Master Thesis

Self-supervised Learning Framework for Imbalanced Positive-Unlabeled Data

Jonas Schweisthal

Supervisors:	Dr. Mina Rezaei, Dr. David Ruegamer
Date:	April 30th, 2022

Declaration of Originality

I confirm that the submitted thesis is original work and was written by me without further assistance. Appropriate credit has been given where reference has been made to the work of others.

Munich, April 30^{th}, 2022

. . . .
Jonas Schweisthal

Abstract

Positive Unlabeled (PU) Learning is a binary classification problem where only positive and unlabeled data are available. Most methods are designed for balanced datasets, but in many applications there exist fewer samples in the positive class, such as in medical diagnosis. Self-supervised representation learning can create embeddings of unlabeled data using pre-text tasks and achieved promising results in imbalanced learning and semi-supervised learning. In this work, we investigate to what extent PU learning on imbalanced data can benefit from self-supervised learning. We apply a two-step approach with decoupled contrastive self-supervised representation learning followed by classifier training adjusted for imbalanced PU data and evaluate them on different image datasets. In addition, we develop a novel contrastive PU loss for imbalanced data that incorporates information about the PU labels already in representation learning. We empirically show that the performance of PU learning can be increased under certain circumstances, especially by self-supervised pre-training without label information.

Contents

Chapter 1

Introduction

Positive-unlabeled (PU) learning is a special case of binary classification, in which only positive and unlabeled samples are available [Jaskie and Spanias, 2019]. For negative observations there is no class information accessible, e.g. due to high costs or missing information in the survey. This differs from semi-supervised learning, where some positive and negative samples are known and the rest of the samples are unlabeled [Ouali et al., 2020]. Typical real world applications of PU data occur in the areas of medical diagnosis [Claesen et al., 2015], fraud detection [Jiang et al., 2020] and knowledge base completion [Arora, 2020], among others. Especially in these areas the classes are often highly imbalanced, with only a few observations in the positive class. While many articles and methods address the problem of PU learning in a balanced setting [Bekker and Davis, 2020], only few explicitly address its application to imbalanced data [Su et al., 2021], Jiang et al. [2020].

Self-supervised Learning can extract information from an unlabeled dataset by applying auxiliary tasks and create helpful representations of the data without requiring the labels of the actual classification task of interest Jing and Tian [2020]. These representations can then be used for training the model on the actual classification task of interest. Contrastive self-supervised learning proved to be a very successful method in this area by applying two data augmentations of one sample and trying to minimize the distance in latent space between related samples while maximizing the distance to the other samples [Chen et al., 2020b], [Jaiswal et al., 2020].

Self-supervised pre-training has proven to be very effective for classification on imbalanced data [Liu et al., 2021], [Yang and Xu, 2020]. In addition, self-supervised learning has been very successful in the area of semi-supervised learning by incorporating information about unlabeled samples into the classification model [Zhai et al., 2019]. Based on this, we want to investigate whether contrastive self-supervised learning can achieve similar success in the related problem of PU learning by using information in unlabeled samples more effectively, especially on imbalanced data, with only a few observations in the positive class.

1.1 Main Contribution

The main contribution of this work can be divided into two parts:

1. We investigate the extent to which positive-unlabeled learning on imbalanced data can benefit from contrastive self-supervised learning using different image datasets. To do so, we develop a two-step framework based on the idea of decoupling representation learning and classification [Kang et al., 2019]. In the first step, representations of the images are learned independently of the label information through contrastive learning. In the second step, we use these representations as input to a simple classification model that is trained with a loss adjusted for imbalanced and PU data [Su et al., 2021]. For this purpose, we compare our method with a comparable model without pre-training and with current state-of-the-art approaches from PU learning.

2. Based on the supervised contrastive loss [Khosla et al., 2020] and the imbalanced non-negative PU loss Su et al. [2021], we develop *connPU*, a contrastive loss for imbalanced PU data that allows to incorporate the label information of the PU problem already during the contrastive learning of the representations. We compare the results of connPU with those of our first approach.

To the best of our knowledge, we are the first to investigate the applicability of self-supervised learning for imbalanced positive-unlabeled data in more detail.

1.2 Thesis Structure

At the beginning, in chapter 2, we explain the basics and methods for the three different areas positive-unlabeled learning (2.1), imbalanced learning (2.2) and self-supervised representation learning (2.3), as well as their overlaps (2.4). In chapter 3, we describe, apply, and evaluate the approach of self-supervised pre-training for imbalanced PU classification. Then, in chapter 4, we elaborate on the development and application of the novel contrastive loss for imbalanced PU data. In chapter 5, we summarize the insights gained from the aforementioned approaches and provide an outlook on possible future research directions based on them.

Chapter 2

Representation Learning from Imbalanced Positive-Unlabeled Data

2.1 Positive-Unlabeled Learning

In a common supervised binary classification problem in machine learning, the data can be divided into two classes, defined as positives and negatives. Here, the labels of the complete training data are known, so each sample can be uniquely assigned to either the positive class or the negative class. Positive-Unlabeled (PU) learning is a special form of binary classification and is characterized by the fact that some samples from the training data set are labeled as positive, but no labels are available for the remaining samples [Liu et al., 2003], [Ouali et al., 2020]. Consequently, these unlabeled samples can belong to either the negative class or the positive class.

2.1.1 Occurrence of Positive-Unlabeled Data

PU data can be found in various application areas in the real world, where a binary classification problem is to be solved, and there are different reasons why only positive labels but no reliable negative labels are collected. Especially in medical data PU learning plays an important role. One example is the identification of genes responsible for diseases [Yang et al., 2012], where it is difficult and costly to identify genes that are definitely not related to the disease. In general, when classifying diseases based on patient data, PU data are often available because patients are not appropriately screened either due to lack of resources or mild or asymptomatic progresses despite having the disease. This is the case, for example, with diabetes [Claesen et al., 2015] which often is not diagnosed, but also with the current issue of Covid-19, where individuals not tested or tested negative by rapid tests due to insufficient sensitivity in some rapid tests [Scheiblauer et al., 2021] should not necessarily be considered negative, but unlabeled. Chen et al. [2020d] show that disease progression, such as Alzheimer's, can also be considered a PU problem, as often early stages of the disease with mild symptoms are incorrectly labeled as healthy in standard binary classification approaches. At the same time, this means that PU learning approaches in this field can be extremely useful for early disease detection without requiring observations at multiple time points for a patient.

Another very current use case of PU learning is recommendation systems [Zhou et al., 2021], which are used primarily by large tech companies and can have a high impact on customer loyalty and sales. In this case, suitable suggestions for future interactions are to be determined and presented to the customer on the basis of previous user data. For example, streaming services generate suggestions for productions that are also interesting for the user on the basis of films and series already seen, or online shopping recommends suitable products on the basis of previous purchasing behavior. Often, only positive samples in the form of purchases and clicks are available, and no negative samples, which means that the rest of the range can be regarded as unlabeled samples[Bekker and Davis, 2020].

Methods from PU learning can also be applied to related classification problems under certain assumptions, such as inlier-based outlier detection [Hido et al., 2008]. Here, verified samples can be considered as labeled positives, whereas the remaining observations are considered unlabeled and the unknown outliers represent the negative class. A very similar problem to which PU learning can be applied is one-class classification [Khan and Madden, 2014]. Here, the classification of a particular group (positives) with only some known samples is of interest and all other groups are considered negative, which at the same time often leads to high heterogeneity in the negative class. Examples of use cases in these areas are the defect detection of machines using only data of correctly functioning systems [Fujimaki et al., 2005], or the detection of malicious users in social networks or knowledge bases [Zheng et al., 2019].

2.1.2 Problem Formulation

PU learning is a large field in machine learning with many different sub-fields, each of which can be described using different assumptions and solved using different methods. In this subsection, we will primarily focus on the concepts that are important for this work. If not stated otherwise, we will follow the contents of Bekker and Davis [2020], which provide an exhaustive overview of the topic.

In the usual binary classification, each observation in the training dataset can be described by the tuple $(\mathbf{x}, y)$, where $\mathbf{x}$ represents the features and $y \in \{0, 1\}$ represents the label. In PU learning, however, the labels are not known for all samples, which is why Elkan and Noto [2008] add the binary variable $s \in \{0, 1\}$ to get the triplet $(\mathbf{x}, y, s)$, where $s = 1$ if the sample is labeled as positive and $s = 0$ if the sample is unlabeled. By the nature of PU learning, the probability $p(y = 1|s = 1) = 1$ and $p(s = 1|y = 0) = 0$.

Labeling Process

Since no direct information about y is available for the training dataset, the auxiliary variable s must be used to determine a suitable classifier via dependencies between s and y. To do this, some distributional assumptions must be made for the labeling mechanism and the data in general. Elkan and Noto [2008] distinguish between two scenarios for the occurrence of PU data. In the *single-training-set-scenario*, the entire training data is assumed to be an i.i.d sample from the true, data-generating distribution $p(\mathbf{x}, y, s)$ where y is not collected. In the *case-control-scenario*, however, the labeled positive data come from their own independent distribution and only the unlabeled data $p(\mathbf{x}|s = 0)$ come from the true distribution. Most algorithms can be applied to both scenarios under certain assumptions, but in this paper we will focus further on the *single-training-set-scenario*, which has also received more attention in previous literature.

At next it is important to understand how the labeling mechanism works, i.e. how the labeled samples are determined from the positive samples. Each sample is selected from the positive samples with the probability $e(\mathbf{x}) = p(s = 1|y = 1, \mathbf{x})$, called propensity score. It is noticeable that $e(\mathbf{x})$ depends on $\mathbf{x}$. This is the case if a *probabilistic gap* is assumed [He et al., 2018], which is given as $\Delta p(\mathbf{x}) = p(y = 1|x) - p(y = 0|x)$. This means that positive samples that are less distinct from negative samples based on the features $\mathbf{x}$ are less likely to be labeled as positive, which is often the case in disease diagnosis.

With the *Selected At Random (SAR)* assumption, the propensity score depends entirely on the features $\mathbf{x}$, but there is no probabilistic gap. The most commonly used assumption is the *Selected Completely At Random (SCAR)* assumption, where the propensity score is independent of $\mathbf{x}$ and thus can be given by

$$e(\mathbf{x}) = p(s = 1|y = 1, \mathbf{x}) = p(s = 1|y = 1) = c \tag{2.1}$$

as a constant label probability. c thus simultaneously corresponds to the proportion of labeled data in the positive data. In the remainder of this thesis, we will make this assumption, in particular for all algorithms described below.

Class Prior

An important aspect in PU learning is the class prior $\pi = p(y = 1)$, which represents the proportion of the positive class in the entire distribution. The prior has a direct relationship with the label probability c, which can be represented as follows:

$$\begin{aligned} c = p(s=1|y=1) &= \frac{p(s=1, y=1)}{p(y=1)} \\ &= \frac{p(s=1)}{p(y=1)} = \frac{p(s=1)}{\pi} \\ \iff \pi &= \frac{p(s=1)}{c} \end{aligned} \tag{2.2}$$

where $p(s = 1)$ can be seen as the proportion of the labeled examples in the whole data.

The prior can be used in several ways to train a suitable classifier. In some applications, however, neither π nor c are known, so π must be estimated from the data. Methods for this are for example approaches via partial matching [Du Plessis and Sugiyama, 2014], decision tree induction [Bekker and Davis, 2018], receiver operating characteristic approaches [Blanchard et al., 2010] or kernel embeddings [Ramaswamy et al., 2016]. In this thesis, however, π is assumed to be known, which is why the estimation of the prior is not discussed in more detail.

Additional Assumptions

For most PU learning algorithms, some additional assumptions about the data are helpful to train a good classifier. Here we address the two most important ones for the later experiments.

The *Seperability* assumption states that there exists a function f and a threshold τ such that for all x_i holds:

$$\begin{aligned} f(x_i) &\geq t, \quad \text{if} \quad y_i = 1 \\ f(x_i) &< t, \quad \text{if} \quad y_i = 0 \end{aligned} \tag{2.3}$$

This property allows to train good classifiers by classifying all positive labeled examples and at the same time as few unlabeled examples as possible as positives [Liu et al., 2002], [Blanchard et al., 2010], or prior knowledge about π can be used to determine the decision boundary in order to optimize the estimation.

Another important assumption for the applications in this thesis is the *Smoothness* assumption. This states that the similarity of two data points x_i and x_j also indicates similar probabilities regarding their true classes $p(y|x_i)$ and $p(y|x_j)$. This is a necessary condition to take advantage of metric learning, in particular self-supervised representation learning, which is described in more detail in section 2.3.

2.1.3 Methods for Positive-Unlabeled Learning

Bekker and Davis [2020] divide the methods for PU learning into 3 or 4. different categories, respectively:

1. *Two-Step Techniques* attempt to determine reliable negative samples (and in some cases further positive observations [Fung et al., 2005]) in the first step, and then in the second step use these to apply traditional semi-supervised learning approaches to the generated positive, negative, and unlabeled observations.

2. *Biased Learning* approaches consider PU learning as a binary classification problem with the unlabeled samples as noisy observations of the negative class. For this, normal binary classifiers are modified such that mis-classifications of positives are penalized harder than those of negatives or hyperparameters of learners are tuned to optimize appropriate PU metrics in the validation dataset.

