Jann Philipp Goschenhofer

Reducing the effort for data annotation - contributions to weakly supervised deep learning

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Eingereicht am 10.03.2023

Jann Philipp Goschenhofer

Reducing the effort for data annotation - contributions to weakly supervised deep learning

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Eingereicht am 10.03.2023

Erster Berichterstatter: Prof. Dr. Bernd Bischl
Zweiter Berichterstatter: Prof. Dr. Thomas Seidl
Dritter Berichterstatter: Prof. Dr. Emmanuel Müller

Tag der Disputation: 20.12.2023

Acknowledgments

Research is a team sport, and this thesis would not have been possible without the support, guidance, and help of many individuals. I would like to express my sincere gratitude to ...

... *my supervisor Prof. Dr. Bernd Bischl for the great collaboration and his trust, support, encouragement, empathy and advice throughout these past years.*

... *Prof. Zsolt Kira who kindly invited me to his lab as a visiting researcher, which was one of the highlights of my Ph.D. experience.*

... *Prof. Dr. Thomas Seidl and Prof. Dr. Emmanuel Müller for their willingness to invest their valuable time as the second and third reviewers for my Ph.D. thesis.*

... *PD Dr. Fabian Scheipl and Prof. Dr. Christian Heumann for their willingness to be part of the examination committee for my doctoral defense.*

... *several companions who made this journey a truly exciting one and supported me a lot in many different ways (in random order with multiple occupations): Matthias, Daniel, Flo, Janek, David, Chris, Kathi, Emilio, Rasmus, Moritz, Paulina, Freddie, Gunnar, Philipp.*

... *all current and former colleagues at the Department of Statistics and the Fraunhofer Institute for Integrated Circuits for the exceptionally friendly, collaborative, and supportive atmosphere.*

... *my loving family Mathis, Gabi, and Wolfgang for their endless support in all endeavors.*

... *Lari who knows the ups and downs the best and was a motivation powerhouse and an indispensable ally not only during this adventure.*

Summary

The rise and success of modern supervised machine and deep learning models, which have become parts of our everyday lives, are partially fueled by the increasing availability of large datasets with high-quality annotations. However, the availability of such annotations, also referred to as labels, remains a critical bottleneck for many machine learning applications as they are a prerequisite for supervised model training. This is particularly evident in domains where the data annotation process is ambiguous and cumbersome or where it requires the knowledge of scarce domain experts. In application domains such as medical imaging or industrial manufacturing, this often leads to the situation where, despite the availability of a large amount of non-annotated data, only a fraction of this data is annotated with appropriate labels. This problem of model training with limited labeled data is the focus of this thesis which covers methods to leverage unlabeled data, samples without annotation, and weakly labeled data, samples with a low-information annotation, for model training. Concretely, it includes contributions to the areas of semi-supervised learning, positive unlabeled learning, constrained clustering, and transfer learning.

First, this thesis introduces the concept of deep semi-supervised learning and provides an overview of recent research on self-training, entropy regularization, consistency regularization, and hybrid approaches. The goal of semi-supervised learning is to train machine learning models on a small dataset of annotated training data while simultaneously using a larger dataset of completely unlabeled data. Since the main developments in this area are driven by the computer vision community, many of these methods have been developed mainly for image data. This motivated one contribution to investigate their application in a time series classification scenario. Another contribution investigates the applicability of semi-supervised learning in a medical imaging context to reduce the data annotation effort in this domain.

Positive unlabeled learning is another exciting sub-field of low-supervised learning. Here, the training data contains only positive or unlabeled samples, while the goal is to learn a binary classifier that can distinguish unseen positive and negative samples. Despite the absence of negative samples during model training, recent positive unlabeled learning methods that use weighted loss functions enable successful model training in this challenging data regime. One contribution to this topic presents a framework that uses explicit estimates of predictive uncertainty to enable self-training in such positive unlabeled settings.

The next section introduces the concept of weakly supervised learning with pairwise binary constraint annotations for constrained clustering. One contribution in this area proposes a method that combines it with concepts from semi-supervised learning to train these models in a semi-constrained manner. This allows the use of large amounts of completely unlabeled data to guide model training on a smaller dataset with pairwise binary constraint annotations. Another contribution in this area leverages the cluster detection capabilities of these models to recognize dynamically changing categories.

The final section includes a description of transfer learning approaches as well as an application of transfer learning with learning tasks of varying granularity in a medical context.

Zusammenfassung

Der Erfolg moderner überwachter Machine- und Deep-Learning-Modelle, die mittlerweile Teil unseres Alltags geworden sind, fußt teilweise auf der zunehmenden Verfgbarkeit großer Datensätze mit hochwertigen Annotationen. Die Verfgbarkeit solcher Annotationen, auch als Labels bezeichnet, bleibt jedoch ein kritisches Bottleneck für viele Anwendungen, da sie eine Voraussetzung für das Training von überwachten Modellen darstellen. Dies ist insbesondere in Domänen ein Problem, in welchen der Prozess der Datenannotation unklar und aufwändig ist oder das Wissen von Fachexperten erfordert. In Anwendungsbereichen wie der medizinischen Bildgebung oder der industriellen Fertigung führt dies oft dazu, dass trotz der Verfügbarkeit einer großen Menge nicht annotierter Daten nur ein Bruchteil dieser Daten mit geeigneten Labels versehen werden kann. Das Modelltraining mit begrenzt annotierten Daten ist der Fokus dieser Arbeit, welche Methoden behandelt, um auch nicht annotierte Daten und schwach annotierte Daten, deren Annotationen geringen Informationsgehalt besitzen, für das Modelltraining zu nutzen. Die vorliegende Arbeit enthält Beiträge zu Semi-supervised Learning, Positive-unlabeled Learning, Constrained Clustering und Transfer Learning.

Zu Beginn wird das Konzept des Semi-supervised Learning vorgestellt und es wird ein Überblick über die aktuelle Forschung zu Self-Training, Entropy-Regularisation, Consistency-Regularisation und hybriden Ansätzen gegeben. Das Ziel von Semi-supervised Learning besteht darin, Modelle auf einem kleinen Datensatz mit annotierten Trainingsdaten zu trainieren, wobei zusätzlich ein größerer Datensatz von nicht annotierten Daten in das Modelltraining mit einbezogen wird. Da die wichtigsten Entwicklungen in diesem Bereich aus dem Bereich Computer Vision getrieben werden, wurden viele dieser Methoden hauptsächlich für Bilddaten entwickelt. Dies motivierte einen Beitrag zur Untersuchung ihrer Anwendung in einem Szenario zur Klassifizierung von Zeitreihen. Ein weiterer Beitrag untersucht die Anwendbarkeit von Semi-supervised Learning in einem medizinischen Bildgebungskontext, um den Aufwand für die Datenannotation zu reduzieren.

Positive Unlabeled Learning ist ein weiteres Teilgebiet von Semi-supervised Learning. Dabei enthält der Trainingsdatensatz nur positive oder nicht annotierte Datenpunkte, während das Ziel darin besteht, einen binären Klassifikator zu lernen, der zwischen positiven und negativen Datenpunkten unterscheiden kann. Trotz des Fehlens annotierter negativer Datenpunkte während des Modelltrainings ermöglichen Positive Unlabeled Learning Methoden ein erfolgreiches Modelltraining in dieser schwierigen Datensituation. Ein Beitrag zu diesem Thema stellt ein Framework vor, welches explizite Schätzungen der Vorhersageunsicherheit verwendet, um Self-Training in solch einem Kontext zu ermöglichen.

Ein weiterer Abschnitt stellt das Konzept des Weakly-supervised Learning mit paarweisen binären Constraint Annotationen für Constrained Clustering vor. Ein Beitrag in diesem Bereich schlägt eine Methode vor, die Constrained Clustering mit Konzepten aus dem Semi-supervised Learning kombiniert, um auch nicht annotierte Daten für das Training dieser Modelle zu verwenden. Dies ermöglicht die Verwendung großer Mengen nicht-annotierter Daten, um das Modelltraining auf einem kleineren Datensatz mit paarweisen binären Constraint Annotationen zu verbessern. Ein weiterer Beitrag in diesem Bereich nutzt die Fähigkeiten dieser Modelle zur Clustererkennung, um dynamisch wechselnde Kategorien in den Daten zu erkennen.

Schließlich enthält ein weiterer Abschnitt eine Beschreibung von Transfer Learning-Ansätzen sowie eine Anwendung von Transfer Learning mit Machine Learning-Problemen unterschiedlicher Granularität in einem medizinischen Kontext.

Contents

1 Introduction

> Data is the new oil. Like oil, data is valuable, but if unrefined [and non-annotated] it cannot really be used [for supervised machine learning].
>
> Extension of the original quote by Clive Humby & Michael Palmer.

Machine learning (ML) and especially deep learning (DL) have seen a strong rise in the past decade. This development has recently culminated in the advent of impressive foundation models such as Chat-GPT (OpenAI, 2023), DALL-E (Ramesh et al., 2022) and Stable Diffusion (Rombach et al., 2022) that have brought DL broad publicity outside academia and earned coverage in leading news outlets. What these models have in common, among other technological developments, is that they have been trained on huge amounts of annotated training data. For instance, Stable Diffusion in its original version has been trained on the LAION-400M dataset (Schuhmann et al., 2021) which consists of 400 million images that are annotated with text descriptions. LAION-400M in turn is a filtered subset of the LAION-5B dataset (Schuhmann et al., 2022) which contains 2.32 billion English image-text pairs. Next to their architectural and technical advancements, this sheer magnitude of annotated training data is one of the key drivers of the impressive performance of these models. A dedicated branch of research is directed at the creation of large high-quality annotated training datasets, partly by the use of a large force of crowd workers for manual annotation (Deng et al., 2014; Antol et al., 2015; Shao et al., 2019) or by automatically crawling annotations from the internet and programmatically cleaning those annotations (Gao et al., 2020a; Srinivasan et al., 2021; Schuhmann et al., 2022; Kreutzer et al., 2022). Collecting data with high-quality annotations is often a hurdle due to limiting factors such as task complexity and time and money constraints, yet it is a core requirement for the training of supervised ML models. As such, it remains a bottleneck for the development of ML applications. In the context of this thesis, labeled, or annotated data refers to data that has been annotated or marked with the correct ground truth output or target, while unlabeled or non-annotated data refers to data that lacks such annotations.

The MS-COCO image benchmark dataset (Lin et al., 2014) is one great example to demonstrate the sheer effort that is required to create large datasets with high-quality annotations. The dataset contains samples from a total of 91 different classes of common everyday objects such as animals, vehicles, or furniture and the authors' goal was to annotate 2.5 million instances distributed over a database of 328k images. It contains data annotations with different degrees of informativeness including instance spotting (where is an instance?), instance-specific class labeling (which classes are depicted in the image?), and instance segmentation (pixel-wise segmentation of instances in the image). The authors designed an elaborate hierarchical annotation pipeline (Deng et al., 2014) to enable efficient annotation by click workers. The annotation of the different informativeness

levels took the following toll: 10k worker hours with an average of 14 seconds per instance and 110 seconds per image for instance spotting, 20k worker hours for instance-specific class labeling with an average of 29 seconds per instance and 220 seconds per image, and 55k worker hours for instance segmentation with an average of 80 seconds per instance and 604 seconds per image. Assuming an hourly wage of 15 USD and leaving aside the effort involved in gathering the raw images, designing, and implementing the annotation pipeline, this results in expenses of roughly 1.28 million USD for the annotation of the MS-COCO dataset. Another example with more fine-grained annotations is the CityScapes dataset (Cordts et al., 2016) which contains fine-grained segmentation masks for 5k images of street scenes and coarsely annotated segmentation masks for 20k street scene images. Fine-grained segmentation annotation took 5400 seconds per image, i.e. 1.5 hours per image, resulting in a total of 7.5k worker hours. The coarse annotation took 420 seconds per image, i.e. 7 minutes per image, resulting in a total of 2.3k worker hours. Following these numbers and the above assumptions, this relatively small segmentation-level dataset incurred costs of roughly 150k USD for the data annotation only. This issue is even more pressing in situations where scarce and hence expensive domain experts are required and the annotation can not be done by trained click workers as in the previous examples. Medical histopathology, the diagnosis and study of diseases of human tissue, is one prominent example where highly specialized experts are required for data annotation. For instance, it takes a histopathologist up to 30 minutes, i.e. 900 seconds, to annotate one malignant lung cancer area in a whole-slide image gathered with high-quality microscopes (Wang et al., 2019a). These examples illustrate that the effort for the creation of high-quality annotated training data can be prohibitively high and it hence remains one major obstacle to developing tailored ML solutions.

In this context, it is interesting to review the infamous quote *"Data is the new oil"* by Clive Humby with the extension of Michael Palmer claiming that *"Data is the new oil. Like oil, data is valuable, but if unrefined it cannot really be used. It has to be changed into gas, plastic, chemicals, etc. to create a valuable entity that drives profitable activity. So, must data be broken down, analysed for it to have value"* (Arthur, 2013). With his concretization of the original quote, Palmer pays tribute to the fact that data is not inherently useful, it has to be refined to deliver real value. In this thesis' context of reducing the need for data annotation for ML model training, this quote could be further extended to *"Data is the new oil. Like oil, data is valuable, but if unrefined [and non-annotated] it cannot really be used [for supervised machine learning]"*. This extension stresses the need for not only preprocessed and cleaned training data but also for training data that has been annotated with high-quality labels.

An alternative approach next to careful and scalable data annotation to overcome this hurdle lies in the development of modeling techniques and training strategies that allow the use of non-annotated or weakly annotated data in the training process. Such non-annotated or weakly annotated data comes at no or low annotation cost and can hence often be gathered with substantially less effort compared to labeled data. These methods lie at the intersection of unsupervised and supervised learning, as they require some degree of label information as opposed to unsupervised learning but not to the extent of supervised learning (Chapelle et al., 2009). This is the core topic of this thesis, which describes different modeling approaches to reduce the need for annotated training data and highlights the scientific contributions to these areas. Specifically, it covers the areas of transfer learning, semi-supervised learning, positive unlabeled learning and constrained clustering as illustrated in Figure 1.1.

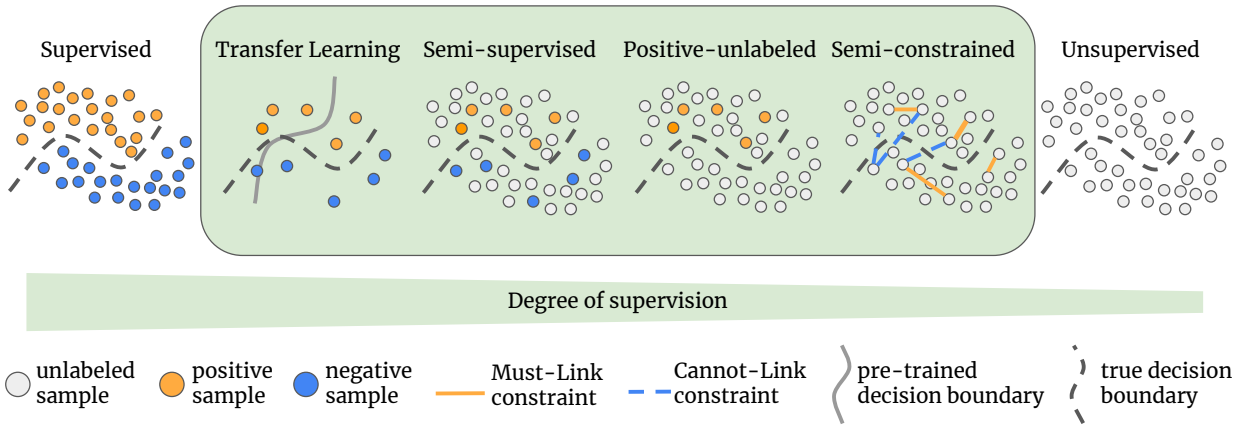


Figure 1.1: Illustration of the different data scenarios and corresponding learning strategies covered in this thesis (illustrated in the green area) at the intersection of supervised and unsupervised learning on a toy dataset. This includes transfer learning, semi-supervised learning, positive unlabeled learning and (semi-)constrained clustering. These methods are ordered from left to right with a decreasing degree of supervision.

- **Transfer Learning** (TL) is a model training strategy that involves using a model that was pre-trained on a source task as a starting point for another similar target task. This warm-starting can substantially reduce the amount of required annotated training data and the training time needed to achieve good performance on the target task.
- **Semi-supervised Learning** (SSL) aims to improve the model performance by leveraging a combination of labeled and unlabeled data. The goal of SSL is to make use of the vast amounts of unlabeled data available in many real-world applications next to a smaller labeled dataset to improve the performance of ML models, hence reducing the amount of required annotated training data.
- **Positive Unlabeled Learning** (PUL) is a subset of SSL that focuses on training binary classification models when only positive labeled samples and unlabeled samples are available. The challenge is to model the underlying class distribution from only the labeled positive and the unlabeled samples, despite the absence of labeled negatives.
- **Constrained Clustering** (CC) is a subset of weakly supervised learning strategies that involves training models on data that is weakly labeled, meaning that it does not have the precise target or output annotations, but rather some form of auxiliary information that can still provide some degree of supervision. Concretely, annotations are used that contain information about the relationship of data samples to one another.

These areas, the scope of this thesis, and the contributions therein are introduced in more detail in the next section followed by a description of the outline of this thesis.

1.1 Scope

The term semi-supervised learning was coined by Merz et al. (1992) according to Chapelle et al. (2009) and has been an active research area for decades of ML research. It is targeted at a

data scenario where a smaller annotated and a larger non-annotated dataset, also referred to as labeled and unlabeled throughout this thesis, are present. The main motivation behind the development of those methods is the observation that often access to labeled data is a bottleneck for ML applications while unlabeled data can be gathered with comparably low effort (Chapelle et al., 2009). The algorithms developed for this data scenario thus try to use the information in the unlabeled dataset to guide the training of an ML model along the labeled data, hence bridging unsupervised and supervised learning which lends the research branch its name. Semi-supervised approaches include the use of neighborhood information between labeled and unlabeled samples, unsupervised regularization strategies, self-training, consistency regularization strategies, and combinations thereof, so-called hybrid methods (Van Engelen and Hoos, 2020). While earlier methods were developed for structured, tabular data, the advent of DL led to the development of potent SSL methods for unstructured data such as texts, images, or time series. This thesis describes two contributions to the field of deep SSL. In Section 4.1 we describe the transfer of recent SSL methods that were mainly developed for image data towards the use with time series data (Goschenhofer et al., 2021). In Section 4.6 we investigate the application of deep SSL methods in the context of histopathology (Dexl et al., 2022).

PUL is directed at similar data scenarios such as SSL with the difference that the labeled dataset only contains positive labeled samples while the unlabeled data consists of both positive and negative samples. Despite this lack of labeled negative training data, PUL aims at learning a binary classifier that can distinguish unseen data into positives and negatives (Bekker and Davis, 2020). Prominent approaches in this area include the use of tailored, weighted loss functions for model training, two-step strategies to identify negatives in the unlabeled data for subsequent model training, and generative approaches. PUL has also been combined with the semi-supervised strategy of self-training (Chen et al., 2020b). In Section 4.3 we propose an approach to improve and simplify this self-training strategy via the explicit inclusion of prediction uncertainty in the self-training process (Dorigatti et al., 2022).

CC subsumes a set of methods that integrate the use of constraint annotations in the clustering process. Contrary to instance-specific class labels, these annotations contain information about the relationship of the training samples to each other. While there exists a broad variety of different constraint types, this thesis is mainly focused on binary pairwise constraints that convey the information on whether two samples are in the same or a different cluster (Zhang et al., 2021b). Originally designed for the integration of constraint information in unsupervised methods such as k-means (Wagstaff et al., 2001), recent methods are tailored towards the use of deep neural networks for CC using specific loss functions and pairwise training strategies (Hsu et al., 2019). Next to their ability to train potent clustering models using weak data annotations only, CC models also retain their ability to detect the number of underlying clusters in the data, the so-called overclustering scenario (Hsu and Kira, 2016). In Section 4.4, we apply this capability for work with short text samples and describe how CC could be used in a scenario with dynamically changing topics (Goschenhofer et al., 2022). Further, we propose a strategy that extends deep CC towards the use of unlabeled data in model training, i.e. a semi-constrained data scenario, in Section 4.2 (Goschenhofer et al., 2023).

TL, also referred to as model pre-training with subsequent fine-tuning, has become a de facto standard procedure for DL in different domains including computer vision (CV) (Mahajan et al., 2018; Dai et al., 2021) or natural language processing (NLP) (Howard and Ruder, 2018; Yamaguchi et al., 2021). It follows the concept to pre-train a DL model on a source task and then fine-tune it

on a target task of interest. Initially mostly used with supervised source tasks, TL is being more and more used with alternative, auxiliary source tasks with the advent of self-supervised learning in recent years. For supervised source tasks, the model pre-training is conducted on a source task for which a large annotated training dataset exists that stems from a similar domain as the target task. Concretely, this could mean that a model is pre-trained on a source task containing a large training dataset with coarse annotations and then fine-tuned on a target task for which a smaller training dataset with more granular, fine-grained annotations exists. In Section 4.5 we describe the use of TL in the context of motor state detection of patients with Parkinson's Disease using sensor movement data (Goschenhofer et al., 2019). Therein, we show that a source task with a large coarsely annotated training dataset can be used to warm-start and improve the final performance of deep time series classification models for the more fine-grained disease status prediction target task.

1.2 Outline

The remainder of this thesis is structured as follows: Section 2.1 introduces the notation used throughout this thesis and Section 2.2 provides a short introduction to deep learning. The core concepts of SSL including its underlying assumptions and an overview of ML developments in this area are described in Section 2.3. This section further includes an overview of the recent developments in deep SSL to which the contributions of this thesis relate such as self-training and consistency regularization. In Section 2.4, PUL as an edge case of semi-supervised binary classification and its core extensions towards DL are introduced followed by an introduction of CC as an approach for weakly supervised learning in Section 2.5. Section 2.6 provides an introduction to TL and describes an application in a medical context. The closing Section 3 concludes this thesis with a reflection on the contributions made throughout this Ph.D. thesis and an outlook on further interesting research questions building atop this work. The concrete scientific contributions of this thesis are listed in Section 4.

2 Methodological Background

2.1 Notation

We define a sample $\boldsymbol{x} = (x_1, ..., x_p)$ as a $p-$dimensional vector of covariates $x_l \in \mathcal{X}_l, l = 1, ..., p$, as realization from the joint input space $\boldsymbol{x} \in \mathcal{X} = (\mathcal{X}_1 \times \cdots \times \mathcal{X}_p) \subseteq \mathbb{R}^p$. Further, $y \in \mathcal{Y}$ denotes the corresponding target value in the target space $\mathcal{Y}$. For classification, the target space results in $\mathcal{Y} = \{1, ..., K\}$ with $K = |\mathcal{Y}|$ different classes and for regression in $\mathcal{Y} = \mathbb{R}$ with $K = 1$. We also define a prediction model $f : \mathcal{X} \mapsto \mathbb{R}^K$ which maps an input sample $\boldsymbol{x} \in \mathcal{X}$ to the respective K prediction scores. The model f is usually parametrized with a parameter vector $\boldsymbol{\theta}$ which is estimated during model training and the estimated parameter vector is denoted as $\hat{\boldsymbol{\theta}}$. Formally, we define the model as a scoring function $f : \mathcal{X} \mapsto \mathbb{R}^K$ that maps an input sample to K scores for classification and to $K = 1$ score for regression tasks and the predicted score vector is denoted as $\hat{\boldsymbol{z}} = f(\boldsymbol{x}|\hat{\boldsymbol{\theta}})$. In the classification case, the class-specific scores $\hat{z}_k$ that correspond to the elements of this score vector $\hat{\boldsymbol{z}}$ are then mapped to predicted class probabilities $\hat{y}_k$ via e.g. the softmax function σ such that $\hat{y}_k = \sigma(\hat{z}_k) = \exp(\hat{z}_k)/\sum_{k=1}^{K} \exp(\hat{z}_k)$ where $\hat{\boldsymbol{y}} = (\hat{y}_1, ..., \hat{y}_K) \in [0, 1]^K$ and $\sum_{k=1}^{K} \hat{y}_k = 1$. The final class prediction for class k then results as $\arg\max_k \hat{y}_k$. In the regression case, $\hat{y} = \hat{z} = f(\boldsymbol{x}|\hat{\boldsymbol{\theta}})$ as no further mapping to probability scores is required. For the sake of readability, we use a simplified notation throughout this thesis and, unless noted otherwise, refer to the model prediction $\hat{\boldsymbol{y}} = f(\boldsymbol{x}|\hat{\boldsymbol{\theta}})$ as a vector of predicted class probabilities with $\hat{y}_k$ being the predicted probability for class k, hence implying the softmax transformation. Situations, where the model f is used as a mere scoring rule are marked and described accordingly throughout the thesis. Further, we define a labeled dataset $D_l = \{(\boldsymbol{x}^{(i)}, y^{(i)}), ..., (\boldsymbol{x}^{(n_l)}, y^{(n_l)})\}$ consisting of n_l tuples of samples and their respective labels as well as an unlabeled dataset D_u which consists of n_u samples $D_u = \{\boldsymbol{x}^{(n_{l+1})}, ..., \boldsymbol{x}^{(n)}\}$ where $n = n_l + n_u$. The goal of semi-supervised learning is to train a prediction model f on a joint dataset $D = D_l \cup D_u$ which consists of a labeled dataset D_l and an unlabeled dataset D_u. Supervised learning instead is directed at training a prediction model f assuming a fully labeled training dataset $D = D_l$. A random subsample of the data is denoted as batch $\mathcal{B} \subset D$.

2.2 A Primer on Deep Learning

This section serves as a primer to neural networks, also referred to as deep neural networks (DNNs), as the basis for the different algorithms explained in the following sections. Specifically, it introduces the concepts of feed-forward neural networks, convolutional neural networks (CNNs) with applications to image and time series data and transformer architectures for NLP applications. This section closely follows the formulations and explanations in the introduction to deep learning book by Goodfellow et al. (2016).

In the contributions of this thesis, we have used various CNN and transformer-based model architectures, and this section will serve as a basis to make them more approachable. While this section explains the basic building blocks of DL architectures, it is by no means a comprehensible introduction to deep learning which would be out of the scope of this thesis. For an in-depth introduction to DL refer to the comprehensive book by (Goodfellow et al., 2016) or recent surveys on transformer architectures (Lin et al., 2022; Tay et al., 2022).

2.2.1 Feed-forward Neural Networks

Feed-forward neural networks, also referred to as multilayer perceptrons (MLPs), are the most basic DL models and aim at learning a parametrized function f with parameters $\boldsymbol{\theta}$ that maps an input data column vector $\boldsymbol{x} = (x_1, x_2, ..., x_p)^\mathsf{T}$ of dimensionality p onto a target $\boldsymbol{y}$. One motivation for MLPs is that through the subsequent coupling of different mathematical transformations of the input data, also termed features, the model can learn different representations of the data which are especially well suited to solve the learning task at hand. An MLP consists of multiple single neurons stacked atop each other, as such they are the most basic unit of a neural network. One neuron accepts an input column vector $\boldsymbol{x}$. The single elements of this input vector are multiplied with a column vector of trainable weight parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_p)^\mathsf{T}$ with $\boldsymbol{\theta} \in \mathbb{R}^p$ and a bias term b is added to the scalar product $\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x}$. The model weights $\boldsymbol{\theta}$ and the bias term b are updated and optimized during the model training. The resulting scalar $r' = \boldsymbol{\theta}^\mathsf{T}\boldsymbol{x} + b$ is then fed into a non-linear and non-parametric activation function σ which provides the output of the neuron such that:

$$r = \sigma(r') = \sigma(\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x} + b) \tag{2.1}$$

There are different activation functions with the ReLU $\sigma(r') = \max\{0, r'\}$ and the sigmoid function $\sigma(r') = 1/(1 + \exp(r'))$ being the most prominent ones for use in simple feed-forward neural networks. This formulation of a single neuron can be easily extended towards the use of multiple neurons, also referred to as a layer. Therefore, we extend the weight vector $\boldsymbol{\theta}$ to a weight matrix $\boldsymbol{\Theta} \in \mathbb{R}^{p \times m}$ and the bias term scalar $b \in \mathbb{R}$ to a vector $\boldsymbol{b} \in \mathbb{R}^m$ to describe the parameters of a layer with m neurons such that $\boldsymbol{r'} = \boldsymbol{\Theta}^\mathsf{T}\boldsymbol{x} + \boldsymbol{b}$. The $i-$th row $\boldsymbol{\Theta}_i \in \mathbb{R}^p$ of the weight matrix $\boldsymbol{\Theta}$ refers to the weight vector of the $i - th$ neuron and the $i-$th scalar element b_i of the bias vector $\boldsymbol{b}$ to the respective bias term. The final output of this layer is then defined as

$$\boldsymbol{r} = \sigma(\boldsymbol{r'}) = \sigma(\boldsymbol{\Theta}^\mathsf{T}\boldsymbol{x} + \boldsymbol{b}) \tag{2.2}$$

where the activation function σ is now applied element-wise. This definition of one layer of different neurons allows us to formulate a one-layer feed-forward neural network for a $K-$class classification problem. We again assume an input vector $\boldsymbol{x} \in \mathbb{R}^p$ which is processed by one layer of m hidden neurons $\boldsymbol{r'} = (r'_1, ..., r'_m) \in \mathbb{R}^m$ with weight matrix $\boldsymbol{\Theta} \in \mathbb{R}^{p \times m}$ that are transformed by the activation function σ_1 such that

$$\boldsymbol{r'} = \sigma_1(\boldsymbol{\Theta}^\mathsf{T} \boldsymbol{x} + \boldsymbol{b}) \tag{2.3}$$

The resulting vector $\boldsymbol{r'} \in \mathbb{R}^m$ is also termed a feature vector as it contains m differently transformed versions of the input $\boldsymbol{x}$. In the final step, we use another weight matrix $\boldsymbol{U} \in \mathbb{R}^{m \times K}$ to transform this feature vector $\boldsymbol{r'}$ to an output vector $\hat{\boldsymbol{y}} \in [0, 1]^K$ such that

$$\hat{\boldsymbol{y}} = \sigma_2(\boldsymbol{U}^\mathsf{T} \boldsymbol{r}) \tag{2.4}$$

using a second activation function σ_2. We use a softmax activation function for σ_2 to yield probabilistic prediction scores $\hat{\boldsymbol{y}}$ as described in the notation Section 2.1.

