphere. We can conclude that, in general, ϵ will be less than 2. If we also take into account the temperature τ, when $\epsilon \leq 2/\tau$. This is always true, however, a stricter upper bound can be found if we consider the geometric properties of the latent space. For example, Graf et al. (2021) show that when the SupCon loss converges to its minimum value, then the representations of the different classes are aligned on a regular simplex. This property could be used to compute a precise upper bound of the ϵ margin, depending on the number of classes in the dataset. We leave further analysis on this matter as future work.

Appendix B

Experimental Setup for Chapter 4

All of our experiments were run using PyTorch 1.10.0. We used a cluster with 4 NVIDIA V100 GPUs and a cluster of 8 NVIDIA A40 GPUs. For consistency, when training with constrastive losses we use a temperature value $\tau = 0.1$ across all of our experiments.

B.1 Generic vision datasets

B.1.1 CIFAR-10 and CIFAR-100

We use the original setup from SupCon (Khosla et al., 2020), employing a ResNet-50, large batch size (1024), learning rate of 0.5, temperature of 0.1 and multiview augmentation, for CIFAR-10 and CIFAR-100. We use SGD as optimizer with a momentum of 0.9, and train for 1000 epochs. Learning rate is decayed with a cosine policy with warmup from 0.01, with 10 warmup epochs.

B.1.2 ImageNet-100

For ImageNet-100 we employ the ResNet50 architecture (He et al., 2016). We use SGD as optimizer, with a weight decay of 10^{-4} and momentum of 0.9, with an initial learning rate of 0.1 and a cosine decay policy. We train for 500 epochs. We train for 100 epochs, and we decay the learning rate by a factor of 0.1 every 30 epochs.

B.2 Biased Datasets

When employing our debiasing term, we find that scaling the ϵ-SupInfoNCE loss by a small factor α (≤ 1) and using λ closer to 1, is stabler than using values of $\lambda >> 1$ (as done for EnD) and tends to produce better results. For biased datasets, we do not make use of the projection head used in Chen et al. (2020); Khosla et al. (2020). For this reason, we also avoid the aggressive augmentation usually employed by contrastive methods (more on this in Sec. C.3). Furthermore, as also done by Hong and Yang (2021), we also experimented with a small contribution of the cross entropy loss for training the model end-to-end; however, we did not find any benefit in doing so, compared to training a linear classifier separately.

B.2.1 Biased-MNIST

We emply the network architecture *SimpleConvNet* proposed by Bahng et al. (2020), consisting of four convolutional layers with 7×7 kernels. We use the Adam optimizer with a learning rate of 0.001, a weight decay of 10^{-4} and a batch size of 256. We decay the learning rate by a factor of 0.1 at 1/3 and 2/3 of the epochs (26 and 53). We train for 80 epochs.

Hyperparameters configuration The hyperparameters are reported in Tab. B.2.1.

Table B.2.1: Biased-MNIST hyperparameters

	Corr (ρ)			
	0.999	0.997	0.995	0.990
α	0.01	0.01	0.03	0.03
λ	0.5	0.5	0.75	0.5
ϵ	0	0	0.5	0.5

B.2.2 Corrupted CIFAR-10

For this dataset we employ the ResNet-18 architecture. We use the Adam optimizer, with an initial learning rate of 0.001, a weight decay of 0.0001. The other Adam parameters are the pytorch default ones ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$). We train for 200 epochs with a batch size of 256. We decay the learning rate using a cosine annealing policy.

Hyperparameters configuration: Table B.2.2 shows the hyperparameters for the results reported in Tab. 4.3.4 of the main paper.

Table B.2.2: Corrupted CIFAR-10 hyperparameters

	Ratio (%)			
	0.5	1.0	2.0	5.0
α	0.1	0.1	0.1	0.1
λ	1.0	1.0	1.0	1.0
ϵ	0.1	0.25	0.5	0.25

B.2.3 bFFHQ

Following Lee et al. (2021),we use the ResNet-18 architecture. We use the Adam optimizer, with an initial learning rate of 0.0001, and train for 100 epochs. For this experiment, we set $\alpha = 0.1$, $\epsilon = 0.25$ and $\lambda = 1.5$. Differently from Lee et al. (2021) we use a batch size of 256 (vs 64) as contrastive losses benefit more from larger batch sizes (Chen et al., 2020; Khosla et al., 2020). Additionally, we also use a weight decay of 10^{-4}, rather than 0. These changes do not provide advantages to

the debiasing task: we obtain an accuracy of 54.8% without FairKL, which is in line with the 56.87% reported for the vanilla model.

Appendix C

Additional Empirical Results for Chapter 4

In this section, we present some additional experiments we conducted, for a more in depth analysis of our proposed framework.

C.1 Complete results for common vision datasets

In Table C.1.1 we report the results on CIFAR-10, CIFAR-100 and ImageNet-100 for different values of ϵ. In Table C.1.2 the full comparison between ϵ-SupCon and ϵ-SupInfoNCE on ImageNet-100 is presented. Our proposed ϵ-SupInfoNCE outperforms SupCon in all datasets for all the ϵ values, reaching the best results. Furthermore, on ImageNet-100, we observe that the lowest accuracy obtained by ϵ-SupInfoNCE (83.02%) is still higher than the best accuracy obtained by ϵ-SupCon (82.83%) on the same dataset, even though ϵ-SupCon is always higher than SupCon. In terms of accuracy, the results in Tab. C.1.2 show that $SupCon \leq \epsilon - SupCon \leq \epsilon - SupInfoNCE$.

Table C.1.1: Complete results for common vision datasets, for different values of ϵ, in terms of top-1 accuracy (%). With every value of ϵ we obtain better results than SupCon (and CE) on the same dataset.

Dataset	CE	SupCon	ϵ-SupInfoNCE		
			$\epsilon = 0.1$	$\epsilon = 0.25$	$\epsilon = 0.5$
CIFAR-10	94.73±0.18	95.64±0.02	95.93±0.02	**96.14**±0.01	<u>95.95</u>±0.12
CIFAR-100	73.43±0.08	75.41±0.19	75.85±0.07	**76.04**±0.01	<u>75.99</u>±0.06
ImageNet-100	82.1±0.59	81.99±0.08	<u>83.25</u>±0.39	83.02±0.41	**83.3**±0.06

C.2 Analysis of ϵ-SupCon for debiasing

We perform a more in-depth analysis of the debiasing capabilities of ϵ-SupInfoNCE and ϵ-SupCon. In Sec. 4.3 of the main text, we hypothesize that the non-constrastive condition of Eq. 4.1.5

$$s_i^+ - s_j^+ \leq 0 \quad \forall i, t \neq i$$

Table C.1.2: Complete comparison of ϵ-SupInfoNCE and ϵ-SupCon on ImageNet-100 in terms of top-1 accuracy (%). The results of ϵ-SupInfoNCE are higher than any results of ϵ-SupCon.