3. *Class Prior Incorporation* uses direct knowledge about the class prior π to generate a good classifier. A distinction can be made between pre-processing and post-processing. In pre-processing a modified dataset is created by rebalancing, incorporation of label probabilities or empirical risk-minimization methods, and directly integrated into the learning process. In post-processing the decision function or the predicted probabilities are adjusted after training.

4. *Other methods* summarize approaches, which cannot be assigned to the upper categories, like e.g. modeling of the densities of the two classes by generative adversarial networks adapted on PU data [Hou et al., 2017], [Guo et al., 2020], [Chiaroni et al., 2018].

In this thesis, we will go into more detail about the use of techniques based on class prior incorporation, in particular the use of cost-sensitive empirical risk-minimization methods, which can be applied very well to unstructured data in the field of deep learning, such as image data.

In the binary classification case, the mis-classification risk $R(g) = \mathbb{E}_{(X,Y)\sim p(x,y)}[\ell(g(X), Y)]$ can be represented as

$$\mathcal{R}_{\text{pn}}(g) = \pi \mathbb{E}_{P(x|y=1)}[\ell(g(x), 1)] + (1-\pi)\mathbb{E}_{P(x|y=0)}[\ell(g(x), 0)] \tag{2.4}$$

where the function $g : \mathbb{R}^d \to (0, 1)$ models $P(y|x)$ and $\ell(\cdot, \cdot)$ is the zero-one loss $l_{01}(g(x), y) = (1 - \lfloor 2g(x) \rfloor)$.

Since there is no direct access to the true label y for the unlabeled data in PU learning, Du Plessis et al. [2014] and Kiryo et al. [2017] show that this risk can be reformulated to be independent from knowledge about true negatives as

$$\mathcal{R}_{\text{pu}}(g) = \pi \mathbb{E}_{P(x|Y=1)}[\ell(g(x), 1)] + \big(\mathbb{E}_{P(x)}[\ell(g(x), 0)] - \pi \mathbb{E}_{P(x|Y=1)}[\ell(g(x), 0)]\big) \tag{2.5}$$

Derived from this equation, Kiryo et al. [2017] developed the following non-negative PU loss:

$$\begin{aligned}\mathcal{L}_{\text{nnpu}}(g) = &\frac{\pi}{n_{s=1}} \sum_{x_i \in X|s=1} \ell\left(g\left(x_i\right), 1\right) + \\ &\max\left(0, \frac{1}{n_{s=0}} \sum_{x_i \in X|s=0} \ell(g(x_i), 0) - \frac{\pi}{n_{s=1}} \sum_{x_i \in X|s=1} \ell\left(g\left(x_i\right), 0\right)\right)\end{aligned} \tag{2.6}$$

where $n_{s=1}$ is the number of positive labeled and $n_{s=0}$ the number of unlabeled samples in the train data.

For training the model $\ell_{01}(g(x), s)$ can be replaced by a surrogate loss for better optimization, e.g. the sigmoid loss

$$\ell_{sig}(g(x), s) = \frac{1}{1 - exp(2(s - 0.5)g(x))} \tag{2.7}$$

The $max(0, \cdot)$ in equation (2.6) was not implemented in the first version of Du Plessis et al. [2014] and was only added by Kiryo et al. [2017] because the second part of equation (2.6) estimates the second part of 2.4, which cannot become negative, in theory. However, since very flexible models like deep neural networks are able to model the empirical risk smaller than 0 for this term, this limit was added to the loss function.

2.1.4 Related Fields

PU learning is usually described as a binary classification problem. This problem can be generalized to *Multi-Positive-Unlabeled Learning*, where the positive class consists of several subclasses that are also to be distinguished. Applications for this are, for example, access permissions to buildings based on face recognition, where several persons have access and the training data consists of several images per person. Approaches for this are described by Xu et al. [2017] and Shu et al. [2020].

The further above mentioned *One-Class Classification* [Perera et al., 2021] differs in the point that not the positive class but the negative class consists of several individual classes. However, since no labels of the negative class are known in PU learning, no trivial subdivision into these subclasses is possible, and it is usually treated as a binary classification problem with adjustment for higher heterogeneity in the negative class, or methods from outlier detection are used [Seliya et al., 2021].

If some negative labels are known, methods from *Semi-supervised Learning* can be applied instead of PU learning. In the field of deep learning often compositions of a supervised loss for the observations with known labels and an unsupervised loss for the unlabeled data [Zhai et al., 2019], [Yang et al., 2021], and/or pseudo-labeling approaches [Rizve et al., 2021] are used. The latter are used in PU learning slightly modified also in the two-step techniques, sometimes also in combination with the help of class prior incorporation [Chen et al., 2020d], [Dorigatti et al., 2022].

2.2 Imbalanced Learning

In supervised machine learning for classification problems, many methods naively assume an approximate balanced distribution of observations per class. In many real world scenarios, however, the classes are strongly imbalanced, with sometimes only extremely few samples in the tail classes. Typical scenarios are object detection in autonomous driving [Carranza-García et al., 2021] or few-shot detection [Ochal et al., 2021]. Imbalanced data does not only occur in multi-class problems, but also in the binary setting, with often only a few samples in the positive minority class. The applications of binary imbalanced data largely overlap with the applications of PU data described in subsection 2.1.1, e.g. in medical diagnosis [Rahman and Davis, 2013] or fraud detection [Makki et al., 2019]. If traditional classification models are not adapted accordingly, the majority class is often wrongly predicted due to its high number of observations in the training of the model, and the minority class is neglected.

2.2.1 Performance Metrics

The confusion matrix for binary classification is displayed in table 2.1. A typical metric for the prediction quality of machine learning models for classification is the accuracy $(TP + TN)/(TP + FP + TN + FN)$. For binary imbalanced data, this method is usually not suitable, since it does not adjust for the different class sizes. Thus, for example, with a high imbalance ratio of 1:19 positives:negatives, a classifier can achieve an accuracy of 95% even if it only ever predicts negative as class. This can lead to problems in applications that put a lot of emphasis on correct prediction of the minority class, such as medical diagnosis. As an alternative measure for imbalanced data, the F1-score, the harmonic mean between recall and precision, is often used: $F1 = 2(recall \cdot precision)/(recall + precision)$. Here $recall = TP/(TP+FN)$ is the proportion of correctly predicted positives out of the positives and $precision = TP/(TP + FP)$ is the proportion of correctly predicted positives out of all positive predictions.

Truth	Prediction	
	Positive	Negative
Positve	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Table 2.1: Confusion matrix of binary classification.

2.2.2 Methods for Imbalanced Learning

Johnson and Khoshgoftaar [2019] divide the approaches for imbalanced learning into 2 categories:

1. *Data-level Methods* try to solve the challenge of imbalanced learning with targeted changes of the data distribution and subsequent training of a traditional classifier, mostly by resampling. Typical applications are simple random under-sampling of the majority class or random over-sampling of the minority class [Van Hulse et al., 2007]. More advanced methods are based on *Synthetic Minority Over-sampling Technique (SMOTE)* [Chawla et al., 2002], where additional artificial samples of the minority class are generated by interpolation between the nearest neighbors of the samples in the minority class.

2. *Algorithm-level Methods* do not change the data structure, but adapt the classification algorithm directly to imbalanced data. Often samples from minority classes are penalized more or the decision threshold is adjusted accordingly. For the latter e.g., Khan et al. [2019] use Bayesian uncertainty estimates, since these correlate directly with the rarity of the classes. As in PU learning (subsection 2.1.3), cost-sensitive learning can use re-weighting to increase the importance of samples from the minority class in classifier training. A method that adjusts for both positive-unlabeled and imbalanced data is described in subsection 2.4.3.

2.3 Self-Supervised visual Representation Learning

2.3.1 Representation Learning

The idea behind *Representation Learning* is to generate mostly low-dimensional representations of data that map information about the latent data structure and are helpful for downstream tasks, such as classification or regression [Bengio et al., 2013]. Especially for unstructured, high-dimensional data and features that cannot be interpreted directly, such as text, audio, or image data, *Deep Representation Learning* has high importance. Use cases with extreme success and progress in recent years include speech recognition and signal processing [Amiriparian, 2019], natural language processing [Mikolov et al., 2013] or object detection [Xie et al., 2021].

Metric Learning
Metric Learning is a subsection of representation learning that attempts to learn good representations in latent space by learning distances between observations [Bellet et al., 2013]. The idea follows the principle of smoothness addressed in subsection 2.1.2. The representations of semantically similar samples should be close to each

other and thus have a lower learned distance or a higher similarity score than different observations. In contrast to e.g. unsupervised generative approaches, metric learning, in a supervised manner, requires information indicating the similarity of observations, such as class labels, link connections or known distances.

2.3.2 Self-Supervised Learning

Compared to some other unsupervised representation learning approaches, deep self-supervised learning does not necessarily focus on the properties of the learned representations, such as their interpretability, but primarily on good performance in downstream tasks. For this, it uses the idea from *Transfer Learning* [Zhuang et al., 2020] that neural networks trained on certain data or tasks in a supervised manner can be very useful for similar data or tasks because the trained weights in the network already model semantic or syntactic representations of the underlying data structure.

In many use cases of representation learning, there exists a large amount of data, but there are no tasks or collected labels available for training supervised learning models. In order to still take advantage of these supervised learning architectures, pretext tasks are used, which can easily be generated automatically from the data. The final performance of the models on these pretext tasks is not important, but only the quality of the representations that are created in the trained weights of the networks.

Typical applications are autoregressive models, i.e. time series data, since the prediction of future or past observations is an auxiliary task, without the need for further tasks or labels through manual annotations by humans or transformations of data. This approach can be found in the field of natural language processing at e.g. next sentence prediction [Devlin et al., 2018], tracking and frame sequence prediction on video data [Wang and Gupta, 2015], or in general in the field of reinforcement learning [Gelada et al., 2019].

In the application to image data, usually pretext tasks are created by different modifications of images, where the type of modification is to be recognized and predicted by the network as a classification task. An example is cutting images into patches, and then predicting the relative position of two patches of the same image [Doersch et al., 2015], or predicting the position of all patches as in a puzzle [Noroozi and Favaro, 2016]. This enables learning the spatial context of the objects in the image. In another method, colored images are transformed to grayscale and the model is trained to reconstruct the original color structure [Zhang et al., 2016].

In general, approaches of *Generative Modeling* can also be seen as self-supervised learning, where the pretext task is the reconstruction of the original sample. Thus, for image datasets, mainly pixel-level contexts are modeled and stored in the representations. Relevant methods are for example the context encoder [Pathak et al., 2016] or bidirectional generative adversarial networks [Donahue et al., 2016].

A broad overview of these and other methods and more detailed insights into self-supervised learning in general can be found at Weng [2019] and Jing and Tian [2020].

2.3.3 Contrastive Learning

Approaches from generative modeling can involve high computational costs, require many samples and often have problems in convergence [Jaiswal et al., 2020], and for many downstream tasks the representations learned at the pixel level may not be optimal. The other pretext tasks described above are quite specific in terms of what information of the data should be stored in the representations, which can harm their generalizability. Accordingly, they must be carefully chosen to contain appropriate added value to the data and downstream tasks of interest [Chen et al., 2020b], [Yamaguchi et al., 2021]. *Contrastive Learning*, as another method of self-supervised learning, has made many advances in the last few years and achieved state-of-the-art performance in many tasks and datasets.

Contrastive learning is a part of the above mentioned metric learning and tries to cluster the representations of similar samples in the embedding space close together, whereas the distance to more different samples should be larger. For this, contrastive learning needs information about which observations belong together, i.e. requiring a class label. The first contrastive loss was described by Chopra et al. [2005] and can be summarized as:

$$\begin{aligned}\mathcal{L}_{\text{con}}\left(\mathbf{x}_i, \mathbf{x}_j, g\right) = & \mathbb{1}_{[y_i=y_j]} \left\|g\left(\mathbf{x}_i\right) - g\left(\mathbf{x}_j\right)\right\|_2^2 + \\ & \mathbb{1}_{[y_i \neq y_j]} \max\left(0, \epsilon - \left\|g\left(\mathbf{x}_i\right) - g\left(\mathbf{x}_j\right)\right\|_2^2\right)\end{aligned} \tag{2.8}$$

where $x_i \in X$ represents the input observations, $y_i \in \{1, ..., L\}$ represents the respective class label, and $g(\cdot) : \mathcal{X} \rightarrow \mathbb{R}^d$ represents the function or trainable neural network that maps the observations to the corresponding representations [Weng, 2021]. Hence, the objective of this loss is to minimize the distance of samples from the same class and maximize the distance to samples from the other classes with the lower bound ϵ so that there's no focus on too easy negative samples.

It is noticeable, that for contrastive learning class labels are needed. These labels can be generated automatically in a self-supervised manner via pretext tasks, too.

The most common technique, especially on image data, uses *Data Augmentation* to generate k noisy versions of each observation. The original N observations are considered to belong each to an own class, and the respective augmentations belong to the same class as their original sample, resulting in N classes with k observations each. Theoretically, one could simply use supervised classification approaches like cross-entropy loss on the augmented data. However, having so many classes with so few samples, these approaches often yield poor results, whereas contrastive learning approaches usually yield way better results [Chen et al., 2020b].