Consequently, the entire network results as $\hat{\boldsymbol{y}} = \sigma_2(\boldsymbol{U}^\mathsf{T}(\sigma_1(\boldsymbol{\Theta}^\mathsf{T} \boldsymbol{x} + \boldsymbol{b})))$. While different activation functions can be used within the neural network, e.g. σ_1, the final activation function is determined by the modeling task. The K-class classification in this case requires the use of a softmax as the final activation function. This very simplistic one-layer neural network can easily be extended towards the subsequent use of multiple layers with matching weight matrices. For instance, a two-layer neural network could follow the structure $\hat{\boldsymbol{y}} = \sigma_3(\boldsymbol{U}^T(\sigma_2(\boldsymbol{\Theta}_2^\mathsf{T}(\sigma_1(\boldsymbol{\Theta}_1^\mathsf{T} + \boldsymbol{b}_1)) + \boldsymbol{b}_2)))$ with two hidden weight matrices $\boldsymbol{\Theta}_1 \in \mathbb{R}^{p \times m}, \boldsymbol{\Theta}_2 \in \mathbb{R}^{m \times m}$ and bias vectors $\boldsymbol{b}_1 \in \mathbb{R}^m, \boldsymbol{b}_2 \in \mathbb{R}^m$. Such stacking of multiple layers increases the depth of the neural network and hence its learning capacity which is why they are also referred to as DNNs.

The training of neural networks is a process that involves two fundamental steps: the forward pass and the backward pass (Goodfellow et al., 2016). Before those steps, the model weights are initialized randomly or using more elaborate procedures such as the Glorot (Glorot and Bengio, 2010) or the He (He et al., 2015) initializations. During the forward pass, the input data is passed through the neural network layer by layer, with each layer performing a set of above-described calculations on the data. These calculations involve weighted sums, added biases and activation functions that transform the input data into a representation that is progressively more suitable for the task at hand. The output of the final layer represents the trained network's prediction or output for the input $\hat{\boldsymbol{y}} = f(\boldsymbol{x}|\hat{\boldsymbol{\theta}})$. The full network is defined over all parameters in $\boldsymbol{\Theta}, \boldsymbol{U}, \boldsymbol{b}$ and with $\boldsymbol{\theta}$ we denote the union of those. The next step is the backward pass, also known as backpropagation, where the network learns from its prediction errors and adjusts its weights to improve its performance (Rumelhart et al., 1986). Therefore, a suitable, differentiable loss function $L(\hat{\boldsymbol{y}}, y)$ is used that measures the distance between the predicted value $\hat{\boldsymbol{y}}$ and the true target value y such that $L : \mathbb{R}^K \times \mathcal{Y} \mapsto \mathbb{R}$. Examples for this loss function are the squared loss $L_2(\hat{y}, y) = (\hat{y} - y)^2$ for regression and the cross entropy loss $L_{CE}(\hat{\boldsymbol{y}}, y) = -\sum_{k=1}^K \mathbb{1}_{[y=k]} \log(\hat{y}_k)$ for classification. Then the partial derivatives, i.e. the gradients, of the loss function with respect to these parameters are calculated using the chain rule. These derivatives are then used to update the parameters in the direction of the negative gradient, to minimize the loss function over the training data (Paszke et al., 2017). The weight updates are performed using iterative optimization algorithms such as stochastic gradient descent or Adam (Kingma and Ba, 2014). These descending

methods adjust the weights in the direction of the negative gradient over multiple iterations, or epochs until the network's performance reaches a satisfactory level. Overall, the training can be seen as an optimization problem, in which the network is attempting to find a set of parameters that minimize the distance between its predictions and the true targets.

2.2.2 Convolutional Neural Networks

CNNs are a specialized type of neural network architecture that are particularly effective at handling data that follows a grid-like topology such as time series or images (Goodfellow et al., 2016). CNNs use the convolution operator in place of general matrix multiplication in at least one of their layers. The convolution allows them to learn to detect features or patterns in the input data that are relevant to the learning task at hand. The convolution operator is a mathematical operation that takes two functions, say x and k, as input and produces a third function, denoted $(x * k)$, as output. In the context of CNNs, the input function is typically the raw input data, while the kernel function is a set of learnable weights that are applied to the input data.

For example, consider a one-dimensional time series $\boldsymbol{x} = (x_1, x_2, ..., x_T)$ with measurements at T time points. The kernel function k of length m can be defined as a vector of learnable weights $\boldsymbol{\theta} = (\theta_1, ..., \theta_m)$, that is applied to subsets of the input time series via element-wise multiplication and summed. For instance, imagine a kernel function $k(t)$ that assigns weight factors $\theta_t = 0.7$ to the current and $\theta_{t-1} = 0.3$ to the previous measurement and a factor of 0 to all other previous measurements $x_1, ..., x_{t-2}$. Formally, the output of the convolution operation between x and k at position t is given by:

$$x'_t = (x * k)(t) = \sum_{i=1}^{t} x_i k_{t-i} \tag{2.5}$$

Intuitively, this operation slides the kernel function over the input time series, computing a weighted sum of the input values at each position. The resulting output $\boldsymbol{x'}$ is a filtered or convolved version of the input time series $\boldsymbol{x}$. In the case of image data, the convolution operator can be extended to two dimensions. Here, the kernel function is a matrix of learnable weights that is applied to subsets of the input image, again via element-wise multiplication and summation. Assume we have an input image $\boldsymbol{X} \in \mathbb{R}^{H \times W}$ and we apply a convolutional filter kernel matrix $\boldsymbol{K} \in \mathbb{R}^{k \times k}$. The output of the convolutional layer is the feature map $\boldsymbol{X'} \in \mathbb{R}^{H_{out} \times W_{out} \times C_{out}}$ where H_{out} and W_{out} are the height and width of the output feature map and determined by the input size, the kernel size, padding, and the stride s, and C_{out} is the number of filters in the layer. The output feature map $\boldsymbol{X'}$ is computed as follows:

$$x'_{i,j,c} = \sigma\left(\sum_{p=1}^{k}\sum_{q=1}^{k} \theta_{p,q,c} x_{i+p\times s, j+q\times s} + b_c\right), \tag{2.6}$$

where $\theta_{p,q,c}$ is the weight of the c-th filter at position (p, q), $x_{i,j}$ is the input pixel at position (i, j), b_c is the bias of the c-th filter, and s is the stride which determines the number of pixels by which the kernel is shifted at each step while scanning the input. The activation function σ is applied element-wise to the output of the convolution operation to finally yield the convolved

version $\boldsymbol{X}'$ of the image $\boldsymbol{X}$. CNNs use the convolution operator in at least one of their layers, typically followed by a non-linear activation function. Similar to MLPs, a CNN architecture usually consists of subsequent blocks of convolutional filters and they include further advances such as skip-connections (He et al., 2016; Tan and Le, 2019) whose descriptions are beyond the scope of this thesis. This allows the network to learn to detect patterns or features in the input data that are relevant to the learning task at hand, with increasing complexity throughout the network architecture. For example, in an image classification task, the network might learn to detect edges, corners, or other visual features in the early stages of the architecture. These might then be combined into more complex features at later stages such as eyes, wheels or noses that are useful for distinguishing between different image classes.

CNNs are trained using backpropagation and gradient descent optimizers, similar to MLP architectures. One challenge in training CNNs and neural networks in general is the issue of overfitting, where the network becomes too specialized to the training data and performs poorly on new, unseen data. To address this, several regularization techniques have been developed, such as dropout (Srivastava et al., 2014), which randomly drops out some of the units in the network during training, and weight decay, which penalizes large weights by adding a regularization term to the loss function.

2.2.3 Transformers

The transformer architecture was introduced in the seminal paper by Vaswani et al. (2017). It has revolutionized the field of (NLP) by achieving state-of-the-art performance on a variety of tasks such as machine translation, text summarization, and sentiment analysis and is the foundation of modern NLP architectures such as GPT (Radford et al., 2018), BERT (Devlin et al., 2019) or T5 (Raffel et al., 2020). The transformer architecture is based on the concept of self-attention, which allows the model to focus on different parts of the input sequence when computing the output. This is in contrast to previous NLP models that used recurrent neural networks (RNNs) such as Long-short-term-memory Networks (LSTMs) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Networks (GRUs) (Cho et al., 2014) to process the input sequence one token at a time. Self-attention in turn extends the concept of attention as introduced by Bahdanau et al. (2015) and Luong et al. (2015).

The core building block of the transformer architecture is the self-attention mechanism. Given an sequence of T input tokens $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T)$, the self-attention mechanism computes a set of output vectors $\boldsymbol{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_T)$, where each output vector $\boldsymbol{z}_i$ is a weighted sum of all input vectors:

$$\boldsymbol{z}_i = \sum_{j=1}^{T} \alpha_{ij} \boldsymbol{x}_j, \tag{2.7}$$

where α_{ij} is the attention weight between input vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The attention weight is computed using a softmax function $\alpha_{ij} = \exp(e_{ij}) / \sum_{k=1}^{T} \exp(e_{ik})$ where e_{ij} is the attention score between input vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The attention score is computed as a dot product between a query vector $\boldsymbol{q}_i$ and a key vector $\boldsymbol{k}_j$ such that $e_{ij} = \boldsymbol{q}_i^\mathsf{T} \boldsymbol{k}_j$, where $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ are learned query and key vectors that are computed from the input vectors using a linear projection. The self-attention mechanism allows the model to attend to different parts of the input token sequence depending

on the task at hand. For example, in a machine translation task, the model can attend to the source language words that are most relevant to the translation of the current target language word. The Transformer architecture typically also includes an MLP and residual connections (He et al., 2016) within each layer where the MLP consists of two linear transformations with a ReLU activation function in between. The residual connections allow the model to learn identity mappings, which helps with training such deep networks, similar to their functionality in CNNs. Further, it typically includes layer normalization which normalizes the activations of each layer to have zero mean and unit variance, which helps with training (Ba et al., 2016) as well as Dropout used to prevent overfitting (Srivastava et al., 2014).

The transformer architecture has become a standard building block for many state-of-the-art NLP models and was also shown to be applicable in CV scenarios with the introduction of the vision transformer (Dosovitskiy et al., 2021). Its success can be attributed to its ability to model long-range dependencies and its parallelizability, which allows for faster training on modern hardware.

2.3 Semi-supervised Learning

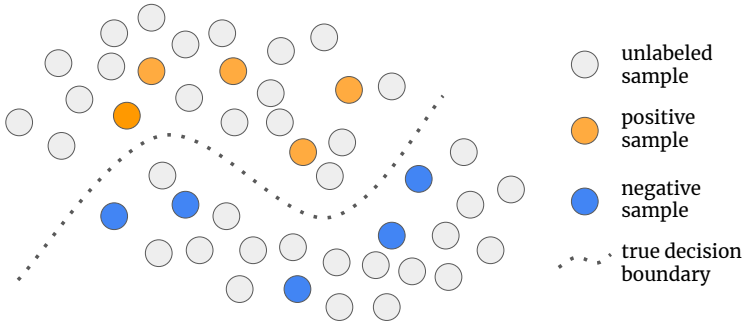


Figure 2.1: Illustration of the SSL setting for a binary classification scenario on a toy dataset. The goal is to train a binary classifier that can learn the true decision boundary using both a labeled dataset D_l, illustrated as colored dots, and an unlabeled dataset D_u, illustrated as gray dots.

The goal and promise of SSL, at the intersection of unsupervised and supervised learning, is to leverage both labeled and unlabeled data for ML tasks as illustrated in Figure 2.1. The expanding research in this field is mainly driven by the sometimes prohibitively high effort involved in annotating large labeled datasets on the one side and the abundance of unlabeled data on the other (Chapelle et al., 2009). Hence, semi-supervised methods mainly focus on settings with few labeled and many unlabeled training data such that $n_l \ll n_u$. While there exists research on SSL for a broad variety of learning tasks, this thesis is focused on semi-supervised classification, the most active area of research. This section introduces the core underlying assumptions of SSL, then provides a short overview of classical approaches, i.e. non-neural network models, which is followed by a comprehensive review of the recent advances in deep SSL and the different concepts applied therein.

SSL relies on three interconnected assumptions (Chapelle et al., 2009; Van Engelen and Hoos, 2020) and many developed approaches rely either on one or a combination of them. While similar in concept, the definitions of Chapelle et al. (2009) and Van Engelen and Hoos (2020) differ slightly. The following structure follows that of the more recent work by Van Engelen and Hoos (2020).

1. **Smoothness assumption**: Two samples $\boldsymbol{x}, \boldsymbol{x}'$ that are close to each other in a high-density region of the input space should have similar labels y, y'. While this assumption is also heavily used in supervised learning, it has even broader implications in SSL as it implies transitivity: if $\boldsymbol{x}$ is close to $\boldsymbol{x}'$ and $\boldsymbol{x}'$ is close to $\boldsymbol{x}^*$, then we can assume the labels y, y^* to be similar as well.

2. **Low-density assumption**: The decision boundary of model f should pass through low-density areas, so-called low-density regions. This adds another perspective to the smoothness assumption as placing the decision boundary in a high-density region would violate this smoothness assumption.

3. **Manifold assumption**: The input space is high-dimensional and consists of multiple lower-dimensional manifolds. All samples lie on those manifolds and samples $\boldsymbol{x}, \boldsymbol{x}'$ that lie on the same manifold have the same labels y, y'.

Furthermore, semi-supervised algorithms can be distinguished between inductive and transductive methods. Inductive learning algorithms aim at learning a general mapping from the data to the target space using the dataset $D = D_l \cup D_u$. After the learning phase at inference time, inductive models along with their estimated model parameters can be used to assign predicted labels to new, unseen data. In that sense, the inductive learning procedure optimizes the model parameters to yield the best possible predictions for unseen data. Contrary to this, transductive methods merely aim at predicting labels for the unlabeled dataset D_u using the labeled dataset D_l without the learning of a general decision rule. Transductive methods optimize directly over the model predictions on the unlabeled data D_u only (Van Engelen and Hoos, 2020). In that sense, induction is more general as it aims at learning general decision rules while transduction tries to reason from the labeled to the specific unlabeled samples.

SSL research traces back to the beginnings of ML research (Dempster et al., 1977) and hence many classical approaches, i.e. non-neural network models, have been developed. Following the taxonomy developed in the standard textbook by Chapelle et al. (2009), these approaches can be distinguished into four model classes: 1) Generative models such as the EM-algorithm for incomplete data (Dempster et al., 1977) that aim at learning the class-conditional density and use the unlabeled data D_u to improve its estimation via better estimation of the marginal. 2) Approaches that follow the low-density separation rationale try to direct the decision boundary through low-density areas following the low-density assumption using the latent information in D_u to identify these areas. This mainly involves max-margin estimators such as the transductive Support Vector Machine (SVM) (Collobert et al., 2006). 3) Graph-based methods that exploit the neighborhood of labeled and unlabeled samples defined via a metric, e.g. defined via a kernel function following the manifold assumption. These neighborhood relationships are then used to propagate class labels from the labeled to their neighboring unlabeled samples. Most of these methods are transductive and Label Propagation (Zhu and Ghahramanih, 2002) is one prominent method of this model class. 4) Change of Representation: two-step approaches that e.g. use D_u in the first step to learn a meaningful data representation which is then tailored towards the learning task using D_l in the second step.

Following the advent of DL in the past decade, a large body of research evolved that combines the SSL paradigm with DL models. In a recent overview, Van Engelen and Hoos (2020) extend the above-mentioned taxonomy of Chapelle et al. (2009) towards the use of neural networks along the dimensions of transduction and induction. Under transduction, they collect mainly graph-based models that leverage joint neighborhood structures in $D = D_l \cup D_u$. With that, they follow the structure of Chapelle et al. (2009) but extend it towards deep graph-based methods such as Deep Label Propagation (Iscen et al., 2019). They further differentiate different learning paradigms that mainly aim at extending existing supervised inductive methods toward using additional unlabeled data D_u next to the labeled data D_l. 1) Self-training methods, also referred to as wrapper methods or pseudo-labeling, use a supervised model f trained on D_l to iterative pseudo-label additional unlabeled samples from D_u to augment the training dataset and then re-train on this expanded labeled dataset. 2) Unsupervised preprocessing methods that use D_u to aid the generation of a meaningful representation of the data in an unsupervised manner. This includes the extraction of meaningful features from the raw data to find an embedding that is favorable for the initial learning task. Such approaches contain but are not limited to dimension reduction techniques such as Principal Component Analysis or autoencoders, again related to the manifold assumption. Further, cluster-then-label approaches use clustering techniques over D or D_u only to facilitate the initial supervised learning task. The final sub-branch of unsupervised preprocessing

methods mainly targets neural network-based methods and summarizes pre-training algorithms that use D_u to initialize the model architecture which is then fine-tuned on D_l. 3) Intrinsically semi-supervised approaches that extend supervised loss functions defined over D_l with tailored loss functions that allow the inclusion of D_u in the training process to enable a semi-supervised model training.

Recent strong-performing SSL methods follow at least one of these paradigms or are combinations of them. The research contributions covered in this thesis are targeted at 1) self-training and 3) intrinsically semi-supervised approaches, that is entropy and consistency regularization and hybrid approaches. Therefore, the remainder of this chapter is focused on providing the reader with an overview of those two areas.

2.3.1 Self-training

Self-training, also referred to as pseudo-labeling or self-learning, is one of the oldest approaches to SSL (Scudder, 1965; Fralick, 1967; Agrawala, 1970) being used for ML modeling (Yarowsky, 1995; Rosenberg et al., 2005). It follows the idea that the model trains itself by iteratively annotating parts of the unlabeled data, see Figure 2.2 for an illustration. The procedure usually alternates between a training and a pseudo-labeling step. After the training step, the model selects unlabeled samples via a selection criterion such as model confidence. These selected samples are then assigned the predicted label and added from D_u to the now updated labeled dataset. The model is then trained on this (pseudo-) labeled dataset and this self-training cycle continues until a stopping criterion, such as the fact that no unlabeled data is left, is reached. Self-training was translated to DL in the pioneering work of Lee (2013) and since then has sparked the development of numerous variants. The pseudo-labeling procedure in these approaches can be broken down into two steps that rely on the predictions from model f to get an overview of recent developments in that area.

The first step is the selection of some unlabeled sample(s) using a selection criterion. There exist different criteria for this selection process. The maximum predicted softmax class probability $\max(\hat{\boldsymbol{y}})$ for an unlabeled sample $\boldsymbol{x} \in D_u$ as a measure of model confidence is the de facto standard selection criterion. This criterion is then often used either with a threshold hyperparameter τ to decide which of the unlabeled samples are selected or by selecting the top k samples, ranked by the selection criterion. The naive version of Lee (2013) and succeeding approaches (Arazo et al., 2020; Laine and Aila, 2017; Tarvainen and Valpola, 2017) select all unlabeled samples for pseudo-labeling. They use a ramp-up function that assigns a gradually increasing weight to the unsupervised loss function and hence balances the impact of pseudo-labeling throughout the process of model training. In contrast, Rizve et al. (2021) select unlabeled samples for pseudo-labeling using a confidence and a prediction uncertainty criterion with two threshold hyperparameters τ, k. More holistic approaches also rely on pseudo-labeling (Sohn et al., 2020; Berthelot et al., 2019b; Xie et al., 2020b,a) using a confidence metric with thresholding for pseudo-label selection. Zhang et al. (2021a) explicitly tackle this thresholding issue and propose flexible, class-specific thresholds that adapt throughout training within their curriculum pseudo-labeling approach. In a similar vein, Cascante-Bonilla et al. (2021) learn a flexible threshold based on extreme value theory.

The second step is the pseudo-label assignment based on the model predictions $\hat{\boldsymbol{y}} = f(\boldsymbol{x}|\hat{\boldsymbol{\theta}})$ for selected unlabeled samples $\boldsymbol{x} \in D_u$, i.e. the mapping of the model prediction $\hat{\boldsymbol{y}}$ to the pseudo-label $\tilde{\boldsymbol{y}}$ which is ultimately used in the unsupervised loss part. The early approach by Lee (2013) uses

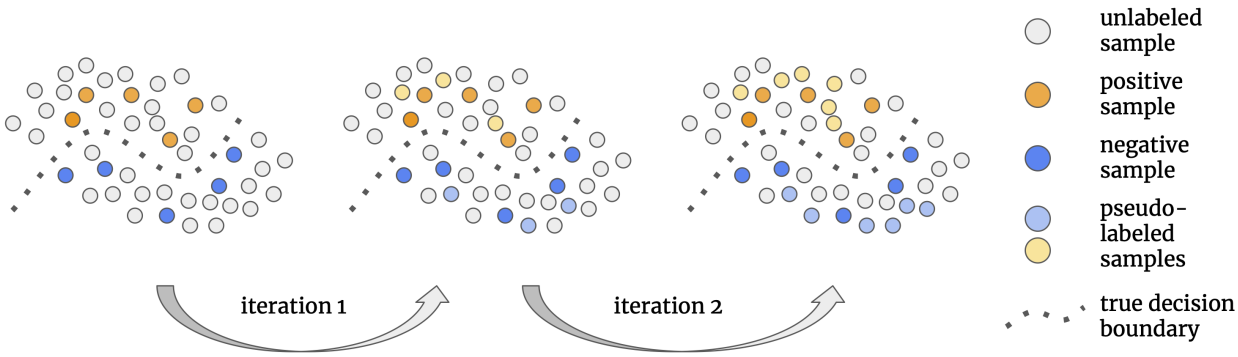


Figure 2.2: Illustration of the iterative self-training method. After an initial model fit on the labeled data, marked as blue and orange dots, parts of the unlabeled data, marked as gray dots, are pseudo-labeled, marked as light blue and orange dots, and additionally used for a model re-fit on this extended (pseudo-) labeled dataset. This iterative procedure continues until a stopping criterion is met.

hard pseudo-labels such that $\tilde{y} = \arg\max_k \hat{y}_k$. Following this, Rizve et al. (2021) also rely on hard pseudo-labels in combination with confidence- and uncertainty-based pseudo-label selection. In contrast, Arazo et al. (2020) use soft pseudo-labels such that $\tilde{\boldsymbol{y}} = \hat{\boldsymbol{y}}$ following observations by Tanaka et al. (2018) on training with noisy labels. They further argue that soft pseudo-labels help integrate the model's confidence into training which ultimately benefits performance. With the noisy-student-framework, Xie et al. (2020b) propose a self-training method that allows using both hard and soft pseudo-labels. Throughout their experiments, they empirically find that soft pseudo-labels lead to slightly better generalization performance, especially for out-of-domain unlabeled data. Parts of the large body of work that combines Self-training with consistency regularization (namely MixMatch (Berthelot et al., 2019b), ReMixMatch (Berthelot et al., 2019a), UDA (Xie et al., 2020a)) also use sharpened soft pseudo-labels. Sharpening is also known as temperature scaling (Guo et al., 2017). It uses a hyperparameter $T \in [0, 1]$ to artificially decrease the entropy of the predicted softmax probability vector $\hat{\boldsymbol{y}}$ such that $\tilde{y}_k = \hat{y}_k^{1/T} / \sum_{k=1}^{K} \hat{y}_k^{1/T}$ where $\tilde{\boldsymbol{y}} = \hat{\boldsymbol{y}}$ for $T = 1$. It is noteworthy that hard pseudo-labels $\tilde{y} = \arg\max_k \hat{\boldsymbol{y}}_k$ are used as targets within FixMatch (Sohn et al., 2020). Within an ablation study, Sohn et al. (2020) investigate the relationship between sharpening via temperature T and confidence-thresholding in combination with hard pseudo-labeling. They find that the sharpening procedure for soft pseudo-labels has no advantage over hard thresholded pseudo-labels, while it comes at the cost of another hyperparameter T. Hence, they use hard pseudo-labels without sharpening and FlexMatch, developed by Zhang et al. (2021a) on top of FixMatch (Sohn et al., 2020), also dispenses with pseudo-label sharpening.

As described above, successful self-training hinges on the interplay of selecting reasonable unlabeled samples via the selection criterion and assigning the correct pseudo-label to them. One prevalent issue that can occur in context is the confirmation bias (Arazo et al., 2020; Li et al., 2019), also referred to as the noise accumulation issue (Zhang et al., 2016). This issue occurs when the model f makes overconfident but wrong predictions on the unlabeled samples. Subsequently, this leads to the selection of such overconfident samples followed by the assignment of a semantically wrong pseudo-label. This in turn creates a wrong training signal and hence confuses semi-supervised model training ultimately leading to model degradation. To put it in another way, if the model f was guaranteed to make correct predictions on all unlabeled samples, self-training could transfer each semi-supervised task into a purely supervised task as all label guesses on D_u would resemble the true underlying but non-observable labels. This confirmation bias is illustrated

in Figure 2.3 where overconfident but semantically wrong predictions on unlabeled samples from the ImageNet-10 dataset (Deng et al., 2009) are depicted. A more detailed description of the confirmation bias along with further illustrations can be found in Section 4.2.

Figure 2.3: Illustration of the confirmation bias. Depicted are example images from the ImageNet-10 dataset (Deng et al., 2009) on which the model makes confident but semantically wrong model predictions. For instance, the model assigns the left image the wrong class label 'Airliner' with high confidence of $\hat{y}_{k=\text{Airliner}} = 0.975$ while the true label y is 'Sports Car'. If these unlabeled samples were selected and used as pseudo-labels in a self-training scenario, their inclusion would have a detrimental impact on model training.

There exist different mechanisms to cope with this confirmation bias, mostly targeted at creating better-calibrated model predictions on the unlabeled samples to mitigate the selection of faulty pseudo-labeled samples. Li et al. (2019) use MC-Dropout (Gal and Ghahramani, 2016) and random data augmentations to yield better-calibrated predictions for use in their student-teacher approach for semi-supervised training. Arazo et al. (2020) propose the use of the mixup data augmentation (Zhang et al., 2018) and the injection of label noise as regularization methods to overcome the confirmation bias. In a similar realm, Rizve et al. (2021) successfully use a combination of prediction confidence and model uncertainty with two distinct thresholds as a pseudo-label selection criterion to overcome this issue. Cascante-Bonilla et al. (2021) take a different perspective and combine curriculum learning with pseudo-labeling. This enables the model to use adaptive thresholds in the selection criterion and leads to on-par performance with more advanced and more complex SSL techniques. Within FlexMatch, Zhang et al. (2021a) also make use of flexible, class-specific thresholds in their extension of FixMatch (Sohn et al., 2020). Next to these extensions, pseudo-labeling remains a crucial component in recent semi-supervised models.

2.3.2 Entropy Regularization

Alternative inherently semi-supervised methods use additional unsupervised loss functions L_u defined over D_u or $D = D_l \cup D_u$ which are combined with the initial, supervised loss function L_l to allow joint model training over both datasets via the combined loss $L = L_l + \lambda L_u$, where hyperparameter λ controls the impact of the unsupervised loss term L_u. This has a regularizing effect and has the benefit that unlabeled samples from D_u can be inherently integrated into model training. For classification problems, typically the cross-entropy loss is used as a supervised loss function:

$$L_l(\hat{\boldsymbol{y}}, y) = -\sum_{k=1}^{K} \mathbb{1}_{[y=k]} \log \hat{y}_k \tag{2.8}$$

where $\hat{\boldsymbol{y}} = f(\boldsymbol{x}|\hat{\boldsymbol{\theta}})$ for $\boldsymbol{x} \in D_l$.

Several versions of the unsupervised loss term L_u have been developed in the literature. These are subsumed under the term unsupervised regularization as they serve as a model regularization without the need for label annotations. One early approach in this context is minimum entropy regularization (MER) introduced for use in SSL by Grandvalet et al. (2005). Thereby, the prediction entropy $H(\hat{\boldsymbol{y}})$ over the unlabeled samples $\boldsymbol{x} \in D_u$ serves as an unsupervised regularization term:

$$L_u(\hat{\boldsymbol{y}}) = -\sum_{k=1}^{K} \hat{y}_k \log \hat{y}_k \tag{2.9}$$

The combination of the supervised and the unsupervised loss terms leads to the formulation of the semi-supervised empirical risk:

$$\begin{aligned} R_{SSL}(f) &= \frac{1}{|B_l|} \sum_{(\boldsymbol{x},y)\in B_l} L_l(\hat{\boldsymbol{y}}, y) + \lambda \frac{1}{|B_u|} \sum_{\boldsymbol{x}\in B_u} L_u(\hat{\boldsymbol{y}}) \\ &= -\frac{1}{|B_l|} \sum_{(\boldsymbol{x},y)\in B_l} \sum_{k=1}^{K} \mathbb{1}_{[y=k]} \log \hat{y}_k - \lambda \frac{1}{|B_u|} \sum_{\boldsymbol{x}\in B_u} \sum_{k=1}^{K} \hat{y}_k \log \hat{y}_k \end{aligned} \tag{2.10}$$

where $B_l = \{(\boldsymbol{x}^{(1)}, y^{(1)}), ..., (\boldsymbol{x}^{(b_l)}, y^{(b_l)})\}$ is a batch of labeled data with batch size $b_l = |B_l|$ and $B_u = \{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(b_u)}\}$ is a batch of unlabeled data with batch size $b_u = |B_u|$. This loss combination forces the model to create sharp, low entropy predictions over the entire dataset and allows the integration of unlabeled samples $\boldsymbol{x} \in D_u$ into model training. MER was developed following the observation that unlabeled data do not contribute to the maximum-likelihood estimation of discriminative, supervised models. Thus, it introduces the regularization term as a prior adding an inductive bias to the model driven by the unlabeled data. The penalization of the model for high-entropy predictions over the unlabeled data potentially pushes the model's decision boundary towards low-density and away from high-density regions (Chapelle et al., 2009). This regularization strategy makes use of the low-density assumption as it "encourages the model to output confident predictions on unlabeled data" (Berthelot et al., 2019b).

Originally developed for logistic regression, MER is also used within neural network-based classifiers due to its generalizable formulation which allows a combination with other SSL approaches. For instance, Miyato et al. (2018) use MER in combination with their virtual adversarial training approach to yield stronger model performance. In MixMatch, Berthelot et al. (2019b) implicitly follow the rationale of MER via the usage of a sharpening function on the model predictions of the unlabeled samples. This can be seen as an indirect application of MER, as the entropy regularization does not come as a direct regularization as in MER but indirectly via the pseudo-labeling scheme. This sharpening functionality is similarly used in UDA (Xie et al., 2020a) and ReMixMatch (Berthelot et al., 2019a). MER is also used as a module in the pseudo-labeling approach by Arazo et al. (2020) to avoid local minima.