Loss	$\epsilon = 0.1$	$\epsilon = 0.25$	$\epsilon = 0.5$
ϵ-SupInfoNCE	**83.25**$\pm$0.39	**83.02**$\pm$0.41	**83.3**$\pm$0.06
ϵ-SupCon	82.83$\pm$0.11	82.54$\pm$0.09	82.77$\pm$0.14

might actually be the reason for the loss of accuracy in ϵ-SupCon when compared to ϵ-InfoNCE, as shown on the analysis on Biased-MNIST in Fig. 4.3.1 of the main text.
In this section, we provide more empirical insights supporting this hypothesis. We plot the similarity of bias-aligned samples ($s^{+,b}$) and bias-conflicting samples ($s^{+,b'}$) during training, to understand how they are affected. Fig. C.2.1 shows the bivariate histogram of the similarities obtained with the two loss functions, at different training epochs and values of ϵ, on Biased-MNIST, with a training ρ of 0.999. Focusing on the bias-aligned samples (first two columns), we observe that, in both cases, most values are close to 1. However, while this is true for most of the shown histograms, the presence of the non-contrastive condition of Eq. 4.1.5 produces a much tighter distribution for ϵ-SupCon, when compared to ϵ-SupInfoNCE. In fact, with ϵ-SupInfoNCE we obtain significantly more bias-aligned samples with a similarity smaller than 1. This is especially evident if we focus on the last training epochs.
More interestingly, if we focus on the bias-conflicting similarities (last two columns), we can also notice how, on average, the distribution of similarities of bias-conflicting samples for ϵ-SupCon tends to be more concentrated around the value of 0. This means that bias-conflicting samples have dissimilar representations even if they are both positives and should be mapped to the same point in the representation space. The effect of the bias is thus still quite important and it has not been discarded. On the other hand, with ϵ-SupInfoNCE, we obtain a much more spread distribution, especially as the number of training epochs increases. This means that a higher number of bias-conflicting samples have a greater similarity (in the representation space), leading to more robust representations.
Clearly, ϵ-SupCon focuses more on bias-aligned samples as most of them have a similarity close to 1, whereas most of the bias-conflicting samples have a similarity close to 0. With our proposed loss ϵ-SupInfoNCE, this behavior is less pronounced, as the lack of the non-contrastive condition leads the model to be less focused on bias-aligned samples. This could explain why ϵ-SupInfoNCE can perform better than ϵ-SupCon in highly biased settings.

C.3 Training with a projection head

In Tab. C.3.1 we show the results on Corrupted CIFAR-10 with and without using a projection head. When employing a projection head, the loss term and the regularization are applied on the projected and original space respectively, and the final classification is performed in the original latent space before the projection head. We conjecture that the loss in accuracy is likely due i.) to the absence of the aggressive augmentation typically used for generating multiviews in contrastive setups, which

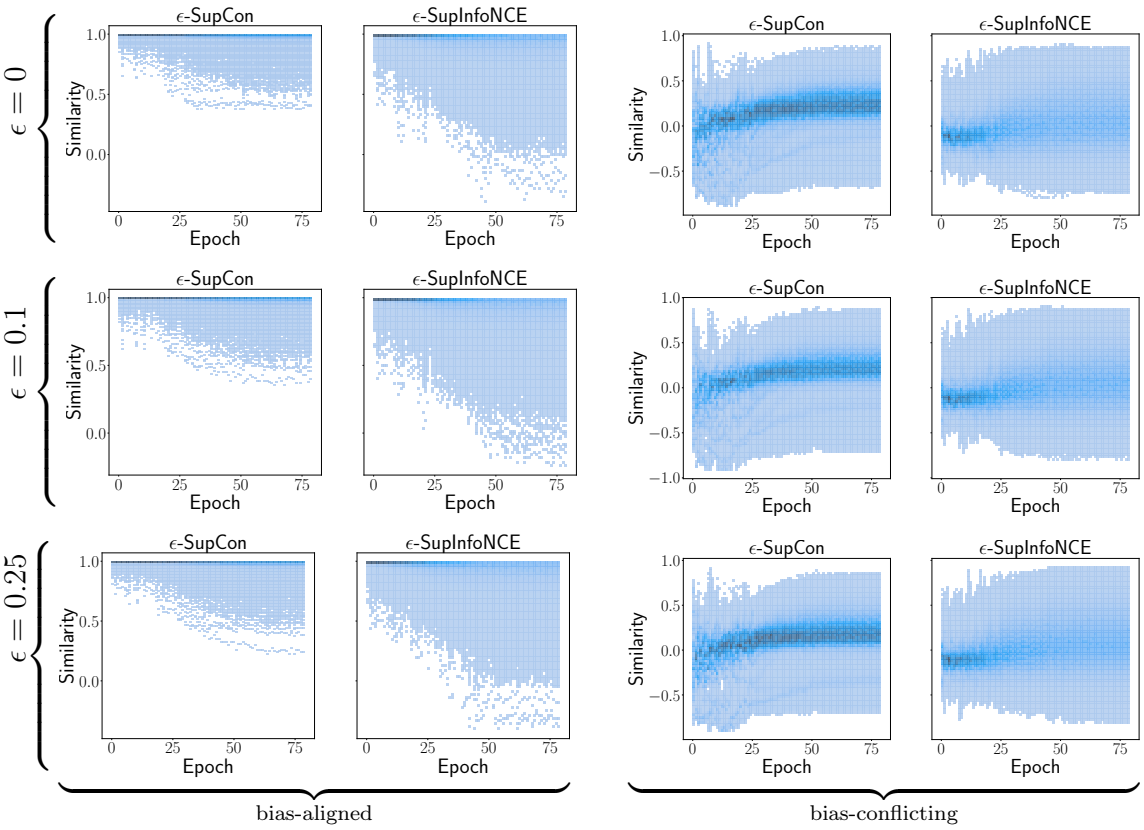


Figure C.2.1: (*first and second columns*) Distribution of positive bias-aligned similarities $s^{+,b}$. Here ϵ-SupCon tends to produce a much tighter distribution, with similarities close to 1; (*third and fourth columns*) Distribution of positive bias-conflicting similarities $s^{+,b'}$. Here ϵ-SupInfoNCE, even if marginally, is able to increase the number of similar bias-conflicting samples. ϵ-SupCon focuses more on bias-aligned samples, resulting in more biased representations. With ϵ-SupInfoNCE, this behavior is less pronounced, as the lack of the non-contrastive condition leads to be less focused on bias-aligned samples and more focused on the bias-conflicting ones. We hypothesize that this is the reason ϵ-SupInfoNCE appears to obtain better results than ϵ-SupCon in more biased datasets.

are probably attenuated by the projection head ii.) minimizing ϵ-SupInfoNCE and the FairKL term on the same latent space rather than two different ones, could be more beneficial for the optimization process.