Similar to the selection of pretext tasks in non-contrastive self supervised learning, the selection of suitable augmentations is very important to generate good representations. Typical augmentations used for image embeddings include random cropping and resizing the image, random color distortions, random color jittering, random Gaussian blur, random horizontal flip, or random grayscale conversion [Weng, 2021]. There are even custom frameworks to develop good data augmentation strategies, such as *AutoAugment* [Cubuk et al., 2018] or *RandAugment* [Cubuk et al., 2020].

A common strategy for training models in contrastive self-supervised learning is guided by the idea of *Siamese Neural Networks* [Bromley et al., 1993]. Two different input samples are sent through a neural network with the same weights so that they have the same forward pass to generate the representations and calculate a distance. Specifically in this application, two different versions of each observation are generated by data augmentation and then passed through a model to get the embeddings, in which the distance is to be minimized or their similarity maximized, respectively.

SimCLR

SimCLR by Chen et al. [2020b] is a framework for visual representation learning that follows this strategy. Figure 2.1 shows the process of the framework. It consists of a data augmentation module that produces two random augmentation transformations $t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$ per image observation in the training batch, a base encoder neural network $f(\cdot)$ generating the representation vectors and a projection head $g(\cdot)$ connected to the representations, which consists of a further simple non-linear neural network on which the contrastive loss function $\mathcal{L}_{NTXent}(z_i, z_j)$ is computed.

The *Normalized Temperature-scaled Cross Entropy* loss (NT-Xent) is a version of the *InfoNCE* loss already used by Oord et al. [2018], to which the temperature parameter τ was added. For each observation, only the two augmented samples whose distance is to be minimized are considered a positive pair, and all other augmented samples of the other observations in the batch are considered negative, to which the distance is to be maximized.

$$\mathcal{L}_{NTXent}(z_i, z_j) = -\log \frac{\exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_k\right)/\tau\right)} \tag{2.9}$$

with z_i, z_j are the projections $g(f(x_i)), g(f(x_j))$ of the augmented samples $\tilde{x_i} = t(x), \tilde{x_j} = t'(x)$ of the observation x, and $sim(z_i, z_j) = z_i^T z_j / (||z_i|| \cdot ||z_j||)$ being the cosine similarity of the two vectors.

Figure 2.2 explains the training process in detail. For the final downstream tasks, such as fine-tuning or linear evaluation, not the projections z are used but

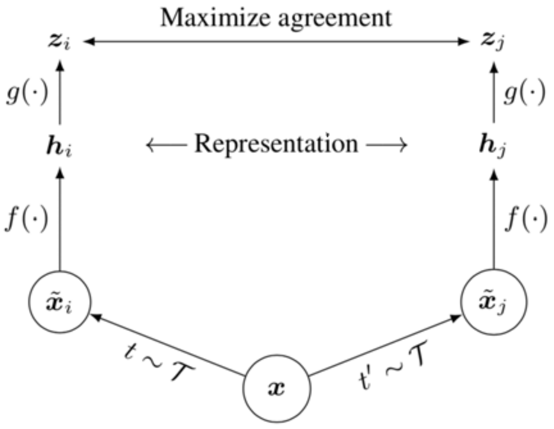


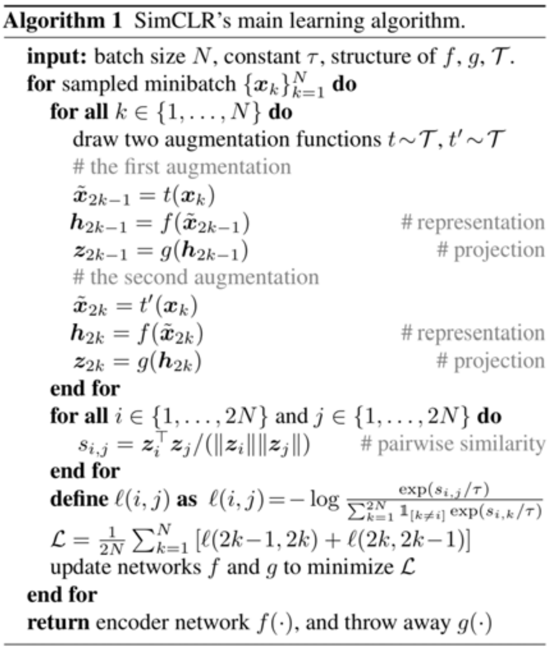
Figure 2.1: Flowchart of SimCLR by Chen et al. [2020b].Two randomly sampled data augmentations t and t' are applied to each observation x to obtain $\tilde{x}_i$ and $\tilde{x}_j$. The encoder $f(\cdot)$ and the projection head $g(\cdot)$ are trained on a the NT-Xent loss to maximize agreement between the outputs z_i and z_j. After training, we pass the original x through $f(\cdot)$ to get the representations h, which can be applied to downstream tasks.

the representations $h = f(x)$. Here, the encoder $f(\cdot)$ is a complex model with many parameters, in this case *ResNet-50* [He et al., 2016], which can learn the representations well. The projector head $g(\cdot)$ is only a 1-hidden-layer network with ReLU non-linearity and is used because it improves the performance of the representations h in the downstream tasks by mitigating the risk of information loss coming from the NT-Xent loss [Chen et al., 2020b]. Chen et al. [2020c] examine the implications for different projection heads in more detail.

In the field of visual self-supervised learning, this twin augmentation strategy is generally a frequently used and very successful technique that achieves excellent results. The applied network architectures, loss functions and training procedures vary. Other well-known and successful methods include *BYOL* [Grill et al., 2020], *Barlow Twins* [Zbontar et al., 2021], *SimSiam* [Chen and He, 2021] or *MoCo* [He et al., 2020], [Chen et al., 2020e].

Supervised Contrastive Learning

The described contrastive learning algorithms are based on the principle of self-supervised learning and define their own class by observation as a pseudo-label. In some use cases of representation learning, however, information about the class labels of the data is available. In *Supervised Contrastive Learning* [Khosla et al., 2020], a new objective takes advantage of contrastive self-supervised learning and extends it by incorporating knowledge about the class labels. Thereby a higher



```
Algorithm 1 SimCLR's main learning algorithm.
input: batch size $N$, constant $\tau$, structure of $f$, $g$, $\mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ do
  for all $k \in \{1, \ldots, N\}$ do
    draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$                # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$                  # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$                  # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$                    # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ do
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$       # pairwise similarity
  end for
  define $\ell(i,j)$ as $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks $f$ and $g$ to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$
```

Figure 2.2: Algorithm of SimCLR by Chen et al. [2020b]. Formal description of the training procedure of SimCLR visualized in figure 2.1.

stability in the model training concerning the selection of hyperparameters as well as the robustness of the representations against natural perturbations in the image data is achieved.

For this purpose, the loss function from equation (2.9) is extended as follows:

$$\mathcal{L}_{supCon} = \sum_{i=1}^{2N} \frac{1}{|J(i)|} \sum_{j \in J(i)} - \log \frac{\exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_k\right)/\tau\right)} \tag{2.10}$$

where $J(i) = \{j \in (2N \setminus \{i\}) : \tilde{y_i} = \tilde{y_j}\}$ is the set of observations with the same label as the sample i , $\tilde{y}$ is the label of the augmented sample $\tilde{x}$ being the same as the label y of the origin sample x.

This means that not all other images except the augmentation from the same original sample x in the batch are considered as negative, but the images with the same original label y are also considered as positive samples. Thus, the distance to them should also be minimized in the embeddings, while the distance would be maximized in the self-supervised setting. Figure 2.3 shows the difference between the two approaches.

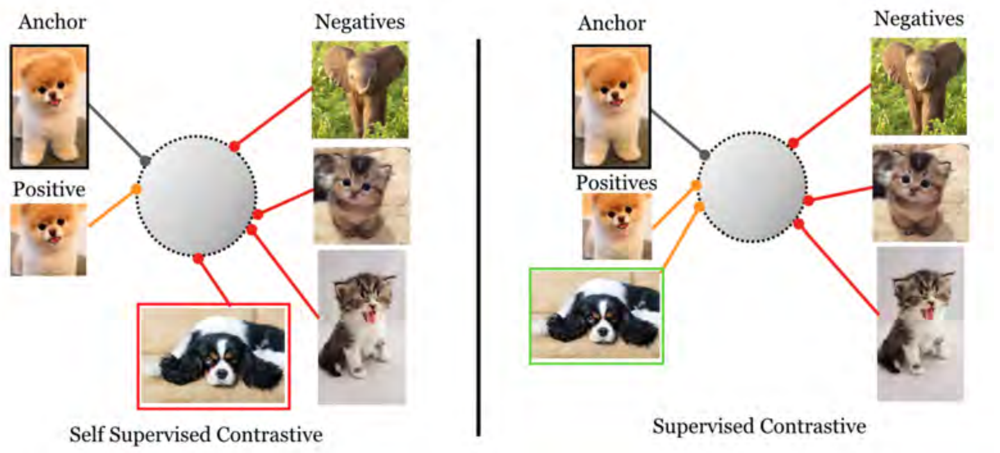


Figure 2.3: Self-supervised contrastive learning vs. supervised contrastive learning by Khosla et al. [2020]. The self-supervised loss treats only the two augmentations of the same original image as positives and minimizes the distance of their representations while maximizing the distance to all other samples in the batch. If there are multiple observations of the same original class in the same batch, the distance to them is also maximized. The supervised contrastive loss, on the other hand, uses class information and minimizes the distance of all observations of the same class.

Supervised and self-supervised contrastive learning is not limited to image data, but is also used in other areas, such as natural language processing or reinforcement learning. An overview of current methods in contrastive learning can be found e.g. at Weng [2021].

2.4 Combination of Approaches

In this thesis, we investigate the benefits of self-supervised learning on PU learning with imbalanced data in more detail. So far, we have explained the problems and approaches to solving its three individual components. In this subsection, we describe previous approaches that combine ideas from each component to solve overlaps in the different challenges.

2.4.1 Self-supervised Learning on Imbalanced Data

Yang and Xu [2020] and Liu et al. [2021] show that architectures leveraging self-supervised pre-training are more robust to class imbalance and achieve better performance than comparable fully-supervised models. One reason for this is that self-supervised pre-training learns general features of the data structure that are independent of the labels and can be transferred from the majority classes to the minority classes. The supervised models, on the other hand, only learn features that are useful for direct classification. This probably leads to an increased risk of overfitting the learned features of the minority class due to lower sample size, which self-supervised learning minimizes by its cross-class feature learning.

There are approaches to further strengthen the robustness of self-supervised learnig for imbalanced data. Liu et al. [2021] introduce *rwSAM*, a reweighting strategy that penalizes loss sharpness to achieve better generealization of representations. Jiang et al. [2021] develop a sampling strategy using additional out-of-distribution data for re-balancing long-tail distributions.

2.4.2 Contrastive Self-supervised Learning and Positive-Unlabeled Learning

As already described in subsection 2.3.3, a limitation in contrastive self-supervised learning like SimCLR is, that all other samples in the batch are considered negative to which the distance should be maximized, although among these negative samples there are usually samples belonging to the same class which should possibly be considered positive. In supervised contrastive learning this is solved by including the known class labels. The basic idea can also be applied to the self-supervised case.

In debiased contrastive learning [Chuang et al., 2020] the principle from PU learning is combined with contrastive learning by considering the other samples in the batch not as negatives but as unlabeled. Thus smaller distances in the representations of very similar samples from the batch are enabled, i.e. a better clustering of classes. Chuang et al. [2020] reformulate the loss from equation (2.9) using the approach from equation (2.5) as follows:

With hyperparameter τ^+ simulating the prior of the classes, i.e. the fraction of class size compared to the whole sample size, the augmented samples M coming from the same original observation as z_i, the other augmented samples U, lower bound $\exp(-1/t)$, and the notation of (2.9), the debiasing term $d_u(z_i)$ can be described as

$$\begin{split} d_u(z_i) = \max\Bigg\{ & \exp(-1/t), \\ & \frac{1}{1-\tau^+}\left(\frac{1}{|U|}\sum_{u\in U}\exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_u\right)/\tau\right) - \tau^+ \frac{1}{|M|}\sum_{m\in M}\exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_m\right)/\tau\right)\right)\Bigg\} \end{split} \tag{2.11}$$

with $|M| = 1$ in most methods like SimCLR, because only two augmentations are generated per sample. This debiasing term can be plugged into the term for negatives in the denominator of equation (2.9), resulting in the debiased contrastive loss:

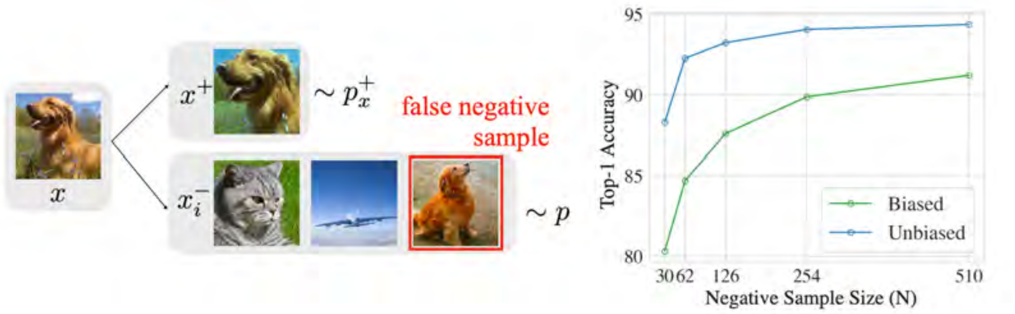


Figure 2.4: Debiased contrastive learning by Chuang et al. [2020]. Self-supervised learning falsely treats the augmentations of the other images in the batch as negative, even if they belong to the same class. Unlike supervised contrastive loss, debiased loss does not need class labels to treat other images of the same class as positives as well, but tries to reduce the erroneous maximization to similar images by debiasing the loss function without using class labels. Treating only samples from different classes as negatives (unbiased) achieves better performance than the standard setting (biased).

$$\mathcal{L}_{deb}(z_i, z_j) = -\log \frac{\exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)/\tau\right)}{\exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)/\tau\right) + |U| \cdot d_u(z_i)} \tag{2.12}$$

The idea of debiasing the contrastive loss is visualized in figure 2.4.