2.3.3 Consistency Regularization

The rationale of unsupervised regularization was further extended within models that aim at creating consistent model predictions over the unlabeled samples, also referred to as perturbation-based methods (Van Engelen and Hoos, 2020). These build up on the smoothness assumption which states that similar data points $\boldsymbol{x}, \boldsymbol{x}'$ that are close to one another in the input space should have similar labels y, y'. Following this assumption, a slightly perturbed version $\boldsymbol{x}' = g(\boldsymbol{x})$ of the input sample $\boldsymbol{x}$ is expected to correspond to the same class as the clean, non-perturbed version $\boldsymbol{x}$, assuming $\boldsymbol{x}$ lies in a high-density region. This expected consistency in model predictions lends this set of methods its name. The perturbation function g can resemble any perturbing process such as the addition of a gaussian noise vector $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_p)$ with $\epsilon_i \sim N(0, 1), i = 1, ..., p$ such that $g(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{\epsilon}$. Consistency regularization is also associated with the manifold assumption (Oliver et al., 2018; Ghosh and Thiery, 2021). This follows the argument that consistency regularization methods are designed in a way that the model learns a more robust mapping of the data to class-specific, low-dimensional manifolds as it has to cope with the induced perturbation. In its most naive form, this consistency requirement is then integrated into semi-supervised model training via a regularization term

$$L_u(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}') = \left\| \hat{\boldsymbol{y}} - \hat{\boldsymbol{y}}' \right\|_2 \tag{2.11}$$

where the model is penalized for quadratic differences in the model predictions over the clean input sample $\hat{\boldsymbol{y}} = f(\boldsymbol{x}|\hat{\boldsymbol{\theta}})$ and its perturbed version $\hat{\boldsymbol{y}}' = f(g(\boldsymbol{x})|\hat{\boldsymbol{\theta}})$ using the L_2-norm. From Equation (2.11) it is evident that no label information is required to calculate this loss and to propagate its gradient back through the model for model training via gradient descent optimization. This makes consistency regularization especially appealing for use in SSL as it does not need labels to regularize the model for inconsistency in model predictions for the unlabeled samples D_u.

In recent years, different perturbation functions g have been developed from the simple addition of gaussian noise to the inputs through to the use of more elaborate methods such as temporal consistency and data augmentation-based consistency (see Table 1 in Sohn et al. (2020) for an overview). An overview of these various approaches is provided in the following.

Noise Perturbation

With the Ladder-Net, Rasmus et al. (2015) introduced an autoencoder-based approach that injects additive gaussian noise at different intermediate representations of the input samples and calculates a regularization term over changes in these representations. This allows them to robustify the model representations and train the model on the joint dataset $D = D_l \cup D_u$ using both the reconstruction loss of the autoencoder as well as the noise-regularization term. The encoder part of the architecture is used at inference time. Instead of random noise, Miyato et al. (2018) propose to add directed adversarial noise to the unlabeled input samples as a regularization mechanism which they coined as virtual adversarial training (VAT). In contrast to the addition of noise to the input sample, the Π-Model adds noise in the form of dropout layers to the model architecture (Laine and Aila, 2017). The regularization term is then calculated over different model prediction samples via the MCDropout algorithm (Gal and Ghahramani, 2016) which simulates an ensemble of models and enforces consistent model predictions across the ensemble members. Park et al.

(2018) propose a combination of the Π-Model with VAT and show small performance gains via this combined method over the solitary approaches.

Temporal Consistency

Methods from another branch of research leverage predictions from different training stages of the model as a perturbation mechanism following the rationale that the model should produce temporally consistent model predictions during training. Temporal ensembling (Laine and Aila, 2017) maintains an exponential moving average of model predictions over stochastically augmented, unlabeled input samples from past training epochs as a consistency target. Thus, the auxiliary target $\tilde{\boldsymbol{y}}_t$ for input sample $\boldsymbol{x}$ is calculated for epoch t as:

$$\tilde{\boldsymbol{y}}_t = \frac{\alpha\hat{\boldsymbol{y}}_t + (1-\alpha)\hat{\boldsymbol{y}}_{t-1}}{(1-\alpha^t)} \tag{2.12}$$

where $\alpha \in [0, 1]$ is a hyperparameter that governs the effect of the past on the present predictions, and the denominator $(1 - \alpha^t)$ is a bias correction term. In the current training epoch t, $\tilde{\boldsymbol{y}}_t$ serves as an auxiliary target for an unlabeled sample $\boldsymbol{x} \in D_u$. The squared distance between the past and the current model predictions is used as an unsupervised loss function such that $L_u(\hat{\boldsymbol{y}}_t, \tilde{\boldsymbol{y}}_t) = \|\hat{\boldsymbol{y}}_t - \tilde{\boldsymbol{y}}_t\|_2$. Tarvainen and Valpola (2017) follow this rationale as well in their mean teacher architecture. Instead of storing past model predictions of D_u, they maintain a teacher version of the initial student model. The parameters of this teacher model are updated using an exponential moving average of the student model's current and the teacher model's previous model parameters. Concretely, the model parameters for the teacher at training step t are calculated as $\hat{\boldsymbol{\theta}}'_t = a\hat{\boldsymbol{\theta}}'_{t-1} + (1-a)\hat{\boldsymbol{\theta}}_t$ where $\hat{\boldsymbol{\theta}}_t$ are the student's parameters at step t. The weights of the student model are directly optimized using a combined loss function: the cross entropy is used as supervised loss L_l over the labeled samples D_l and a squared loss is used as consistency loss L_u over samples from the combined dataset D. Concretely, the consistency loss for a (un-)labeled sample $\boldsymbol{x} \in D$ at training step t is $L_u(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}') = \|\hat{\boldsymbol{y}} - \hat{\boldsymbol{y}}'\|_2$ where $\hat{\boldsymbol{y}} = f(\boldsymbol{x}|\hat{\boldsymbol{\theta}}_t)$ and $\hat{\boldsymbol{y}}' = f(\boldsymbol{x}|\hat{\boldsymbol{\theta}}'_t)$. The basic principle remains: the model is trained to make consistent predictions for the unlabeled samples throughout the training process, and this signal is used to incorporate unlabeled data into the model training.

This concept of teacher-student models is an important training paradigm for SSL and inspired a broader body of semi-supervised training approaches. For instance, the iterative noisy student (Xie et al., 2020b) uses a teacher model that is trained on clean, labeled samples only to pseudo-label unlabeled samples. These are then used as auxiliary training targets for the student model. After training the student on this combined (un-)labeled dataset, the current student takes over the role of the teacher to teach the next, novel student model. Based on this and additional tricks such as noise injection in both the model and the data, this approach can leverage large amounts of unlabeled data. Other examples are the unbiased teacher approaches that train a student and a slowly progressing teacher model for semi-supervised object detection (Liu et al., 2021, 2022). The teacher model is gradually trained via exponential moving averaging of the student's model weights. Along with other tricks including data augmentation and class balancing, the unbiased teacher yields strong performance with only a fraction of samples being annotated compared to the supervised baseline. Luo et al. (2018) couple the student-teacher paradigm with a graph-based approach where the teacher model is used to create clusters of the unlabeled data, leading

to more expressive features. Ke et al. (2019) identify the close coupling of the teacher and student models as a bottleneck in the mean teacher and propose the dual student architecture as an improvement.

Data Augmentation

Data augmentation strategies have become a core component of recent deep-learning approaches to increase model performance in settings with large and small annotated training datasets (Lemley et al., 2017; Cubuk et al., 2019, 2020) to overcome overfitting and enhance model generalization (Zhang et al., 2018). In supervised settings, data augmentation serves as a regularization mechanism that prevents the model from overfitting to the training examples and teaches it invariances in the training data (Cubuk et al., 2019; Ghosh and Thiery, 2021). Next to the application in CV, data augmentation also plays a crucial component in NLP (Yu et al., 2018), speech recognition (Park et al., 2019), and time series classification (Iwana and Uchida, 2021) settings. As mentioned earlier, we can define a data augmentation strategy g that creates an augmented version $\boldsymbol{x}'$ of the original sample $\boldsymbol{x}$ in a label-preserving manner. This means that the semantic meaning, as expressed in the corresponding label, of $\boldsymbol{x}$ and the augmented version $\boldsymbol{x}'$ is preserved and remains the same despite the augmentation. Simple augmentations for image data include e.g. cropping, translations, flipping, warping, and rotation of the original image.

Interpreted as a smart perturbation noise, data augmentation strategies easily fit the rationale of consistency regularization. Hence, they also play a crucial role in consistency regularization for SSL, next to supervised learning. For instance, one core component of MixMatch's pseudo-labeling strategy is the repeated random augmentation of the unlabeled samples before the label guessing step (Berthelot et al., 2019b). It furthermore makes use of the data augmentation strategy MixUp (Zhang et al., 2018) to mix and match the augmented pseudo-labeled unlabeled samples and the originally labeled samples. With the extension ReMixMatch, Berthelot et al. (2019a) replace the standard augmentations in MixMatch (Berthelot et al., 2019b) with augmentation anchoring. Therefore, they use the model prediction for a weakly augmented sample (e.g. simple cropping or flipping of the images) as targets for predictions over strongly augmented versions of the same sample. For the strong augmentations, they use a customized version of the AutoAugment strategy (Cubuk et al., 2019) which learns an augmentation policy throughout model training. Within their work on unsupervised data augmentation (UDA), Xie et al. (2020a) show the potential of specifically designed data augmentation strategies in both supervised learning and SSL settings for image and text-based learning tasks. Across a variety of experiments, they show that especially the RandAugment (Cubuk et al., 2020) strategy leads to model performance on par with fully supervised learning with a fraction of labeled data. RandAugment is a simplified version of AutoAugment which randomly chooses some from a larger set of augmentation functions and hence alleviates the overhead required to learn the augmentation policy in AutoAugment, facilitating its use in settings with few data annotations. FixMatch (Sohn et al., 2020), a successor of ReMixMatch (Berthelot et al., 2019a), further builds on top of the weak- and strong augmentations paradigm using sets of weak and strong data augmentations. The rationale is to use confident predictions on weakly augmented unlabeled samples as pseudo-labels which then serve as targets in an unsupervised cross-entropy loss function. Predictions over strongly augmented versions of these samples serve as input to this unsupervised loss function. Next to the AutoAugment strategy (Cubuk et al., 2019), Sohn et al. (2020) find that the simpler RandAugment (Cubuk et al., 2020) strategy also applies very well to this setting. This relatively straightforward use of the weak- and

strong augmentations training scheme lead to substantial improvement over its predecessors such as MixMatch (Sohn et al., 2020), ReMixMatch (Berthelot et al., 2019a), and UDA (Xie et al., 2020a) despite its simplification. These examples demonstrate the potential and the crucial role of data augmentation strategies in SSL.

2.3.4 Hybrid Approaches

The careful reader might have noticed that the same approaches have been cited as examples of the different SSL strategies self-training, entropy regularization, and consistency regularization described in this chapter. This is due to the recent trend that the best-performing SSL approaches often combine these different complementary strategies. A prominent example of this development is the family of "-Match" papers that have been introduced by various research groups in recent years. Starting with MixMatch, Berthelot et al. (2019b) combine elaborate data augmentation with pseudo-labeling, entropy regularization via a sharpening function, and Mixup (Zhang et al., 2018) as a holistic approach to SSL. Model prediction vectors over differently augmented versions of an unlabeled sample are averaged, sharpened via a temperature scaling mechanism, and then used as pseudo-labels. Subsequently, a batch of labeled and pseudo-labeled data are combined via Mixup to create synthetic training samples with synthetic labels which are then fed into an unsupervised loss function L_u. This combination of different SSL paradigms allows MixMatch to achieve impressive predictive performance with a low degree of supervision. In their follow-up work ReMixMatch, Berthelot et al. (2019a) further improve MixMatch by distribution alignment to align the distribution of pseudo-labels with the distribution of labeled data, and augmentation anchoring to stabilize the pseudo-labeling scheme by introducing the weak and strong data augmentation scheme. With FixMatch, Sohn et al. (2020) further improve upon these results using weak and strong data augmentations: pseudo-labels from weakly augmented unlabeled samples $\boldsymbol{x} \in D_u$ are selected based on a prediction confidence criterion and serve as training targets in the auxiliary classification loss L_u. Model predictions over exaggeratedly strong augmented versions of these samples are then used as input to this loss function, allowing model training on both D_u and D_l. This idea has sparked a lot of further research such as FeatMatch (Kuo et al., 2020) which uses data augmentation in the manifold space or FlexMatch (Zhang et al., 2021a) which combines this concept with curriculum learning.

2.3.5 Contributions

Despite its rise in CV and NLP, the application of deep SSL for time series classification remains a somewhat under-investigated area. This motivated our work in Section 4.1 which translates recent SSL approaches from image to time series classification (Goschenhofer et al., 2021). It, therefore, describes the necessary adaptations for this domain switch, that is the choice of a suitable backbone architecture and suitable data augmentation techniques. The efficacy of this translation is then evaluated across a broader set of time series classification benchmark datasets empirically showing that SSL does well apply in this domain but with smaller relative performance gains compared to the image classification tasks. A series of additional experiments also sheds light on the effect of different time series-specific backbone architectures and data augmentation strategies of varying complexity.

In Section 4.6, we investigated the suitability of deep SSL in the context of medical imaging on the task of tissue classification for colon cancer histology in a low-label regime (Dexl et al., 2022). Especially in medical scenarios the availability of trained professionals for data annotation tasks often constitutes a bottleneck for the development of DL solutions. To address this, we compared semi-supervised with supervised models with varying degrees of supervision and amounts of classes in this scenario. Furthermore, we investigated the robustness of these models towards domain shifts concerning different scanners used for data collection and demonstrated the effectiveness of customized data augmentation strategies in medical imaging.

2.4 Positive Unlabeled Learning

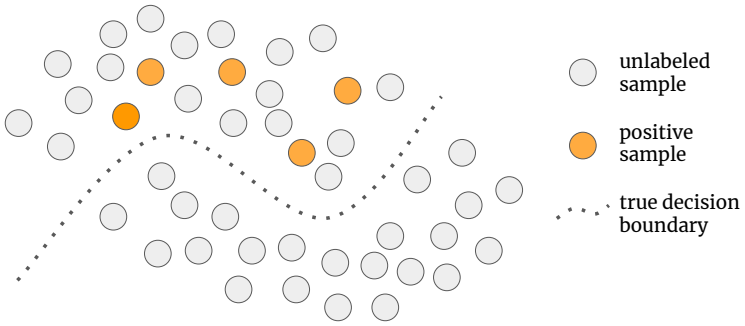


Figure 2.4: Illustration of the positive-unlabeled learning (PUL) setting on a toy dataset. The goal of PUL is to train a positive-negative classifier that can learn the true decision boundary, illustrated as a dashed gray line, from a dataset that contains only positive, marked as orange dots, or unlabeled data, marked as gray dots. This is a particularly challenging learning objective as the model does not have access to any labeled negative samples during training.

Positive-unlabeled learning (PUL) is a binary classification ML scenario where only positive and unlabeled samples are present for model training resulting in a positive dataset D_p and an unlabeled dataset D_u (Bekker and Davis, 2020) as illustrated in Figure 2.4. The true underlying labels of the unlabeled samples in D_u could be both positive or negative but there exist no labeled negative samples. This constraint that the only labeled samples are positive separates PUL from regular SSL where a small fraction of labeled positives and negatives is present as D_l next to the larger D_u. Next to its close relation to SSL, PUL differs from one-class classification (OCC) where binary training data is given as there is very little data support for the positive class in OCC which relates it with e.g. anomaly detection (Khan and Madden, 2014).

PUL was getting increasing attention from the research community in recent years, partially driven by the appearance of many real-world data problems that naturally are PUL problems. This includes applications in diverse areas such as bioinformatics (Lan et al., 2016), biomedical imaging (Zhao et al., 2023), fake review detection (Li et al., 2014) and audio processing (Ito and Sugiyama, 2022). One illustrative example would be medical records. Those records contain information if patients have been diagnosed with certain diseases (positives). Still, they usually do not provide a list of diseases that the patient has not been diagnosed with but has been tested for (negatives). Plainly speaking, the absence of a diagnosis does not imply that the patient does not suffer from the disease. Following this, a medical record naturally is either positive (disease diagnosed in the medical record) or unlabeled (the disease is not mentioned in the medical record). In a scenario where we would now like to train a model that predicts the disease diagnosis from other features in the medical records, we would be facing a natural PUL problem (Bekker and Davis, 2020; Chen et al., 2020a). Another example is that of personalized advertisements where clicks on certain ads are used as positives, signaling the interest of the user in the advertisement. Though, a non-click on an advertisement does not imply that it is not interesting and hence a negative. The user could have simply overlooked this respective advertisement and it should thus be treated as unlabeled (Bekker and Davis, 2020).

Formally, we observe samples from an input space $\boldsymbol{x} \in \mathcal{X}$ and a binary target space $\mathcal{Y} = \{+1, -1\}$. We aim at learning a binary classifier $f : \mathcal{X} \mapsto \mathbb{R}$ parametrized with $\boldsymbol{\theta}$ that predicts an input sample $\boldsymbol{x}$ to be either positive or negative where the final class prediction is obtained using a

monotonic transformation function such as the signum, tanh or sigmoid functions to map the predictions to the final target space (Chen et al., 2020b). We have access to one positive dataset $D_p = \{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(n_p)}\}$ of $n_p = |D_p|$ samples where we know that these samples correspond to a positive label. Further, a second unlabeled dataset $D_u = \{\boldsymbol{x}^{(n_l)}, ..., \boldsymbol{x}^{(n)}\}$ of $n_u = |D_u|$ unlabeled samples is given with unobserved targets from the binary target space $y \in \mathcal{Y}$ and $n = n_p + n_p$ describes the overall amount of samples in $D = D_p \cup D_u$. There exist two scenarios for the data generating process in PUL (Bekker and Davis, 2020). The single-training-set scenario assumes that both D_p and D_u are subsets of the same dataset D. The case-control scenario in turn assumes that D_l and D_u are from two different datasets. Refer to (Bekker and Davis, 2020) for a more thorough discussion of both scenarios including illustrative examples. Next to the single-training-set assumption, most PUL approaches make another assumption about the labeling mechanism (Bekker and Davis, 2020). The selected completely at random (SCAR) mechanism assumes that labeled positives are selected completely at random, irrespective of their features, from the underlying distribution. This means that the probability for a sample $\boldsymbol{x}$ to be labeled as positive is proportional to its probability of being positive. On the contrary, the selected at random (SAR) assumption states that the labeling of positive samples is dependent on their feature values. The majority of research in this area that is relevant to the contribution of this thesis assumes the single-training-set scenario and the SCAR labeling mechanism, which is where this section is also focused on. The class prior $\pi : P(y = +1)$ describes the probability that a random sample from D is positive such that $P(\boldsymbol{x}) = \pi P(\boldsymbol{x}|y = +1) + (1 - \pi)P(\boldsymbol{x}|y = -1)$. It is often assumed that the true positive class prior π is known a priori or that it can be estimated from the data (Christoffel et al., 2016; Bekker and Davis, 2020). From these definitions, it is evident that we do not have access to negatives during model training, making PUL an especially challenging task not only for model training but also for model evaluation. Most current research assumes positive-negative labeled validation and test sets to evaluate and tune the model (Kiryo et al., 2017; Chen et al., 2020b; Acharya et al., 2022). Despite this, Jain et al. (2017) introduced an approach to estimate the AUC performance of a model on positive unlabeled validation and test sets.

2.4.1 Methods

Importance reweighting methods that treat unlabeled samples as weighted negative samples have become the standard methods for modern PUL. The unbiased PU loss (uPU) (Du Plessis et al., 2014) was the first development in this direction and defines the empirical risk for a classifier f as follows (Kiryo et al., 2017; Chen et al., 2020b):

$$R_{uPU}(f) = \frac{\pi}{n_p} \sum_{\boldsymbol{x} \in D_p} L\left(\hat{y}, +1\right) + \left(\frac{1}{n_u} \sum_{\boldsymbol{x} \in D_u} L\left(\hat{y}, -1\right) - \frac{\pi}{n_p} \sum_{\boldsymbol{x} \in D_p} L\left(\hat{y}, -1\right) \right) \tag{2.13}$$

where $f : \mathcal{X} \mapsto \mathbb{R}$ is the classifier and $L : \mathbb{R} \times \{+1, -1\} \mapsto \mathbb{R}$ a differentiable loss function that measures the loss resulting from predicting an output $\hat{y} = f(\boldsymbol{x}|\hat{\boldsymbol{\theta}})$ while the true label is y. A sigmoid loss function, which is the horizontally mirrored version of the logistic loss, is often chosen for $L(\hat{y}, y)$ (Kiryo et al., 2017) such that

$$L_{sig}(\hat{y}, y) = \frac{1}{1 + \exp(\hat{y}y)} \tag{2.14}$$

where $y \in \{+1, -1\}$.

In a follow-up work on the uPU loss, Kiryo et al. (2017) show this risk formulation in Equation (2.13) could become negative via the second loss term and is prone to overfitting. This is especially becoming a problem when working with over-parametrized neural network architectures in the context of DL. They propose a simple yet effective alternative instead, the non-negative PU loss (nnPU). The resulting empirical risk follows the definition:

$$R_{nnPU}(f) = \frac{\pi}{n_p} \sum_{\boldsymbol{x} \in D_p} L\left(\hat{y}, +1\right) + \max\left(0, \frac{1}{n_u} \sum_{\boldsymbol{x} \in D_u} L\left(\hat{y}, -1\right) - \frac{\pi}{n_p} \sum_{\boldsymbol{x} \in D_p} L\left(\hat{y}, -1\right)\right) \tag{2.15}$$

The nnPU loss has become the standard loss formulation for PUL and is used as a module in various follow-up works on PUL using DNNs (Xu et al., 2019; Chen et al., 2020b; Luo et al., 2021)

It has also sparked further research on the development of reweighting-based loss functions for different PUL scenarios. Kato et al. (2019) develop a novel loss formulation on top of the nnPU loss which allows robust PU learning with a selection bias (PUSB) in the presence of a selection bias in the training data. PUSB addresses a setting where there is a labeling bias in the positives such that the prior for the training data does not match that of the test data. Hsieh et al. (2019) introduce a scenario where biased negative (bN) data can be collected alongside the standard PUL data. They propose the PUbN loss as an extension of the nnPU loss tailored to this special data setting. Su et al. (2021) propose a version of the nnPU loss that can handle imbalanced data scenarios with underrepresented positives. This scenario corresponds to a low prior π and hence a small ratio of positives in the combined dataset D. Su et al. (2021) argue that this setting is more realistic than general, balanced PUL scenarios as, for instance, disease diagnoses in medical records or fraud in financial data are usually underrepresented compared to healthy patients or correct financial transactions.

Next to these reweighting-based loss formulations, earlier two-step approaches focused on the identification of reliable negatives within D_u and the subsequent training of (semi-)supervised classification models on these reliable negatives along positives and eventually unlabeled samples (Bekker and Davis, 2020). Initially developed for text classification problems, methods for the first step often focus on metrics that measure the distance between D_u and D_p to identify reliable negatives within D_u. For instance, Li and Liu (2003) learn prototypes of positive and negative samples and then use the cosine distance to identify reliable negatives. Subsequently, an SVM classifier is trained on this newly constructed dataset of positives and reliable negatives. They also show that the first step can be combined with k-means clustering to improve the selection of reliable negatives. Similarly, Zhang and Zuo (2009) use an approach based on k-means that uses the distance to the nearest k positives as a metric to select reliable negatives with subsequent SVM classification and Liu et al. (2002) use a Naive Bayes classifier combined with smart thresholding to detect reliable negatives. Yu (2005) proposed an iterative approach using SVMs which shares similarities with the self-training procedure described in Section 2.3.1.

The formulation of the nnPU loss brought a model-agnostic view on PUL and enabled the integration of PUL training within DNNs (Kiryo et al., 2017). Following this, generative approaches using GANs for PUL emerged (Hou et al., 2018; Chiaroni et al., 2018; Liu et al., 2019) where the generator was trained to hallucinate positive and negative samples which would then serve

as a training dataset for a binary classifier. With VPU, another DL approach was introduced by Chen et al. (2020a) which replaced the nnPU loss with a variational loss formulation that allows the implicit estimation of π during model training. Similarly, Yoo et al. (2021) introduced an approach for graph-based PUL, where the relationships between samples are known and embedded in a graph structure, which also does not require information about π. Within Split-PU, Xu et al. (2022) combine best practices from SSL such as teacher-student modeling and the weak-strong augmentation scheme for consistency regularization for PUL, both of which were introduced in Section 2.3. Chen et al. (2020b) introduced the SSL concept of self-training to PUL. With Self-PU, they proposed an approach where self-paced learning, a confidence-weighting scheme based on the model predictions, and a teacher-student distillation approach are combined.

2.4.2 Contribution

In Section 4.3 we describe one of the few approaches that apply self-training to PUL settings to further leverage the unlabeled data next to their use in the weighted loss function (Dorigatti et al., 2022). Thereby, we overcome the problem of overconfident pseudo-labels via explicit modeling of uncertainty rather than the teacher-student paradigm (Chen et al., 2020b). While overconfident pseudo-labels in self-training are a general issue, it is getting exacerbated in low-labeled data scenarios and settings with high class imbalancedness. Hence, we propose an uncertainty-aware approach that uses model ensembling as a means to yield well-calibrated predictions. This helps overcome this issue and enables a substantial performance increase over state-of-the-art baselines in both heavily imbalanced and general PUL scenarios.

2.5 Constrained Clustering

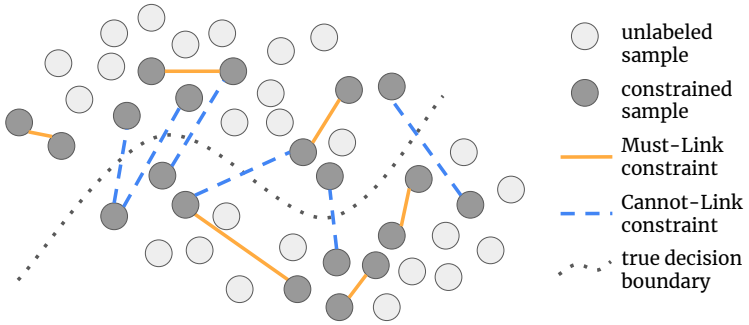


Figure 2.5: Illustration of the (semi-) constrained clustering scenario for a binary clustering task on a toy dataset. In CC, the label information is provided as constraints between samples instead of instance-level class labels, pairwise binary constraints in this setting. The dark gray dots resemble constrained samples and hence the classic CC setting while the addition of unlabeled samples (light gray dots) extends it to the semi-constrained setting.

The term constrained clustering refers to a subset of weakly-supervised learning methods in which the label information is provided in the form of constraints that describe the relationship between samples, as opposed to instance-level class labels, which are commonly assumed in classification settings. The constraint annotations, also referred to as weak annotations, typically contain less information about the respective constrained samples than instance-level class labels (Zhang et al., 2021b). This is illustrated in Figure 2.5, which depicts pairwise binary constraints, such as Must-Link and Cannot-Link constraints, that carry the information of whether two constrained samples belong to the same or different clusters, respectively. As indicated by the method's name, constraint annotations are used to guide the clustering of the data. Many standard ML algorithms have been extended towards the use of constraints such as k-means clustering (Wagstaff et al., 2001; Davidson and Ravi, 2005b), the EM algorithm (Basu et al., 2004), spectral clustering (Wang and Davidson, 2010) or hierarchical clustering (Davidson and Ravi, 2005a). Additionally, CC has been integrated with deep neural networks as introduced by Hsu and Kira (2016). Constraints can come in different forms, including binary pairwise constraints (Wagstaff and Cardie, 2000), instance difficulty constraints, triplet constraints, and global size constraints, among others (Zhang et al., 2021b). This chapter and the related contributions focus on binary pairwise constraints, referred to as constraints in the following for the sake of readability.

Constraints can be a useful, weaker alternative to instance-specific class labels in certain data annotation scenarios. Class label annotation requires the classes to be non-ambiguous and clearly defined with the cardinality of the class label space being fixed. However, this may not always be feasible, particularly in situations where the exact grouping and number of final classes are not known or the class cardinality is too large for a human annotator to remember during the data annotation process. One example of this difficulty is in text settings, such as the Yahoo! example (Cohn et al., 2003), where the task is to group different texts into reasonable clusters to create a searchable taxonomy despite neither knowing the exact grouping nor the number of final classes of the text samples. Another example is when the goal is to build a classifier that detects persons based on portraits (Georghiades et al., 2001). In this setting, a human annotator would have to select one out of a multitude of persons for each portrait during the data annotation process, requiring her to keep all potential persons in mind during this task. Pairwise binary constraint

annotation can be a remedy to these difficulties as the annotator is tasked with assigning a binary constraint to pairs of data, rather than selecting a specific class label. This allows for the annotation of samples despite only having a vague understanding of the different categories and also alleviates the annotator from keeping all potential classes in mind. Refer to Davidson and Basu (2007) for a comprehensive collection of use cases for CC from different domains.

2.5.1 Methods

Wagstaff and Cardie (2000) introduced the use of binary pairwise constraints and spurred the adaptation of many existing algorithms towards the use of constraints including k-means clustering (Wagstaff et al., 2001; Davidson and Ravi, 2005b), the EM algorithm (Basu et al., 2004), spectral clustering (Wang and Davidson, 2010) or hierarchical clustering (Davidson and Ravi, 2005a). In another line of research, constraints are used to learn a pairwise distance metric (Xing et al., 2002; Davis et al., 2007; Anand et al., 2013) which can subsequently be used in a separate clustering step. In contrast to this, methods such as MPCK-Means (Bilenko et al., 2004) or CECM (Antoine et al., 2012) integrate pairwise constraints into both metric learning and clustering. There also exist several comprehensive overview papers for the algorithmic developments next to this overview on classical ML methods for CC (Davidson and Basu, 2007; Dinler and Tural, 2016; Zhang et al., 2021b). Next to pairwise binary constraints, there is also a variety of different constraint types such as triplet constraints that describe the relationship of a triple of instances where samples are annotated as an anchor, a positive or a negative (Zhang et al., 2021b). The semantic meaning is that the anchor is expected to be closer to the positive than the negative sample. Another example is instance difficulty constraints that indicate whether specific samples are hard or easy to cluster (Zhang et al., 2021b). The remainder of this section is focused on pairwise binary constraints.