Table C.3.1: Accuracy on Corrupted CIFAR-10 with and without projection head

	Ratio (%)			
	0.5	1.0	2.0	5.0
With Head	30.85±0.19	32.75±0.57	37.95±0.14	45.67±0.66
Without Head	**33.33**±0.38	**36.53**±0.38	**41.45**±0.42	**50.73**±0.90

C.4 Ablation study of debiasing regularization

We perform an ablation study of our debiasing regularization on Corrupted CIFAR-10 and on bFFHQ. We test two variants of the regularization term:

1. Only with the conditions on the mean of the representations μ_+ and μ_- (Eq. 4.2.2), similarly to EnD, but with the differences in formulations of Sec. 4.2.2;

2. Full FairKL debiasing term of Eq. 4.2.3.

The results are shown in Tab. C.4.1. As it can be easily observed, employing the full regularization constraint consistently results in better accuracy.

Table C.4.1: Ablation study of $\mathcal{R}^{FairKL}$ on Corrupted CIFAR-10 and bFFHQ

	Corrupted CIFAR-10 Ratio (%)				bFFHQ Ratio (%)
	0.5	1.0	2.0	5.0	0.5
FairKL (mean)	32.37±1.72	35.65±0.75	39.94±0.50	50.25±0.16	60.55±1.05
FairKL (full)	**33.33**±0.38	**36.53**±0.38	**41.45**±0.42	**50.73**±0.90	**63.70**±0.90

C.5 Importance of the regularization weight

We conduct an analysis on the importance and stability of the weights α and λ of Eq. 4.2.4. We perform multiple experiments selecting $\alpha \in \{0.01, 0.1, 1.0\}$. For simplicity, we fix $\epsilon = 0$, and we report the accuracy scored on the Biased-MNIST test. The results are show in Tab. C.5.1. There seems to be a correlation between the value of α and the strength of the bias: for stronger biases it is better to give more importance to the regularization term rather than the target loss function. Additionaly, we also find that α depends on the complexity of the dataset: for example on Corrupted-CIFAR10 and bFFHQ we use $\alpha = 0.1$, for 9-Class ImageNet we use $\alpha = 0.5$.

Table C.5.1: Importance of the weights α and λ.

	$\lambda = 0.5$			$\lambda = 1$		
Corr.\ α	0.01	0.1	1.0	0.01	0.1	1.0
0.999	**89.55**$\pm$1.43	31.63$\pm$2.30	38.38$\pm$1.26	<u>84.98</u>$\pm$2.29	43.21$\pm$10.08	31.84$\pm$4.31
0.997	**94.08**$\pm$0.10	82.11$\pm$2.48	78.91$\pm$2.48	<u>91.08</u>$\pm$0.82	84.98$\pm$10.23	79.03$\pm$3.15
0.995	<u>92.42</u>$\pm$3.76	90.60$\pm$3.35	86.63$\pm$2.26	88.39$\pm$5.00	**93.97**$\pm$1.83	88.27$\pm$1.72
0.99	95.00$\pm$0.21	<u>96.60</u>$\pm$0.17	93.75$\pm$0.25	90.72$\pm$0.51	**97.13**$\pm$0.38	94.74$\pm$0.40

Appendix D

Additional Empirical Results for Chapter 5

In this section we provide some additional results about our debiasing technique, mainly focusing on the worst-case scenarios described in Section 5.1.4.

D.1 Debiasing on an unbiased dataset

Here we show that the supervised EnD regularization does not deteriorate the final results if applied to a training set that is not biased. Table D.1.1 shows the results of training with EnD on Biased-MNIST with $\rho = 0.1$. In this setting, applying the regularization term is not harmful toward obtaining good generalization: this is because in a supervised setting we still have access to the correct color labels, thus we do not perform disentanglement over any useful features for the network. This is a trivial result, however, with this demonstrated we can now focus on an unbiased training set in the unsupervised case.

ρ	Vanilla	EnD
0.1	99.21	$99.24_{\pm 0.05}$

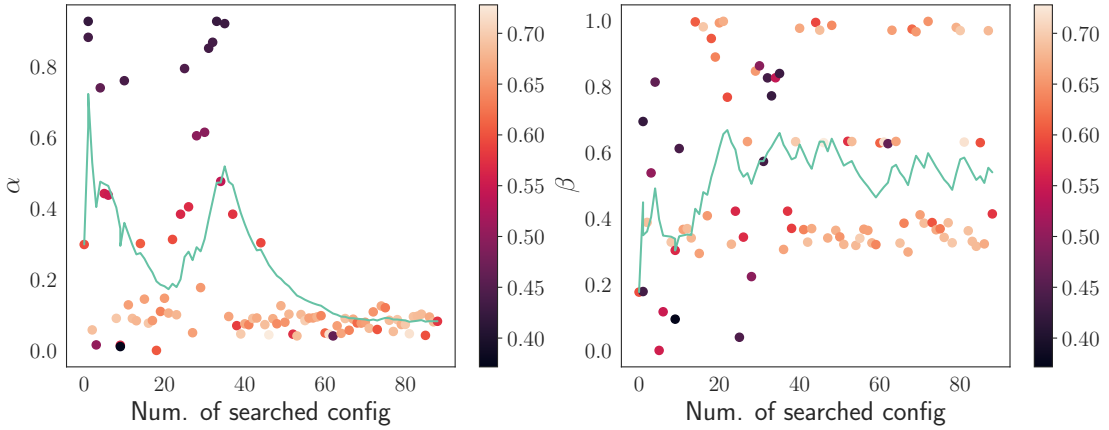
Table D.1.1: Debiasing on an unbiased training set ($\rho = 0.1$)

D.2 Debiasing with wrong pseudo-labels

We now assume that the pseudo-labels we compute are not representative of the true bias attributes. Using Biased-MNIST as case study, we identify that the worst-case

ρ	Vanilla	EnD (target)	EnD (random)
0.995	$72.10_{\pm 1.90}$	$72.25_{\pm 0.56}$	$66.68_{\pm 0.35}$

Table D.2.1: Debiasing on incorrect bias labels. Target means that target labels are also used as bias label (i.e. $t_i = b_i$, worst case), random means that bias labels are assigned randomly.