Thus, it was shown how self-supervised representation learning can benefit from approaches of PU learning. On the other hand, we are not aware of any application that explicitly investigates the effect of self-supervised learning for solving a PU problem. We apply this approach in this thesis in chapter 3.

2.4.3 Positive-Unlabeled Learning on Imbalanced Data

In many of the application areas of PU learning mentioned in subsection 2.1.1, there is a high class imbalance with usually little data belonging to the positive class, such as in disease diagnosis or fraud detection. However, the usually successful approaches using the nnPU loss from equation (2.6) by Kiryo et al. [2017] are not adjusted for class imbalance.

One possible approach would be to oversample the minority positive class $y = 1$, but since only the labeled positives $s = 1$ are known and there could be other positives $(y = 1|s = 0)$ in the unlabeled class, such strategies cannot be applied directly to PU data (figure 2.5). Su et al. [2021] develop a reweighting strategy for imbalanced PU learning for this purpose, which simulates oversampling of the true

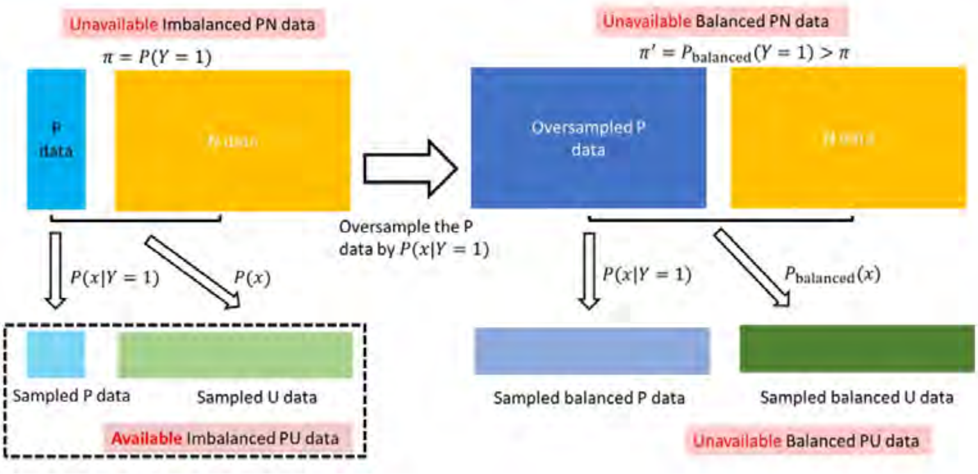

Unavailable Imbalanced PN data
π = P(Y = 1)
P data
N data
Oversample the P data by P(x|Y = 1)
P(x|Y = 1)
P(x)
Sampled P data
Sampled U data
Available Imbalanced PU data
Unavailable Balanced PN data
π' = P_balanced(Y = 1) > π
Oversampled P data
N data
P(x|Y = 1)
P_balanced(x)
Sampled balanced P data
Sampled balanced U data
Unavailable Balanced PU data

Figure 2.5: Data generating process of imbalanced PU data the imbalanced nnPU loss is trying to exploit for training a fair PU classifier Su et al. [2021]. The true underlying imbalanced PN data, from which the imbalanced PU data is sampled, could generate a balanced PN dataset using oversampling, from which balanced PU data could be sampled. However, only imbalanced PU data is available.

minority positive class and thus outperforms other approaches on evaluation metrics for imbalanced data, like the F1-score. The new loss results as

$$\begin{aligned}\mathcal{L}_{\text{ImbnnPU}}(g) =& \frac{\pi}{n_{s=1}} \sum_{x_i \in X|s=1} \ell\left(g\left(x_i\right), 1\right) + \max\Bigg(0, \\ & \frac{1-\pi'}{n_{s=0}(1-\pi)} \sum_{x_i \in X|s=0} \ell(g(x_i), 0) - \frac{(1-\pi')\pi}{n_{s=1}(1-\pi)} \sum_{x_i \in X|s=1} \ell\left(g\left(x_i\right), 0\right)\Bigg)\end{aligned} \tag{2.13}$$

where the parameter $\pi' \mathrel{\widehat{=}} P_{balanced}(Y = 1)$ represents the proportion of the true positive samples in the distribution of the simulated balanced dataset and, hence, is set to 0.5 by the authors. The other components correspond to those of equation (2.6).

Approaches that attempt to solve the problem of imbalanced PU learning in other ways include *ProbTagging* [Jiang et al., 2020], an aggregation method using probabilities calculated based on the similarity to nearest neighbours, or an AUC optimization method adjusted for PU data [Sakai et al., 2018].

Chapter 3

Novel Self-supervised Approach for Imbalanced PU Learning

3.1 Background

As discussed in subsection 2.4.1, contrastive self-supervised pre-training is very effective for classification on imbalanced data and does not require prior knowledge of class labels. Kang et al. [2019] showed that decoupling a representation learning network $f(\cdot)$ and then training a linear classifier $g(\cdot)$ can yield excellent results in this context. In particular, this is the case even if only $g(\cdot)$ is adjusted on the imbalanced setting, for example by reweighting or resampling, and $f(\cdot)$ is not adjusted with respect to class imbalance. Some promising methods for classification on long-tailed data are based on this two-step approach [Chen et al., 2022], [Marrakchi et al., 2021], [Li et al., 2021], using variations of supervised contrastive learning for the representation training of $f(\cdot)$.

The method of Su et al. [2021] described in subsection 2.4.3 improves the performance of nnPU of Kiryo et al. [2017] (subsection 2.1.3) in a PU scenario with a low proportion of positive samples. However, it does not solve some challenges of imbalanced PU learning with deep neural networks, such as sufficient feature learning of the underrepresented positive class, stable and convergent learning of the PU classifier, or robustness to mis-specification of the class prior π, which is often not known exactly in real-world use cases.

Since most of the data is unlabeled in this setting with only a few positive labeled samples, a contrastive self-supervised objective could help to extract useful additional information for a good classifier. In the following, we develop a novel framework to investigate to what extent PU learning on imbalanced data can benefit from self-supervised pre-training. To the best of our current knowledge, there are no published articles for this use case and we are the first to investigate this approach in more detail.

3.2 Methodology

In our method, we follow the idea of Kang et al. [2019] to train a feature extractor $f(\cdot)$ in the first step and then train a simple classifier $g(\cdot)$ on the encoded features for class prediction of the PU data in the second step.

The training process of our following two-step framework is shown in Figure 3.1. For training $f(\cdot)$, we use the backbone of the *SimCLR* method discussed in subsection 2.3.3 [Chen et al., 2020b], which creates two noisy versions x_i and x_j for the observations x via an augmentation module $\mathcal{T}$. Their representations h_i and h_j are created by $f(\cdot)$. From these, a simple shallow projector network $p(\cdot)$ is used to create the projections z_i and z_j. On top of that, the debiased contrastive loss $\mathcal{L}_{deb}(z_i, z_j)$ by Chuang et al. [2020] (see equation (2.12) and subsection 2.4.2) is computed.

After $f(\cdot)$ is trained, the representations $h = f(x)$ of the original data x are generated and given as input to the classifier $g(\cdot)$. Here, $g(\cdot)$ is just a linear layer that produces the 1-dimensional output $g(h)$. On this, the loss $\mathcal{L}_{imbnnPU}(g(h), s)$ is computed for the PU classification problem, where s is the label of the PU problem, with $s = 1$ if x is positive and labeled, and $s = 0$ if x is unlabeled. For $\mathcal{L}_{imbnnPU}(g(h), s)$, the imbalanced nnPU loss [Su et al., 2021] from equation (2.13) is used which proved to be successful on imbalanced PU data, with the sigmoid loss from equation (2.7) used as the surrogate loss $l(\cdot, \cdot)$. In this second step, only $g(\cdot)$ is trained, in the encoder $f(\cdot)$ the weights are frozen and it is only used in the forward pass to compute h.

3.3 Experiments

3.3.1 Image Augmentation

For the image augmentation module $\mathcal{T}$, we use augmentations with the same parameters as *SimCLR* [Chen et al., 2020b], which have proven to be most successful for good representation learning: 1. *random cropping* and resizing of the crop to the original image size [Szegedy et al., 2015] with random flip. 2. *color distortion*, consisting of color dropping, where the image is turned to grayscale with a selected probability, and color jittering, doing random changes of brightness, contrast, saturation and hue in the images [Howard, 2013].

3.3.2 Deep Representation Network Architecture

As feature extractor $f(\cdot)$ we use ResNet-50 [He et al., 2016], as in the default SimCLR. ResNet-50 is a 50-layer deep residual convolutional neural network for image

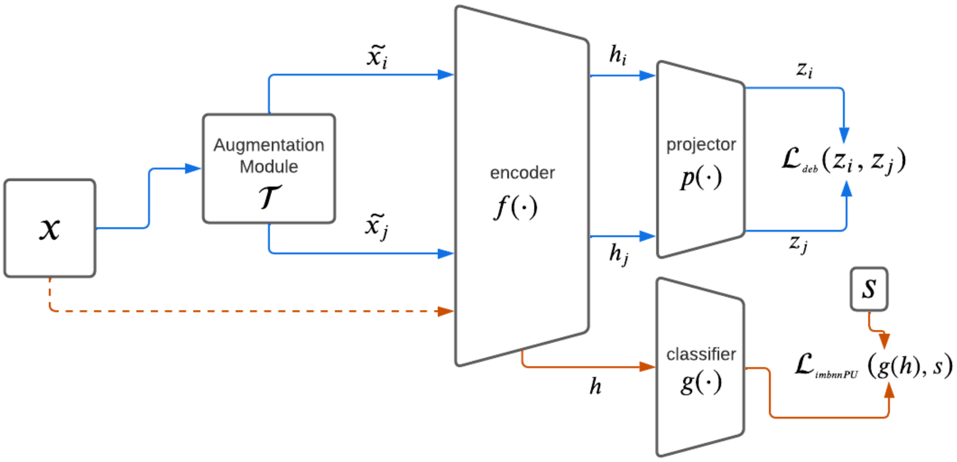


Figure 3.1: Training procedure of self-supervised representation learning for imbalanced PU learning. Step 1: Pre-training (blue). From the observations x, the augmentations $\tilde{x}_i$ and $\tilde{x}_j$ are generated and sent through an encoder $f(\cdot)$ and a projector $p(\cdot)$. On the outputs z_i and z_j, the debiased loss is applied for clustering of similar observations, independent of the PU label s. Step 2: Classifier training (orange). After pre-training, the weigths of $f(\cdot)$ are frozen and the representations h of the original observations x are generated (dotted). On h, the linear classifier $g(\cdot)$ is trained using the imbalanced nnPU loss.

recognition, which has achieved state of the art results in many different challenges in classification and object detection, among others.

Also like in SimCLR, for representation learning in pre-training the default linear classification head of ResNet-50 with 2048 → 1000 dimensions is replaced by the 1-hidden-layer network projection head $p(\cdot)$ with $2048 \rightarrow 2048 \rightarrow ReLU \rightarrow 128$ dimensions with non-linear $ReLU$ activation function [Nair and Hinton, 2010], to calculate the projections z. As explained in subsection 2.3.3, the performance of representations h on downstream tasks improves by introducing this non-linear projection head $p(\cdot)$ before computing the contrastive loss.

3.3.3 Optimization

For self-supervised pre-training we use a batch size of 128, which leads to 256 different samples in the batch due to the random twin augmentations from 3.3.1. As hyperparameters of the debiased loss, we choose $\tau^+ = 0.1$ and $\tau = 0.5$ after Chuang et al. [2020]. As optimizer we use the base *Adam* optimizer [Kingma and Ba, 2014] with a learning rate $\gamma = 3e^{-4}$. We train $f(\cdot)$ for 100 epochs.