Formally, we define an sample from the input space $\boldsymbol{x} \in \mathcal{X}$ and a target space $\mathcal{Y} = \{1, ..., K\}$ of $K = |\mathcal{Y}|$ potential clusters. Analogous to supervised classification introduced in the notation Section 2.1, we aim at training a clustering model f parametrized with $\boldsymbol{\theta}$ such that $f : \mathcal{X} \mapsto \mathbb{R}^K$. The model predicts a probability distribution over cluster assignments $\hat{\boldsymbol{y}} = f(\boldsymbol{x}|\hat{\boldsymbol{\theta}})$ for sample $\boldsymbol{x}$ where $\hat{y}_k$ denotes the predicted probability of $\boldsymbol{x}$ belonging to cluster $k \in 1, ..., K$. The final cluster assignment prediction for cluster k then results as $\arg\max_k \hat{y}_k$. For model training, we consider an initial dataset of unlabeled samples $\boldsymbol{x} \in D$ from which n_c pairwise samples are annotated with a corresponding pairwise binary constraint $c \in \{0, 1\}$, forming the constrained dataset $D_c = \{(\boldsymbol{x}, \boldsymbol{x}', c)^{(l)} | l = 1, ..., n_c, \boldsymbol{x} \in D, \boldsymbol{x}' \in D, \boldsymbol{x} \neq \boldsymbol{x}'\}$. These constraint annotations, later referred to as constraints, describe that both samples either correspond to the same cluster $c = 1$, termed Must-Link constraint, or to different clusters, $c = 0$, termed Cannot-Link constraint. At this point, it is important to mention that CC models aim at predicting K potential cluster assignments despite being trained with binary pairwise targets $c \in \{0, 1\}$ only. Note that when K is unknown, the model may have a larger number of outputs m than the ground truth number of clusters resulting in the mapping $f : \mathcal{X} \mapsto \mathbb{R}^m$. This is referred to as the overclustering scenario. In the case of semi-constrained clustering, we use an additional unlabeled dataset D_u that contains the remaining samples $\boldsymbol{x} \in D, \boldsymbol{x} \notin D_c$ which are not part of any constraint pair. For semi-constrained clustering, we use both the constrained dataset D_c and the unlabeled dataset D_u for model training.

In addition to the previously mentioned approaches, Hsu and Kira (2016) proposed a method for CC using DNNs. The proposed method utilizes a pairwise training technique with batches

of pairwise samples and their associated binary constraint in the form of $(\boldsymbol{x}, \boldsymbol{x}', c)$ as introduced above. They also introduced the Kullback-Leibler Constrained Loss (KCL) as a loss function for the pairwise training of CC models. The KCL is a pairwise loss formulation that is built on top of the Kullback-Leibler distance and is defined as:

$$L_{KCL}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}', c) = L(\hat{\boldsymbol{y}}|\hat{\boldsymbol{y}}', c, q) + L(\hat{\boldsymbol{y}}'|\hat{\boldsymbol{y}}, c, q) \tag{2.16}$$

where

$$L(\hat{\boldsymbol{y}}|\hat{\boldsymbol{y}}', c, q) = c \ \mathrm{KL}\left(\hat{\boldsymbol{y}}|\hat{\boldsymbol{y}}'\right) + (1-c) \max\left(0, q - \mathrm{KL}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{y}}')\right) \tag{2.17}$$

with $q > 0$ being a margin hyperparameter and

$$\mathrm{KL}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}') = \sum_{l=1}^{K} \hat{y}_k \log \frac{\hat{y}_k}{\hat{y}'_k} \tag{2.18}$$

being the Kullback-Leibler distance between the predicted cluster assignment vectors $\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}'$ of the input pair $\boldsymbol{x}, \boldsymbol{x}'$. The KCL penalizes the model for differences in cluster assignment predictions for a pair with a Must-Link constraint $c = 1$ using the Kullback-Leibler distance as a distance metric for the predicted cluster assignments. Similarly, a large loss is incurred for a pair with a Cannot-Link constraint $c = 0$ if the model predicts similar cluster assignments for both input samples. The final loss in Equation (2.16) is computed as the sum of the Kullback-Leibler distances, an asymmetric distance metric, to yield a symmetric loss formulation. This loss formulation alongside the pairwise training strategy allows the training of a regular multi-class classification network f with pairwise constraints only. Hsu et al. (2018) further used this loss formulation for unsupervised domain adaptation and unsupervised clustering with unseen categories, so-called cross-task learning. In follow-up work, Hsu et al. (2019) introduced the Meta-Classification Likelihood (MCL) as an improved version of the KCL. Similar to the KCL, it also allows pairwise training of a CC neural network with binary constraints. The MCL loss is inspired by problem reduction strategies (Allwein et al., 2000) that reduce the initial multiclass classification problem to a binary meta-problem. Prominent examples of such problem-reduction strategies are one-vs-all or one-vs-one settings where the multiclass classification problem is reduced to multiple binary classification problems. Following this interpretation, the multiclass classification task encapsulates multiple binary classification problems. Hsu et al. (2019) reverse this order such that the multiclass classification task of detecting multiple clusters in the data is wrapped by a binary classification task. This re-formulation allows them to solve this task with pairwise binary constraints while still yielding a discriminative CC model, similar to the KCL. Motivated by a maximum-likelihood formulation, the pairwise MCL loss function formulates as:

$$L_{MCL}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}', c) = -c \log \hat{s} - (1-c) \log(1-\hat{s}) \tag{2.19}$$

where the dot product of the two predicted cluster assignment vectors $\hat{s} = \hat{\boldsymbol{y}}^{\mathsf{T}} \hat{\boldsymbol{y}}' \in [0, 1]$ is the input and the constraint annotation c the target of the loss function. We can see the correspondence to the binary cross entropy loss $L_{BCE}(\hat{y}, y) = -y \log \hat{y} - (1-y) \log(1-\hat{y})$ from this formulation in Equation (2.19). Hsu et al. (2019) show empirically that the MCL allows more stable model

training with better model performance across a variety of datasets, making the MCL the successor of the KCL. They also provide an intuition for this observation by visualizing the loss landscapes and show that similar to the KCL, the MCL allows the training of CC models in the overclustering scenario.

2.5.2 Contributions

In Section 4.2, we describe an approach that addresses a core limitation of deep CC models (Hsu and Kira, 2016; Hsu et al., 2019), which is their strict requirement for constrained data only, and hence the inability to use unlabeled data for model training (Goschenhofer et al., 2023). To address this limitation, we propose the ConstraintMatch model architecture that allows for training with a combined constrained and unlabeled dataset $D = D_c \cup D_u$ in a semi-constrained setting. This architecture is inspired by recent developments in SSL such as the weak- and strong augmentation scheme (Sohn et al., 2020) and builds upon unsupervised clustering methods (Van Gansbeke et al., 2020). We demonstrate how the introduction of a pseudo-constraining mechanism can overcome the confirmation bias in this scenario which leads to improved performance across a series of CV benchmarks.

In Section 4.4, we explore the cluster discovery capabilities of CC models for short texts in the context of NLP (Goschenhofer et al., 2022). Specifically, we first show that deep CC applies well to topic detection in short texts compared to unsupervised and supervised baselines in both the regular and the overclustering scenario. Having shown these capabilities, we propose a dynamic topic discovery scenario where a second dataset with novel topics occurs sometime after the initial training dataset. We propose the use of deep CC and empirically show its usefulness for such a dynamic scenario.

2.6 Transfer Learning

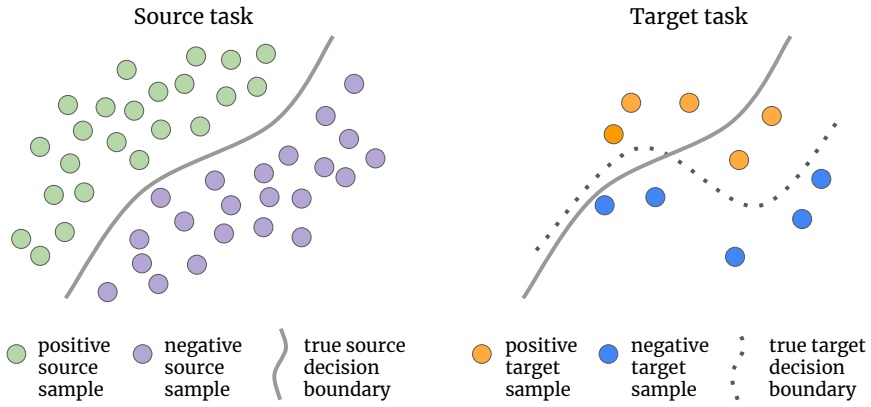


Figure 2.6: Illustration of the TL setting on two toy datasets. TL is directed at pre-training a model on a source task (left) and subsequently fine-tuning it on the target task (right), to support model training on the target task via the transfer of abstract knowledge from the source task, depicted as the source decision boundary (gray line). Depending on the similarity of the source and target tasks, TL can support the training of models on target tasks with few annotated samples.

2.6.1 Methods

Pre-training DNNs has become a de facto standard procedure before training on the DL task of interest across different data modalities such as CV (Sun et al., 2017; Mahajan et al., 2018; Dai et al., 2021), NLP (Mou et al., 2016; Howard and Ruder, 2018; Yamaguchi et al., 2021) or multimodal applications (Gan et al., 2022). For instance, using ImageNet pre-trained models in CV or pre-trained BERT embeddings in NLP has become a widely used practice. This strategy, termed transfer learning, is motivated by the concept of human learning and the transfer of knowledge from one task to a similar second task. For example, a musician who can play the violin may use her knowledge of music, such as a sense of rhythm or the ability to read sheet music, to master playing the piano more quickly than a person who has never played an instrument before. TL and SSL both aim to train models for target tasks with a limited amount of labeled data. While SSL exploits large amounts of unlabeled data from the target task, TL is directed at tasks where even unlabeled data from the target task is hard to gather and hence it reverts to data from a similar yet different source task. In that sense, the process of TL is to pre-train a model on a source task and then fine-tune it on the target task, hoping that abstractions learned in the source task provide a warm-start for the target task compared to random initialization of the model parameters (Zhuang et al., 2020). Intuitively, one expects that the similarity of the source and target tasks determines the benefit of TL suggesting that a similar source task is more beneficial for a target task than a dissimilar one. While this is often the case, it is not a universal rule (He et al., 2019) and there exist scenarios where the closeness in both tasks confuses the model fine-tuning on the target task, also referred to as negative transfer (Wang et al., 2019b). The nature of the source task in TL can also differ. Supervised source tasks are DL tasks with annotated data as prediction targets such as the infamous ImageNet pre-training for image classification (Mahajan et al., 2018; Sun et al., 2017). Other source tasks cover auxiliary problems that should teach the model an abstract understanding of the feature domain. Examples are word2vec (Mikolov et al.,

2013) in the NLP space, which has evolved into masked language modeling with the advent of transformer architectures (Yamaguchi et al., 2021), and the paradigm of self-supervised learning in CV (Jaiswal et al., 2020; Dai et al., 2021).

TL is also partially described as a form of weakly supervised learning. In that sense, the labels of the source dataset are usually more coarse (Taherkhani et al., 2019) or cover slightly different concepts (He et al., 2019) than that of the target task. Hence, they are also referred to as weak labels to differentiate from the strong, more informative, and specific labels of the target task. Taherkhani et al. (2019) introduce one illustrative example of such coarse labels using the ImageNet 2010 dataset to construct a two-level hierarchical dataset. Therein, the first-level source dataset contains 143 class annotations, also termed super-classes, and the granular second-level target dataset contains 387 classes. Concretely, two target images with classes "Cheetah" and "Leopard" share the same coarse label "Big Cat" from the source dataset. They then propose a model architecture tailored towards the use of weak supervision to leverage the learned concepts of the coarse source task for the more granular target task. With HTrans, Banerjee et al. (2019) propose a weakly supervised TL strategy for a similar situation in the context of NLP. This concept of pre-training on a weak and coarsely annotated source task has also been successfully applied in medical imaging scenarios where the lack of large annotated target datasets is especially immanent due to the scarcity of medical experts required for data annotation. For instance, (Ke et al., 2020) and (Ezhov et al., 2019) propose the use of a large set of coarse segmentation masks as a source task and the subsequent fine-tuning using fine-granular segmentation masks for the target task in the context of microscopy and teeth segmentation. Furthermore, (Hosseinzadeh et al., 2021) systematically benchmark pre-training on the ImageNet (Deng et al., 2009) and the iNat (Van Horn et al., 2021) datasets with both supervised and self-supervised pre-training on seven subsequent target tasks in the medical domain and find that more fine-grained source tasks support the fine-tuning on the target task better.

2.6.2 Contribution

In Section 4.5, we describe the application of a supervised TL scheme in the context of motor state prediction for patients with Parkinson's Disease using DL models for the classification of sensor movement data (Goschenhofer et al., 2019). The target task in this setting was the exact prediction of the disease progression on a clinical scale and, as often in the medical domain, we had limited access to annotated target data. Hence, we adopted above described TL scheme and used a larger second dataset with coarse, weak annotations as the source task. This source dataset contained movement data from persons with and without diagnosed Parkinson's Disease which allowed us to create the supervised source task of predicting whether a given movement data window corresponds to a healthy or a diagnosed person. Using this weakly supervised TL approach helped increase the performance of the model on the target task, potentially due to the similar domain of both tasks.

3 Conclusion and Outlook

3.1 Conclusion

The examples cited in the introduction of this thesis stress the need for large annotated training datasets and the effort involved in creating those, which is one key bottleneck for the development of tailored DL solutions. Throughout the subsequent chapters, various alternative concepts at the intersection of unsupervised and supervised learning were introduced, all directed at the same goal: reducing the need and hence the effort for data annotation. The key conclusion from those chapters is that recent developments in SSL, PUL, CC, and TL are suitable means to use an untapped treasure that comes as an abundant resource: unlabeled or weakly labeled data. While all aimed at the same target, these concepts help reduce the effort for data annotation in different ways: SSL is directed at including abundant unlabeled data to guide supervised model training which enables the training of potent models with few labeled data only. PUL allows the training of binary classification models despite the absence of negative data annotations using unlabeled data. CC allows the use of pairwise constraint annotations which are weaker and hence less effortful to gather compared to instance-specific class labels. TL allows the pre-training of DL models on coarse source tasks which can then be fined tuned on fine-grained target tasks with relatively few training data annotations.

To summarize the content of this thesis, Section 2.3 presented an overview of SSL with a focus on recent developments such as self-training and consistency regularization. In this context, contributions in this thesis expanded modern SSL from the image to the time series modality (Goschenhofer et al., 2021) and investigated the applicability, the robustness and the crucial role of data augmentation for SSL in the medical imaging domain (Dexl et al., 2022).
PUL was introduced in Section 2.4 as an edge case of binary semi-supervised classification where the labeled dataset contains exclusively positive samples while the unlabeled data consists of both positive and negative samples. Recent DL-based approaches for PUL combine reweighting-based loss functions with self-training strategies. With the PUUPL architecture, we proposed an improved version of self-training for PUL via the explicit inclusion of model uncertainty in the pseudo-labeling process (Dorigatti et al., 2022).
The concept of CC as a weakly supervised learning strategy was introduced in Section 2.5, explaining both the foundations and the use of DL-based approaches for model training with pairwise binary constraints. With ConstraintMatch, we propose a potent alternative to deep CC that allows the inclusion of unlabeled samples for model training and uses a pseudo-constraining mechanism to overcome the confirmation bias, one main shortcoming in self-training (Goschenhofer et al., 2023). We further demonstrated and exploited the cluster detection capabilities of CC models in an NLP context, where we applied CC to detect dynamically changing topics in short texts (Goschenhofer et al., 2022).
Finally, the concept of TL was introduced in Section 2.6 as an approach to leverage features ingrained in the model by pre-training on a source task. This pre-trained model is subsequently

trained on the target task of interest, which helps retain strong model performance despite few annotated training samples for the target task and speeds up model training. We applied this powerful paradigm within a time series classification problem in a medical context (Goschenhofer et al., 2019).

3.2 Outlook

Shaping a weakly supervised future for ML requires further research on the effective use of un- or weakly annotated data in different contexts. Some potential future ideas that build upon the contributions of this thesis are outlined in the following.

Pseudo-constraining beyond constrained clustering: The pseudo-constraining mechanism introduced in the Section 4.2 was primarily designed and investigated in the context of semi-constrained clustering (Goschenhofer et al., 2023). In this context it proved to be an effective approach to overcome the confirmation bias, leading to increased performance compared to the pseudo-labeling approach usually used in semi-supervised classification (Berthelot et al., 2019a; Sohn et al., 2020; Zhang et al., 2021a). Using the problem reduction concept of pseudo-constraining on top of pseudo-labels also in semi-supervised classification instead of semi-constrained clustering would be an interesting avenue. In this setting, the labeled data would contain instance-specific class labels while the unlabeled data would be fed to the model in a pairwise manner, using pseudo-constraints as targets for a pairwise loss function such as the MCL (Hsu et al., 2019). One problem that occurs in practical semi-supervised classification is a potential class distribution mismatch between the labeled and the unlabeled data that is hard to detect due to the non-annotated nature of the unlabeled dataset (Chen et al., 2020c). The proposed inclusion of a CC component into semi-supervised classification models could not only help overcome the confirmation bias but would also equip such a model with a cluster detection capability. Such an architecture could then also be trained in the overclustering scenario to detect the underlying clusters in the potentially different D_u. Potentially, this would then allow extending such semi-supervised classification models also for use in settings with unknown cardinality of the class target space.

Uncertainty quantification for CC: Uncertainty quantification, i.e. the calibration of classification model predictions, is an active and important research field with major developments in the past years including the introduction of MCDropout (Gal and Ghahramani, 2016), model ensembling (Lakshminarayanan et al., 2017) or conformal predictions (Angelopoulos and Bates, 2021) for DNNs. It is driven by the practical need for model prediction scores that can be interpreted as valid probability values for decision support systems in safety-critical applications such as healthcare or autonomous driving. While there exists a plethora of research on uncertainty quantification for classification tasks, it is yet an underexplored area in the context of weakly supervised learning and CC specifically. Future research on uncertainty quantification for CC could fill this important gap and facilitate the application of such models in different safety-critical domains.

Human-in-the-loop weakly supervised learning: Active learning deals with the same data situation as SSL with the difference that an oracle, most often a human annotator, can be queried to annotate a select set of unlabeled data. Hence, the focus lies on the development of methods to optimally select unlabeled samples that improve the model performance the most in a model-driven way. Active and semi-supervised learning are therefore also described as two sides of the

same coin where active learning aims at exploring yet unknown spaces in the unlabeled data while SSL is directed at exploiting what the model already knows about them (Settles, 2009). This similarity suggests the combination of both concepts. First attempts in that direction have been made (Mittal et al., 2019; Gao et al., 2020b; Bengar et al., 2021), mostly focused on semi-supervised classification settings. Following these approaches, it would be exciting to investigate whether active learning can also be combined with weakly supervised approaches beyond instance-level class labels as used in CC. Also, using such human-in-the-loop approaches would allow for the combination of different methods, i.e. mixed supervision. A first step in that direction would be the design of an active learning strategy that intelligently queries weak constraint annotations or instance-specific class labels, or both. This could enable a model-driven balancing of the tradeoff between ease of annotation, i.e. quantity, as constraints are more effortless to annotate than class labels, and the annotation informativeness, i.e. quality, as instance-specific class labels contain more information than binary constraints.

To conclude this thesis, I want to return to the introduction, where I expanded on Humby and Palmer's famous quote by saying that *"Data is the new oil. Like oil, data is valuable, but if unrefined and non-annotated it cannot really be used for supervised machine learning"*. While acknowledging the importance of data with high-quality annotations, this extended quote refers only to the context of supervised ML and limits the training resource to annotated data only. However, this thesis sheds light on the fact that unlabeled or weakly labeled data is an alternative and more accessible resource than labeled data, which requires less refinement and can fuel various applications beyond supervised learning. This is analogous to the current societal transition from unsustainable, hard-to-obtain natural resources such as oil to sustainable, renewable, and thus abundant green resources. While this current transition phase requires the use of both resource sources, I argue that the future of energy is in regenerative, sustainable resources, and the future of ML is in leveraging un- and weakly annotated data.

References

Anish Acharya, Sujay Sanghavi, Li Jing, Bhargav Bhushanam, Dhruv Choudhary, Michael Rabbat, and Inderjit Dhillon. 2022. Positive unlabeled contrastive learning. *International Conference on Machine Learning*.

A Agrawala. 1970. Learning with a probabilistic teacher. *IEEE Transactions on Information Theory*, 16(4):373–379.

Erin L Allwein, Robert E Schapire, and Yoram Singer. 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of machine learning research*, 1(Dec):113–141.

Saket Anand, Sushil Mittal, Oncel Tuzel, and Peter Meer. 2013. Semi-supervised kernel mean shift clustering. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1201–1215.

Anastasios Nikolas Angelopoulos and Stephen Bates. 2021. A gentle introduction to conformal prediction and distribution-free uncertainty quantification.

Violaine Antoine, Benjamin Quost, M-H Masson, and Thierry Denoeux. 2012. Cecm: Constrained evidential c-means algorithm. *Computational Statistics & Data Analysis*, 56(4):894–914.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Charles Arthur. 2013. Tech giants may be huge, but nothing matches big data.

Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *ArXiv*, abs/1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*.

Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, and Kostas Tsioutsiouliklis. 2019. Hierarchical transfer learning for multi-label text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6295–6300, Florence, Italy. Association for Computational Linguistics.

Sugato Basu, Mikhail Bilenko, and Raymond J Mooney. 2004. A probabilistic framework for semi-supervised clustering. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 59–68.

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

Javad Zolfaghari Bengar, Joost van de Weijer, Bartlomiej Twardowski, and Bogdan Raducanu. 2021. Reducing label effort: Self-supervised meets active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1631–1639.

David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. 2019a. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019b. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32.

Mikhail Bilenko, Sugato Basu, and Raymond J Mooney. 2004. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 11.

Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. 2021. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. *AAAI*.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning. *Cambridge, Massachusettes: The MIT Press View Article*, 20(3):542–542.

Hui Chen, Fangqing Liu, Yin Wang, Liyue Zhao, and Hao Wu. 2020a. A variational approach for learning from positive and unlabeled data. In *Advances in Neural Information Processing Systems*, volume 33, pages 14844–14854. Curran Associates, Inc.

Xuxi Chen, Wuyang Chen, Tianlong Chen, Ye Yuan, Chen Gong, Kewei Chen, and Zhangyang Wang. 2020b. Self-pu: Self boosted and calibrated positive-unlabeled training. In *International Conference on Machine Learning*, pages 1510–1519. PMLR.

Yanbei Chen, Xiatian Zhu, Wei Li, and Shaogang Gong. 2020c. Semi-supervised learning under class distribution mismatch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3569–3576.

Florent Chiaroni, Mohamed-Cherif Rahal, Nicolas Hueber, and Frédéric Dufaux. 2018. 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.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Marthinus Christoffel, Gang Niu, and Masashi Sugiyama. 2016. Class-prior estimation for learning from positive and unlabeled data. In *Asian Conference on Machine Learning*, volume 45 of *Proceedings of Machine Learning Research*, pages 221–236.

References

David Cohn, Rich Caruana, and Andrew McCallum. 2003. Semi-supervised clustering with user feedback. *Constrained clustering: advances in algorithms, theory, and applications*, 4(1):17–32.

Ronan Collobert, Fabian Sinz, Jason Weston, Léon Bottou, and Thorsten Joachims. 2006. Large scale transductive svms. *Journal of Machine Learning Research*, 7(8).

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123.

Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, volume 33, pages 18613–18624. Curran Associates, Inc.

Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. 2021. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1601–1610.

Ian Davidson and Sugato Basu. 2007. A survey of clustering with instance level constraints. *ACM Transactions on Knowledge Discovery from data*, 1(1-41):2–42.

Ian Davidson and SS Ravi. 2005a. Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 59–70. Springer.

Ian Davidson and SS Ravi. 2005b. Clustering with constraints: Feasibility issues and the k-means algorithm. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 138–149. SIAM.

Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. 2007. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

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

Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Li Fei-Fei. 2014. Scalable multi-label annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3099–3102.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jakob Dexl, Michaela Benz, Petr Kuritcyn, Thomas Wittenberg, Volker Bruns, Carol Geppert, Arndt Hartmann, Bernd Bischl, and Jann Goschenhofer. 2022. Robust colon tissue cartography with semi-supervision. *Current Directions in Biomedical Engineering*, 8(2):344–347.

Derya Dinler and Mustafa Kemal Tural. 2016. A survey of constrained clustering. In *Unsupervised learning algorithms*, pages 207–235. Springer.

Emilio Dorigatti, Jann Goschenhofer, Benjamin Schubert, Mina Rezaei, and Bernd Bischl. 2022. Positive-unlabeled learning with uncertainty-aware pseudo-label selection. *arXiv preprint arXiv:2201.13192*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*.

Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. 2014. Analysis of learning from positive and unlabeled data. *Advances in neural information processing systems*, 27:703–711.

Matvey Ezhov, Adel Zakirov, and Maxim Gusarev. 2019. Coarse-to-fine volumetric segmentation of teeth in cone-beam ct. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 52–56. IEEE.

S Fralick. 1967. Learning to recognize patterns without a teacher. *IEEE Transactions on Information Theory*, 13(1):57–64.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, Jianfeng Gao, et al. 2022. Vision-language pre-training: Basics, recent advances, and future trends. *Foundations and Trends® in Computer Graphics and Vision*, 14(3–4):163–352.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020a. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ö Arık, Larry S Davis, and Tomas Pfister. 2020b. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 510–526. Springer.

Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):643–660.

Atin Ghosh and Alexandre H Thiery. 2021. On data-augmentation and consistency-based semi-supervised learning. *ICLR*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

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

Jann Goschenhofer. 2022. Deep semi-supervised learning for time-series classification. In *Deep Learning Applications, Volume 4*, pages 361–384. Springer.

Jann Goschenhofer, Bernd Bischl, and Zsolt Kira. 2023. Constraintmatch for semi-constrained clustering. *International Joint Conference on Neural Networks (IJCNN)*.

Jann Goschenhofer, Rasmus Hvingelby, David Rügamer, Janek Thomas, Moritz Wagner, and Bernd Bischl. 2021. Deep semi-supervised learning for time series classification. In *20th IEEE International Conference on Machine Learning and Applications (ICMLA)*.

Jann Goschenhofer, Franz MJ Pfister, Kamer Ali Yuksel, Bernd Bischl, Urban Fietzek, and Janek Thomas. 2019. Wearable-based parkinson's disease severity monitoring using deep learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 400–415. Springer.

Jann Goschenhofer, Pranav Ragupathy, Christian Heumann, Bernd Bischl, and Matthias Assenmacher. 2022. Cc-top: Constrained clustering for dynamic topic discovery. *Proceedings of the First Workshop on Ever Evolving NLP (EvoNLP)*, page 26–34.

Yves Grandvalet, Yoshua Bengio, et al. 2005. Semi-supervised learning by entropy minimization. *NeurIPS*, 367:281–296.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Kaiming He, Ross Girshick, and Piotr Dollar. 2019. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Taher Hosseinzadeh, Reza Mohammad, Fatemeh Haghighi, Ruibin Feng, Michael B Gotway, and Jianming Liang. 2021. A systematic benchmarking analysis of transfer learning for medical image analysis. In *Domain Adaptation and Representation Transfer, and Affordable Healthcare and AI for Resource Diverse Global Health: Third MICCAI Workshop, DART 2021, and First MICCAI Workshop, FAIR 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27 and October 1, 2021, Proceedings 3*, pages 3–13. Springer.

Ming Hou, Brahim Chaib-Draa, Chao Li, and Qibin Zhao. 2018. Generative adversarial positive-unlabelled learning. *AAAI*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *Proceedings of the 56th Annual Meeting of the As- sociation for Computational Linguistics*, 1:328–339.

Yu-Guan Hsieh, Gang Niu, and Masashi Sugiyama. 2019. Classification from positive, unlabeled and biased negative data. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2820–2829. PMLR.

Yen-Chang Hsu and Zsolt Kira. 2016. Neural network-based clustering using pairwise constraints. *ICLR Workshop*.

Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. 2018. Learning to cluster in order to transfer across domains and tasks. *ICLR*.

Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. 2019. Multi-class classification without multi-class labels. *ICLR*.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. 2019. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5070–5079.

Nobutaka Ito and Masashi Sugiyama. 2022. Audio signal enhancement with learning from positive and unlabelled data. *arXiv preprint arXiv:2210.15143*.

Brian Kenji Iwana and Seiichi Uchida. 2021. An empirical survey of data augmentation for time series classification with neural networks. *Plos one*, 16(7):e0254841.

Shantanu Jain, Martha White, and Predrag Radivojac. 2017. Recovering true classifier performance in positive-unlabeled learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2.

Masahiro Kato, Takeshi Teshima, and Junya Honda. 2019. Learning from positive and unlabeled data with a selection bias. In *International conference on learning representations*.

Rihuan Ke, Aurélie Bugeau, Nicolas Papadakis, Peter Schuetz, and Carola-Bibiane Schönlieb. 2020. Learning to segment microscopy images with lazy labels. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 411–428. Springer.

Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson WH Lau. 2019. Dual student: Breaking the limits of the teacher in semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6728–6736.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ryuichi Kiryo, Gang Niu, Marthinus C. du Plessis, and Masashi Sugiyama. 2017. Positive-unlabeled learning with non-negative risk estimator. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1674–1684, Red Hook, NY, USA. Curran Associates Inc.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. 2020. Featmatch: Feature-based augmentationfor semi-supervised learning. In *European Conference on Computer Vision*. Springer.

Samuli Laine and Timo Aila. 2017. Temporal ensembling for semi-supervised learning. *ICLR*.

Balaji Lakshminarayanan, A. Pritzel, and C. Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*.

Wei Lan, Jianxin Wang, Min Li, Jin Liu, Yaohang Li, Fang-Xiang Wu, and Yi Pan. 2016. Predicting drug–target interaction using positive-unlabeled learning. *Neurocomputing*, 206:50–57.

Dong-Hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

Joseph Lemley, Shabab Bazrafkan, and Peter Corcoran. 2017. Smart augmentation learning an optimal data augmentation strategy. *Ieee Access*, 5:5858–5869.

Huayi Li, Zhiyuan Chen, Bing Liu, Xiaokai Wei, and Jidong Shao. 2014. Spotting fake reviews via collective positive-unlabeled learning. In *2014 IEEE international conference on data mining*, pages 899–904. IEEE.

Xiaoli Li and Bing Liu. 2003. Learning to classify texts using positive and unlabeled data. In *IJCAI*, volume 3, pages 587–592. Citeseer.

Yiting Li, Lu Liu, and Robby T Tan. 2019. Certainty-driven consistency loss for semi-supervised learning. *CVPR*.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2022. A survey of transformers. *AI Open*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Bing Liu, Wee Sun Lee, Philip S Yu, and Xiaoli Li. 2002. Partially supervised classification of text documents. In *ICML*, volume 2, pages 387–394. Sydney, NSW.

Fangqing Liu, Hui Chen, Liyue Zhao, and Hao Wu. 2019. Discriminative adversarial networks for positive-unlabeled learning. *ArXiv*, abs/1906.00642.

Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. 2021. Unbiased teacher for semi-supervised object detection. *ICLR*.

Yen-Cheng Liu, Chih-Yao Ma, and Zsolt Kira. 2022. Unbiased teacher v2: Semi-supervised object detection for anchor-free and anchor-based detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9819–9828.

Chuan Luo, Pu Zhao, Chen Chen, Bo Qiao, Chao Du, Hongyu Zhang, Wei Wu, Shaowei Cai, Bing He, Saravanakumar Rajmohan, et al. 2021. Pulns: Positive-unlabeled learning with effective negative sample selector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8784–8792.

Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. 2018. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8896–8905.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. 2018. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196.

Christopher J Merz, DC St Clair, and William E Bond. 1992. Semi-supervised adaptive resonance theory (smart2). In *IJCNN International Joint Conference on Neural Networks*, volume 3, pages 851–856. IEEE.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Sudhanshu Mittal, Maxim Tatarchenko, Özgün Çiçek, and Thomas Brox. 2019. Parting with illusions about deep active learning. *arXiv preprint arXiv:1912.05361*.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

References

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111*.

Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. 2018. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems*, 31.

OpenAI. 2023. Chatgpt. *chat.openai.com/*.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*.

Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon. 2018. Adversarial dropout for supervised and semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. *NeurIPS Autodiff Workshop*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. 2015. Semi-supervised learning with ladder networks. *Advances in neural information processing systems*, 28.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. 2021. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *ICLR*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. 2005. Semi-supervised self-training of object detection models. *Seventh IEEE Workshops on Applications of Computer VIsion (WACV)*, 1.

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

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

Henry Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371.

Burr Settles. 2009. Active learning literature survey.

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. 2019. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *CoRR*, abs/2001.07685.

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2443–2449.

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

Guangxin Su, Weitong Chen, and Miao Xu. 2021. Positive-unlabeled learning from imbalanced data. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.

Fariborz Taherkhani, Hadi Kazemi, Ali Dabouei, Jeremy Dawson, and Nasser M Nasrabadi. 2019. A weakly supervised fine label classifier enhanced by coarse supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6459–6468.

Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28.

Jesper E Van Engelen and Holger H Hoos. 2020. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. 2020. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pages 268–285. Springer.

Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. 2021. Benchmarking representation learning for natural world image collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12884–12893.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Kiri Wagstaff and Claire Cardie. 2000. Clustering with instance-level constraints. *AAAI/IAAI*, 1097:577–584.

Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. 2001. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584.

Xi Wang, Hao Chen, Caixia Gan, Huangjing Lin, Qi Dou, Efstratios Tsougenis, Qitao Huang, Muyan Cai, and Pheng-Ann Heng. 2019a. Weakly supervised deep learning for whole slide lung cancer image analysis. *IEEE transactions on cybernetics*, 50(9):3950–3962.

Xiang Wang and Ian Davidson. 2010. Flexible constrained spectral clustering. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 563–572.

Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. 2019b. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11293–11302.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020a. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020b. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698.

Eric Xing, Michael Jordan, Stuart J Russell, and Andrew Ng. 2002. Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, 15.

Chengming Xu, Chen Liu, Siqian Yang, Yabiao Wang, Shijie Zhang, Lijie Jia, and Yanwei Fu. 2022. Split-pu: Hardness-aware training strategy for positive-unlabeled learning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2719–2729.

Miao Xu, Bingcong Li, Gang Niu, Bo Han, and Masashi Sugiyama. 2019. Revisiting sample selection approach to positive-unlabeled learning: Turning unlabeled data into positive rather than negative. *arXiv preprint arXiv:1901.10155*.

Atsuki Yamaguchi, George Chrysostomou, Katerina Margatina, and Nikolaos Aletras. 2021. Frustratingly simple pretraining alternatives to masked language modeling. *arXiv preprint arXiv:2109.01819*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196.

Jaemin Yoo, Junghun Kim, Hoyoung Yoon, Geonsoo Kim, Changwon Jang, and U Kang. 2021. Accurate graph-based pu learning without class prior. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 827–836. IEEE.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *ICLR*.

Hwanjo Yu. 2005. Single-class classification with mapping convergence. *Machine Learning*, 61(1):49–69.

Bangzuo Zhang and Wanli Zuo. 2009. Reliable negative extracting based on knn for learning from positive and unlabeled examples. *Journal of Computers*, 4(1):94–101.

Bowen Zhang, Yidong Wang, Wenxin Hou, HAO WU, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021a. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, volume 34, pages 18408–18419. Curran Associates, Inc.

Hongjing Zhang, Tianyang Zhan, Sugato Basu, and Ian Davidson. 2021b. A framework for deep constrained clustering. *Data Mining and Knowledge Discovery*, 35(2):593–620.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. Mixup: Beyond empirical risk minimization. *ICLR*.

Zixing Zhang, Fabien Ringeval, Bin Dong, Eduardo Coutinho, Erik Marchi, and Björn Schüller. 2016. Enhanced semi-supervised learning for multimodal emotion recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5185–5189. IEEE.

Zipei Zhao, Fengqian Pang, Yaou Liu, Zhiwen Liu, and Chuyang Ye. 2023. Positive-unlabeled learning for binary and multi-class cell detection in histopathology images with incomplete annotations. *Journal of Machine Learning for Biomedical Imaging*.

Xiaojin Zhu and Zoubin Ghahramanih. 2002. Learning from labeled and unlabeled data with label propagation. *Technical Report*.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.

4 Contributions

4.1 Deep Semi-supervised Learning for Time Series Classification

Contributing article:

Jann Goschenhofer, Rasmus Hvingelby, David Rügamer, Janek Thomas, Moritz Wagner, and Bernd Bischl. 2021. Deep semi-supervised learning for time series classification. In *20th IEEE International Conference on Machine Learning and Applications (ICMLA)*

Following the above main publication, an extended version of this paper culminated in the book chapter:

Jann Goschenhofer. 2022. Deep semi-supervised learning for time-series classification. In *Deep Learning Applications, Volume 4*, pages 361–384. Springer

Both original articles are attached in the following.

Author contributions:

Jann Goschenhofer was responsible for the conceptualization of this paper (i.e. idea, goal, and scope), with feedback and input provided by Janek Thomas, David Rügamer, and Bernd Bischl. The software for the experiments was implemented by Jann Goschenhofer with support from Moritz Wagner and Rasmus Hvingelby, who implemented the self-supervised baseline method and assisted in coding reviews. All experiments were run by Jann Goschenhofer, except those experiments on different backbone architectures which were run by Rasmus Hvingelby. Computing resources that were provided by Bernd Bischl. Jann Goschenhofer was responsible for the writing of the manuscript with support from David Rügamer, Janek Thomas, Rasmus Hvingelby, and Bernd Bischl.

Copyright information:

2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)

Deep Semi-supervised Learning for Time Series Classification

Jann Goschenhofer[1,2], Rasmus Hvingelby[2], David Ruegamer[1], Janek Thomas[1], Moritz Wagner[2], Bernd Bischl[1,2]
LMU Munich, Munich, Germany[1]
Fraunhofer Institute for Integrated Circuits (IIS), Erlangen, Germany[2]
jann.goschenhofer@stat.uni-muenchen.de

2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA) | 978-1-6654-4337-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICMLA52953.2021.00072

***Abstract*—While deep semi-supervised learning has gained much attention in computer vision, limited research exists on its applicability in the time series domain. In this work, we investigate the transferability of state-of-the-art deep semi-supervised models from image to time series classification. We discuss the necessary model adaptations, in particular an appropriate model backbone architecture and the use of tailored data augmentation strategies. Based on these adaptations, we explore the potential of deep semi-supervised learning in the context of time series classification by evaluating our methods on large public time series classification problems with varying amounts of labeled samples. We perform extensive comparisons under a decidedly realistic and appropriate evaluation scheme with a unified reimplementation of all algorithms considered, which is yet lacking in the field. We find that these transferred semi-supervised models show significant performance gains over strong supervised, semi-supervised and self-supervised alternatives, especially for scenarios with very few labeled samples.**

***Index Terms*—Semi-supervised Learning, Time Series Classification, Data Augmentation**

I. Introduction

Time series classification (TSC) spans many real-world applications in domains from healthcare [1] over cybersecurity [2] to manufacturing [3]. Several algorithms for TSC have been proposed over the years [4] [5].

In many real-world scenarios, time series data can be collected easily, but acquiring labels for this data is costly. For instance, in disease monitoring, sensor data are collected with low effort but the labelling of this data requires time-consuming work by medical experts [6]. Semi-supervised learning (SSL) addresses this by leveraging large amounts of unlabeled data in combination with a small amount of labeled data when training machine learning (ML) models.

Especially in computer vision, the advances in deep neural networks and the promised label efficiency of SSL have lead to the introduction of several innovative approaches for image data [7]. While there is much work on classical semi-supervised models for TSC, research on the use of neural network-based SSL algorithms for TSC is still limited.

This motivates our main research question that we approach holistically in this work: *Can we transfer well established deep semi-supervised models from the image to the time series domain?* More specifically, we answer this question for the most prominent state-of-the-art SSL approaches, by proposing adaptions for MixMatch [8], Virtual Adversarial Training [9], the Mean Teacher [10] and the Ladder Net [11]. These include the modification of a suitable backbone architecture as well as adaptions of an appropriate data augmentation strategy to account for the domain transfer of these models. For demonstration of the efficacy of our proposed frameworks we adhere to best practices for realistic evaluation of semi-supervised models and provide a fair and reliable model comparison with a high degree of practicality [12].

A. Related Work

a) Time Series Classification: Over the past years, a variety of methods has been developed for TSC. A detailed overview on classical ML methods that were specifically developed for TSC [13], [14], [15] is provided in [4]. An alternative approach towards TSC consists in the extraction of statistical features from the raw time series as the basis for training any strong classifier for tabular data [16]. Also in deep learning, specific methods for time series classification have been developed [17], [18], [19]. A comprehensive overview on these recent developments can be found in [5].

b) Semi-Supervised Learning: There exists a plethora of different concepts that extract additional information from unlabeled data via semi-supervision. These range from the extension of supervised ML methods such as the semi-supervised Support Vector Machine [20] or semi-supervised Boosting [21] to inherently semi-supervised methods such as Label Propagation [22], Manifold Regularization [23] or Co-Training [24]. [25] provide a detailed overview on these semi-supervised approaches. There is also growing research on deep semi-supervised learning, mainly driven by the computer vision community. A recent overview and taxonomy on these developments are provided by [7]. Amongst these are graph-based methods such as Deep Label Propagation [26], SNTG [27] or the extension of pseudo-labelling for deep learning [7]. Further, there is growing research on regularization-based approaches following the rationale of adding an additional unsupervised regularization loss term to the initial supervised loss. The Mean Teacher [10] and its predecessors, Temporal Ensembling and the Π-Model [28], employ a consistency loss over the unlabeled samples to reward similar predictions for differently augmented versions of the same unlabeled sample. To overcome one drawback of those methods, the need for domain-dependent data augmentation strategies, Virtual Adversarial Training (VAT) [9] adds small perturbations to the input data to create an auxiliary unsupervised training

978-1-6654-4337-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ICMLA52953.2021.00072

target. MixMatch [8] in turn combines different regularization strategies in one common framework. These regularization-based approaches yield state-of-the-art performance on image classification benchmarks.

c) SSL for TSC: Different classical semi-supervised models have been developed for TSC. In their foundational work, [29] propose an approach that combines pseudo-labelling with a nearest-neighbor model for imbalanced, binary TSC tasks. This cluster-then-label [7] rationale for labeled and unlabeled time series via custom distance metrics is also employed in approaches such as DTW-D [30], SUCCESS [31] or LCLC [32]. Graph-based label propagation [22] is combined with time-series-specific distance metrics by [33] and [34] introduced the shapelet-based SSSL.

d) Deep SSL for TSC: There has been recent developments on neural net-based approaches. A customized version of the LadderNet [11] based on the FCN architecture [17] was applied by [35] on three multivariate human activity recognition (HAR) datasets. They report relative gains of the semi-supervised model over the supervised baselines for small amounts of labeled samples. To the best of our knowledge, [35] are the first to evaluate SSL methods on large, multivariate TSC datasets. A self-supervised approach, where the model is jointly trained on an auxiliary forecasting task over the whole dataset next to the initial supervised classification task on the labeled data only, was introduced by [36]. They build upon the benchmark of [34] on a subset of smaller, univariate TSC datasets from the UCR repository [3] and report state-of-the-art performance compared to the majority of above methods as well as a customized variant of the $\Pi-$Model [28] that works on time series problems. In alignment with [35], they report particularly strong model performance for the deep supervised baseline FCN [17] trained on few labeled samples only reporting it to outperform all above mentioned classical semi-supervised models. This deep learning baseline outperforms all of the classical semi-supervised models and almost always beats the $\Pi-$Model. We include this approach as a self-supervised baseline in our experiments.

e) Limitations: All existing model comparisons for semi-supervised TSC, despite the work of [35], are limited to univariate time series datasets with a maximal size of 1000 training samples. In contrast to computer vision research on SSL [7], these model comparisons are conducted for one fixed relative amount of labeled samples in the vast majority of experiments, making it hard to deduce general information for different data situations. They also do not align with the guidelines established by [12] for SSL on image data and do not include repeated model runs to account for randomness in the selection of labeled samples. Another issue is the lack of publicly accessible implementations of the classical approaches to semi-supervised TSC, making it impossible to validate against these approaches. This in turn leads to the problem that model comparisons with existing methods solely rely on values reported in former work for the same datasets with partially opaque dataset splits and unlabelling procedures.

Our **main contributions** can be summarized as follows: 1) We propose four new deep SSL algorithms for TSC and describe tuning parameters and meaningful data augmentation strategies. 2) We investigate the applicability of deep SSL in the domain of TSC and provide insights in which settings the proposed methods work well and how they compare to existing approaches. 3) Through these experiments we are able to identify two out of our four proposed methods that notably improve over existing approaches.

II. From Images to Time Series

A. Problem Formulation

We define an equidistant time series as $x^{(i)} = \{\{x^{(i)}_{1,1}, ..., x^{(i)}_{1,t}\}, ..., \{x^{(i)}_{c,1}, ..., x^{(i)}_{c,t}\}\}$, where t describes the length and c the amount of covariates such that $x^{(i)} \in \mathcal{X} \subseteq \mathbb{R}^{c \times t}$. For $c = 1$ the time series is called univariate and for $c > 1$ multivariate. Next to the input space $\mathcal{X}$, we use $y^{(i)} \in \mathcal{Y}$ to denote a categorical variable in the target space $\mathcal{Y}$. The goal of SSL is to train a prediction model $f : \mathcal{X} \mapsto \mathcal{Y}$ on a dataset $\mathcal{D} = (\mathcal{D}^l, \mathcal{D}^u)$ which consists of a labeled dataset $\mathcal{D}^l = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n_l}$ and an unlabeled dataset $\mathcal{D}^u = \{x^{(i)}\}_{i=n_l+1}^{n}$ where $n = n_l + n_u$. We consider the case where $n_l \ll n_u$, as usual in SSL. Further, we define one batch of data as $\mathcal{B} \subset \mathcal{D}$, where $\mathcal{B}^l \subseteq \mathcal{D}^l$ contains the labeled samples and $\mathcal{B}^u \subseteq \mathcal{D}^u$ the unlabeled samples in that batch such that $\mathcal{B} = (\mathcal{B}^l, \mathcal{B}^u)$.

B. Backbone Architecture

A basic building block in deep learning for images is a 3-dimensional tensor, whereas time series can be represented as 2-dimensional tensors with channels corresponding to the number of covariates. The extension of building blocks of powerful image classification architectures to TSC is thus straightforward, yet the right choice of a backbone architecture is crucial. We propose the use of the Fully Convolutional Network (FCN) [17] as a backbone architecture as it was shown to outperform a variety of models on 44 different TSC problems and is used in related work on semi-supervised TSC [36]. In all regularization-based semi-supervised methods discussed in Section II-D, except for the Ladder Net [11], the network architecture can be decoupled from the model training strategy. This allows us to replace the backbone architecture of many of the established SSL methods from image classification with the FCN. In case of the Ladder Net, we design the decoder as a mirrored version of the FCN encoder (see Section II-D).

C. Data Augmentation

One crucial component of regularization-based semi-supervised methods is the injection of random noise into the model. Data augmentation strategies $g(x^{(i)}), g : \mathcal{X} \mapsto \mathcal{X}$ should be designed such that they perturbate the input $x^{(i)}$ of a sample while preserving the meaning of its label $y^{(i)}$. This can be achieved by utilizing inherent invariances in the data, e.g., rotations of images usually preserve the meaning of an image. For images, invariances can be easily understood visually. In the time series domain, such invariances are not

straightforward to understand, rendering the design of reasonable data augmentation strategies in this domain challenging. A set of data augmentation strategies for multivariate time series classification was introduced by [37] and evaluated on one HAR task. They show that the majority of strategies are beneficial, but some can deteriorate the model performance. To overcome the additional burden of choosing the right strategy, we propose the use of the RandAugment strategy [38] which removes the need for a separate search phase. For each training batch, N augmentation strategies are randomly chosen out of a set of K possible policies. Next to N, a $magnitude$ hyperparameter is introduced which controls the augmentation intensity of the selected policies. We use the following set of augmentation policies [37]: warping in the time dimension, warping the magnitude, addition of Gaussian Noise and random rescaling. We use RandAugment in this context following the rationale that even if a augmentation strategy is (not) label preserving, training with RandAugment with $N = 1$ will still produce correct model updates in at least $\frac{K-1}{K}$ of the forward passes. Early experiments in a fully supervised setting showed that the application of this data augmentation strategy improves model performance across all datasets used in our experiments.

D. Methods

The Mean Teacher [10] is the successor of a series of consistency-regularization-based models such as Temporal Ensembling or the Π-Model [28] for SSL and was empirically shown to outperform its predecessors [12]. Thereby, a teacher model, that is an average of the consecutive student models, is used to enforce consistency in model predictions over the course of model training.

Virtual Adversarial Training (VAT) [9] also focuses on consistency regularization. Similar to adversarial examples [39], a small data perturbation is learned such that its addition to the initial data point is expected to yield the maximum change in the model's prediction. These perturbed model predictions are used as auxiliary labels for the unlabeled samples within a regularization term to enable model training on the whole data set. This approach is particularly interesting for the time series domain where visual inspection of the appropriateness of data augmentation policies is difficult, as it does not rely on data augmentation techniques.

In MixMatch, various semi-supervised techniques such as data augmentation for consistency regularization, Mixup training [40] and pseudo-labeling are combined within one holistic approach [8]. It was empirically shown to perform well on image data, motivating our use of it in this work [8].

The Ladder Net by [11] is a reconstruction-based SSL model and is inspired by denoising autoencoders [41]. In its core, it extends a supervised encoder model with a corresponding decoder network which allows for the calculation of an unsupervised reconstruction loss over the unlabeled samples enabling training on the whole dataset. The Ladder Net was previously extended to TSC problems [35] and is thus also part of this study.

III. Experimental Design

A. Baseline Models

Next to shapelet- and distance-based methods [4], fitting standard ML methods on hand-crafted statistical features has been a widely used approach for TSC before the introduction of specific deep learning architectures for TSC [17] [18]. We include a Random Forest and a Logistic Regression trained on features, extracted via the tsfresh framework [16] from the time series, as baselines.
In addition, we train the FCN architecture [17] on the labeled samples $\mathcal{D}^l$ based on the cross entropy loss as a supervised deep learning baseline model for our experiments. To ensure a fair and reliable model comparison, we explicitly use the same architecture of this supervised baseline model as the backbone for all SSL approaches. We also use the performance of a supervised FCN trained on the fully labeled datasets as an estimated upper bound for the model performance.
Furthermore, we evaluate the performance of the self-supervised approach that was recently introduced for TSC by [36]. Thereby, an auxiliary forecasting task from the time series data $\mathcal{D}$ is created and combined with the initial classification task as a surrogate supervision signal allowing the use of unlabeled data in model training. The model is then jointly trained on both tasks simultaneously. Next to its re-implementation, we further extend their approach for multivariate TSC by increasing the amount of neurons in the surrogate model head accordingly. The direct comparison with this self-supervised approach is of special interest as it was shown to outperform classical semi-supervised approaches in a set of experiments on smaller TSC datasets [36].

B. Data Sets

We evaluate the performance of the above described semi-supervised models on 6 publicly available datasets. In contrast to previous work [33], [34], [36], we explicitly focus on large datasets with at least 1000 observations. Their main characteristics are described in Table I.

TABLE I: Characteristics of the used data sets where *c* refers to the amount of covariates, *Size* to the size of the whole training data set and *Length* to the length of the time series.

Name	Classes	Size	Length	c	Balanced
Crop	24	7,200	46	1	✓
ElectricDevices	7	8,926	96	1	✗
FordB	2	3,636	500	1	✓
Pamap2	13	11,313	100	6	✗
WISDM	6	10,727	80	3	✗
Balanced SITS	6	35,064	46	1	✓

With Crop, ElectricDevices and FordB we include three of the largest datasets from the UCR Time Series Classification Repository [3]. In addition, we use the two multivariate HAR datasets Pamap2 [42] and WISDM [43]. We also evaluate the models on a class-balanced version of the Satellite Image Time Series (SITS) dataset [44].

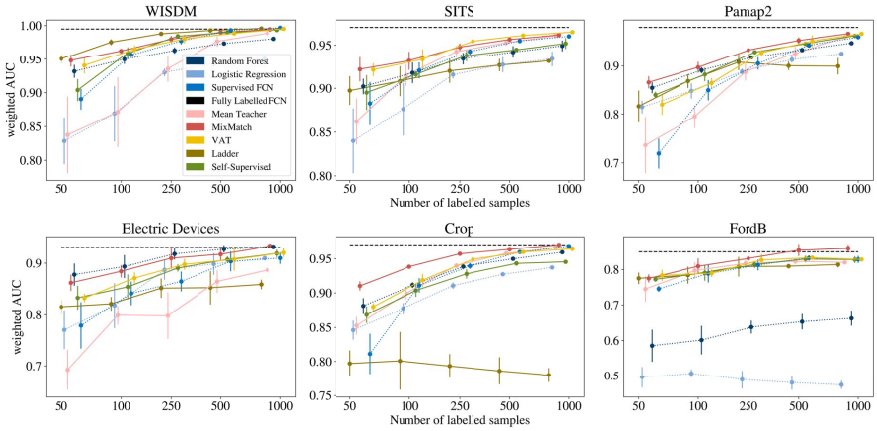


Fig. 1: Performance of all models on the 6 different datasets over various n_l as presented in Table II in the appendix. The horizontal line marks the performance of the fully labeled baseline, i.e. the supervised FCN model trained on the fully labeled dataset. Dots represent the mean wAUC and the vertical lines the standard deviation over 5 repeated *unlabeling* steps. The performance of the baseline models are depicted as dotted, those of the semi-supervised models as solid lines. Semi-supervised models clearly outperform the baseline models in settings with few labeled samples $n_l \in \{50, 100\}$ on all but the Electric Devices dataset.

C. Evaluation, Tuning and Implementation

Due to special factors, such as the selection of the labeled data points, an unbiased and fair model comparison is particularly crucial to get a realistic perspective on the performance of the semi-supervised models [7]. We adhere to the guidelines for realistic evaluation of semi-supervised models by [12] to guarantee reliable and fair experimental results. For performance evaluation of SSL models, the standard procedure is to split a fully labeled dataset $\mathcal{D}$ into labeled and unlabeled datasets $\mathcal{D}^l$ and $\mathcal{D}^u$ via artificial *unlabeling* of n_u randomly drawn samples [7]. This way, semi-supervised data settings for different amounts of labeled samples l are simulated. We unlabel in a stratified manner to retain the datasets' label distributions. For the following experiments, we split the evaluation of one model f on one data set D in two distinct phases.

a) Tuning Phase: In the tuning phase, we tuned the model f with one fixed amount of labeled samples to yield an optimal set of hyperparameters θ^*. Thereby, f was trained on a training dataset $\mathcal{D}_{train} = (\mathcal{D}^l_{train}, \mathcal{D}^u_{train})$, where we fixed $|\mathcal{D}^l_{train}| = 500$, and validated on a labeled holdout validation set $\mathcal{D}_{val}$. The choice of the size of $\mathcal{D}_{val}$ is subject to recent discussions [11], [12], [45]. Large $\mathcal{D}_{val}$ are expected to yield stable results for model tuning, which is important for many hyperparameter-sensitive semi-supervised models, but stands in contrast to the promised practicality of these models in settings with few labeled data. First insights on this trade-off are are given by [12] and [45], which empirically show in smaller experiments $|\mathcal{D}_{val}| = 1000$ to be a validation set size where variance in the performance estimates is still low enough to allow for reasonable model selection. Following this, we set the size of the labeled validation set to $|\mathcal{D}_{val}| = 1000$ which is rather small compared to recent literature where $|\mathcal{D}_{val}| \geq 4000$ [9], [28], [10]. A separate labeled test set $\mathcal{D}_{test}$ with $|\mathcal{D}_{test}| = 2000$ is kept aside for the evaluation phase. Hyperband [46] with random sampling as implemented in the Optuna framework [47] was used for tuning, with a fixed budget of 100 GPU hours for each deep learning model and dataset. We measure model performance in terms of weighted Area under the Curve (wAUC) to account for model calibration and class imbalance.

b) Evaluation Phase: In the evaluation phase, we train $f(\theta^*)$ on $\mathcal{D}_{train}$ with varying amounts of $n_l \in \{50, 100, 250, 500, 1000\}$ for a maximum of 25000 model update steps, assuming θ^* is also a suitable hyperparameter set for amounts of labels $n_l \neq 500$ on which the model was not specifically tuned. This evaluation scheme is in line with previous work on SSL for image data [8], [12]. Model performance is tracked on $\mathcal{D}_{val}$ and the model checkpoint with the best validation performance is used for inference on the holdout $\mathcal{D}_{test}$. The selection of especially (un-)informative labeled samples can have a major effect on the model performance, especially for small n_l. To account for potentially (un-)lucky selection of $\mathcal{D}^l_{train}$ in the *unlabelling* split of $\mathcal{D}_{train} = (\mathcal{D}^l_{train}, \mathcal{D}^u_{train})$, we repeat this *unlabelling* step 5 times. In case of the ML baseline models, we use a Random Search with a budget of 100 model evaluations for the tuning phase and evaluate them on the same set of values for n_l

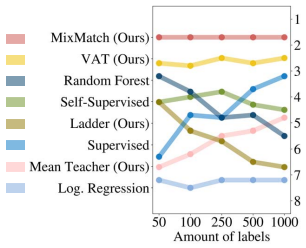


Fig. 2: Average ranks of all models based on the wAUC over the 6 datasets for varying n_l. Models are sorted by their strongest performance on $n_l = 50$ and plotted with decreasing rank as indicated on the right vertical axis.

in the evaluation phase. See Table IV in the appendix for the specific ranges. All deep learning models were implemented in a unified codebase[1] and trained using the Adam optimizer [48] with all parameters set to default values except the learning rate and weight decay. We implemented all deep learning models from scratch in one unified framework and validated our implementations based on performance metrics reported on image classification tasks..

IV. Experimental Results

Experimental findings are visualized in Figure 1 and Table II in the appendix. The ranking of the various models for different n_l, averaged over the datasets, is shown in Figure 2 and Table III in the appendix.

a) Semi-supervised models outperform supervised baselines: Overall, our results show that semi-supervised models outperform baseline models especially for small amounts of labeled data. This relative performance gain of semi-supervised over supervised models is decreasing with an increase in n_l and we find that all models benefit from more labeled samples in most cases. This is in line with literature on SSL [7].

b) Deep SSL translates well to TSC: Following our experimental results in Figure 1, we deduct that *transferring well-established semi-supervised models from the image to the time series domain is indeed possible.* We find that the deep semi-supervised models, especially the transferred MixMatch and VAT, show impressive performance gains over the deep supervised baseline model over all datasets up to $n_l = 500$, even reaching the performance of the fully labeled baseline in few cases. For instance, the Mixmatch model exceeds the deep supervised baseline by 0.16 wAUC on the Pamap2 and by 0.10 wAUC on the Crop dataset for $n_l = 50$. These findings again encourage our proposed transfer.

c) Strong baselines are crucial: We find the use of strong baselines crucial for a realistic perspective on semi-supervised learning performance. For instance, the Mean Teacher shows weak performance on the majority of datasets, often performing even worse than the supervised baseline. This is in line with results of [36]. The strong performance of the Random Forest for small n_l on the other hand also stresses the need for realistically strong supervised baselines.

d) Proposed methods outperform existing semi-supervised approaches: While our results on the Ladder Net outperforming other supervised methods align with those of [35], we also observe that the Ladder Net is notably worse compared to alternative SSL algorithms we propose. This varying performance might be grounded in the large amount of hyperparameters of the Ladder Net and its sensitivity to different settings of those.

e) Proposed methods outperform self-supervised modeling: Similar to [36], we find their self-supervised approach to perform better or at least equally well compared to the deep supervised baseline model. Additionally, we are able to show that our extension towards multivariate time series also works well on the two multivariate datasets, WISDM and Pamap2. The proposed approaches MixMatch and VAT furthermore consistently outperform this self-supervised approach across different amounts of labels on all 6 datasets.

f) Ranking of model performance similar to image domain: In terms of model performance ranking, literature suggests that MixMatch performs better than VAT which again outperforms the Mean Teacher and the Ladder Net [8], [12]. When ranking the algorithms across the datasets in Figure 2, we confirm this ranking in the TSC setting.

Our results show that the promised label efficiency of modern, deep semi-supervised model approaches translates well to TSC problems. Furthermore, these findings suggest the use of strong semi-supervised models from the image domain as these transferred models show stronger performance than the currently existing semi- and self-supervised approaches tailored towards TSC. We believe that this work, also thanks to a strong focus on a fair and reliable model comparison, can serve as the basis for future research advances in semi-supervised learning for time series classification.

[1] https://github.com/Goschjann/ssltsc

Acknowledgements

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy through the Center for Analytics – Data – Applications (ADA-Center) within the framework of BAYERN DIGITAL II (20-3410-2-9-8) as well as the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A.

References

[1] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, *et al.*, "Scalable and accurate deep learning with electronic health records," *NPJ Digital Medicine*, vol. 1, no. 1, p. 18, 2018.

[2] G. A. Susto, A. Cenedese, and M. Terzi, "Time-series classification methods: Review and applications to power systems data," in *Big data application in power systems*, pp. 179–220, Elsevier, 2018.

[3] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The ucr time series archive," 2019.

[4] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017.