(a) (b)

Figure D.2.1: Evolution of (a) α and (b) β versus the number of searched configs during the hyperparameters optimization with incorrect pseudo-labels. We can observe how the optimization process drives α towards 0, while β does not seem to be relevant. The point color indicates the accuracy on the unbiased test set, while the line shows the trend as an exponentially weighted moving average computed with a smoothing factor of 0.1.

scenario for the pseudo-labeling step corresponds to using a completely unbiased dataset (i.e. $\rho = 0.1$) for training the biased encoder. Taking into account the results shown in Section D.1, performing the pseudo-labeling step in this setting will most likely result in pseudo-labels corresponding to the actual target class rather than the background color. We emulate this event by setting the bias label b_i equal to the target label t_i for every sample in the dataset, and then we the apply EnD algorithm. To test this worst-case with EnD, we choose $\rho = 0.995$ as it provides a way for the final accuracy to both decrease or increase with respect to a vanilla model. The results are reported in Table D.2.1 and noted as *target*. Even in this case, we are able to retain the baseline performances, although we do not obtain any significant improvement. This is thanks to the hyperparameter optimization policy that we employ (recall that we assume an unbiased validation set - even if small - is available).

Figure D.2.1 visualizes the evolution of the hyperparameters α and β while searching for possible configurations. In this setting, α represents the most dangerous term, as it enforces decorrelation among samples with the same class, conflicting with the cross-entropy term. However, the optimization process drives α towards 0, making it effectively non-influent on the loss term. On the other hand, the entangling term β does not bring any contribution to the learning process: it is, in fact, useless as there is full alignment between target and bias labels, hence there are no positive bias-conflicting samples.

A possible scenario in which β would not have null influence is if we do not impose $t_i = b_i$. We explore this extreme setting by assigning a random value to b_i for every sample i. The results are reported as *random* in Table D.2.1. In this case, it is possible to observe a drop in performance with respect to the baseline. However, we argue that random pseudo-labels would be the result of poor representations due to

possibly underfitting models or lack of sufficient training data - which, in a practical setting, would be a more pressing issue.

Appendix E

Appendix for Covid-19 Detection

E.1 Complete results for direct diagnosis

Here we report the complete results of all the experiments with performed for direct classification of Covid-19 on the CORDA-CDSS dataset. With respect to Sec. 7.1, we report additional evaluation metrics. They are summarized in Tab. E.1.1. Tables E.1.2 and E.1.3 shows the results for ResNet-18; Tables E.1.4 and E.1.5 show the result for ResNet-50 and DenseNet-121.

Table E.1.1: Evaluated metrics.

Metric	**Definition**	**Meaning**
TP		True positives
TN		True negatives
FP		False positives
FN		False negatives
TPR		*same as Sensitivity*
FPR		*same as Specificity*
AUC		Aggregate measure of performance across all possible classification thresholds
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Fraction of correct predictions made by the model
Sensitivity	$\frac{TP}{TP+FN}$	How many positive samples were correctly identified (also called *recall*)
Specificity	$\frac{TN}{TN+FP}$	How many true negatives were correctly identified
BA	$\frac{TPR+TNR}{2}$	Balanced accuracy, accounts for imbalanced sets
F-Score	$\frac{2TP}{2TP+FP+FN}$	Harmonic mean between precision and recall, where precision is defined as $\frac{TP}{TP+FP}$. A value of 1 means perfect precision and recall
DOR	$\frac{TPR \cdot TNR}{(1-TPR)(1-TNR)}$	Effectiveness of a diagnostic test (Glas et al., 2003)

Pre-trained encoder	Training dataset	Test dataset	Sensitivity	Specificity	F-Score	Accuracy	BA	AUC	DOR
none	AC	A	0.56	0.42	0.60	0.51	0.49	0.52	0.91
		AB	0.56	0.22	0.15	0.26	0.39	0.33	0.36
		AC	0.56	0.96	0.49	0.95	0.76	0.95	34.23
		AD	0.52	0.48	0.58	0.51	0.50	0.53	1.00
	A	A	0.56	0.58	0.63	0.56	0.57	0.59	1.71
		AB	0.56	0.37	0.18	0.39	0.46	0.43	0.74
		AC	0.56	0.38	0.08	0.39	0.47	0.46	0.76
		AD	0.56	0.58	0.63	0.57	0.57	0.59	1.76
	AD	A	0.58	0.64	0.66	0.60	0.61	0.63	2.48
		AB	0.58	0.63	0.27	0.63	0.61	0.63	2.37
		AC	0.58	0.54	0.11	0.54	0.56	0.58	1.62
		AD	0.57	0.66	0.66	0.60	0.61	0.64	2.57
	D	A	0.91	0.11	0.77	0.64	0.51	0.54	1.28
		AB	0.91	0.66	0.41	0.69	0.78	0.87	19.56
		AC	0.91	0.11	0.09	0.14	0.51	0.45	1.22
		AD	0.91	0.18	0.78	0.67	0.55	0.58	2.22
	AB	A	0.88	0.18	0.77	0.64	0.53	0.58	1.55
		AB	0.88	0.94	0.76	0.93	0.91	0.97	112.93
		AC	0.88	0.14	0.09	0.17	0.51	0.42	1.14
		AD	0.87	0.20	0.77	0.65	0.54	0.60	1.67

Table E.1.2: Results obtained training a ResNet-18 with no pre-train. Dataset naming follows Table 7.1.1