We then train the linear classifier $g(\cdot)$ for 100 epochs as well, with a batchsize of 256, the optimizer is set to Adam with $\gamma = 3e^{-4}$. As hyperparameters of the imbalanced nnPU loss $\mathcal{L}_{imbnnPU}(g(h), s)$ (2.13) we follow [Su et al., 2021] and set $\pi' = 0.5$ and $\pi = p(y = 1|s = 0)$ by proportion of positives in the unlabeled samples per dataset.

3.3.4 Datasets and Tasks

We want to investigate how well our framework performs on classifying imbalanced positive unlabeled image data. For this we use the well-studied open source datasets CIFAR-10 and CIFAR-100[1]. We also explore the application to a medical image dataset for glaucoma classification, following Diaz-Pinto et al. [2019].

CIFAR-10

The train dataset of CIFAR-10 consists of 50,000 images from 10 different classes with 5,000 images per class, the test dataset consists of 10,000 images, also class balanced. In previous studies for PU learning [Kiryo et al., 2017], [Chen et al., 2020d], [Chen et al., 2020a], [Dorigatti et al., 2022], the 10 classes were divided into the 2 super classes "vehicles" (4 classes) and "animals" (6 classes), and one of the two super classes was defined as positive. Of the positive class, a fraction c was considered positively labeled $s = 1$ to mimic the label probability $c = p(s = 1|y = 1)$, and the remainder was considered unlabeled $s = 0$.

However, this setting produces an approximately balanced data set between positives and negatives (2:3), whereas we want to investigate the approach to imbalanced data. Consequently, we define the "vehicles" (4 classes) as the positive class and downsample the positives in the train dataset to only 3,000 samples, resulting in a 1:10 positives : negatives ratio. Like Su et al. [2021] we set c to 0.2, so that we have a total of 600 labeled positives and 32,400 unlabeled observations in the train dataset. In the test dataset, we continue to use the nearly balanced distribution of the two classes, so that evaluation via naive performance metrics such as the *accuracy* is still possible.

CIFAR-100

CIFAR-100 consists of a train dataset with 50,000 images and a test dataset with 10,000 images which can be divided into 100 balanced classes. These classes can be grouped into 20 balanced super classes containing 5 classes each. We define the two similar super classes "vehicles 1" and "vehicles 2" as positive and the remaining 18 super classes as negative. Thus we achieve a positives : negatives ratio of 1:9 and no downsampling has to be done. We set c to 0.2 again, so in total there are 1,000 labeled positives and 49,000 unlabeled samples in the train dataset. In this case,

[1] available at: `https://www.cs.toronto.edu/~kriz/cifar.html`

the imbalanced ratio also exists in the test dataset, so that metrics suitable for an imbalanced scenario must be used for evaluation, such as the F1-score.

GLAUCOMA
Glaucoma is an eye disease that can lead to blindness. In fundus images showing the retina of patients, in addition to arteries and veins, the optic disc is visible. The optic disc can be divided into optic cup, a bright center, and neuro-retinal rim, a slightly darker area around the center. Here, an abnormal size of the optic cup compared to the optic disc is an indication of glaucoma disease, which should be detected [Diaz-Pinto et al., 2019].

As dataset, we use the labeled observations of the dataset used by Diaz-Pinto et al. [2019]. This merges several glaucoma datasets [Zhang et al., 2010], [Sivaswamy et al., 2014], [Medina-Mesa et al., 2016] [Köhler et al., 2013] into one, since the individual datasets contain relatively few observations. In total there are 2,357 samples, 956 with glaucoma (positive) and 1401 without glaucoma (negative). In the absence of a test dataset, we randomly select 85% of the samples as the train dataset and 15% as the test dataset, and again label $c = 0.2$ of the positive samples. The final result is 163 labeled positive and 1,840 unlabeled observations in the train dataset.

For the evaluation, we use the same procedure as most other papers in PU learning, such as Kiryo et al. [2017][Chen et al., 2020d], [Chen et al., 2020a], [Dorigatti et al., 2022]. Here, the classifier is trained on the positive and unlabeled data from the train dataset and the performance is reported on the fully labeled test dataset, exclusively. The artificially generated unlabeled samples from the train dataset and their actually known labels are not included in the evaluation.

3.4 Results

In the following, we describe the performance of our framework under different scenarios, hereafter referred to as *debiased+imbnnPU*. As a comparison method, in the following called *imbnnPU*, we use the loss $\mathcal{L}_{imbnnPU}$ of Su et al. [2021] directly for training a model in a 1-step-manner, without pre-training and without decoupling feature extractor and classifier, as usual before. For good comparability, we also use ResNet-50 as the backbone for this model, which means that in both methods the final model has the same number of parameters. For optimization we also use the settings from subsection 3.3.3.

3.4.1 Classification Performance

Table 3.1 shows the results of our method debiased+imbnnPU on the test datasets. For our artificially generated imbalanced PU datasets CIFAR-10 and CIFAR-100, our method clearly achieves better results in terms of accuracy, F1-score and AUC of the test dataset compared to the simple ResNet-50 model with imbnnPU loss without pre-training. For CIFAR-10 there are performance improvements of 8.8%, 11.0% and 5.4%, for CIFAR-100 2.4%, 18.5% and 12.7%. For CIFAR-100, the test data set is imbalanced, so the accuracy and the minor improvement of only 2.4% should not be overvalued because it does not adjust for the different number of samples per class.

In addition, for CIFAR-10 we performed the pre-training with the original NT-Xent loss (2.9) used in SimCLR instead of the debiased loss, with the same settings. In comparison to NT-Xent, the debiased loss achieves an improvement of 1-1.5% in the metrics.

For Glaucoma, we obtain hardly any differences in performance between models with and without pre-training. This suggests that in the self-supervised pre-training no additional information helpful for classification was stored in the representations, but at the same time decoupling does not degrade performance. It should be noted that the used self-supervised architecture of Chen et al. [2020b] with choice of image augmentations was optimized for typical benchmark datasets like CIFAR-10 and CIFAR-100, while the transferability to other datasets with different underlying structures, like in medical imaging, is not ensured.

	method	Accuracy	F1	AUC
CIFAR-10	imbnnPU	86.5	83.0	93.6
	NT-Xent + imbnnPU	94.3	92.5	97.9
	debiased + imbnnPU	**95.3**	**94.0**	**99.0**
CIFAR-100	imbnnPU	*86.7*	44.1	82.9
	debiased + imbnnPU	***89.1***	**62.6**	**95.6**
Glaucoma	imbnnPU	**75.0**	67.0	77.7
	NT-Xent + imbnnPU	74.6	67.3	**77.8**
	debiased + imbnnPU	74.2	**68.3**	77.3

Table 3.1: Results of self-supervised pre-training for imbalanced PU learning vs. imbalanced PU learning without pre-training. Best performances per dataset and measure are **bold**.

Competitors

In PU learning, there are only few approaches so far that focus on the application to imbalanced data with few positive samples. Su et al. [2021] have shown that the imbalanced nnPU loss outperforms many other methods in this setting, such as *nnPU* [Kiryo et al., 2017], *self-PU* [Chen et al., 2020d], *SMOTE* [Chawla et al., 2002], or *SSImbalance* [Yang and Xu, 2020].

For the usual balanced scenario, there are some state-of-the-art methods that report their results for CIFAR-10 trained on the full train dataset with the approximately balanced split "vehicles" : "animals" with ratio 2 : 3, but they are not explicitly suitable for imbalanced learning. Since our debiased+imbnnPU method, as described above, far outperformed the simple ResNet-50 with imbnnPU method, we want to check whether this two-step architecture with pre-training and a separate classifier can achieve similar performances to current SOTA-PU methods even when training on less data overall and with class imbalance.

As comparison methods, we select *VPU* [Chen et al., 2020a], *PAN*[Hu et al., 2021], *Self-PU* [Chen et al., 2020d] and *PUUPL* [Dorigatti et al., 2022] and use the performance reported in the articles for CIFAR-10 on the same class split "vehicles" vs. "animals". Here, all 20,000 samples of the positive "vehicles" class are used for training or validation, of which 3,000 (or 1,000) are known to be labeled, whereas we use only 3,000 positive samples with 600 labeled. The test dataset remains the same in both scenarios.

Table 3.2 shows the accuracy of the competitors. Our method clearly surpasses the other baselines, and can improve the accuracy by 3.9% compared to the previous best method PUUPL. Moreover, we only need 66% of the train samples, 15% of the positive samples, and 20% of the labeled samples that PUUPL uses.

Method	Train Samples	Positives Labeled	Accuracy
VPU	50,000	3,000	89.5
PAN	50,000	1,000	89.7
Self-PU	50,000	3,000	90.8
PUUPL	50,000	3,000	91.4
imbnnPU	**33,000**	**600**	86.5
debiased+imbnnPU	**33,000**	**600**	**95.3**

Table 3.2: Performance of SOTA-competitors trained on balanced PU CIFAR-10. Best performance and fewest resources needed are **bold**.

Comparison to supervised baseline

The loss $\mathcal{L}_{imbnnU}$ (2.13) we use for our experiments actually tries to minimize the empirical risk (2.4) on the underlying true binary classification problem with the true label y by reweighting and reformulating the loss $l(\cdot, \cdot)$ on the PU scenario using the PU label s and adjusting for imbalance. Consequently, the performance of a model with the same architecture trained on the actually unknown y can be viewed as the upper bound we are trying to achieve.

To investigate how close our framework comes to this upper bound and whether pre-training can be helpful to reduce the gap between PU and fully supervised performance, we train both models in a supervised setting on the true labels y using a *weighted binary cross-entropy* (wBCE) loss on the output $g(\cdot)$

$$\ell_{wCE}(g_i, y_i) = -w_{pos} y_i \cdot \log \sigma\left(g_i\right) + \left(1 - y_i\right) \cdot \log\left(1 - \sigma\left(g_i\right)\right) \tag{3.1}$$

where $\sigma(g_i) = \frac{1}{1+exp(-g_i)}$ and $w_{pos} = |positves|/|negatives|$ per dataset. In 3.3 the performances of the supervised models on both methods are displayed.

As expected, also in the imbalanced supervised setting the debiased+wBCE variant achieves better results than the simple training of the weighted BCE. However, it is also noticeable that the difference in the evaluation metrics within the same architecture between PU setting and binary setting is smaller for the methods with debiased pre-training than for the single model methods. For example, the difference in F1-score and AUC for CIFAR-10 (CIFAR-100) in the two-step setting is 0.9% and 0.4% (4.2% and 1.5%), whereas in the simple training it is 5.3% and 3.2% (9.2% and 5.3%). This shows that self-supervised pre-training is not only useful for the problem of class imbalance, but additionally applied to PU learning, it helps to reduce the gap in performance to supervised learning approaches.

	data	method	Accuracy	F1	AUC
CIFAR-10	PU	imbnnPU	86.5	83.0	93.6
		debiased + imbnnPU	95.3	94.0	99.0
	supervised	weighted BCE	91.0	88.3	96.8
		debiased + wBCE	95.9	94.9	99.4
CIFAR-100	PU	imbnnPU	*86.7*	44.1	82.9
		debiased + imbnnPU	*89.1*	62.6	95.6
	supervised	weighted BCE	*91.0*	53.3	88.2
		debiased + wBCE	*91.7*	68.8	97.1

Table 3.3: Performance with and without debiased pre-training for PU vs. supervised data.

3.4.2 Quality of Learned Representations

The main feature of our debiased+imbnnPU method compared to the competitors is the representation learning that is decoupled from the actual classification task in the first step. In the second step, it even uses the same architecture in the forward pass as our implementation of imbnnPU, except that only the weights in the last linear layer are updated, whereas imbnnPU trains the complete ResNet-50 architecture. Since our method achieves much better results in the evaluation, the quality of the representation before the last linear layer is obviously significant for the quality of the classifier.

Accordingly, the representations for the two models are examined in more detail below. Figure 3.2 shows the t-SNE [Van der Maaten and Hinton, 2008] visualizations of the 2048-dimensional representations of the test dataset of CIFAR-10 of both models, trained on the imbalanced PU CIFAR-10 train dataset. t-SNE is a method for visualizing high-dimensional data that clusters similar observations together.

On the right side, one can see the observations color-coded according to their binary classes. It can be seen that there is slightly less overlap between the two classes in debiased pre-training than in imbnnPU. This is the case even though debiased pre-training has no information about the class labels. One reason for this may be the lack of ability to generalize in the low-labeled positive class. ImbnnPU is driven in updating the weights of its network only by its loss function, which depends only on the label information. This can limit the ability to create good representations even in normal imbalanced binary settings, which is why a decoupling between representation learning and classifer learning can be helpful [Kang et al., 2019]. This effect can be amplified in the PU setting, where the lack of generalization of the representations makes it harder to recognize unlabeled positives as positives despite adjusting the loss function, which further complicates to achieve separability between the two classes. Better discriminance of the representations before the linear classifier is helpful to determine a good classifier, as stated in subsection 2.1.2.