[5] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[6] J. Goschenhofer, F. M. Pfister, K. A. Yuksel, B. Bischl, U. Fietzek, and J. Thomas, "Wearable-based parkinson's disease severity monitoring using deep learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 400–415, Springer, 2019.

[7] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[8] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems 32*, pp. 5049–5059, 2019.

[9] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2019.

[10] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems 30*, pp. 1195–1204, 2017.

[11] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Advances in Neural Information Processing Systems 28*, pp. 3546–3554, 2015.

[12] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *Advances in Neural Information Processing Systems 31*, pp. 3235–3246, 2018.

[13] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, "Learning time-series shapelets," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 392–401, 2014.

[14] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with cote: the collective of transformation-based ensembles," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2522–2535, 2015.

[15] R. J. Kate, "Using dynamic time warping distances as features for improved time series classification," *Data Mining and Knowledge Discovery*, vol. 30, no. 2, pp. 283–312, 2016.

[16] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package)," *Neurocomputing*, pp. 72–77, 2018.

[17] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585, 2017.

[18] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, pp. 1936–1962, 2020.

[19] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, 2018.

[20] V. Vapnik, *Statistical learning theory*. Wiley New York, 1998.

[21] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu, "Semiboost: Boosting for semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 11, pp. 2000–2014, 2008.

[22] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," tech. rep., 2002.

[23] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, no. Nov, pp. 2399–2434, 2006.

[24] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, 1998.

[25] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-supervised learning*. MIT Press, 2006.

[26] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5070–5079, 2019.

[27] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang, "Smooth neighbors on teacher graphs for semi-supervised learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8896–8905, 2018.

[28] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[29] L. Wei and E. Keogh, "Semi-supervised time series classification," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 748–753, 2006.

[30] Y. Chen, B. Hu, E. Keogh, and G. E. Batista, "Dtw-d: time series semi-supervised learning from a single example," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 383–391, 2013.

[31] K. Marussy and K. Buza, "Success: a new approach for semi-supervised classification of time-series," in *International Conference on Artificial Intelligence and Soft Computing*, pp. 437–447, 2013.

[32] M. N. Nguyen, X.-L. Li, and S.-K. Ng, "Positive unlabeled learning for time series classification," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, , IJCAI-11*, 2011.

[33] Z. Xu and K. Funaya, "Time series analysis with graph-based semi-supervised learning," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–6, 2015.

[34] H. Wang, Q. Zhang, J. Wu, S. Pan, and Y. Chen, "Time series feature learning with labeled and unlabeled data," *Pattern Recognition*, vol. 89, pp. 55–66, 2019.

[35] M. Zeng, T. Yu, X. Wang, L. T. Nguyen, O. J. Mengshoel, and I. Lane, "Semi-supervised convolutional neural networks for human activity recognition," in *2017 IEEE International Conference on Big Data (Big Data)*, pp. 522–529, 2017.

[36] S. Jawed, J. Grabocka, and L. Schmidt-Thieme, "Self-supervised learning for semi-supervised time series classification," in *Proceedings of the 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2020*, pp. 499–511, 2020.

[37] T. T. Um, F. M. J. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulic, "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks," in *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pp. 216–220, 2017.

[38] 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*, pp. 702–703, 2020.

[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, pp. 2672–2680, 2014.

[40] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *6th International Conference for Learning Representations, ICLR 2018*, 2018.

[41] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.," *Journal of Machine Learning Research*, vol. 11, no. 12, pp. 3371–3408, 2010.

[42] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers*, pp. 108–109, 2012.

[43] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[44] F. Petitjean, J. Inglada, and P. Gancarski, "Satellite image time series analysis under time warping," *IEEE transactions on geoscience and remote sensing*, vol. 50, no. 8, pp. 3081–3095, 2012.

[45] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1476–1485, 2019.

[46] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, 2017.

[47] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631, 2019.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations, ICLR 2015*, 2015.

APPENDIX

A. Model Performance

TABLE II: Results for models over datasets with varying numbers of labels n_l. Performance is measured as weighted AUC. The best results for each n_l-dataset-combination are emphasized in bold with standard deviations over 5 replications in brackets.

Number of labels	50						100					
Dataset	Crop	Electric Devices	FordB	Pamap2	SITS	WISDM	Crop	Electric Devices	FordB	Pamap2	SITS	WISDM
Model												
Ladder	0.797 (0.019)	0.815 (0.013)	0.775 (0.016)	0.816 (0.033)	0.897 (0.017)	0.95 (0.007)	0.801 (0.042)	0.821 (0.013)	0.784 (0.024)	0.869 (0.014)	0.908 (0.017)	0.976 (0.004)
Logistic Regression	0.846 (0.014)	0.771 (0.037)	0.496 (0.029)	0.815 (0.013)	0.84 (0.037)	0.829 (0.034)	0.877 (0.008)	0.817 (0.043)	0.507 (0.012)	0.848 (0.016)	0.876 (0.03)	0.869 (0.041)
Mean Teacher	0.853 (0.013)	0.692 (0.039)	0.745 (0.036)	0.737 (0.058)	0.862 (0.026)	0.838 (0.057)	0.899 (0.009)	0.799 (0.018)	0.797 (0.023)	0.795 (0.023)	0.915 (0.012)	0.871 (0.052)
MixMatch	**0.910 (0.007)**	0.862 (0.016)	0.775 (0.012)	**0.866 (0.013)**	**0.922 (0.014)**	**0.948 (0.009)**	**0.938 (0.003)**	0.884 (0.012)	**0.809 (0.022)**	**0.897 (0.011)**	0.932 (0.011)	0.963 (0.003)
Random Forest	0.881 (0.011)	**0.878 (0.022)**	0.585 (0.045)	0.855 (0.011)	0.902 (0.009)	0.932 (0.01)	0.911 (0.004)	**0.893 (0.022)**	0.601 (0.04)	0.891 (0.006)	0.918 (0.012)	0.95 (0.007)
Self-Supervised	0.868 (0.012)	0.832 (0.023)	0.772 (0.007)	0.84 (0.007)	0.895 (0.02)	0.904 (0.016)	0.904 (0.009)	0.854 (0.024)	0.79 (0.022)	0.882 (0.014)	0.916 (0.003)	0.959 (0.009)
Supervised	0.811 (0.03)	0.779 (0.045)	0.745 (0.009)	0.719 (0.031)	0.883 (0.025)	0.89 (0.016)	0.911 (0.01)	0.841 (0.023)	0.788 (0.025)	0.85 (0.021)	0.921 (0.009)	0.957 (0.004)
VAT	0.88 (0.005)	0.832 (0.009)	**0.783 (0.016)**	0.82 (0.022)	0.921 (0.007)	0.941 (0.012)	0.919 (0.009)	0.871 (0.012)	0.789 (0.011)	0.865 (0.01)	**0.933 (0.012)**	**0.965 (0.006)**

Number of labels	250						500					
Dataset	Crop	Electric Devices	FordB	Pamap2	SITS	WISDM	Crop	Electric Devices	FordB	Pamap2	SITS	WISDM
Model												
Ladder	0.793 (0.017)	0.851 (0.019)	0.808 (0.028)	0.908 (0.017)	0.92 (0.014)	0.988 (0.001)	0.786 (0.02)	0.852 (0.033)	0.809 (0.007)	0.901 (0.009)	0.927 (0.007)	0.993 (0.001)
Logistic Regression	0.911 (0.005)	0.887 (0.019)	0.489 (0.025)	0.888 (0.006)	0.916 (0.005)	0.93 (0.006)	0.927 (0.002)	0.898 (0.021)	0.48 (0.019)	0.913 (0.006)	0.927 (0.009)	0.944 (0.004)
Mean Teacher	0.94 (0.004)	0.798 (0.045)	0.817 (0.011)	0.888 (0.017)	0.943 (0.007)	0.936 (0.019)	0.958 (0.004)	0.864 (0.022)	0.823 (0.007)	0.927 (0.014)	0.953 (0.003)	0.977 (0.008)
MixMatch	**0.957 (0.003)**	0.910 (0.019)	**0.831 (0.023)**	**0.934 (0.007)**	**0.948 (0.004)**	0.980 (0.005)	**0.964 (0.003)**	0.917 (0.013)	**0.854 (0.016)**	**0.953 (0.006)**	0.956 (0.002)	0.990 (0.003)
Random Forest	0.939 (0.004)	**0.918 (0.011)**	0.638 (0.018)	0.918 (0.005)	0.934 (0.007)	0.963 (0.005)	0.950 (0.003)	**0.928 (0.007)**	0.653 (0.022)	0.934 (0.004)	0.942 (0.007)	0.974 (0.001)
Self-Supervised	0.928 (0.007)	0.891 (0.007)	0.814 (0.01)	0.930 (0.001)	0.938 (0.007)	**0.984 (0.001)**	0.943 (0.003)	0.907 (0.006)	0.829 (0.01)	0.947 (0.004)	0.945 (0.002)	0.990 (0.002)
Supervised	0.939 (0.004)	0.864 (0.019)	0.812 (0.015)	0.905 (0.013)	0.943 (0.004)	0.977 (0.005)	0.960 (0.002)	0.904 (0.019)	0.832 (0.008)	0.943 (0.004)	0.955 (0.001)	**0.993 (0.004)**
VAT	0.949 (0.002)	0.898 (0.011)	0.827 (0.017)	0.929 (0.006)	0.954 (0.004)	0.98 (0.006)	0.960 (0.003)	0.907 (0.022)	0.833 (0.005)	0.952 (0.012)	**0.961 (0.001)**	0.989 (0.002)

Number of labels	1000					
Dataset	Crop	Electric Devices	FordB	Pamap2	SITS	WISDM
Model						
Ladder	0.78 (0.011)	0.858 (0.009)	0.814 (0.008)	0.899 (0.017)	0.932 (0.001)	0.996 (0.001)
Logistic Regression	0.937 (0.001)	0.91 (0.003)	0.474 (0.011)	0.926 (0.004)	0.934 (0.009)	0.950 (0.003)
Mean Teacher	0.966 (0.001)	0.887 (0.013)	0.82 (0.006)	0.96 (0.002)	0.96 (0.002)	0.989 (0.005)
MixMatch	**0.970 (0.001)**	**0.933 (0.003)**	**0.859 (0.010)**	**0.967 (0.004)**	0.961 (0.002)	0.994 (0.001)
Random Forest	0.959 (0.001)	0.932 (0.004)	0.663 (0.021)	0.948 (0.003)	0.949 (0.006)	0.981 (0.001)
Self-Supervised	0.945 (0.001)	0.919 (0.007)	0.828 (0.010)	0.963 (0.004)	0.952 (0.006)	0.994 (0.001)
Supervised	0.968 (0.001)	0.910 (0.011)	0.828 (0.010)	0.960 (0.003)	0.960 (0.002)	**0.997 (0.001)**
VAT	0.964 (0.001)	0.920 (0.008)	0.828 (0.006)	0.967 (0.004)	**0.965 (0.002)**	0.995 (0.001)

B. Model Ranking

TABLE III: The average rank of all models based on the wAUC over the 6 different datasets for various amounts of labels n_l. Lower rank indicates stronger model performance. Ranks are shown with decimals due to averaging over datasets.

	Number of labels				
	50	100	250	500	1000
MixMatch	**1.7**	**1.7**	**1.7**	**1.7**	**1.7**
VAT	2.7	2.8	2.5	2.7	2.5
MeanTeacher	6.7	6.2	5.5	5.3	4.8
Self-supervised	4.2	4.0	3.8	4.3	4.5
Ladder	4.2	5.3	5.7	6.5	6.7
Supervised	6.3	4.7	4.8	3.7	3.2
Random Forest	3.2	3.8	4.8	4.7	5.5
Logistic Regression	7.2	7.5	7.2	7.2	7.2

C. Hyperparameters

TABLE IV: Hyperparameter ranges used for tuning of the different models. Deep Learning models were tuned via Hyperband as described in Section 3 while the Random Forest and the Logistic Regression were tuned via Random Search with a bugdet of 100 model evaluations each.

Parameter	Range	Scale
Shared		
Weight decay	$[1e^{-6}; 1e^{-2}]$	log
Learning rate	$[1e^{-5}; 1e^{-2}]$	log
Rampup length	[5000; 25000]	linear
Magnitude (RandAug)	[1; 10]	linear
N (RandAug)	[1; 6]	linear
VAT		
ϵ	[0.1; 10.0]	linear
α	[0.1; 5.0]	linear
MixMatch		
α	[0.5; 1.0]	linear
λ_u	[0.0; 150.0]	linear

Parameter	Range	Scale
Self-Supervised Learning		
λ	[0.1; 10]	log
horizon h	[0.1, 0.2, 0.3]	discrete
stride s	[0.05, 0.1, 0.2, 0.3]	discrete
Ladder Net		
Noise ratio	[0.1, 0.3, 0.45, 0.6]	discrete
Loss weights	[0.1; 10.0]	log
Mean Teacher		
α_{ema}	[0.9; 1.0]	log
w_{max}	[0; 10]	linear
Random Forest		
Number of trees	[100; 1000]	linear
Max. tree depth	[3; 25]	linear
Logistic Regression		
Regularization term	[None, L_1, L_2]	discrete

Deep Semi-supervised Learning for Time-Series Classification

Jann Goschenhofer

Abstract While deep semi-supervised learning has gained much attention in computer vision, limited research exists on its applicability in the time-series domain. In this work, we investigate the transferability of state-of-the-art deep semi-supervised models from image to time-series classification. We discuss the necessary model adaptations, in particular, an appropriate model backbone architecture and the use of tailored data augmentation strategies. Based on these adaptations, we explore the potential of deep semi-supervised learning in the context of time-series classification by evaluating our methods on large public time-series classification problems with varying amounts of labeled samples. We perform extensive comparisons under a decidedly realistic and appropriate evaluation scheme with a unified reimplementation of all algorithms considered, which is yet lacking in the field. Further, we shed light on the effect of different data augmentation strategies and model architecture backbones in this context within a series of experiments. We find that these transferred semi-supervised models show substantial performance gains over strong supervised, semi-supervised and self-supervised alternatives, especially for scenarios with very few labeled samples.

1 Introduction

Time-series classification (TSC) spans many real-world applications in domains from healthcare [34] over cybersecurity [38] to manufacturing (Dau et al. [12]). Several algorithms for TSC have been proposed over the years (Bagnall et al. [2]; Fawaz et al. [13]).

In many real-world scenarios, time-series data can be collected easily, but acquiring labels for this data is costly. For instance, in disease monitoring, sensor data are collected with low effort but the labeling of this data requires time-consuming work by medical experts (Goschenhofer et al. [16]). Semi-supervised learning (SSL)

J. Goschenhofer (✉)
University of Colombo School of Computing, Colombo, Sri Lanka
e-mail: jann.goschenhofer@stat.uni-muenchen.de

M. A. Wani and V. Palade (eds.), *Deep Learning Applications, Volume 4*,
Advances in Intelligent Systems and Computing 1434,
https://doi.org/10.1007/978-981-19-6153-3_15

addresses this by leveraging large amounts of unlabeled data in combination with a small amount of labeled data when training machine learning (ML) models.

Especially in computer vision, the advances in deep neural networks and the promised label efficiency of SSL have led to the introduction of several innovative approaches for image data (Van Engelen and Hoos [42]). While there is much work on classical semi-supervised models for TSC, research on the use of neural network-based SSL algorithms for TSC is still limited.

This motivates our main research question that we approach holistically in this work: *Can we transfer well-established deep semi-supervised models from the image to the time-series domain?* More specifically, we answer this question for the most prominent state-of-the-art SSL approaches, by proposing adaptions for FixMatch (Sohn et al. [37]), MixMatch (Berthelot et al. [6]), Virtual Adversarial Training (Miyato et al. [29]), the Mean Teacher (Tarvainen and Valpola [40]) and the Ladder Net (Rasmus et al. [35]). These include the modification of a suitable backbone architecture as well as the adaptions of an appropriate data augmentation strategy to account for the domain transfer of these models. For demonstration of the efficacy of our proposed frameworks, we adhere to best practices for realistic evaluation of semi-supervised models and provide a fair and reliable model comparison with a high degree of practicality (Oliver et al. [31]).

1.1 Related Work

1.1.1 Time-Series Classification

Over the past years, a variety of methods have been developed for TSC. A detailed overview on classical ML methods that were specifically developed for TSC (Grabocka et al. [17]; Bagnall et al. [3]; Kate [21] is provided in [2]. An alternative approach towards TSC consists in the extraction of statistical features from the raw time series as the basis for training any strong classifier for tabular data (Christ et al. [10]). Also in deep learning, specific methods for time-series classification have been developed (Wang [4, 14, 46]). A comprehensive overview on these recent developments can be found in [13].

1.1.2 Semi-supervised Learning

There exists a plethora of different concepts that extract additional information from unlabeled data via semi-supervision. These range from the extension of supervised ML methods such as the semi-supervised Support Vector Machine (Vapnik [43]) or semi-supervised Boosting (Mallapragada et al. [27]) to inherently semi-supervised methods such as Label Propagation (Zhu and Ghahramani [53]), Manifold Regularization (Belkin et al. [5]) or Co-Training (Blum and Mitchell [7]). [8] provide a detailed overview on these semi-supervised approaches. There is also growing

research on deep semi-supervised learning, mainly driven by the computer vision community. A recent overview and taxonomy on these developments are provided by (Van Engelen and Hoos [42]). Amongst these are graph-based methods such as Deep Label Propagation (Iscen et al. [18]), SNTG (Luo et al. [26]) or the extension of pseudo-labelling for deep learning (Van Engelen and Hoos [42]). Further, there is growing research on regularization-based approaches following the rationale of adding an additional unsupervised regularization loss term to the initial supervised loss. The Mean Teacher (Tarvainen and Valpola [40]) and its predecessors, Temporal Ensembling and the Π-Model (Laine and Aila [24]), employ a consistency loss over the unlabeled samples to reward similar predictions for differently augmented versions of the same unlabeled sample. To overcome one drawback of those methods, the need for domain-dependent data augmentation strategies, Virtual Adversarial Training (VAT) (Miyato et al. [29]) adds small perturbations to the input data to create an auxiliary unsupervised training target. MixMatch (Berthelot et al. [6]) in turn combines different regularization strategies in one common framework. These regularization-based approaches yield state-of-the-art performance on image classification benchmarks.

1.1.3 Semi-supervised Time Series Classification

Different classical semi-supervised models have been developed for TSC. In their foundational work, [47] propose an approach that combines pseudo-labeling with a nearest-neighbor model for imbalanced, binary TSC tasks. This cluster-then-label (Van Engelen and Hoos [42]) rationale for labeled and unlabeled time series via custom distance metrics is also employed in approaches such as DTW-D (Chen et al. [9]), SUCCESS (Chen et al. [28]) or LCLC (Nguyen et al. [30]). Graph-based label propagation (Zhu and Ghahramani [53]) is combined with time-series-specific distance metrics by (Xu and Funaya [49]) and (Wang et al. [45]) introduced the shapelet-based SSSL. Furthermore, there has been a surge of research on neural-net-based approaches for TSC. A customized version of the LadderNet (Rasmus et al. [35]) based on the FCN architecture (Rasmus et al. [46]) was applied by [50] on three multivariate human activity recognition (HAR) datasets. They report relative gains of the semi-supervised model over the supervised baselines for small amounts of labeled samples. In this context [50] are the first to evaluate SSL methods on large, multivariate TSC datasets. A self-supervised approach, where the model is jointly trained on an auxiliary forecasting task over the whole dataset next to the initial supervised classification task on the labeled data only, was introduced by [20]. They build upon the benchmark of [45] on a subset of smaller, univariate TSC datasets from the UCR repository (Dau et al. [12]) and report state-of-the-art performance compared to the majority of above methods as well as a customized variant of the Π−Model (Laine and Aila [24]) that works on time-series problems. In alignment with [50], they report particularly strong model performance for the deep supervised baseline FCN (Wang et al. [46]) trained on few labeled samples only reporting it to outperform all above-mentioned classical semi-supervised models.

This deep learning baseline outperforms all of the classical semi-supervised models and almost always beats the $\Pi-$Model. We include this approach as a self-supervised baseline in our experiments.

1.1.4 Limitations

All existing model comparisons for semi-supervised TSC, despite the work of [50], are limited to univariate time-series datasets with a maximal size of 1000 training samples. In contrast to computer vision research on SSL (Van Engelen and Hoos [42]), these model comparisons are conducted for one fixed relative amount of labeled samples in the vast majority of experiments, making it hard to deduce general information for different data situations. They also do not align with the guidelines established by [31] for SSL on image data and do not include repeated model runs to account for randomness in the selection of labeled samples. Another issue is the lack of publicly accessible implementations of the classical approaches to semi-supervised TSC, making it impossible to validate against these approaches. This in turn leads to the problem that model comparisons with existing methods solely rely on values reported in former work for the same datasets with partially opaque dataset splits and unlabeling procedures.

1.1.5 Contributions

Our **main contributions** can be summarized as follows: (1) We propose five new deep SSL algorithms for TSC and describe tuning parameters and meaningful data augmentation strategies. Further, we (2) investigate the applicability of deep SSL in the domain of TSC and provide insights into which settings the proposed methods work well and how they compare to existing approaches. Through these experiments we are (3) able to identify three out of our five proposed methods that notably improve over existing approaches and in a series of additional experiments, we (4) provide insights on the role of different data augmentation strategies and architecture backbones in this context.

2 From Images to Time Series

2.1 Problem Formulation

We define an equidistant time series as $x^{(i)} = \{\{x_{1,1}^{(i)}, ..., x_{1,t}^{(i)}\}, ..., \{x_{c,1}^{(i)}, ..., x_{c,t}^{(i)}\}\}$, where t describes the length and c the amount of covariates such that $x^{(i)} \in \mathcal{X} \subseteq \mathbb{R}^{c \times t}$. For $c = 1$ the time series is called univariate and for $c > 1$ multivariate. Next to the input space $\mathcal{X}$, we use $y^{(i)} \in \mathcal{Y}$ to denote a categorical variable in the target space $\mathcal{Y}$.

The goal of SSL is to train a prediction model $f : \mathcal{X} \mapsto \mathcal{Y}$ on a dataset $\mathcal{D} = (\mathcal{D}^l, \mathcal{D}^u)$ which consists of a labeled dataset $\mathcal{D}^l = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n_l}$ and an unlabeled dataset $\mathcal{D}^u = \{x^{(i)}\}_{i=n_l+1}^{n}$ where $n = n_l + n_u$. We consider the case where $n_l \ll n_u$, as usual in SSL. Further, we define one batch of data as $\mathcal{B} \subset \mathcal{D}$, where $\mathcal{B}^l \subseteq \mathcal{D}^l$ contains the labeled samples and $\mathcal{B}^u \subseteq \mathcal{D}^u$ the unlabeled samples in that batch such that $\mathcal{B} = (\mathcal{B}^l, \mathcal{B}^u)$.

2.2 Backbone Architecture

A basic building block in deep learning for images is a three-dimensional tensor, whereas time series can be represented as two-dimensional tensors with channels corresponding to the number of covariates. The extension of building blocks of powerful image classification architectures to TSC is thus straightforward, yet the right choice of a backbone architecture is crucial. We propose the use of the Fully Convolutional Network (FCN) (Wang et al. [46]) as a backbone architecture as it was shown to outperform a variety of models on 44 different TSC problems and is used in related work on semi-supervised TSC (Jawed et al. [20]). In all regularization-based semi-supervised methods discussed in Sect. 2.4, except for the Ladder Net (Rasmus et al. [35]), the network architecture can be decoupled from the model training strategy. This allows us to replace the backbone architecture of many of the established SSL methods from image classification with the FCN. In case of the Ladder Net, we design the decoder as a mirrored version of the FCN encoder (see Sect. 2.4).

Within a larger benchmark for different handwriting-recognition tasks [32] proposed the *CNN-LSTM*: an architecture for multivariate TSC consisting of CNN layers similar to the FCN followed by an LSTM head to capture temporal structure in the CNN features. Through an extensive benchmark, the CNN-LSTM showed consistently strong model performance. Next to the CNN-LSTM [14] successfully introduced the inception modules (Szegedy et al. [39]) from image to time-series classification, which has become another strong performing TSC architectures over time.

While we rely on the FCN as backbone for our main experiments in Sect. 1, we provide additional experimental results with the CNN-LSTM and the InceptionTime in Sect. 5. Therein, we use the same hyperparameters for the CNN-LSTM as [32] and set $nf = 64$ and $depth = 12$ for InceptionTime.

2.3 Data Augmentation

One crucial component of regularization-based semi-supervised methods is the injection of random noise into the model. Data augmentation strategies $g(x^{(i)})$, $g : \mathcal{X} \mapsto \mathcal{X}$ should be designed such that they perturbate the input $x^{(i)}$ of a sample while

preserving the meaning of its label $y^{(i)}$. This can be achieved by utilizing inherent invariances in the data, e.g., rotations of images usually preserve the meaning of an image. For images, invariances can be easily understood visually. In the time-series domain, such invariances are not straightforward to understand, rendering the design of reasonable data augmentation strategies in this domain challenging. A set of data augmentation strategies for multivariate time-series classification was introduced by [41] and evaluated on one human-activity-recognition task. They show that the majority of strategies are beneficial, but some can deteriorate the model performance. Iwana and Uchida [19] include these strategies in their review and introduce a taxonomy for different augmentation strategies for TSC and provide an empirical overview on the 128 datasets from the UCR repository (Dau et al. [12]). The results from their large empirical study show that while some augmentation strategies improve model performance and others are detrimental, the final impact of the data augmentation strategies heavily depends on and varies over the respective domain and application. Refer to [48] for similar yet less comprehensive overview which also includes forecasting next to classification tasks.

To overcome the additional burden of choosing the right strategy, we propose the use of the RandAugment strategy (Cubuk et al. [11]) which removes the need for a separate search phase. For each training batch, N augmentation strategies are randomly chosen out of a set of K possible policies. Next to N, a $magnitude$ hyperparameter is introduced which controls the augmentation intensity of the selected policies. We use the following set of augmentation policies following [41]: warping in the time dimension, warping the magnitude, addition of Gaussian Noise and random rescaling. We use RandAugment in this context following the rationale that even if one augmentation strategy is not label preserving, training with RandAugment with $N = 1$ will still produce correct model updates in at least $\frac{K-1}{K}$ of the forward passes. Early experiments in a fully supervised setting showed that the application of this data augmentation strategy improves model performance across all datasets used in our experiments.

In a series of additional experiments in Sect. 5, we further investigate the role of the size of the pool for RandAugment on the model performance on both supervised and semi-supervised models on a subset of datasets.

2.4 Methods

The Mean Teacher (Tarvainen and Valpola [40]) is the successor of a series of consistency-regularization-based models such as Temporal Ensembling or the Π-Model (Laine and Aila [24]) for SSL and was empirically shown to outperform its predecessors (Oliver et al. [31]). Thereby, a teacher model, that is an average of the consecutive student models, is used to enforce consistency in model predictions over the course of model training.

Virtual Adversarial Training (VAT) also focuses on consistency regularization (Miyato et al. [29]). Similar to adversarial examples (Goodfellow et al. [15]), a

small data perturbation is learned such that its addition to the initial data point is expected to yield the maximum change in the model's prediction. These perturbed model predictions are used as auxiliary labels for the unlabeled samples within a regularization term to enable model training on the whole data set. This approach is particularly interesting for the time-series domain where visual inspection of the appropriateness of data augmentation policies is difficult, as it does not rely on data augmentation techniques.

The Ladder Net by [35] is a reconstruction-based SSL model and is inspired by denoising autoencoders (Vincent et al. [44]). In its core, it extends a supervised encoder model with a corresponding decoder network which allows for the calculation of an unsupervised reconstruction loss over the unlabeled samples enabling training on the whole dataset. The Ladder Net was previously extended to TSC problems (Zeng et al. [50]) and is thus also part of this study.

In MixMatch, various semi-supervised techniques such as data augmentation for consistency regularization, Mixup training (Zhang et al. [52]) and pseudo-labeling are combined within one holistic approach (Berthelot et al. [6]). It was empirically shown to perform well on image data, motivating our use of it in this work (Berthelot et al. [6]).

FixMatch further builds upon this consistency regularization rationale via a combination with pseudo-labeling, alleviating the need for Mixup regularization over labeled and unlabeled samples (Sohn et al. [37]). Therefore, it uses confident predictions of weakly augmented versions of the unlabelled samples as pseudo-labels and combines them with model predictions over strongly augmented versions of the same samples within an unsupervised loss function. Hence, it alleviates the need for Mixup training at the cost of having to create a set of weak and strong data augmentation strategies. We include FixMatch as well in this work as it can be seen as the successor of MixMatch.

3 Experimental Design

3.1 Baseline Models

Next to shapelet- and distance-based methods (Bagnall et al. [2]), fitting standard ML methods on hand-crafted statistical features has been a widely used approach for TSC before the introduction of specific deep learning architectures for TSC (Wang et al. [46]) (Wang et al. [14]). We include a Random Forest and a Logistic Regression trained on features, extracted via the tsfresh framework (Christ et al. [10]) from the time series, as baselines.

In addition, we train the FCN architecture (Wang et al. [46]) on the labeled samples $\mathcal{D}^l$ based on the cross entropy loss as a supervised deep learning baseline model for our experiments. To ensure a fair and reliable model comparison, we explicitly use the same architecture of this supervised baseline model as the backbone for all SSL

approaches. We also use the performance of a supervised FCN trained on the fully labeled datasets as an estimated upper bound for the model performance.

Furthermore, we evaluate the performance of the self-supervised approach that was introduced for TSC by [20]. Thereby, an auxiliary forecasting task from the time-series data $\mathcal{D}$ is created and combined with the initial classification task as a surrogate supervision signal allowing the use of unlabeled data in model training. The model is then jointly trained on both tasks simultaneously. Next to its re-implementation, we further extend their approach for multivariate TSC by increasing the amount of neurons in the surrogate model head accordingly. The direct comparison with this self-supervised approach is of special interest as it was shown to outperform classical semi-supervised approaches in a set of experiments on smaller TSC datasets (Jawed et al. [20]).

3.2 Data Sets

We evaluate the performance of the above-described semi-supervised models on six publicly available datasets. In contrast to previous work (Jawed et al. [20, 45, 49]), we explicitly focus on large datasets with at least 1000 observations. Their main characteristics are described in Table 1.

With Crop, ElectricDevices and FordB, we include three of the largest datasets from the UCR Time Series Classification Repository (Dau et al. [12]). In addition, we use the two multivariate HAR datasets Pamap2 (Reiss and Stricker [36]) and WISDM (Kwapisz et al. [23]). We also evaluate the models on a class-balanced version of the Satellite Image Time Series (SITS) dataset (Petitjean et al. [33]).