Pre-trained encoder	Training dataset	Test dataset	Sensitivity	Specificity	F-Score	Accuracy	BA	AUC	DOR
C	AC	A	0.68	0.44	0.69	0.60	0.56	0.61	1.68
		AB	0.68	0.22	0.18	0.27	0.45	0.49	0.59
		AC	0.68	0.90	0.37	0.89	0.79	0.90	19.82
		AD	0.67	0.50	0.70	0.61	0.58	0.63	2.03
	A	A	0.54	0.80	0.66	0.63	0.67	0.72	4.78
		AB	0.54	0.31	0.16	0.34	0.43	0.48	0.54
		AC	0.54	0.55	0.10	0.55	0.55	0.61	1.48
		AD	0.57	0.76	0.67	0.63	0.67	0.72	4.20
	AD	A	0.70	0.49	0.72	0.63	0.59	0.67	2.23
		AB	0.70	0.30	0.20	0.34	0.50	0.59	0.98
		AC	0.70	0.37	0.10	0.39	0.53	0.61	1.37
		AD	0.71	0.52	0.73	0.65	0.61	0.70	2.65
	D	A	0.94	0.09	0.79	0.66	0.52	0.57	1.66
		AB	0.94	0.61	0.39	0.65	0.78	0.92	26.24
		AC	0.94	0.08	0.09	0.12	0.51	0.58	1.50
		AD	0.95	0.14	0.80	0.68	0.54	0.62	3.09
	AB	A	0.82	0.38	0.77	0.67	0.60	0.63	2.81
		AB	0.82	0.95	0.75	0.94	0.89	0.97	89.14
		AC	0.82	0.30	0.10	0.32	0.56	0.59	1.98
		AD	0.83	0.38	0.78	0.68	0.60	0.64	2.99
B	AC	A	0.86	0.31	0.78	0.67	0.58	0.60	2.67
		AB	0.86	0.29	0.24	0.36	0.58	0.48	2.47
		AC	0.86	0.95	0.61	0.95	0.90	0.97	122.64
		AD	0.82	0.38	0.77	0.67	0.60	0.61	2.79
	A	A	0.54	0.58	0.62	0.56	0.56	0.67	1.64
		AB	0.54	0.37	0.17	0.39	0.46	0.49	0.70
		AC	0.54	0.73	0.15	0.72	0.64	0.72	3.21
		AD	0.56	0.62	0.64	0.58	0.59	0.70	2.08
	AD	A	0.71	0.49	0.72	0.64	0.60	0.67	2.35
		AB	0.71	0.25	0.20	0.31	0.48	0.51	0.83
		AC	0.71	0.47	0.11	0.48	0.59	0.64	2.16
		AD	0.73	0.52	0.74	0.66	0.62	0.70	2.93
	D	A	0.91	0.20	0.79	0.67	0.56	0.61	2.56
		AB	0.91	0.70	0.44	0.73	0.81	0.89	24.38
		AC	0.91	0.15	0.09	0.19	0.53	0.55	1.83
		AD	0.92	0.28	0.81	0.71	0.60	0.66	4.47
	AB	A	0.88	0.24	0.78	0.67	0.56	0.66	2.32
		AB	0.88	0.94	0.77	0.94	0.91	0.97	122.67
		AC	0.88	0.24	0.10	0.27	0.56	0.67	2.26
		AD	0.88	0.26	0.78	0.67	0.57	0.68	2.58

Table E.1.3: Results obtained training a ResNet-18 with a pre-trained encoder. Dataset naming follows Table 7.1.1

Pre-trained encoder	Training dataset	Test dataset	Sensitivity	Specificity	F-Score	Accuracy	BA	AUC	DOR
C	AC	A	0.74	0.49	0.74	0.66	0.62	0.65	2.79
		AB	0.74	0.40	0.24	0.44	0.57	0.64	1.92
		AC	0.74	0.92	0.43	0.91	0.83	0.93	31.76
		AD	0.70	0.54	0.73	0.65	0.62	0.66	2.74
	A	A	0.61	0.71	0.70	0.64	0.66	0.67	3.87
		AB	0.61	0.40	0.20	0.43	0.51	0.53	1.06
		AC	0.61	0.58	0.12	0.58	0.60	0.63	2.20
		AD	0.62	0.74	0.71	0.66	0.68	0.69	4.64
	AD	A	0.53	0.64	0.62	0.57	0.59	0.64	2.07
		AB	0.53	0.56	0.22	0.56	0.55	0.58	1.47
		AC	0.53	0.57	0.10	0.57	0.55	0.58	1.53
		AD	0.55	0.68	0.64	0.59	0.61	0.66	2.60
	D	A	0.97	0.04	0.79	0.66	0.51	0.57	1.35
		AB	0.97	0.45	0.32	0.51	0.71	0.89	23.29
		AC	0.97	0.09	0.09	0.13	0.53	0.56	2.91
		AD	0.97	0.10	0.80	0.68	0.54	0.62	3.59
	AB	A	0.76	0.33	0.72	0.61	0.54	0.65	1.55
		AB	0.76	0.95	0.72	0.93	0.85	0.97	63.61
		AC	0.76	0.36	0.10	0.38	0.56	0.63	1.75
		AD	0.76	0.32	0.72	0.61	0.54	0.64	1.49
B	AC	A	0.73	0.40	0.72	0.62	0.57	0.58	1.83
		AB	0.73	0.25	0.20	0.31	0.49	0.44	0.92
		AC	0.73	0.96	0.58	0.95	0.85	0.97	68.71
		AD	0.70	0.46	0.71	0.62	0.58	0.60	1.99
	A	A	0.64	0.56	0.69	0.61	0.60	0.65	2.27
		AB	0.64	0.49	0.24	0.51	0.57	0.61	1.72
		AC	0.64	0.63	0.14	0.63	0.64	0.69	3.06
		AD	0.67	0.60	0.72	0.65	0.64	0.69	3.05
	AD	A	0.63	0.38	0.65	0.55	0.51	0.63	1.05
		AB	0.63	0.46	0.22	0.48	0.55	0.61	1.46
		AC	0.63	0.62	0.14	0.62	0.63	0.70	2.86
		AD	0.65	0.44	0.67	0.58	0.55	0.66	1.46
	D	A	0.98	0.13	0.81	0.70	0.56	0.61	6.77
		AB	0.98	0.72	0.48	0.75	0.85	0.90	112.57
		AC	0.98	0.11	0.10	0.15	0.55	0.61	5.59
		AD	0.98	0.20	0.82	0.72	0.59	0.65	12.25
	AB	A	0.81	0.29	0.75	0.64	0.55	0.64	1.74
		AB	0.81	0.94	0.73	0.93	0.88	0.97	73.35
		AC	0.81	0.25	0.09	0.28	0.53	0.57	1.43
		AD	0.80	0.30	0.74	0.63	0.55	0.64	1.71

Table E.1.4: Results obtained training a ResNet-50 model. Dataset naming follows Table 7.1.1