In contrast, the robustness of contrastive self-supervised pre-training on imbalanced data has been well studied [Yang and Xu, 2020], [Liu et al., 2021]. In our application, the debiased contrastive loss using randomly augmented inputs generate robust representations on sample level, but also allow clustering of similar samples without requiring label information. Our method also detects similar features of substructures within the two classes, which is not directly incentivized by the label-based imbnnPU loss function. On the left side the same representations are shown as on the right side, but this time color-coded by their original 10 subclasses. You can see that with imbnnPU there are hardly any clusters within the two classes with respect to the labels of the subclasses. In debiased contrastive pre-training, on the other hand, clusters form around the 10 subclasses, even if the 4 positive vehicle-classes having less training samples have slightly worse boundaries than the 6 negative animal-classes.

These substructures in the representations could help to identify the correct positives in the unlabeled ones, even with only a few labeled positive samples. This principle is also found, among others, in few-shot learning [Wang et al., 2020], where a classifier is also trained using representation learning and few labeled data. Moreover, in our framework, decoupling the feature extractor from the classifier training prevents the representations from being altered by the imbnnPU loss and thus becoming less intra-class distinguishable between the subclasses.

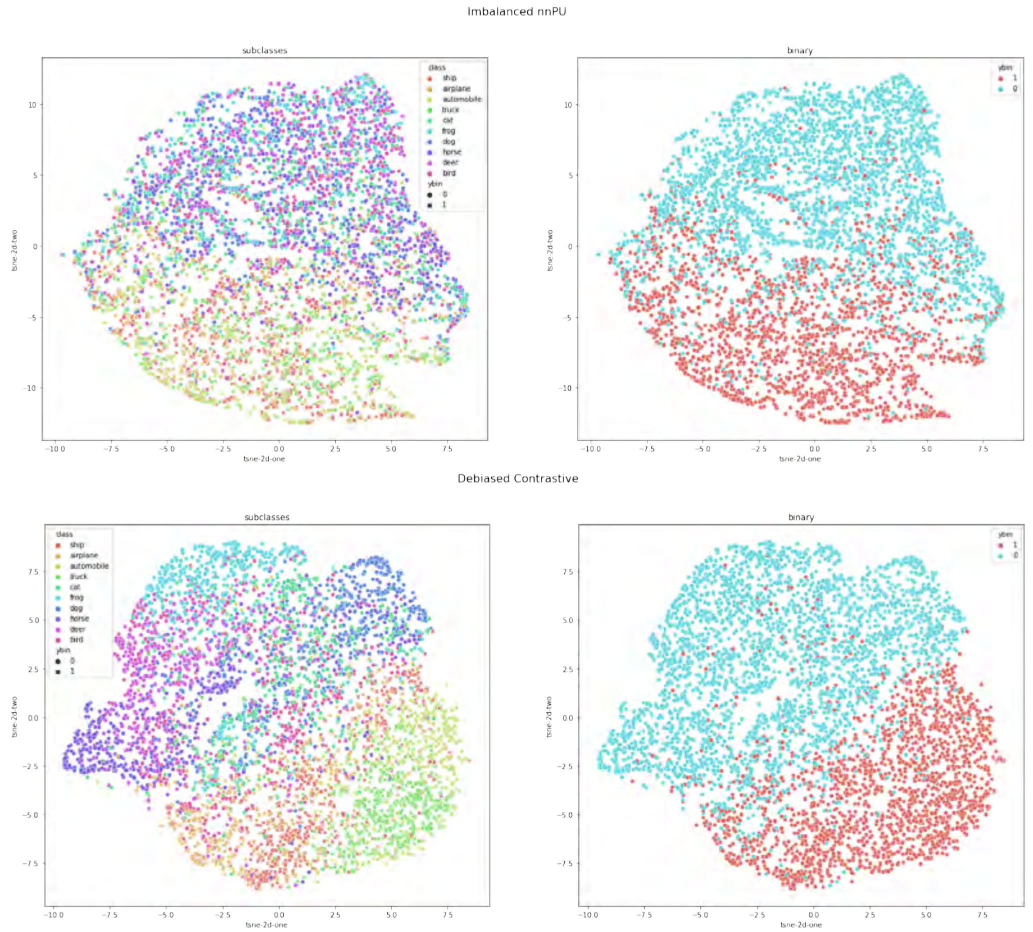


Figure 3.2: t-SNE visualization of representations on test dataset of CIFAR-10. Top: ResNet-50 trained on imbalanced nnPU loss without pre-training. Bottom: After pre-training on debiased contrastive loss. Color-coded for underlying 10 subclasses (left) and binary classes (right).

3.4.3 Robustness Against Mis-specification of the Class Prior

In the previous analyses, we always assumed the class prior π to be known, and in the imbnnPU loss for each dataset to be set to the proportion of positive samples in the unlabeled samples, following Su et al. [2021] and Kiryo et al. [2017] . In real-world applications, however, this proportion is often not known and must be estimated using domain knowledge or other methods, as discussed earlier in subsection 2.1.2. This can often result in erroneous estimates that can degrade the performance of the model. In the following, we investigate how mis-specification of π in $\mathcal{L}_{imbnnPU}$ changes the goodness of the model.

The accuracy and F1-score depend strongly on the chosen decision boundary deciding when an output of a classifier $g(\cdot)$ is predicted to be positive or negative. By using an incorrect class prior, the outputs of the model can be shifted and yield worse results in these metrics, even if the monotony of the score $g(\cdot)$ is further preserved, and thus the model could still be suitable as a ranking model. To be able to evaluate the goodness of the models under different false priors independently of the decision boundary, the AUC is used as a metric.

In figure 3.3, for the two models imbnnPU and debiased+imbnnPU the course of the AUC on the test dataset of CIFAR-10 is displayed over the 100 training epochs of the classifier, using distorted priors. In the figure there is one run for each distortion factor b_{dis} of the true prior π, so that in each case in $\mathcal{L}_{imbnnPU}$ the prior π was replaced by the distorted prior $\pi_{dis} = b_{dis} \cdot \pi$. Here, b_{dis} varies from 0.1 to 10.0, with 1.0 yielding the model with the unbiased π.

In general, it is noticeable that our method clearly has a higher AUC than the model without self-supervised pre-training, even under different mis-specified priors. The variance is also clearly smaller and the course is more stable, since only the weights of the classifier and not the entire ResNet-50 parameters are trained. For both models, no deterioration of the performance can be observed for an underestimation of π with $b_{dis} < 1$. For an overestimation with $b_{dis} > 1$, a deterioration is visible for $b_{dis} \geq 5$ for both models, although the deterioration of the performance is still clearly lower for our method. At an extreme overestimation of $b_{dis} = 10$, strong deviations in stability are shown and the AUC of our model starts to decrease during training. At the same time, the AUC at the beginning is above 95% due to the previous representation learning and also in the remaining cases, hardly any improvement is seen after a few epochs. Therefore, it can be summarized that in our method debiased+imbnnPU, with sufficient representation learning in the first step, the training of the classifier could be limited to a few epochs. So even with strong mis-specification of the class prior, high stability and performance could be observed.

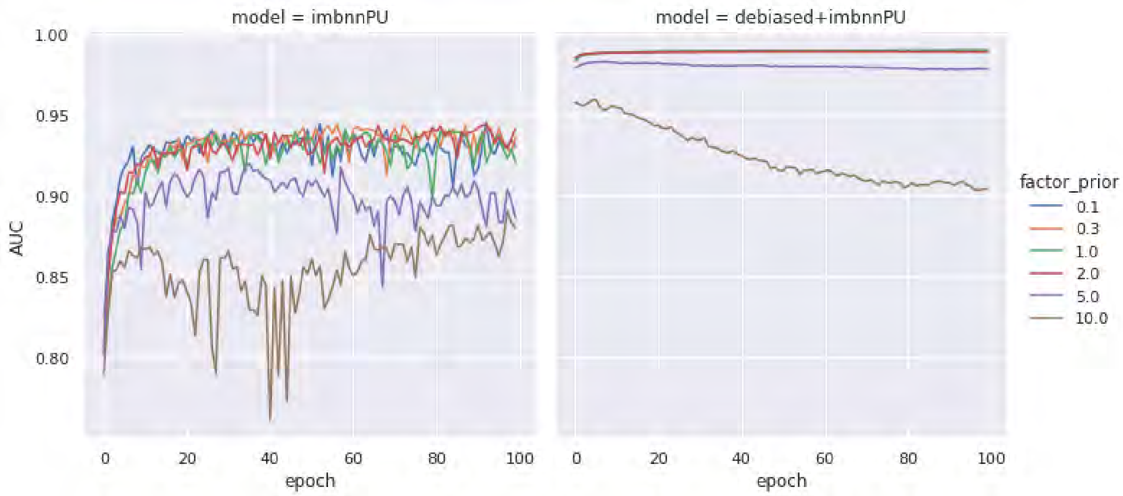


Figure 3.3: Robustness against prior mis-specification for different distortion factors b_{dis} of the prior. AUC of the test dataset of CIFAR-10 is plotted over 100 training epochs for imbnnPU (left) and for the classifier of debiased+imbnnPU after finished pre-training (right).

Chapter 4

Novel Contrastive Method for Learning Imbalanced PU Distribution

4.1 Background

In the previous chapter we showed how PU learning on imbalanced data can benefit from decoupled contrastive representation learning. This pre-training was done in a self-supervised manner completely without label information, although for some samples the positive class label is known. Khosla et al. [2020] showed that by including label information in the pre-training, supervised contrastive learning can further increase the quality of the representations compared to self-supervised pre-training. Therefore, in other approaches that also use decoupled two-step architectures for classification on imbalanced data, the supervised contrastive loss from equation (2.10) is used for feature learning in the first step [Marrakchi et al., 2021], [Chen et al., 2022]. Thus, at the same time, in pre-training the class labels can already be used to further improve the representations of the minority classes, e.g., via random oversampling, reweighting, or other adjustments of the supervised contrastive loss [Kang et al., 2020], [Wang et al., 2021], [Cui et al., 2021], [Li et al., 2021].

In the case of PU data, a naive application of the supervised contrastive loss does not make sense or is not directly possible, since the true class y is not known for the unlabeled data. If one naively considers the unlabeled observations as negatives, the supervised contrastive loss erroneously minimizes the distance of the representations of the unlabeled positives to the negatives, making a subsequent classification of the two classes difficult. Accordingly, in the following we want to investigate to what extent the supervised contrastive loss can be adjusted for imbalanced PU data to enable a more discriminative representation learning.

4.2 Methodology

Novel contrastive imbalanced nnPU loss

In supervised loss (2.10), the loss for an observation x_i or its projection z_i can be represented as:

$$\mathcal{L}_{supCon}(z_i, s) = \frac{1}{|J|} \sum_{j \in J} - \log \frac{\exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right) / \tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_k\right) / \tau\right)} \tag{4.1}$$

where $J = J(i, s) = \{j \in (2N \setminus \{i\}) : s_j = s\}$ and usually, the class label s is set to be the class label s_i of observation x_i, so the distance of projection z_i of observation x_i to all samples with the same class is minimized. For s, however, a different class can also be selected, so that the distance of z_i to observations of a different class $z_j, j \in J(i, s)$ is minimized.

For our representation learning method, we would like to take advantage of $\mathcal{L}_{imbnnPU}$ (2.13) as it adjusts for PU data and class imbalance at the same time. $\mathcal{L}_{imbnnPU}$ uses a surrogate loss $\ell(g(x_i), s)$, which is usually the loss between the predicted probability of classifier $g(x_i)$ for input x and PU label s, e.g. the sigmoid loss (2.7). $\ell(g(x_i), s)$ and $\mathcal{L}_{supCon}(z_i, s)$ follow a similar behavior, because $\ell(g(x_i), s_i)$ minimizes the prediction error between prediction $g(x_i)$ and label s, whereas $\mathcal{L}_{supCon}(z_i, s)$ minimizes the distance of projection z_i to projections $z_j, j \in J(i, s)$ of class s. Hence, we plug $\mathcal{L}_{supCon}(z_i, s)$ as surrogate loss $l(\cdot, \cdot)$ into $\mathcal{L}_{imbnnPU}$ in equation (2.13).

With positive samples $P = \{j \in 2N : s_j = 1\}$, unlabeled samples $U = \{j \in 2N : s_j = 0\}$, $n_p = |P|$, $n_u = |U|$, we thus obtain the novel *contrastive imbalanced nnPU* (connPU) loss

$$\begin{aligned} \mathcal{L}_{connPU} = & \frac{\pi^{'}}{n_p(n_p - 1)} \sum_{i \in P} \sum_{j \in P, i \neq j} l(z_i, z_j) + max\big(0, \\ & \frac{(1 - \pi^{'})}{n_u(n_u - 1)(1 - \pi)} \sum_{i \in U} \sum_{j \in U, i \neq j} l(z_i, z_j) - \frac{(1 - \pi^{'})\pi}{n_p n_u (1 - \pi)} \sum_{i \in P} \sum_{j \in U} l(z_i, z_j)\big) \end{aligned} \tag{4.2}$$

with

$$l(z_i, z_j) = - \log \frac{\exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right) / \tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp\left(\operatorname{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_k\right) / \tau\right)}$$

and $\operatorname{sim}(z_i, z_j) = z_i^T z_j / (||z_i|| \cdot ||z_j||)$ being the cosine similarity of the two vectors.