Table 1 Characteristics of the used data sets where *c* refers to the amount of covariates, *Size* to the size of the whole training data set and *Length* to the length of the time series

Name	Classes	Size	Length	c	Balanced
Crop	24	7,200	46	1	✓
ElectricDevices	7	8,926	96	1	✗
FordB	2	3,636	500	1	✓
Pamap2	13	11,313	100	6	✗
WISDM	6	10,727	80	3	✗
Balanced SITS	6	35,064	46	1	✓

3.3 Evaluation, Tuning and Implementation

Due to special factors, such as the selection of the labeled data points, an unbiased and fair model comparison is particularly crucial to get a realistic perspective on the performance of the semi-supervised models (Van Engelen and Hoos [42]). We adhere to the guidelines for realistic evaluation of semi-supervised models by [31] to guarantee reliable and fair experimental results. For performance evaluation of SSL models, the standard procedure is to split a fully labeled dataset $\mathcal{D}$ into labeled and unlabeled datasets $\mathcal{D}^l$ and $\mathcal{D}^u$ via artificial *unlabeling* of n_u randomly drawn samples (Van Engelen and Hoos [42]). This way, semi-supervised data settings for different amounts of labeled samples l are simulated. We unlabel in a stratified manner to retain the datasets' label distributions. For the following experiments, we split the evaluation of one model f on one data set D in two distinct phases.

3.3.1 Tuning Phase

In the tuning phase, we tuned the model f with one fixed amount of labeled samples to yield an optimal set of hyperparameters θ^*. Thereby, f was trained on a training dataset $\mathcal{D}_{train} = (\mathcal{D}^l_{train}, \mathcal{D}^u_{train})$, where we fixed $|\mathcal{D}^l_{train}| = 500$, and validated on a labeled holdout validation set $\mathcal{D}_{val}$. The choice of the size of $\mathcal{D}_{val}$ is subject to recent discussions (Rasmus et al. [35]; Oliver et al. [31]; Zhai et al. [51]. Large $\mathcal{D}_{val}$ are expected to yield stable results for model tuning, which is important for many hyperparameter-sensitive semi-supervised models, but stands in contrast to the promised practicality of these models in settings with few labeled data. First insights on this trade-off are given by [31, 51], which empirically show in smaller experiments $|\mathcal{D}_{val}| = 1000$ to be a validation set size where variance in the performance estimates is still low enough to allow for reasonable model selection. Following this, we set the size of the labeled validation set to $|\mathcal{D}_{val}| = 1000$ which is rather small compared to recent literature where $|\mathcal{D}_{val}| \geq 4000$ (Miyato et al. [29]; Laine and Aila [24]; Tarvainen and Valpola [40]). A separate labeled test set $\mathcal{D}_{test}$ with $|\mathcal{D}_{test}| = 2000$ is kept aside for the evaluation phase. Hyperband (Li et al. [25]) with random sampling as implemented in the Optuna framework (Akiba et al. [1]) was used for tuning, with a fixed budget of 100 GPU hours for each deep learning model and dataset. We measure model performance in terms of weighted Area under the Curve (wAUC) to account for model calibration and class imbalance.

3.3.2 Evaluation Phase

In the evaluation phase, we train $f(\theta^*)$ on $\mathcal{D}_{train}$ with varying amounts of $n_l \in \{50, 100, 250, 500, 1000\}$ for a maximum of 25000 model update steps, assuming θ^* is also a suitable hyperparameter set for amounts of labels $n_l \neq 500$ on which the model was not specifically tuned. This evaluation scheme is in line with previous work

on SSL for image data (Berthelot et al. [6]; Oliver et al. [31]). Model performance is tracked on $\mathcal{D}_{val}$ and the model checkpoint with the best validation performance is used for inference on the holdout $\mathcal{D}_{test}$. The selection of especially (un-)informative labeled samples can have a major effect on the model performance, especially for small n_l. To account for potentially (un-)lucky selection of $\mathcal{D}^l_{train}$ in the *unlabelling* split of $\mathcal{D}_{train} = (\mathcal{D}^l_{train}, \mathcal{D}^u_{train})$, we repeat this *unlabelling* step 5 times. In case of the ML baseline models, we use a Random Search with a budget of 100 model evaluations for the tuning phase and evaluate them on the same set of values for n_l in the evaluation phase. See Table 6 in the appendix for the specific ranges. All deep learning models were implemented in a unified codebase[1] and trained using the Adam optimizer (Kingma and Ba [22]) with all parameters set to default values except the learning rate and weight decay. We implemented all deep learning models from scratch in one unified framework and validated our implementations based on performance metrics reported on image classification tasks.

4 Experimental Results

Experimental findings are visualized in Fig. 1 and Table 4 in the appendix. The ranking of the various models for different n_l, averaged over the six datasets is shown in Fig. 2 and in Table 5 in the appendix.

4.1 Main Results

4.1.1 Semi-supervised Models Outperform Supervised Baselines

Overall, our results show that semi-supervised models outperform baseline models especially for small amounts of labeled data. This relative performance gain of semi-supervised over supervised models is decreasing with an increase in n_l and we find that all models benefit from more labeled samples in most cases, a finding that is also line with literature on SSL [42].

4.1.2 Deep SSL Translates well to TSC

Following our experimental results in Fig. 1, we deduct that *transferring well-established semi-supervised models from the image to the time-series domain is indeed possible.* We find that the deep semi-supervised models, especially the transferred MixMatch, FixMatch and VAT, show substantial performance gains over the deep supervised baseline model over all six datasets up to $n_l = 500$, even reaching

[1] https://github.com/Goschjann/ssltsc.

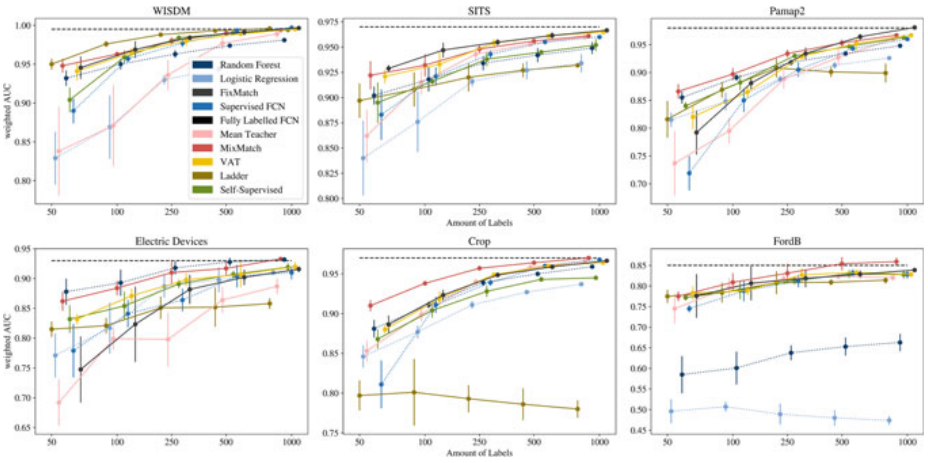


Fig. 1 Performance of all models on the six different datasets over various n_l as presented in Table 4 in the appendix. The horizontal line marks the performance of the fully labeled baseline, i.e. the supervised FCN model trained on the fully labeled dataset. Dots represent the mean wAUC and the vertical lines the standard deviation over five repeated *unlabeling* steps. The performance of the baseline models is depicted as dotted, those of the semi-supervised models as solid lines. Semi-supervised models clearly outperform the baseline models in settings with few labeled samples $n_l \in \{50, 100\}$ on all but the Electric Devices dataset

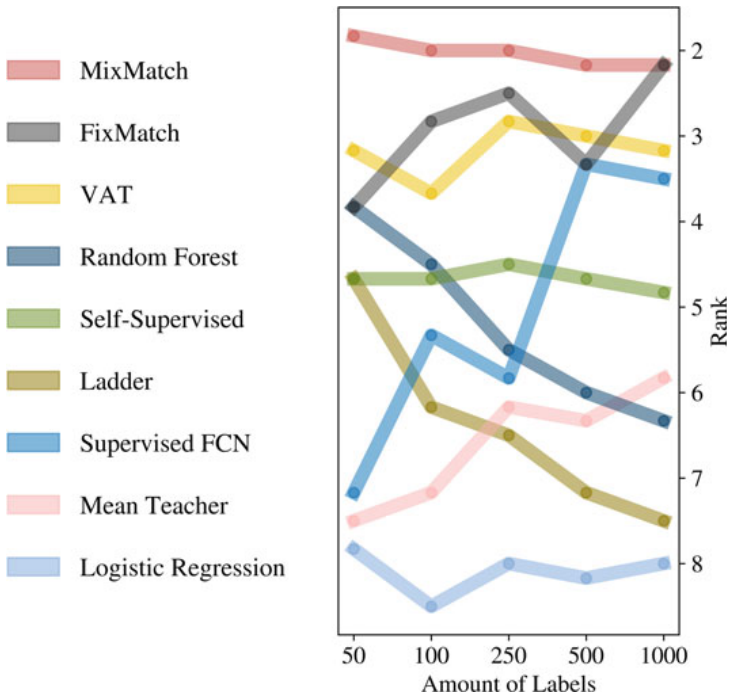


Fig. 2 Average ranks of all models based on the wAUC over the six datasets for varying n_l. Models are sorted by their strongest performance on $n_l = 50$ and plotted with decreasing rank as indicated on the right vertical axis

the performance of the fully labeled baseline in few cases. For instance, the Mixmatch model exceeds the deep supervised baseline by 0.16 wAUC on the Pamap2 and by 0.10 wAUC on the Crop dataset for $n_l = 50$. Overall, these findings encourage the usage of semi-supervised learning paradigms in the context of TSC, and we hope that this motivates further research on this exciting area.

4.1.3 Strong Baselines Are Crucial

The use of strong baselines crucial to gain a realistic perspective on the performance of semi-supervised learning approaches (Oliver et al. [31]) and our experimental results support this need for strong baselines further. For instance, the Mean Teacher shows weak performance on the majority of datasets, often performing even worse than the supervised baseline. These experimental results are in line with those of [20] who also include the Π-Model, next to deep supervised baselines, in their model comparison which is very similar to the Mean-Teacher in design. The strong performance of the Random Forest for small n_l on the other hand also stresses the need for realistically strong supervised baselines.

4.1.4 Proposed Methods Outperform Existing Semi-supervised Approaches

In alignment with [50], we also find the Ladder Net outperforms other supervised methods. Despite this performance gain over the supervised baselines, we observe that the Ladder Net performs notably worse compared then alternative proposed semi-supervised algorithms such as FixMatch, MixMatch and VAT. This varying performance might be grounded in the large amount of hyperparameters of the Ladder Net and its sensitivity to different settings of those. We further interpret this somehwat expected result as an expression of a general shift away from purely reconstruction-based approaches over to the development of more sophisticated self-supervised approaches that include complex pretext tasks.

4.1.5 Proposed Methods Outperform Self-supervised Modeling

Similar to [20], we find their self-supervised approach to perform better or at least equally well compared to the deep supervised baseline model. Additionally, we are able to show that our naive extension of their approach towards multivariate time series also works well on the two multivariate datasets, WISDM and Pamap2. Despite this performance gain over the supervised baslines, the proposed approaches Mix-Match, FixMatch and VAT furthermore consistently outperform this self-supervised approach across different amounts of labels on all six datasets. We hypothesize that this further demonstrates the capabilities of consistency regularization. Despite this, we also want to stress that consistency regularization as used within FixMatch

Table 2 Augmentation Procedures within RandAugment assigned to augmentation pools of varying size

Augmentation	Pool size		
Procedure	Small	Medium	Large
Time noise	✓	✓	✓
Magnitude noise	✓	✓	✓
Magnitude scaling	✓	✓	✓
Time warping		✓	✓
Magnitude warping		✓	✓
Time cutout			✓
Random crop			✓

and MixMatch and self-supervised learning as introduced by [20] are orthogonal approaches in this context and the combination of both could potentially enable further performance increases.

4.1.6 Ranking of Model Performance Similar to Image Domain

In terms of model performance ranking, literature suggests that MixMatch and FixMatch perform better than VAT which again outperforms the Mean Teacher and the Ladder Net [6, 31]. When ranking the algorithms across the datasets in Fig. 2, we confirm this ranking in the TSC setting in our experimental setup. One exception in this context is the FixMatch architecture which does unexpectedly not manage to outperform MixMatch as suggested in literature (Sohn et al. [37]) and ranks second. This further stresses the need to compare and evaluate modeling approaches across different domains and data modalities.

4.2 Additional Experiments

In addition to the main results in Sect. 4, we want to provide more insights on the use of (a) different data augmentation procedures and (b) different model architectures with more complexity than the FCN by [46]. For these additional experiments, we use the Electric Devices, Pamap2 and SITS datasets. We decidedly chose these three datasets as they offer relatively large training datasets, do not contain too many classes and span uni- and multivariate time series across the different domains remote sensing, production and human activitiy recognition. Further, we hold the amount of labeled samples fix to $n_l = 500$.

4.2.1 Data Augmentation

As motivated in Sect. 2, we use RandAugment as a wrapper over various data augmentation strategies tailored towards TSC. The choice of the most suitable data augmentation procedure for TSC remains an open research question with recent benchmarks providing unclear recommendations due to the heavy dependence of the domain of the respective TSC task (Iwana and Uchida [19]) (Wen et al. [48]). Still, we want to shed light on the size and hence the complexity of the pool of single data augmentation procedures across datasets from different domains. Therefore, we created a small, a medium and a large pool of procedures with increasing complexity as summarized in Table 2, refer to [19] for a detailed description of the single procedures. We fixed the RandAugment hyperparameters $N = 2, magnitude = 3$ as these were revealed as reasonable default values via our extensive tuning phase and evaluated all models on the test set following the procedure described in Sect. 3.3. The FCN architecture is used as backbone across all three models.

In Fig. 3, we illustrate the performance of two semi-supervised models, FixMatch and VAT, and the supervised model over varying RandAugment pool sizes across the Electric Devices, Pamap2 and SITS datasets for a fixed amount of $n_l = 500$ labels. Especially on the Electric Devices and the Pamap2 datasets, we find Fixmatch to benefit from an increase in pool size and thus data augmentation complexity the most. We hypothesize that this is due to the heavy use of consistency regularization which relies on heavy data augmentation within FixMatch, as it was shown to benefit from heavy data augmentation on image data as well [37]. Also, Pamap2 and ElectricDevices contain rather long and in the case of Pamap2 even multivariate time series as opposed to SITS. Hence, these two datasets might benefit more from stronger data augmentation as they are inherently more complex, it is important to note that the efficacy of data augmentation methods in TSC is highly domain-dependent (Iwana and Uchida [19]). Further, we observe that increasing the pool size from medium to large has a slightly detrimental effect on the supervised model while VAT does not benefit from a more complex set of procedures nor is it detrimental to its performance. Across all three datasets, we observe that the medium pool size leads to stronger model performance compared to the small pool. From a practitionerÃ¢â‚¬â„¢s view, this smaller series of experiments suggests that one should

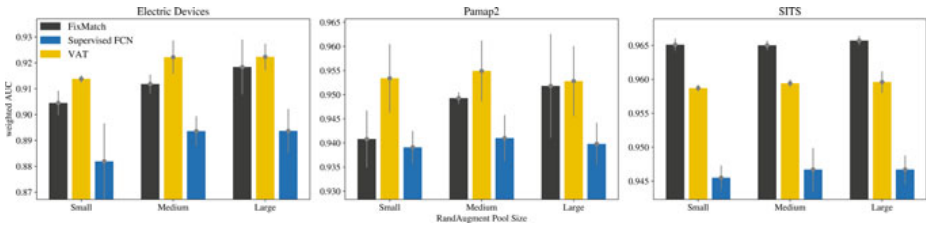


Fig. 3 Effect of different sizes of the data augmentation pool for the RandAugment strategy. We compare a small, medium and large pool over the Supervised FCN, VAT and FixMatch on three datasets with standard deviations reported over three repetitions each

invest in the creation of a data augmentation pool with some degree of complexity as this favors model performance across training paradigms. Further, the creation of overly complex augmentation pools is one of the key drivers of FixMatch's success on both, image and time-series classification tasks.

4.2.2 Backbone Architectures

We used the FCN [45] as backbone architecture in our main results in Sect. 4 as its strong baseline performance across different benchmarks next to its simplicity in design makes it a reasonable default backbone for TSC [13]. In addition to the FCN, we investigate the suitability of more complex backbone architectures for both supervised and semi-supervised learning, i.e. MixMatch and VAT. Therefore, we experiment with the CNN-LSTM [32] as a more advanced version of the FCN and the InceptionTime backbone [14] which introduces the inception module from computer vision based architectures [39] to TSC. All models were tuned with the same time budget on $n_l = 500$ following the evaluation strategy outlined in Sect. 3.3 and we present results for the $n_l = 500$ scenario across the three datasets.

From Table 3, we first observe that while the choice of the backbone architecture has an effect on the respective model performances, this effect is not substantial. For instance, we find the largest absolute performance gap of 0.016 wAUC with the supervised model on the Electric Devices dataset. We further confirm the role of the FCN architecture as a strong baseline following [46] and backing our choice of it as the backbone for our main experiments. Across all three models, the use of elaborate backbone architectures can still yield notable performance gains. For instance, this can be observed for MixMatch on the Pamap2 dataset where the InceptionTime backbone leads to an absolute performance increase of 0.011 wAUC over the CNN-LSTM backbine. From a practitionerÃ¢â‚¬â„¢s perspective, these results suggest that while the optimal backbone architecture can boost performance to some extent, the FCN backbone is a reasonable architecture choice for both supervised as well as semi-supervised time-series classification.

Table 3 Effect of the backbone architectures FCN (Wang et al. [46]), CNN-LSTM (Ott et al. [32]) and InceptionTime (Fawaz et al. [14]) on the performance of the Supervised Model, VAT and MixMatch across three datasets for fixed $n_l = 500$

	Electric devices		
	Supervised	VAT	MixMatch
FCN	0.904 (0.019)	0.907 (0.022)	**0.917 (0.013)**
CNN-LSTM	**0.920 (0.004)**	**0.918** (0.004)	0.905 (0.004)
InceptionTime	**0.920 (0.009)**	0.913 (0.012)	0.916 (0.003)
	Pamap2		
	Supervised	VAT	MixMatch
FCN	0.943 (0.004)	0.952 (0.012)	0.934 (0.007)
CNN-LSTM	**0.959 (0.002)**	0.929 (0.005)	0.915 (0.005)
InceptionTime	0.958 (0.004)	**0.955 (0.005)**	**0.946 (0.001)**
	SITS		
	Supervised	VAT	MixMatch
FCN	0.955 (0.001)	**0.961 (0.001)**	0.956 (0.006)
CNN-LSTM	**0.964 (0.001)**	0.960 (0.003)	**0.959 (0.001)**
InceptionTime	0.962 (0.001)	0.959 (0.001)	0.956 (0.001)

5 Conclusion

In this work, we investigated the transferability of modern semi-supervised learning approaches from their initial domain of computer vision towards the classification of time-series data. We explored this potential within a series of experiments across six challenging benchmark datasets and describe the necessary changes that enable this transfer. Thereby, we further shed light on the role of data augmentation strategies and backbone model architectures in this context. Our results show that the promised label efficiency of modern, deep semi-supervised model approaches translates well to TSC problems. Furthermore, these findings suggest the use of strong semi-supervised models from the image domain as these transferred models show stronger performance than the currently existing semi- and self-supervised approaches tailored towards TSC. We believe that this work, also thanks to a strong focus on a fair and reliable model comparison, can serve as the basis for future research advances in semi-supervised learning for time-series classification.

Appendix

5.1 Model Performance

See Table 4.

Table 4 Results for models over datasets with varying numbers of labels n_l. Performance is measured as weighted AUC. The best results for each n_l-dataset-combination are emphasized in bold with standard deviations over five replications in brackets

Number of labels	50						100					
Dataset	Crop	Electric devices	FordB	Pamap2	SITS	WISDM	Crop	Electric devices	FordB	Pamap2	SITS	WISDM
Model												
Ladder	0.797 (0.019)	0.815 (0.013)	0.775 (0.016)	0.816 (0.033)	0.897 (0.017)	**0.950 (0.007)**	0.801 (0.042)	0.821 (0.013)	0.784 (0.024)	0.869 (0.014)	0.908 (0.017)	**0.976 (0.004)**
Logistic regression	0.846 (0.014)	0.771 (0.037)	0.496 (0.029)	0.815 (0.013)	0.840 (0.037)	0.829 (0.034)	0.877 (0.008)	0.817 (0.043)	0.507 (0.012)	0.848 (0.016)	0.876 (0.03)	0.869 (0.041)
Mean teacher	0.853 (0.013)	0.692 (0.039)	0.745 (0.036)	0.737 (0.058)	0.862 (0.026)	0.838 (0.057)	0.899 (0.009)	0.799 (0.018)	0.797 (0.023)	0.795 (0.023)	0.915 (0.012)	0.871 (0.052)
MixMatch	**0.910 (0.007)**	0.862 (0.016)	0.775 (0.012)	**0.866 (0.013)**	0.922 (0.014)	0.948 (0.009)	**0.938 (0.003)**	0.884 (0.012)	**0.809 (0.022)**	**0.897 (0.011)**	0.932 (0.011)	0.963 (0.003)
Random forest	0.881 (0.011)	**0.878 (0.022)**	0.585 (0.045)	0.855 (0.011)	0.902 (0.009)	0.932 (0.01)	0.911 (0.004)	**0.893 (0.022)**	0.601 (0.04)	0.891 (0.006)	0.918 (0.012)	0.95 (0.007)
Self-supervised	0.868 (0.012)	0.832 (0.023)	0.772 (0.007)	0.84 (0.007)	0.895 (0.02)	0.904 (0.016)	0.904 (0.009)	0.854 (0.024)	0.79 (0.022)	0.882 (0.014)	0.916 (0.003)	0.959 (0.009)
Supervised	0.811 (0.03)	0.779 (0.045)	0.745 (0.009)	0.719 (0.031)	0.883 (0.025)	0.89 (0.016)	0.911 (0.01)	0.841 (0.023)	0.788 (0.025)	0.85 (0.021)	0.921 (0.009)	0.957 (0.004)
VAT	0.880 (0.005)	0.832 (0.009)	**0.783 (0.016)**	0.82 (0.022)	0.921 (0.007)	0.941 (0.012)	0.919 (0.009)	0.871 (0.012)	0.789 (0.011)	0.865 (0.01)	0.933 (0.012)	0.965 (0.006)
FixMatch	0.886 (0.011)	0.748 (0.056)	0.776 (0.053)	0.792 (0.040)	**0.929 (0.003)**	0.946 (0.015)	0.923 (0.007)	0.823 (0.063)	0.807 (0.041)	0.881 (0.011)	**0.947 (0.007)**	0.969 (0.011)

(continued)

Table 4 (continued)

Number of labels	250						500					
Dataset	Crop	Electric devices	FordB	Pamap2	SITS	WISDM	Crop	Electric devices	FordB	Pamap2	SITS	WISDM
Model												
Ladder	0.793 (0.017)	0.851 (0.019)	0.808 (0.028)	0.908 (0.017)	0.920 (0.014)	**0.988 (0.001)**	0.786 (0.02)	0.852 (0.033)	0.809 (0.007)	0.901 (0.009)	0.927 (0.007)	**0.993 (0.001)**
Logistic Regression	0.911 (0.005)	0.887 (0.019)	0.489 (0.025)	0.888 (0.006)	0.916 (0.005)	0.93 (0.006)	0.927 (0.002)	0.898 (0.021)	0.48 (0.019)	0.913 (0.006)	0.927 (0.009)	0.944 (0.004)
Mean Teacher	0.940 (0.004)	0.798 (0.045)	0.817 (0.011)	0.888 (0.017)	0.943 (0.007)	0.936 (0.019)	0.958 (0.004)	0.864 (0.022)	0.823 (0.007)	0.927 (0.014)	0.953 (0.003)	0.977 (0.008)
MixMatch	**0.957 (0.003)**	0.910 (0.019)	**0.831 (0.023)**	**0.934 (0.007)**	0.948 (0.004)	0.980 (0.005)	**0.964 (0.003)**	0.917 (0.013)	**0.854 (0.016)**	0.953 (0.006)	0.956 (0.002)	0.990 (0.003)
Random Forest	0.939 (0.004)	**0.918 (0.011)**	0.638 (0.018)	0.918 (0.005)	0.934 (0.007)	0.963 (0.005)	0.950 (0.003)	**0.928 (0.007)**	0.653 (0.022)	0.934 (0.004)	0.942 (0.007)	0.974 (0.001)
Self-Supervised	0.928 (0.007)	0.891 (0.007)	0.814 (0.01)	0.930 (0.001)	0.938 (0.007)	0.984 (0.001)	0.943 (0.003)	0.907 (0.006)	0.829 (0.01)	0.947 (0.004)	0.945 (0.002)	0.990 (0.002)
Supervised	0.939 (0.004)	0.864 (0.019)	0.812 (0.015)	0.905 (0.013)	0.943 (0.004)	0.977 (0.005)	0.960 (0.002)	0.904 (0.019)	0.832 (0.008)	0.943 (0.004)	0.955 (0.001)	**0.993 (0.004)**
VAT	0.949 (0.002)	0.898 (0.011)	0.827 (0.017)	0.929 (0.006)	0.954 (0.004)	0.980 (0.006)	0.960 (0.003)	0.907 (0.022)	0.833 (0.005)	0.952 (0.012)	0.961 (0.001)	0.989 (0.002)
FixMatch	0.949 (0.003)	0.882 (0.024)	0.819 (0.022)	**0.934 (0.011)**	**0.955 (0.003)**	0.984 (0.001)	0.959 (0.001)	0.902 (0.013)	0.828 (0.009)	**0.964 (0.005)**	**0.962 (0.003)**	0.991 (0.004)

(continued)

Table 4 (continued)

Number of labels	1000						
Dataset	Crop	Electric devices	FordB	Pamap2	SITS	WISDM
Model						
Ladder	0.78 (0.011)	0.858 (0.009)	0.814 (0.008)	0.899 (0.017)	0.932 (0.001)	0.996 (0.001)
Logistic Regression	0.937 (0.001)	0.91 (0.003)	0.474 (0.011)	0.926 (0.004)	0.934 (0.009)	0.950 (0.003)
Mean Teacher	0.966 (0.001)	0.887 (0.013)	0.820 (0.006)	0.960 (0.002)	0.960 (0.002)	0.989 (0.005)
MixMatch	**0.970 (0.001)**	**0.933 (0.003)**	**0.859 (0.010)**	0.967 (0.004)	0.961 (0.002)	0.994 (0.001)
Random Forest	0.959 (0.001)	0.932 (0.004)	0.663 (0.021)	0.948 (0.003)	0.949 (0.006)	0.981 (0.001)
Self-Supervised	0.945 (0.001)	0.919 (0.007)	0.828 (0.010)	0.963 (0.004)	0.952 (0.006)	0.994 (0.001)
Supervised	0.968 (0.001)	0.910 (0.011)	0.828 (0.010)	0.960 (0.003)	0.960 (0.002)	**0.997 (0.001)**
VAT	0.964 (0.001)	0.920 (0.008)	0.828 (0.006)	0.967 (0.004)	0.965 (0.002)	0.995 (0.001)
FixMatch	0.967 (0.001)	0.916 (0.005)	0.839 (0.005)	**0.981 (0.001)**	**0.967 (0.002)**	**0.997 (0.001)**

5.2 Model Ranking

See Table 5.

Table 5 The average rank of all models based on the wAUC over the six different datasets for various amounts of labels n_l. Lower rank indicates stronger model performance. Ranks are shown with decimals due to averaging over datasets

	Number of labels				
	50	100	250	500	1000
MixMatch	**1.8**	**2.0**	**2.0**	**2.2**	**2.2**
FixMatch	3.8	2.8	2.5	3.3	**2.2**
VAT	3.2	3.7	2.8	3.0	3.2
MeanTeacher	7.5	7.2	6.2	6.3	5.8
Self-supervised	4.7	4.7	4.5	4.7	4.8
Ladder	4.7	6.2	6.5	7.2	7.5
Supervised	7.2	5.3	5.8	3.3	3.5
Random forest	3.8	4.5	5.5	6.0	6.3
Logistic regression	7.8	8.5	8.0	8.2	8.0

5.3 Hyperparameters

See Table 6.