Pre-trained encoder	Training dataset	Test dataset	Sensitivity	Specificity	F-Score	Accuracy	BA	AUC	DOR
C	AC	A	0.68	0.51	0.71	0.62	0.59	0.64	2.20
		AB	0.68	0.22	0.18	0.27	0.45	0.43	0.58
		AC	0.68	0.93	0.44	0.92	0.80	0.93	27.98
		AD	0.67	0.54	0.71	0.63	0.60	0.65	2.38
	A	A	0.77	0.38	0.74	0.64	0.57	0.63	1.99
		AB	0.77	0.08	0.18	0.16	0.42	0.31	0.29
		AC	0.77	0.37	0.11	0.39	0.57	0.62	1.97
		AD	0.77	0.42	0.75	0.65	0.59	0.66	2.42
	AD	A	0.60	0.64	0.68	0.61	0.62	0.68	2.72
		AB	0.60	0.36	0.19	0.39	0.48	0.51	0.84
		AC	0.60	0.54	0.11	0.54	0.57	0.63	1.73
		AD	0.61	0.68	0.69	0.63	0.65	0.71	3.32
	D	A	0.87	0.11	0.75	0.61	0.49	0.62	0.81
		AB	0.87	0.37	0.26	0.43	0.62	0.70	3.80
		AC	0.87	0.11	0.09	0.14	0.49	0.49	0.79
		AD	0.88	0.18	0.77	0.65	0.53	0.66	1.61
	AB	A	0.81	0.31	0.75	0.64	0.56	0.67	1.94
		AB	0.81	0.93	0.71	0.92	0.87	0.97	61.00
		AC	0.81	0.13	0.08	0.16	0.47	0.47	0.62
		AD	0.82	0.30	0.76	0.65	0.56	0.67	1.95
B	AC	A	0.67	0.56	0.71	0.63	0.61	0.66	2.50
		AB	0.67	0.36	0.21	0.39	0.51	0.48	1.11
		AC	0.67	0.98	0.63	0.96	0.82	0.98	90.25
		AD	0.62	0.60	0.68	0.61	0.61	0.66	2.45
	A	A	0.63	0.62	0.70	0.63	0.63	0.70	2.84
		AB	0.63	0.34	0.19	0.37	0.49	0.52	0.88
		AC	0.63	0.45	0.10	0.46	0.54	0.59	1.42
		AD	0.66	0.64	0.72	0.65	0.65	0.73	3.45
	AD	A	0.63	0.62	0.70	0.63	0.63	0.68	2.84
		AB	0.63	0.47	0.23	0.49	0.55	0.63	1.56
		AC	0.63	0.61	0.13	0.61	0.62	0.70	2.74
		AD	0.65	0.66	0.71	0.65	0.66	0.71	3.61
	D	A	0.99	0.07	0.81	0.68	0.53	0.61	6.36
		AB	0.99	0.62	0.41	0.66	0.80	0.91	142.68
		AC	0.99	0.04	0.09	0.08	0.51	0.53	3.41
		AD	0.99	0.14	0.82	0.71	0.56	0.65	16.12
	AB	A	0.78	0.44	0.76	0.67	0.61	0.69	2.80
		AB	0.78	0.96	0.75	0.94	0.87	0.97	86.56
		AC	0.78	0.37	0.11	0.39	0.57	0.66	2.02
		AD	0.80	0.50	0.78	0.70	0.65	0.72	4.00

Table E.1.5: Results obtained training a DenseNet-121 model. Dataset naming follows Table 7.1.1

E.2 Details on CheXpert pretraining

E.2.1 Network Architecture

As the backbone of our model, we used state-of-the-art convolutional neural networks such as ResNet (He et al., 2016) and DenseNet (Huang et al., 2017). Specifically, we first tested a smaller ResNet-18 in order to assess the benefits of this approach compared to the method explained in Sec. 7.1. Then, we switched to a larger DenseNet-121, as the much larger dataset allowed us to exploit larger models.

The encoder is then followed by a two-layer fully connected classifier. The classifier architecture we designed reflects the hierarchy of the different lung pathologies presented in Fig. 7.2.1. As shown in Fig. E.2.1, the classifier is constructed by stacking two fully-connected layers, and makes use of connectivity patterns similar to "dense connections" as proposed by Huang et al. (2017). The first fully-connected layer (*FC1*) is used to classify the 8 top-level classes from the extracted features. Output logits are then concatenated with the extracted image features, and the second fully-connected layer (*FC2*) is used to predict the remaining 6 children pathologies. A sigmoid layer is used to obtain the probability for each class. We call this architecture *Hierarchical Classifier* (HC).

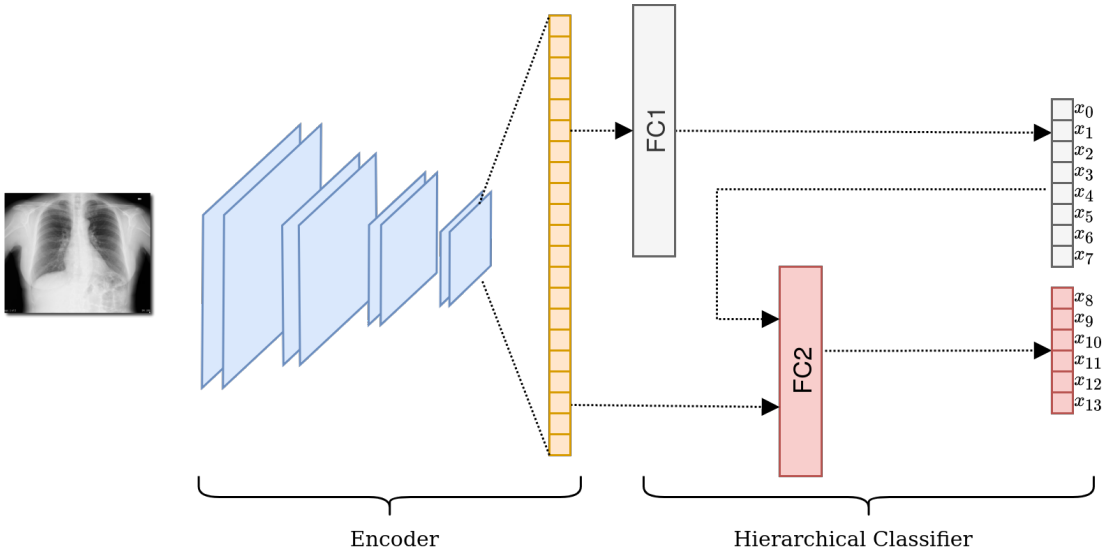


Figure E.2.1: Complete trained framework. After the encoder extracts deep features from the CXR, the Hierarchical Classifier provides outcome on the found pathologies.

The models have been trained using the standard weighted binary cross entropy loss (BCE)

$$L_n = -w_n \cdot [y_n \log(x_n) + (1 - y_n) \log(1 - x_n)] \tag{E.2.1}$$

where, for a given sample, y_n is the ground truth label for the n-th class, x_n is the model probability prediction and w_n is the weight associated to the n-th class. Weights can be used to address imbalances in the labels distribution, by giving more importance to minority classes.

Dealing with uncertainty To address the uncertain labels, multiple approaches have been suggested by Irvin et al. (2019): for example, ignoring all the uncertain labels, or considering them as either positive or negative are two main-stream solutions. The approach we followed consists in mapping the uncertain labels to maximum uncertainty (0.5). Through loss weighting, we can not only address the class unbalance issue (as discussed in Sec. E.2.1), but we can also control the influence of uncertain labels to the learning process. The following weighting schema has been used:

$$w_n = \begin{cases} 1 + S_n^+ / S_n^- & \text{if } y_n = 0 \\ 1 + S_n^- / S_n^+ & \text{if } y_n = 1 \\ 1 & \text{if } y_n = 0.5 \end{cases} \tag{E.2.2}$$

where S_n^- and S_n^+ respectively represent the cardinality of negative and positive samples for the n-th class. Hence, uncertain samples will have a lower influence during the training process, while being pushed either towards 0 or 1 by the higher weight certain samples in the same class. All of the remaining blank labels are ignored when computing the BCE loss, considering them as missing labels.