In theory, $\mathcal{L}_{connPU}$ thus tries to cluster the labeled positives together, as in supervised contrastive loss, but instead of clustering all unlabeled samples together, it also tries to shift unlabeled observations that are similar to the positive class

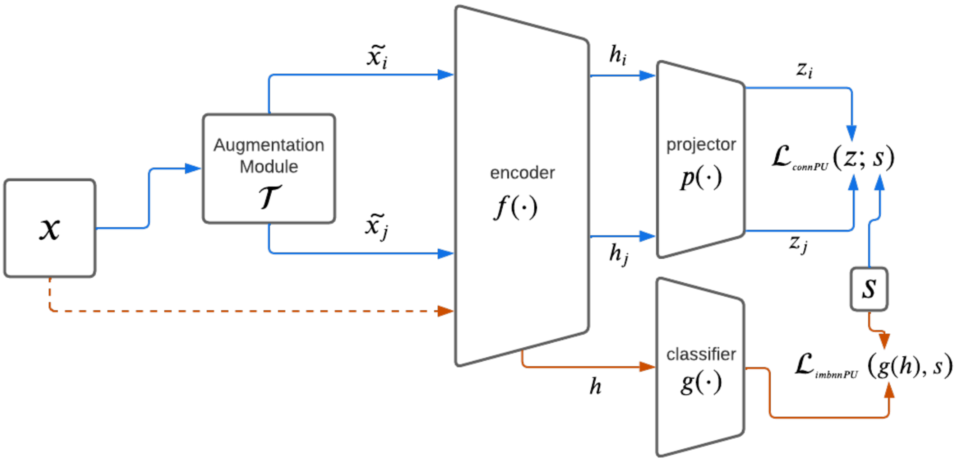


Figure 4.1: Training procedure of connPU for imbalanced PU learning. Most of the training procedure remains as shown in Figure 3.1. Instead of debiased loss, connPU loss is substituted for representation learning. Here, the knowledge about the PU lables s is already included in the pre-training of $f(\cdot)$ (blue). There are no changes in the training of $g(\cdot)$ in the second step (orange).

towards the positive class. In addition, the reweighting gives more weight to the minority positive class to form better representations.

Training Procedure

For the training procedure we stick to the same two-step setup with decoupled feature extractor $f(\cdot)$ and classifier $g(\cdot)$ as for the self-supervised approach described in subsection 3.2. Here, only when training $f(\cdot)$, the debiased loss $\mathcal{L}_{deb}$ is replaced by our new contrastive imbalanced nnPU loss $\mathcal{L}_{connPU}$. Accordingly, knowledge about the PU label s is already included in the pre-training loss and hence the learning of the representations. The slight modification of the training procedure is shown in Figure 4.1. In the following, our new approach will be referred to as *connPU+imbnnPU*.

4.3 Experiments

For image augmentation and the architecture of $f(\cdot)$ and $g(\cdot)$ we choose the same settings as described in the self-supervised framework in subsections 3.3.1 and 3.3.2.

For optimization of $\mathcal{L}_{connPU}$, we also set $\pi' = 0.5$ and π to the fraction of positives in the unlabeled samples per dataset. In addition, we train the classifier $g(\cdot)$ for only 5 epochs after pre-training, based on the findings from the previous chapter that the maximum classifier performance is reached already after a few epochs. Furthermore, label information is already leveraged in pre-training, which should incentivize a direct subdivision into the two classes in the representations. We choose the remaining settings and hyperparameters as described in the self-supervised setting in subsection 3.3.3.

Datasets
We use the same datasets from subsection 3.3.4 as for the previous models. As another dataset, we introduce *CIFAR-2*, an artificially created imbalanced subset of CIFAR-10, consisting of only 2 of the original 10 classes. Here, all 5000 images of the class "bird" are defined as true negatives (unlabeled), and 750 images of the class "plane" are defined as positives, of which 150 are considered labeled. The idea behind this is that in the splits of CIFAR-10 and CIFAR-100 used above, different subclasses are combined into positives and negatives, making the classes heterogeneous. We additionally want to investigate how connPU+immbnnPU performs on a dataset with higher homogeneity within the two classes and at the same time using clearly less training samples (5,750 with only 150 labeled compared to 33,000 with 600 labeled). The test dataset used is the 1,000 samples from each of the two classes from the balanced test dataset of CIFAR-10. To adjust for the lower number of samples, we train $f(\cdot)$ for 500 epochs for this dataset to ensure a similar number of update steps as in the other scenarios.

4.4 Results

4.4.1 Classfication Performance

Table 4.1 shows the performance of our method connPU+imbnnPU against the two methods imbnnPU and debiased+imbnnPU from the previous chapter. Debiased+imbnnPU still achieves the best results for the CIFAR-X datasets, with a huge performance boost of over 13% in all three metrics in the CIFAR-2 dataset compared to imbnnPU. For CIFAR-10 and CIFAR-2, also connPU+imbnnPU clearly outperforms imbnnPU with improvements in accuracy, F1-score and AUC for CIFAR-2 (CIFAR-10) of 9.3% (3.5%), 7.8% (4.2%) and 12.3% (2.3%), respectively. However, compared to debiased+imbnnPU, connPU+imbnnPU performs 4-7% worse in each of these data sets.

What stands out is the low performance of connpu+imbnnPU on the CIFAR-100 dataset, where it itself is far below imbnnPU. The problem with our pre-train loss $\mathcal{L}_{connPU}$ for this dataset might be that our loss tries to divide the representations into two clusters and to minimize the distances of the samples within them. In CIFAR-100, with 100 distinct subclasses, and 80 of them in the negative class, there is a high intra-class heterogeneity, especially in the negative class, which can lead to problems in representation learning and consequently to difficulties in classifier training for this method.

For Glaucoma, connPU+imbnnPU also achieves the lowest performance compared to the other methods. One reason for this may be that when even debiased+imbnnPU already has difficulties learning helpful representations at the sample level, clustering the representations into the two classes and trying to identify positives in the unlabeled observations further weakens the discriminance.

With a difference of about 5% in the three metrics, connPU+imbnnPU comes closest to the performance of debiased+imbnnPU in CIFAR-2, with only one subclass within positives and negatives. This supports the assumption that connPU+ imbnnPU particularly benefits from high homogeneity within the positive and negative classes.

		Accuracy	F1	AUC
	imbnnPU	86.5	83.0	93.6
CIFAR-10	debiased + imbnnPU	95.3	94.0	99.0
	connPU + imbnnPU	90.0	87.2	95.9
	imbnnPU	*86.7*	44.1	82.9
CIFAR-100	debiased + imbnnPU	*89.1*	62.6	95.6
	connPU + imbnnPU	*66.0*	32.4	78.5
	imbnnPU	75.0	67.0	77.7
Glaucoma	debiased + imbnnPU	74.2	68.3	77.3
	connPU + imbnnPU	63.7	60.4	65.8
	imbnnPU	76.8	78.2	80.8
CIFAR-2	debiased + imbnnPU	91.4	91.2	97.5
	connPU + imbnnPU	86.1	86.0	93.1

Table 4.1: Results for connPU pre-training vs. debiased pre-training and no pre-training on different datasets.

4.4.2 Quality of Learned Representations

As already mentioned, in the first step our framework connPU+imbnnPU tries to maximize the distances of the representations between both classes and to minimize them within the true binary classes by using the loss $\mathcal{L}_{connPU}$, which does not explicitly promote the separability of the representations on sample level and with respect to substructures within the class. This pattern is also evident in the visualization of the learned features. Figure 4.2 shows the t-SNE projection of the representations from pre-training with connPU for the test dataset of CIFAR-10. Compared to the debiased pre-training, there are no major differences in the discriminance of the represented features on the binary labels. However, if we look at the underlying 10 subclasses, we notice that the clusters of the subclasses are much weaker with connPU than with debiased representation learning. However, these structures can be helpful in training the classifier.

Another problem may be that when clustering the observations within the two classes is difficult, the focus of the model is more on minimization of the representations within classes (while additionally trying to minimize the distance to some unlabeled positives) than on maximization of the distance between classes. This can further diminish the discriminance of the representations, especially regarding the binary classes. This problem arises when there is large heterogeneity of observations in at least one of both classes, such as in CIFAR-100 or more generally when PU learning is applied to one-class-classification scenarios 2.1.1, with high variance in the negative class.

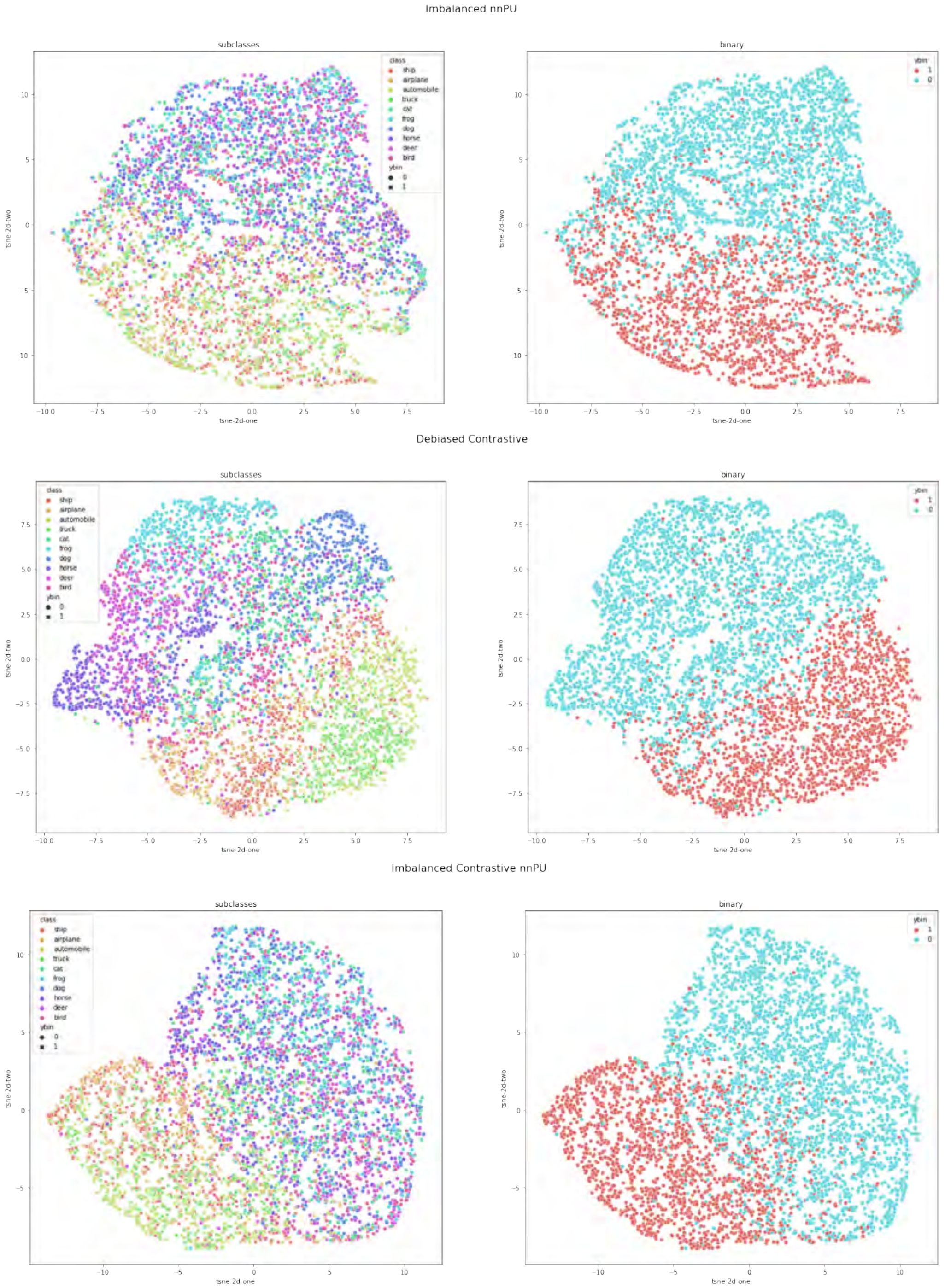


Figure 4.2: t-SNE visualization of representations on test dataset of CIFAR-10. Top: ResNet-50 trained on imbalanced nnPU loss without pre-training. Middle: After pre-training on debiased contrastive loss. Bottom: After pre-training on connPU loss. Color-coded for underlying 10 subclasses (left) and binary classes (right).

Chapter 5

Conclusion

In this thesis we investigated the application of self-supervised learning to the problem of positive-unlabeled learning on imbalanced data. We could show that by decoupling representation learning using contrastive learning in the first step and subsequent PU learning with re-weighting on a linear classifier in the second step, the performance on two image datasets could be clearly improved compared to training a simple model. Furthermore, we could show that the performance gap between PU learning and supervised learning could be further closed by this method and that the classifier is more stable in the training process and more robust against misspecification of the class prior. We were even able to outperform state-of-the-art PU frameworks on balanced data with our method for imbalanced data, requiring less training data and fewer labeled samples.

As another contribution, we developed and evaluated the connPU loss, an extension of the supervised contrastive loss for the scenario of imbalanced PU data. In our experiments, we were able to show that connPU benefits from high intra-class homogeneity during representation learning and runs into problems when intra-class heterogeneity is high.

Overall, the application of connPU could not achieve the results of self-supervised learning. Thus, we conclude that PU learning on imbalanced data can benefit from representation learning without incorporating label knowledge in the first step, and training a simple PU classifier in the second step.

However, as a limitation of our method, suitable objectives for self-supervised learning and the selection of data pre-processing and data augmentation methods are extremely important for good representation learning and consequently for a good classification result. By applying the default procedure and settings of SimCLR to a medical image dataset for glaucoma classification, we could not achieve any improvement over the model without pre-training.

5.1 Future Direction

One advantage of our work is the simple two-step methodology. In the pre-training step, our implementation of SimCLR with debiased contrastive loss can be replaced by any representation learning method. Thus, the method can also be applied to the problem of PU learning in other application domains with different data structures, such as in the fields of 3D imaging, video, natural language processing, or signal processing.

Another promising direction may be the study of the classification head. Since representation learning compresses unstructured data into a smaller vector format, more classical PU methods, such as bagging support vector machines [Mordelet and Vert, 2010], can be applied to it. Moreover, we adopted the SCAR assumption in our work for the labeling mechanism. An interesting research approach would be to investigate the applicability of our method in scenarios with more realistic labeling mechanisms, such as under the probabilistic gap assumption.

List of Figures

List of Tables

Bibliography

S. Amiriparian. *Deep representation learning techniques for audio signal processing.* PhD thesis, Technische Universität München, 2019.

S. Arora. A survey on graph neural networks for knowledge graph completion. *arXiv preprint arXiv:2007.12374*, 2020.

J. Bekker and J. Davis. Estimating the class prior in positive and unlabeled data through decision tree induction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

J. Bekker and J. Davis. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4):719–760, 2020.

A. Bellet, A. Habrard, and M. Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

G. Blanchard, G. Lee, and C. Scott. Semi-supervised novelty detection. *The Journal of Machine Learning Research*, 11:2973–3009, 2010.

J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using aßiamesetime delay neural network. *Advances in neural information processing systems*, 6, 1993.

M. Carranza-García, P. Lara-Benítez, J. García-Gutiérrez, and J. C. Riquelme. Enhancing object detection for autonomous driving by optimizing anchor generation and addressing class imbalance. *Neurocomputing*, 449:229–244, 2021.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

H. Chen, F. Liu, Y. Wang, L. Zhao, and H. Wu. A variational approach for learning from positive and unlabeled data. *Advances in Neural Information Processing Systems*, 33: 14844–14854, 2020a.

K. Chen, D. Zhuang, and J. M. Chang. Supercon: Supervised contrastive learning for imbalanced skin lesion classification. *arXiv preprint arXiv:2202.05685*, 2022.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020b.

T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020c.

X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

X. Chen, W. Chen, T. Chen, Y. Yuan, C. Gong, K. Chen, and Z. Wang. Self-pu: Self boosted and calibrated positive-unlabeled training. In *International Conference on Machine Learning*, pages 1510–1519. PMLR, 2020d.

X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020e.

F. Chiaroni, M.-C. Rahal, N. Hueber, and F. Dufaux. Learning with a generative adversarial network from a positive unlabeled dataset for image classification. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1368–1372. IEEE, 2018.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

C.-Y. Chuang, J. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

M. Claesen, F. De Smet, P. Gillard, C. Mathieu, and B. De Moor. Building classifiers to predict the start of glucose-lowering pharmacotherapy using belgian health expenditure data. *arXiv preprint arXiv:1504.07389*, 2015.

E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

J. Cui, Z. Zhong, S. Liu, B. Yu, and J. Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 715–724, 2021.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

A. Diaz-Pinto, A. Colomer, V. Naranjo, S. Morales, Y. Xu, and A. F. Frangi. Retinal image synthesis and semi-supervised learning for glaucoma assessment. *IEEE transactions on medical imaging*, 38(9):2211–2218, 2019.

C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

E. Dorigatti, J. Goschenhofer, B. Schubert, M. Rezaei, and B. Bischl. Positive-unlabeled learning with uncertainty-aware pseudo-label selection. *arXiv preprint arXiv:2201.13192*, 2022.

M. C. Du Plessis and M. Sugiyama. Class prior estimation from positive and unlabeled data. *IEICE TRANSACTIONS on Information and Systems*, 97(5):1358–1362, 2014.

M. C. Du Plessis, G. Niu, and M. Sugiyama. Analysis of learning from positive and unlabeled data. *Advances in neural information processing systems*, 27, 2014.

C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220, 2008.

R. Fujimaki, T. Yairi, and K. Machida. An approach to spacecraft anomaly detection problem using kernel feature space. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 401–410, 2005.

G. P. C. Fung, J. X. Yu, H. Lu, and P. S. Yu. Text classification without negative examples revisit. *IEEE transactions on Knowledge and Data Engineering*, 18(1):6–20, 2005.

C. Gelada, S. Kumar, J. Buckman, O. Nachum, and M. G. Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, pages 2170–2179. PMLR, 2019.

J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

T. Guo, C. Xu, J. Huang, Y. Wang, B. Shi, C. Xu, and D. Tao. On positive-unlabeled classification in gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8385–8393, 2020.

F. He, T. Liu, G. I. Webb, and D. Tao. Instance-dependent pu learning by bayesian optimal relabeling. *arXiv preprint arXiv:1808.02180*, 2018.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori. Inlier-based outlier detection via direct density ratio estimation. In *2008 Eighth IEEE international conference on data mining*, pages 223–232. IEEE, 2008.

M. Hou, B. Chaib-Draa, C. Li, and Q. Zhao. Generative adversarial positive-unlabelled learning. *arXiv preprint arXiv:1711.08054*, 2017.

A. G. Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013.

W. Hu, R. Le, B. Liu, F. Ji, J. Ma, D. Zhao, and R. Yan. Predictive adversarial learning from positive and unlabeled data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7806–7814, 2021.

A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

K. Jaskie and A. Spanias. Positive and unlabeled learning algorithms and applications: A survey. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–8. IEEE, 2019.

L. Jiang, D. Li, Q. Wang, S. Wang, and S. Wang. Improving positive unlabeled learning: Practical aul estimation and new training method for extremely imbalanced data sets. *arXiv preprint arXiv:2004.09820*, 2020.

Z. Jiang, T. Chen, T. Chen, and Z. Wang. Improving contrastive learning on imbalanced seed data via open-world sampling. *arXiv preprint arXiv:2111.01004*, 2021.

L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020.

J. M. Johnson and T. M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

B. Kang, Y. Li, S. Xie, Z. Yuan, and J. Feng. Exploring balanced feature spaces for representation learning. In *International Conference on Learning Representations*, 2020.

S. Khan, M. Hayat, S. W. Zamir, J. Shen, and L. Shao. Striking the right balance with uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 103–112, 2019.

S. S. Khan and M. G. Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374, 2014.

P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

R. Kiryo, G. Niu, M. C. Du Plessis, and M. Sugiyama. Positive-unlabeled learning with non-negative risk estimator. *Advances in neural information processing systems*, 30, 2017.

T. Köhler, A. Budai, M. F. Kraus, J. Odstrčilik, G. Michelson, and J. Hornegger. Automatic no-reference quality assessment for retinal fundus images using vessel segmentation. In *Proceedings of the 26th IEEE international symposium on computer-based medical systems*, pages 95–100. IEEE, 2013.

T. Li, P. Cao, Y. Yuan, L. Fan, Y. Yang, R. Feris, P. Indyk, and D. Katabi. Targeted supervised contrastive learning for long-tailed recognition. *arXiv preprint arXiv:2111.13998*, 2021.

B. Liu, W. S. Lee, P. S. Yu, and X. Li. Partially supervised classification of text documents. In *ICML*, volume 2, pages 387–394. Sydney, NSW, 2002.

B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu. Building text classifiers using positive and unlabeled examples. In *Third IEEE international conference on data mining*, pages 179–186. IEEE, 2003.

H. Liu, J. Z. HaoChen, A. Gaidon, and T. Ma. Self-supervised learning is more robust to dataset imbalance. *arXiv preprint arXiv:2110.05025*, 2021.

S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine. An experimental study with imbalanced classification approaches for credit card fraud detection. *IEEE Access*, 7:93010–93022, 2019.

Y. Marrakchi, O. Makansi, and T. Brox. Fighting class imbalance with contrastive learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 466–476. Springer, 2021.

E. Medina-Mesa, M. Gonzalez-Hernandez, J. Sigut, F. Fumero-Batista, C. Pena-Betancor, S. Alayon, and M. Gonzalez de la Rosa. Estimating the amount of hemoglobin in the neuroretinal rim using color images and oct. *Current Eye Research*, 41(6):798–805, 2016.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

F. Mordelet and J.-P. Vert. A bagging svm to learn from positive and unlabeled examples. *arXiv preprint arXiv:1010.0772*, 2010.

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

M. Ochal, M. Patacchiola, A. Storkey, J. Vazquez, and S. Wang. Few-shot learning with class imbalance. *arXiv preprint arXiv:2101.02523*, 2021.

A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Y. Ouali, C. Hudelot, and M. Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020.

D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

P. Perera, P. Oza, and V. M. Patel. One-class classification: A survey. *arXiv preprint arXiv:2101.03064*, 2021.

M. M. Rahman and D. N. Davis. Addressing the class imbalance problem in medical datasets. *International Journal of Machine Learning and Computing*, 3(2):224, 2013.

H. Ramaswamy, C. Scott, and A. Tewari. Mixture proportion estimation via kernel embeddings of distributions. In *International conference on machine learning*, pages 2052–2060. PMLR, 2016.

M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021.

T. Sakai, G. Niu, and M. Sugiyama. Semi-supervised auc optimization based on positive-unlabeled learning. *Machine Learning*, 107(4):767–794, 2018.

H. Scheiblauer, A. Filomena, A. Nitsche, A. Puyskens, V. M. Corman, C. Drosten, K. Zwirglmaier, C. Lange, P. Emmerich, M. Müller, et al. Comparative sensitivity evaluation for 122 ce-marked rapid diagnostic tests for sars-cov-2 antigen, germany, september 2020 to april 2021. *Eurosurveillance*, 26(44):2100441, 2021.

N. Seliya, A. Abdollah Zadeh, and T. M. Khoshgoftaar. A literature review on one-class classification and its potential applications in big data. *Journal of Big Data*, 8(1):1–31, 2021.

S. Shu, Z. Lin, Y. Yan, and L. Li. Learning from multi-class positive and unlabeled data. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1256–1261. IEEE, 2020.

J. Sivaswamy, S. Krishnadas, G. D. Joshi, M. Jain, and A. U. S. Tabish. Drishti-gs: Retinal image dataset for optic nerve head (onh) segmentation. In *2014 IEEE 11th international symposium on biomedical imaging (ISBI)*, pages 53–56. IEEE, 2014.

G. Su, W. Chen, and M. Xu. Positive-unlabeled learning from imbalanced data. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence, Virtual Event*, 2021.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, pages 935–942, 2007.

P. Wang, K. Han, X.-S. Wei, L. Zhang, and L. Wang. Contrastive learning based hybrid networks for long-tailed image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 943–952, 2021.

X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2015.

Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

L. Weng. Self-supervised representation learning. *lilianweng.github.io*, 2019. URL `https://lilianweng.github.io/posts/2019-11-10-self-supervised/`.

L. Weng. Contrastive representation learning. *lilianweng.github.io*, 2021. URL `https://lilianweng.github.io/posts/2021-05-31-contrastive/`.

J. Xie, X. Zhan, Z. Liu, Y. S. Ong, and C. C. Loy. Unsupervised object-level representation learning from scene images. *Advances in Neural Information Processing Systems*, 34: 28864–28876, 2021.

Y. Xu, C. Xu, C. Xu, and D. Tao. Multi-positive and unlabeled learning. In *IJCAI*, pages 3182–3188, 2017.

S. Yamaguchi, S. Kanai, T. Shioda, and S. Takeda. Image enhanced rotation prediction for self-supervised learning. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 489–493. IEEE, 2021.

P. Yang, X.-L. Li, J.-P. Mei, C.-K. Kwoh, and S.-K. Ng. Positive-unlabeled learning for disease gene identification. *Bioinformatics*, 28(20):2640–2647, 2012.

X. Yang, Z. Song, I. King, and Z. Xu. A survey on deep semi-supervised learning. *arXiv preprint arXiv:2103.00550*, 2021.

Y. Yang and Z. Xu. Rethinking the value of labels for improving class-imbalanced learning. *Advances in Neural Information Processing Systems*, 33:19290–19301, 2020.

J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1476–1485, 2019.

R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

Z. Zhang, F. S. Yin, J. Liu, W. K. Wong, N. M. Tan, B. H. Lee, J. Cheng, and T. Y. Wong. Origa-light: An online retinal fundus image database for glaucoma analysis and research. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 3065–3068. IEEE, 2010.

P. Zheng, S. Yuan, X. Wu, J. Li, and A. Lu. One-class adversarial nets for fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1286–1293, 2019.

Y. Zhou, J. Xu, J. Wu, Z. Taghavi, E. Korpeoglu, K. Achan, and J. He. Pure: Positive-unlabeled recommendation with generative adversarial network. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2409–2419, 2021.

F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.