E.2.2 Results

For the following discussion about radiological findings detection, we present the results obtained with DenseNet-121, as, obviously, it outperformed ResNet-18 given the greater size of the CheXpert dataset. In Sec. 7.2, however, we also review the final results achieved with ResNet-18, in order to provide a more in-depth comparison with the direct approach.
Tab. 7.2.1 of the main text shows the results of the HC model presented in the previous section, evaluated on the chosen CheXpert test classes[1] of the validation set, in terms of AUC. The results we obtained are in line with those proposed by the Stanford team: 0.83 AUC for Atelectasis, 0.79 for Cardiomegaly, 0.93 for Consolidation, 0.93 for Edema and 0.93 for Pleural Effusion.

To further test the reliability of the lung pathologies detection step, we computed a prediction for all of the lung pathologies on the CORDA-CDSS dataset. We also employed a manually annotated label "RX" for CORDA-CDSS, which indicates whether the patient is completely healthy (0) or presents any kind of radiological finding (1). This label is non-specific to Covid-19. Figure E.2.2 shows the correlation between the predicted pathologies and the COVID and RX labels. As expected, the "No Finding" and the RX labels show a quite strong negative correlation, meaning that the model is generalizing well on unseen data.
Figure E.2.3 shows the confusion matrix obtained by predicting the RX label using the probability score obtained for the "No Finding" class (RX- when "No Finding" presented the highest probability and RX+ otherwise), reaching a sensitivity of 0.75 and a specificity of 0.79.
It also interesting to notice how both the RX and COVID labels exhibit the highest correlation values with "Lung Opacity", "Edema", "Consolidation", "Pneumonia",

[1] https://stanfordmlgroup.github.io/competitions/chexpert/

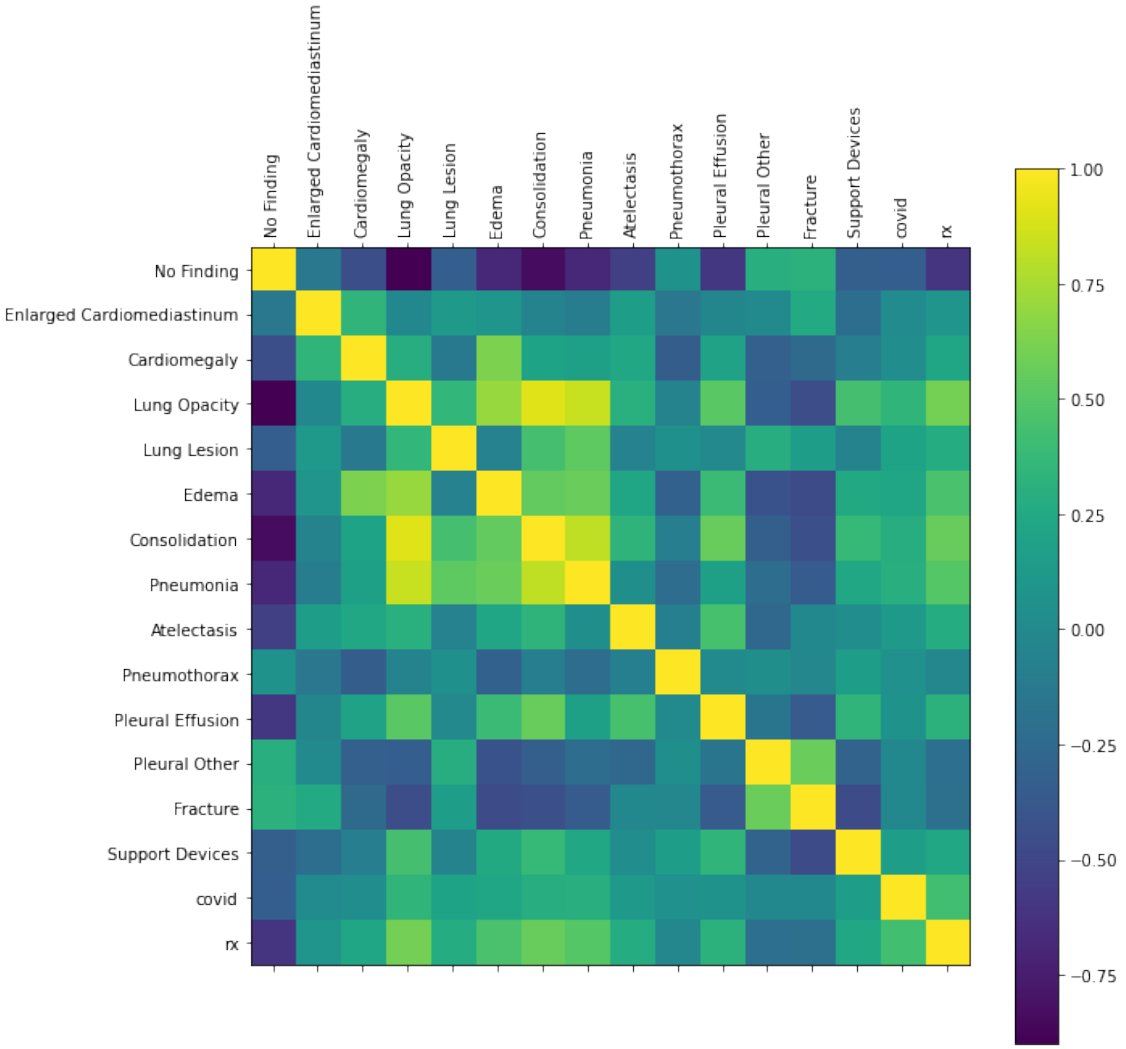


Figure E.2.2: Correlation between predicted lung pathologies and labels from CORDA dataset on the CORDA-CDSS dataset

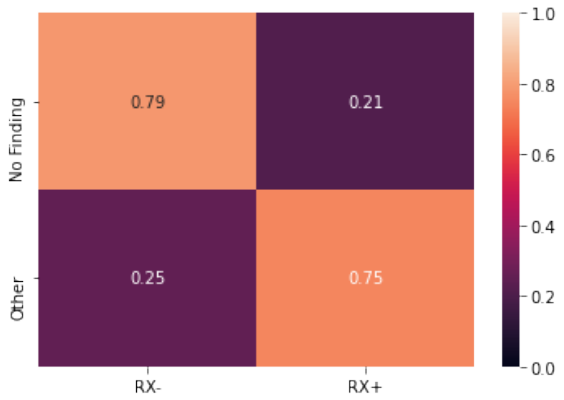


Figure E.2.3: Confusion matrix using the "No Finding" CheXpert class to predict the CORDA-CDSS RX label.

"Atelectasis" and "Pleural Effusion", which is coherent with what mentioned in Section E.3.

E.3 Analysis of classification trees for Covid-19 prediction

As said in Sec. 7.2, training the tree model on the probability outputs results in a very good interpretability. Fig. E.3.1 graphically shows the decision tree: this provides a very clear interpretation for the decision process. Each box in the tree represents a splitting criterion based on a certain radiological finding: the *Gini* index, that has been used in the CART algorithm, indicates the impurity of a split, *value* represents the obtained partition between COVID negative (first element of the vector) and positive samples respectively. A path from the root to a leaf node represents a number of sequential decisions taken on a given sample, in order to make a final prediction. From the clinical and radiological perspective, these data are consistent with the Covid-19 CXR semeiotics that radiologists are used to deal with.

The edema feature, although unspecific, is strictly related to the interstitial involvement that is typical of Covid-19 infections and it has been largely reported in the recent literature (Guan et al., 2020). Indeed, in recent Covid-19 radiological papers, interstitial involvement has been reported as ground glass opacity (GGO) appearance (Wong et al., 2020). However this definition is more pertinent to the CT imaging setting rather than CXR; the "edema" feature (according to the CheXpert definition) can be compatible, from the radiological perspective, to the interstitial opacity of Covid-19 patients.

Furthermore, the not irrelevant role of cardiomegaly (or more in general enlarged cardiomediastinum) in the decision tree can be interesting from the clinical perspective. In fact, this can be read as an additional proof that established cardiovascular disease can be a relevant risk factor to develop Covid-19.[2]

E.4 Analysis of Covid-19 detection on pathological patients

In addition to the results presented in Sec. 7.2, we could determine whether the model was able to discriminate between COVID-19 positives and negatives or whether it was just exploiting stratification in the data (e.g. healthy vs non-healthy) thanks to the RX label present in CORDA-CDSS (see Sec. E.2.2).

By inspecting the subset of RX-positive images in the test set, we were able to assess spurious correlations between ill patients and COVID-19 positive ones. Results for the RX+ test subset are shown in Table E.4.1. We can notice how the tree model

[2] https://www.escardio.org/Education/Covid-19-and-Cardiology/ESC-Covid-19-Guidance

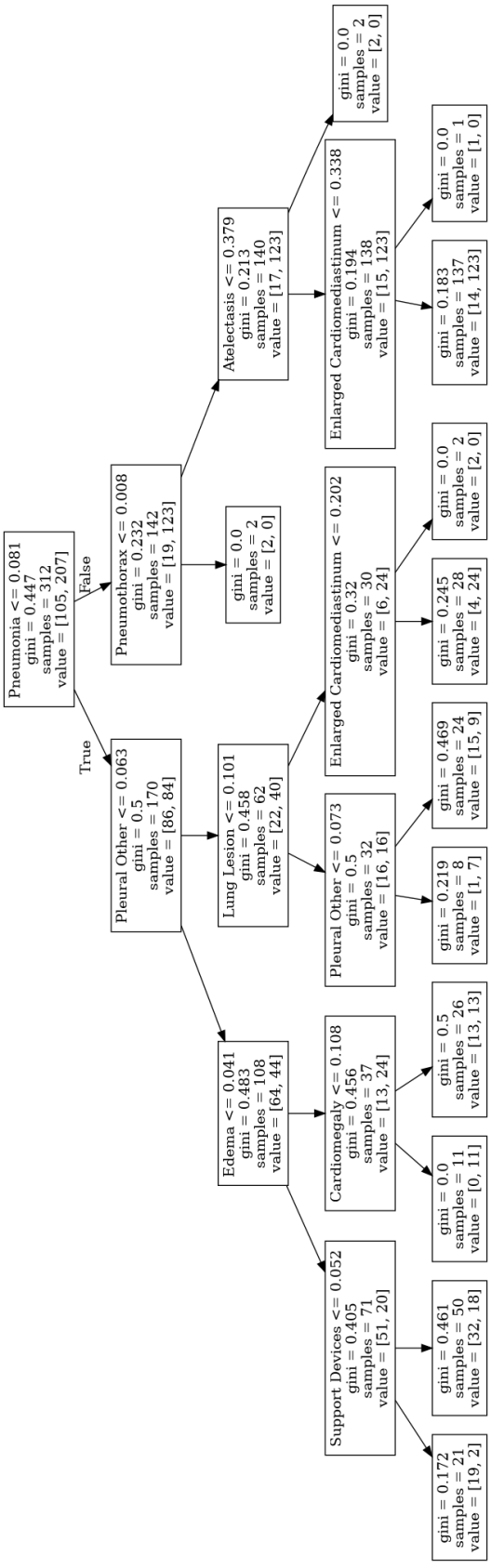


Figure E.3.1: Decision Tree obtained for Covid-19 classification based on the probabilities for the 14 classes of findings.

Table E.4.1: Stratified analysis of the two-step method on CORDA-CDSS RX+.

Model	**Sensitivity**	**Specificity**	**BA**	**AUC**
DenseNet-121+Tree	0.89	0.27	0.58	0.62
DenseNet-121+FC	0.82	0.73	0,78	0.86

lacks of specificity among ill patients. This is however expected, as making a diagnosis solely based on the presence of certain lung pathologies might increase false positives rate. In order to discriminate between COVID-19 and any other disease, it is useful to exploit the richer features extracted by the encoder, which also contain information about the appearance of the findings. Promising results are in fact obtained by the fully-connected classifier trained on top of the encoder, reaching a quite high sensitivity of 0.82 while retaining a good specificity of 0.73.

Again, this is the result of a trade-off between interpretability and discriminative power: while existing techniques like Grad-CAM (Selvaraju et al., 2017) might help in explaining deep model predictions, the insights they provide are limited when compared to simpler models like decision trees. Grad-CAM highlights the region of the input image which is more relevant to the final prediction.

It is worth mentioning that with *DenseNet-121+FC* we also achieved a sensitivity of 0.50 (specificity 0.79) on the RX-negative images, which did not show apparent lung pathologies according to the radiologists. This is an obviously harder task, and, while the sensitivity of our model is certainly not high, these images would have been completely discarded by a first radiological examination. This approach might aid radiologists in better identifying COVID-positive patients.