Table 6 Hyperparameter ranges are used for tuning of the different models. Deep Learning models were tuned via Hyperband as described in Sect. 3 while the Random Forest and the Logistic Regression were tuned via Random Search with a bugdet of 100 model evaluations each

Parameter	Range	Scale
Shared		
Weight decay	$[1e^{-6}; 1e^{-2}]$	log
Learning rate	$[1e^{-5}; 1e^{-2}]$	log
Rampup length	[5000; 25000]	linear
Magnitude (RandAug)	[1; 10]	linear
N (RandAug)	[1; 6]	linear
VAT		
ϵ	[0.1; 10.0]	linear
α	[0.1; 5.0]	linear
MixMatch		
α	[0.5; 1.0]	linear
λ_u	[0.0; 150.0]	linear
FixMatch		
τ	[0.75; 0.99]	linear
λ_u	[0.0; 10.0]	linear
Self-Supervised Learning		
λ	[0.1; 10]	log
horizon h	[0.1, 0.2, 0.3]	discrete
stride s	[0.05, 0.1, 0.2, 0.3]	discrete
Ladder Net		
Noise ratio	[0.1, 0.3, 0.45, 0.6]	discrete
Loss weights	[0.1; 10.0]	log
Mean Teacher		
α_{ema}	[0.9; 1.0]	log
w_{max}	[0; 10]	linear
Random Forest		
Number of trees	[100; 1000]	linear
Max. tree depth	[3; 25]	linear
Logistic Regression		
Regularization term	[None, L_1, L_2]	discrete

References

1. Akiba, T. et al.: Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACMSIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2623–2631 (2019)
2. Bagnall, A., Lines, J., Bostrom, A., et al.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining Knowl. Discov. **31**(3), 606–660 (2017)
3. Bagnall, A., Lines, J., Hills, J., et al.: Time-series classification with COTE: the collective of transformation-based ensembles. IEEE Trans. Knowl. Data Eng. **27**(9), 2522–2535 (2015)
4. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. In: CoRR (2018)
5. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J. Mach. Learn. Res. **7**, 2399–2434 (2006)
6. Berthelot, D., et al.: MixMatch: a holistic approach to semi-supervised learning. Adv. Neural Inf. Process. Syst. **32**, 5049–5059 (2019)
7. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with cotraining. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 92–100 (1998)
8. Chapelle, O., Scholkopf, B., Zien, A.: Semi-supervised Learning. MIT Press, p. 508 (2006)
9. Chen, Y. et al.: DTW-D: time series semi-supervised learning from a single example. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 383–391 (2013)
10. Christ, M. et al.: Time series feature extraction on basis of scalable hypothesis tests (tsfresh-A Python package). In: Neurocomputing, pp. 72–77 (2018)
11. Cubuk, E.D. et al.: Randaugment: practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 702–703 (2020)
12. Dau, H.A. et al.: The UCR time series archive (2019). arXiv:1810.07758 [cs.LG]
13. Fawaz, H.I., Forestier, G., et al.: Deep learning for time series classification: a review. Data Mining Knowl. Discov. **33**(4), 917–963 (2019)
14. Fawaz, H.I., Lucas, B., et al.: Inceptiontime: Finding alexnet for time series classification. Data Mining Knowl. Discov. **34**, 1936–1962 (2020)
15. Goodfellow, I., et al.: Generative adversarial nets. Adv. Neural Inf. Process. Syst. **27**, 2672–2680 (2014)
16. Goschenhofer, J. et al.: Wearable-based Parkinson's disease severity monitoring using deep learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 400–415. Springer (2019)
17. Grabocka, J. et al.: Learning time-series shapelets. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 392–401 (2014)
18. Iscen, A. et al.: Label propagation for deep semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5070–5079 (2019)
19. Iwana, B.K., Uchida, S.: An empirical survey of data augmentation for time series classification with neural networks. Plos one **16**(7), e0254841 (2021)
20. Jawed, S., Grabocka, J., Schmidt-Thieme, L.: Self-supervised learning for semi-supervised time series classification. In: Proceedings of the 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2020, pp. 499–511 (2020)
21. Kate, R.J.: Using dynamic time warping distances as features for improved time series classification. Data Mining Knowl. Discov. **30**(2), 283–312 (2016)
22. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference for Learning Representations, ICLR 2015 (2015)
23. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. ACM SigKDD Explorat. Newsl. **12**(2), 74–82 (2011)
24. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: 5th International Conference on Learning Representations, ICLR 2017 (2017)

25. Li, L., et al.: Hyperband: a novel bandit-based approach to hyperparameter optimization. J. Mach. Learn. Res. **18**(1), 6765–6816 (2017)
26. Luo, Y. et al.: Smooth neighbors on teacher graphs for semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8896–8905 (2018)
27. Mallapragada, P.K., et al.: Semiboost: boosting for semi-supervised learning. IEEE Trans. Pattern Anal. Mach. Intell. **31**(11), 2000–2014 (2008)
28. Marussy, K., Buza, K.: SUCCESS: a new approach for semi-supervised classification of time-series. In: International Conference on Artificial Intelligence and Soft Computing, pp. 437–447 (2013)
29. Miyato, T., et al.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE Trans. Pattern Anal. Mach. Intell. **41**(8), 1979–1993 (2019)
30. Nguyen, M.N., Li, X.-L., Ng, S.-K.: Positive unlabeled learning for time series classification. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, IJCAI-11 (2011)
31. Oliver, A., et al.: Realistic evaluation of deep semi-supervised learning algorithms. Adv. Neural Inf. Process. Syst. **31**, 3235–3246 (2018)
32. Ott, F. et al.: Benchmarking online sequence-to-sequence and character based handwriting recognition from IMU-enhanced pens. arXiv preprint arXiv:2202.07036
33. Petitjean, F., Inglada, J., Gancarski, P.: Satellite image time series analysis under time warping. IEEE Trans. Geosci. Remote Sens. **50**(8), 3081–3095 (2012)
34. Rajkomar, A., et al.: Scalable and accurate deep learning with electronic health records. NPJ Digital Med. **1**(1), 18 (2018)
35. Rasmus, A., et al.: Semi-supervised learning with ladder networks. Advan. Neural Inf. Process. Syst. **28**, 3546–3554 (2015)
36. Reiss, A., Stricker, D.: Introducing a newbenchmarked dataset for activity monitoring. In: 2012 16th International Symposium on Wearable Computers, pp. 108–109 (2012)
37. Sohn, K. et al.: FixMatch: simplifying semi-supervised learning with consistency and confidence. In: CoRR (2020). arXiv:2001.07685
38. Susto, G.A., Cenedese, A., Terzi, M.: Time-series classification methods: reviewand applications to power systems data. In: Big Data Application in Power Systems, pp. 179–220. Elsevier (2018)
39. Szegedy, C. et al.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
40. Tarvainen, A., Valpola, H.: Mean teachers are better role models: weightaveraged consistency targets improve semi-supervised deep learning results. Adv. Neural Inf. Process. Syst. **30**, 1195–1204 (2017)
41. Um, T.T. et al.: Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks. In: Proceedings of the 19th ACM International Conference on Multimodal Interaction, pp. 216–220 (2017)
42. Van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Mach. Learn. **109**(2), 373–440 (2020)
43. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
44. Vincent, P., et al.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**(12), 3371–3408 (2010)
45. Wang, H., et al.: Time series feature learning with labeled and unlabelled data. Pattern Recognit. **89**, 55–66 (2019)
46. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks:Astrong baseline. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 1578–1585 (2017)
47. Wei, L., Keogh, E.: Semi-supervised time series classification. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 748–753 (2006)
48. Wen, Q. et al.: Time series data augmentation for deep learning: a survey (2020). arXiv preprint arXiv:2002.12478

49. Xu, Z., Funaya, K.: Time series analysis with graph-based semisupervised learning. In: 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 1–6 (2015)
50. Zeng, M. et al.: Semi-supervised convolutional neural networks for human activity recognition. In: 2017 IEEE International Conference on Big Data (Big Data), pp. 522–529 (2017)
51. Zhai, X. et al.: S4L: self-supervised semi-supervised learning. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1476–1485 (2019)
52. Zhang, H. et al.: mixup: beyond empirical risk minimization. In: 6th International Conference for Learning Representations, ICLR 2018 (2018)
53. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Technical report (2002)

4.2 ConstraintMatch for Semi-constrained Clustering

Contributing article:

Jann Goschenhofer, Bernd Bischl, and Zsolt Kira. 2023. Constraintmatch for semi-constrained clustering. *International Joint Conference on Neural Networks (IJCNN)*

Author contributions:

This work was initiated by Jann Goschenhofer and the main body of work was done under the close supervision of Zsolt Kira during Jann Goschenhofer's stay as a visiting researcher at Zsolt Kira's research lab. Jann Goschenhofer was responsible for the conceptualization of the idea, the implementation of the different approaches into one holistic codebase, and running the experiments. Zsolt Kira greatly contributed to the development of the concept and the experimental design via his supervision. Bernd Bischl contributed via supervision, manuscript editing and providing access to computing resources. Jann Goschenhofer was responsible for writing the original draft, creating the included figures, reviewing, and editing. Zsolt Kira contributed to manuscript writing, proofreading and editing, and the formulation of the reviewer responses.

Copyright information:

ConstraintMatch for Semi-constrained Clustering

Jann Goschenhofer*†‡, Bernd Bischl*†‡, Zsolt Kira§
LMU Munich*
Fraunhofer Institute for Integrated Circuits†
Munich Center for Machine Learning‡
Georgia Institute of Technology§

***Abstract*—Constrained clustering allows the training of classification models using pairwise constraints only, which are weak and relatively easy to mine, while still yielding full-supervision-level model performance. While they perform well even in the absence of the true underlying class labels, constrained clustering models still require large amounts of binary constraint annotations for training. In this paper, we propose a semi-supervised context whereby a large amount of *unconstrained* data is available alongside a smaller set of constraints, and propose *ConstraintMatch* to leverage such unconstrained data. While a great deal of progress has been made in semi-supervised learning using full labels, there are a number of challenges that prevent a naive application of the resulting methods in the constraint-based label setting. Therefore, we reason about and analyze these challenges, specifically 1) proposing a *pseudo-constraining* mechanism to overcome the confirmation bias, a major weakness of pseudo-labeling, 2) developing new methods for pseudo-labeling towards the selection of *informative* unconstrained samples, 3) showing that this also allows the use of pairwise loss functions for the initial and auxiliary losses which facilitates semi-constrained model training. In extensive experiments, we demonstrate the effectiveness of ConstraintMatch over relevant baselines in both the regular clustering and overclustering scenarios on five challenging benchmarks and provide analyses of its several components.**

I. Introduction

Manual annotation of class labels is a tedious and labor-intensive task that can constitute a significant obstacle in applications, particularly in situations where the annotator has to select from a large number of potential class labels or where the annotation is ambiguous due to the task complexity. Additionally, supervised classification models require knowledge of the total number of classes present in the respective application, i.e. the cardinality of the label space. *Constrained clustering* offers a remedy for this as model training in this weakly supervised regime requires only weak, pairwise constraint relations (i.e. similar/dissimilar) which incur less annotation effort compared to instance-specific class labels [47]. These models can also learn meaningful cluster representations even in the overclustering scenario without knowledge of the underlying amount of clusters [14]. The majority of research on constrained clustering focuses on the *constrained* scenario, where each data point is associated with at least one constraint pair. As this setting still requires large sets of given constraints and hence incurs high annotation effort, we focus on the *semi-constrained* setting where a clustering model is trained on both a small dataset of pairwise constraints and a large dataset of unconstrained samples.

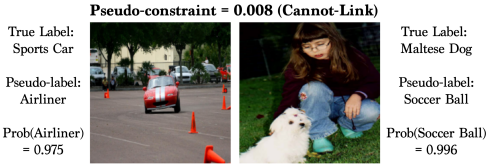


Fig. 1: Illustration of pseudo-constraining. While the model creates overconfident, wrong pseudo-labels for both unlabeled samples, it still yields a semantically correct pseudo-constraint.

While a great deal of progress has been made in semi-supervised learning when class labels are provided, we identify through analysis a number of challenges when applying such methods to the constrained clustering setting. One of the most effective methods in semi-supervised learning, pseudo-labeling, utilizes confident predictions on unlabeled data in training and is therefore prone to *confirmation bias*. Specifically, unlabeled samples that were confidently assigned the wrong class label by the model are selected as pseudo-labels, which leads to subsequent model degradation [1]. We analyze this issue in the context of constraint labels and propose a *pseudo-constraining* mechanism that we show can mitigate it, by generating pseudo-constraints from the pseudo-labels (see Fig. 1). Further, we argue that a confidence-based pseudo-label selection criterion is inappropriate in this setting as it leads to the unnecessary de-selection of unconstrained samples that contain valuable information for subsequent pseudo-constraining. We, therefore, propose an entropy-based criterion to select *informative* unconstrained samples and show its superiority. The combination of these two methods, *ConstraintMatch*, facilitates effective pseudo-labeling and unifies the initial and auxiliary learning task. We show that ConstraintMatch is able to outperform several state-of-the-art baselines using only a few constraint annotations by substantial margins, even in the more challenging overclustering scenario.

Contributions We 1) propose *ConstraintMatch* as a method for semi-constrained training of clustering models leveraging a large set of unconstrained samples next to a small set of pairwise constraints. Within a series of experiments, we 2) specifically make the case for pseudo-constraints over naive pseudo-labels and provide a detailed analysis of ConstraintMatch's several components. Furthermore, we 3) empirically

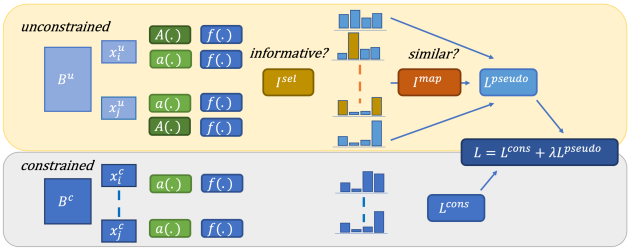


Fig. 2: ConstraintMatch combines pairwise training on batches of constrained (gray) and unconstrained (yellow) samples leveraging weak and strong data augmentations $a(), A()$. The criterion $\mathcal{I}^{sel}$ is used to select *informative* pseudo-labels from unconstrained samples which are then mapped to pairwise pseudo-constraints via $\mathcal{I}^{map}$ to overcome the *confirmation bias*. Predictions from model $f()$ over strongly augmented versions of these samples serve as inputs to the auxiliary loss $\mathcal{L}^{pseudo}$ to enforce consistency in predicted cluster assignments. ConstraintMatch is trained on a combination of the pseudo-constrained and the constrained loss $\mathcal{L}^{cons}$.

prove the strong performance of ConstraintMatch of up to 16.75% NMI over the constrained baseline on a series of five challenging benchmark datasets in both the regular and the overclustering scenario. Thereby, we evaluate models in different settings to unify the evaluation of modern deep clustering approaches and 4) release our source code[1] for future research on semi-constrained clustering.

II. Related Work

We provide an overview of the context of ConstraintMatch at the intersection of deep clustering, constrained clustering, and semi-supervised learning in the following.

Deep Clustering Early methods for deep clustering combine a reconstruction target with a clustering loss to learn expressive clustering features via reconstruction [10], [23], [37]. Subsequent approaches shift this focus toward low-level features via alternating cluster assignments with those provided by traditional clustering algorithms [3], [45]. More recent research is directed at mapping the data onto low-dimensional representations which serve as a training target for similarity-based losses and as cluster predictions during model inference [4], [18], [24], [29], [44]. Van Gansbeke et al. [40] found these approaches are prone to learning low-level features which lack meaning for semantic clustering next to heavy dependence on network initialization. Therefore, they propose SCAN as a two-step approach where feature representations learned via contrastive pretext tasks [5], [11] are used to mine nearest neighbors of the unlabeled samples. The model is then trained via a clustering loss which maximizes the alignment of their joint feature representations and enables clustering in the absence of the true underlying amount of clusters. SCAN was decidedly evaluated on test datasets only to prove its efficacy on new, unseen data. While this is appealing from a modeling perspective, it prevents direct comparison with prior work where clustering models are evaluated on the union of training and test datasets. We unify this model comparison in our experiments and find SCAN to perform on par with subsequent approaches TCC [30], CC [24], and MICE [38]. Hence, we use SCAN as a starting point for ConstraintMatch.

Constrained Clustering The introduction of binary instance-level constraints for clustering [41] led to the adaptation of existing clustering methods towards the use of constraints [42], see [9] for an overview. With the proposal of the KCL loss, Hsu et al. [14] introduced constrained clustering and overclustering to deep learning. They further showed its applicability to transfer learning [15] and introduced the Meta-classification-likelihood (MCL) for improved model training with pairwise constraints [16] and both loss formulations can not be used with unconstrained data. Zhang et al. [47] provide a framework to work with various types of constraints.

Semi-supervised Learning In semi-supervised classification, the rationale of consistency regularization lead to substantial improvements over supervised baselines [2], [21], [33], [46]. Among these, FixMatch [33] yields state-of-the-art model performance even in settings with very low supervision. It combines confidence-based pseudo-labeling [22] with consistency regularization using the weak-and strong augmentation scheme over unlabeled samples.

Semi-constrained Clustering With S^3C^2, a two-stage approach was proposed that leverages pseudo-constraints mined from a siamese network trained on few constraints [32]. While this approach was shown to perform well on simple benchmarks, it lacks end-to-end training and requires the true amount of clusters as input. Similarly, the approach by Fogel et al. [8] requires said amount of true clusters as input next to being a transductive method prohibiting inference on unseen data without access to the training data after training. PCOG [28] is another transductive method that also requires the true amount of cluster while its spectral decomposition component hinders it from scaling to large datasets. The approach of Shukla et al. [31] relies on a few class labels, rendering it non-applicable for scenarios where only constraints are present. In contrast to these approaches, we introduce a semi-constrained

[1] https://anonymous.4open.science/r/constraintmatch

clustering method that only relies on constraint annotations, leverages unconstrained data, is inductive, and works well without knowledge w.r.t the true amount of clusters.

III. Method

A. Notation

We consider a dataset $\mathcal{D}$ which consists of constrained and unconstrained datasets $\mathcal{D}^c$ and $\mathcal{D}^u$. $\mathcal{D}^c$ contains n_c constrained pairs of the form $x_{ij}^c = (x_i^c, x_j^c, c_{ij}) \in \mathcal{D}^c$ where x_i^c, x_j^c refer to two input samples and $c_{ij} \in \{0, 1\}$ to the associated binary constraint. These constraints describe that both samples either correspond to the same cluster $c_{ij} = 1$, *Must-Link* constraints (ML), or to different clusters $c_{ij} = 0$, *Cannot-Link* constraints (CL). $\mathcal{D}^u$ consists of n_u unconstrained input samples $x_i^u \in \mathcal{D}^u$. Further, we denote $\mathcal{B} \subset \mathcal{D}, \mathcal{B}^c \subset \mathcal{D}^c, \mathcal{B}^u \subset \mathcal{D}^u$ as batches of input samples x_i of the respective datasets. We refer to true class labels as $y_i \in \mathcal{Y}$ where $K = |\mathcal{Y}|$ describes the amount of true classes, i.e. the amount of underlying clusters K, in the dataset. Note that when K is not known, the model may have a different number of outputs n_{out} than the ground truth number of clusters. We aim at training a clustering model f in the form of a neural network with its final head consisting of n_{out} output neurons followed by a softmax layer, i.e. the model predicts a probability distribution over cluster assignments $\hat{y}_i = f(x_i)$ where $\hat{y}_{il}$ denotes the predicted probability of x_i belonging to cluster $l \in 1, ..., n_{out}$. Similarly, we refer to pseudo-labels as $\tilde{y}_i$ and to pseudo-constraints as $\tilde{c}_{ij} \in [0, 1]$. Further, we introduce the criterion $\mathcal{I}^{sel}$ which selects a subset of informative pseudo-labels $\tilde{y}_i$ based on their predicted cluster assignments $\hat{y}_i$ from $\mathcal{B}$. From pairs of these selected pseudo-labels $\tilde{y}_i, \tilde{y}_j$ we construct pseudo-constraints $\tilde{c}_{ij}$ using a second criterion $\mathcal{I}^{map}$.

B. Algorithm

ConstraintMatch is an annotation-efficient method that can leverage large unconstrained (i.e. unlabeled) data $\mathcal{D}^u$ next to few constraint pairs $\mathcal{D}^c$ to train a clustering model f. It uses unsupervised clustering in a pretraining step and combines training strategies from constrained clustering [14], [16] with the state-of-the-art semi-supervised learning method Fixmatch [33], refer to Fig. 2 for illustration. We use SCAN [40] for the pretraining step but other pretraining methods would also be applicable. Specifically, pseudo-labeling (i.e. self-training) has proven itself an effective method for leveraging unlabeled data and is a key component of recent semi-supervised classification models [21], [33]. In naive pseudo-labeling, confident model predictions over unlabeled samples are used as pseudo-targets in an auxiliary classification loss to guide model training next to the initial supervised loss, assuming that model confidence is associated with model correctness [22], [39]. Adapting this concept to constrained clustering, we identified three main weaknesses which we overcome: 1) **Pseudo-constraining:** Prediction errors in the selected pseudo-labels can amplify during training, potentially leading to model degradation, also known as *confirmation bias* [1]. We, therefore, propose the generation of *pseudo-constraints*, relying on the fact that pairwise constraints result in a simpler problem reduction [16]. 2) **Informativeness criterion** to carry information of whether two samples x_i and x_j are predicted to be in the same or a different cluster, which cannot be done via maximal prediction probability [33] or alternative uncertainty metrics [1], and 3) **Unification of losses** by utilizing a constraint-based loss for the unlabeled set.

Our overall algorithm processes unconstrained batches $\mathcal{B}^u$ via an unconstrained branch and constrained batches $\mathcal{B}^c$ within a constrained branch to enable training of clustering model f in this semi-constrained data scenario. The constrained branch is trained via a pairwise objective $\mathcal{L}^{cons}$ which allows the training of the model f on binary pairwise constraints $x_{ij}^c = (x_i^c, x_j^c, c_{ij}) \in \mathcal{D}^c$. Therefore, we combine the predictions from model f over weakly augmented constrained samples $a(x_i^c), a(x_j^c)$ along the associated constraint c_{ij} within the pairwise loss function $\mathcal{L}^{cons}$. For the unconstrained branch, we build upon the intuition of consistency regularization via weak and strong data augmentation strategies $a()$ and $A()$ [33] as follows. Given a pair of unconstrained samples x_i^u, x_j^u, we use the selection criterion $\mathcal{I}^{sel}$ to select *informative* model predictions over weakly augmented versions of those samples $(f(a(x_i^u)), f(a(x_j^u)))$ as pseudo-labels $(\tilde{y}_i, \tilde{y}_j)$. These are then combined into pseudo-constraints $\tilde{c}_{ij}$ via $\mathcal{I}^{map}$ and used as targets within the auxiliary loss function $\mathcal{L}^{pseudo}$. Model predictions over strongly augmented versions of the unconstrained pair $(\hat{y}_i, \hat{y}_j) = (f(A(x_i^u)), f(A(x_j^u)))$ serve as inputs for $\mathcal{L}^{pseudo}$. ConstraintMatch is trained using the combined loss function $\mathcal{L} = \mathcal{L}^{cons} + \lambda \mathcal{L}^{pseudo}$. The components of ConstraintMatch are explained in the following.

1) Pseudo-Label Selection Semi-supervised approaches use model confidence as measured via the maximal prediction probability [33] or alternative uncertainty metrics [1] as selection criteria. Model confidence assumes uni-modal model predictions, i.e. the model is confident that sample x_i belongs to class $\hat{y}_i = l$. In contrast to pseudo-labeling, we do not need the information of whether one sample x_i is confidently predicted to be in class $\hat{y}_i = l$ but the information of whether samples x_i and x_j are predicted to be in the same or a different cluster. Filtering for model confidence de-selects multi-modal model predictions that would, for instance, be assigned to two clusters with high probability each - pseudo-constraining allows us to use such multi-modal predictions.

Given a batch of model predictions over weakly-augmented, unconstrained samples, we aim to select those that are important for the subsequent pseudo-constraint generation. We propose measuring such *informativeness* of a probability vector $\hat{y}_i$ using the *normalized entropy*:

$$\mathcal{H}_n(\hat{y}_i) = -\frac{1}{\log(n_{\text{out}})} \sum_{l=1}^{n_{out}} p(\hat{y}_{il}) \log(p(\hat{y}_{il})) \qquad (1)$$

with $\mathcal{H}_n(\hat{y}_i) \in [0; 1]$ where $\mathcal{H}_n(\hat{y}_i) = 1$ describes the minimum level of information and maximal entropy and $\mathcal{H}_n(\hat{y}_i) =$

0 the maximum level of potential informativeness and minimal entropy where the model places the entire probability mass in one cluster. Hence, we use the normalized entropy in combination with a threshold hyperparameter $\tau \in [0;1]$ as criterion to select pseudo-labels:

$$\mathcal{I}_{\tau}^{sel}(\hat{y}_i) = \mathbb{1}(\mathcal{H}_n(\hat{y}_i) < \tau) \tag{2}$$

where $\mathbb{1}$ is an indicator function. In the experiment section, we provide an empirical analysis of the suitability of this criterion next to a sensitivity analysis of τ.

2) Pseudo-Constraining Confirmation bias is a critical problem in pseudo-labeling methods [1], and in constraint-based clustering, there is an opportunity to alleviate this. As an illustration, refer to the two unconstrained samples x_i^u, x_j^u from Fig. 1: the true label y_i of x_i^u would be "sports car" while the model wrongly but confidently assigns it to the "airliner" cluster and similarly x_j^u is assigned the wrong cluster "soccer ball" instead of the true y_j "maltese dog" (see Fig. 6 in the Appendix for more examples). While this prediction error would lead to a wrong prediction target in naive pseudo-labeling and hence confuse model training, the resulting pseudo-constraint $\tilde{c}_{ij} = 0.008$ would still be correctly assigned as Cannot-Link, as $\tilde{y}_j \neq \tilde{y}_i$ in both situations. Therefore, we create pseudo-constraints from the pseudo-labels to drive the loss function $\mathcal{L}^{pseudo}$. Given a batch of informative pseudo-labels, we next combine pseudo-label pairs $\tilde{y}_i, \tilde{y}_j$ into pseudo-constraints $\tilde{c}_{ij}$, expressing the (dis)-similarity of those samples. As $\tilde{y}_i, \tilde{y}_j$ are probability vectors, we propose to use a divergence measure to quantify this distance and derive a meaningful pseudo-constraint. The Jensen-Shannon-Distance [25] allows the symmetric mapping of two probability vectors onto a similarity score:

$$JSD(\tilde{y}_i, \tilde{y}_j) = \sqrt{((KL(\tilde{y}_i|m) + KL(\tilde{y}_j|m))/2} \tag{3}$$

where $m = (\tilde{y}_i + \tilde{y}_j)/2$ and $KL(y_i|m)$ refers to the Kullback-Leibler Distance between $\hat{y}_i$ and m and $JSD(\tilde{y}_i, \tilde{y}_j) \in [0,1]$. We exploit this property and use the inverse Jensen-Shannon-Distance to calculate soft pseudo-constraints $\tilde{c}_{ij} = 1 - JSD(\tilde{y}_i, \tilde{y}_j) \in [0;1]$ where $\tilde{c}_{ij} = 0.0$ resembles a Cannot-Link and $\tilde{c}_{ij} = 1.0$ a Must-Link pseudo-constraint over all pairwise combinations of the informative pseudo-labels. We refer to this inverse Jensen-Shannon-Distance as $\mathcal{I}^{map}(\tilde{y}_i, \tilde{y}_j)$ in Fig. 2. Pseudo-constraints are generated over the combined batch $\mathcal{B} = \mathcal{B}^c \cup \mathcal{B}^u$, treating the samples in $\mathcal{B}^c$ as unconstrained.

3) Pairwise Loss Function There exists a variety of loss functions that can deal with pairwise constraints [47] with the KCL [14] and the MCL [16] being the most prominent ones. Following the findings of Hsu et al. [16] and guided by preliminary experimental results, we propose the use of the MCL as a pairwise loss function, as it was shown to result in higher model performance, and smoother model training, and is hyperparameter-free. The MCL is aligned on the binary cross-entropy loss and follows the definition:

$$\mathcal{L}(c_{ij}, \hat{c}_{ij}) = -\sum_{ij} c_{ij} \log(\hat{c}_{ij}) + (1 - c_{ij}) \log(1 - \hat{c}_{ij}) \tag{4}$$

where $\hat{c}_{ij} = \langle \hat{y}_i, \hat{y}_j \rangle$ combines the individual predicted cluster assignment vectors into an alignment score and $c_{ij} \in \{0,1\}$ refers to the pairwise constraint $c_{ij} = 0$ for Cannot-Link and $c_{ij} = 1$ for Must-Link constraints. The MCL allows training with soft constraints $c_{ij} \in [0,1]$, similar to the use of soft labels in the cross-entropy loss [27]. We use this property for the processing of soft pseudo-constraints $\tilde{c}_{ij}$ as explained above and analyzed further in the experiments section. ConstraintMatch is trained on the combined loss:

$$\begin{aligned} \mathcal{L} = & \sum_{x_i, x_j \in \mathcal{B}^c} \mathcal{L}^{cons}\left(c_{ij}, \langle f(a(x_i)), f(a(x_j)) \rangle\right) + \\ & \lambda \sum_{x_i, x_j \in \mathcal{B}} \mathcal{L}^{pseudo}\left(\tilde{c}_{ij}, \langle f(A(x_k)), f(A(x_l)) \rangle\right) \end{aligned} \tag{5}$$

where hyperparameter $\lambda \geq 0$ controls the impact of the pseudo-constraint loss and is tuned on the validation set.

IV. Experiments

In this section, we compare the performance of ConstraintMatch with prior work on five challenging benchmark datasets and provide empirical evidence for the effectiveness of pseudo-constraining. This includes i) the relative improvement of ConstraintMatch across various baselines, ii) an empirical analysis of the benefit of pseudo-constraining and iii) its robustness w.r.t annotation noise, iv) analyses of the algorithmic choices made for its several components, and v) an evaluation of ConstraintMatch in the overclustering scenario.

A. Experimental Setup

Datasets and Constraint Mining We use the Cifar10 [19], Cifar100 [19], STL10 [6], ImageNet-10 [4] and ImageNet-Dogs [4] datasets to demonstrate the effectiveness of the proposed method. We use the 20 superclasses in Cifar100 as ground truth labels for constraint mining following prior work [24], [40], declaring it as Cifar100-20 in the following. We adhere to the provided train/test splits to enable comparison with prior work evaluated on separate test datasets [16], [40]. Recent deep clustering approaches instead are evaluated on the training set or the union of training and test datasets [24], [30], [38]. To be comparable with both bodies of literature, we provide benchmark results in both settings marked as "Test" and "Train(+Test)" in Table I following [24]. An overview of the used datasets, the amount of sampled constraints as well as the training and validation splits for hyperparameter tuning is provided in Table V in the Appendix. For constraint-sampling, we randomly sample n_c constraints from each dataset via the following procedure: n_c samples are randomly sampled without replacement as constraint members (x_i^c, y_i) and for each of those samples, a second pair member (x_j^c, y_j) is randomly chosen with replacement from the remaining training samples to create the constraint pair (x_i^c, x_j^c, c_{ij}),

TABLE I: Comparison of ConstraintMatch with relevant baselines (B), competitors (C), and upper bound models (U) across datasets and varying amounts of constraints n_c. Performance metrics were averaged over five folds and calculated on separate test splits in the upper part and in the Train(+Test) setting in the lower part. Best results comparing ConstraintMatch with the baseline and competitor models are shown in bold and † denotes values reported in the literature. Statistical significance for differences in model performance between ConstraintMatch and the constrained competitor for $n_c \in \{5\text{k}, 10\text{k}\}$ respectively established using the Wilcoxon signed-rank test [7], [43] (significance code $^* : p < 0.05$).

				Cifar 10			Cifar 100-20			STL 10			ImageNet 10			ImageNet Dogs		
Split	Model		n_c	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
Test	Supervised†	U		93.80	86.20	87.00	80.00	68.00	63.20	80.60	65.90	63.10	-	-	-	-	-	-
	Fully Constrained	U		94.86	88.39	89.11	77.99	68.37	61.94	90.49	80.94	80.47	96.64	93.45	92.60	67.10	73.52	58.13
	SCAN† [40]	B	0	87.60	78.70	75.80	45.90	46.80	30.10	76.70	68.00	61.60	86.20	81.57	75.71	47.20	**55.42**	35.87
	Constrained	C	5k	90.12	80.52	80.02	50.99	46.23	32.63	85.90	74.62	72.51	93.12	87.43	85.49	44.08	43.27	28.92
	ConstraintMatch		5k	92.23*	84.64*	84.27*	54.19*	52.74*	37.84*	88.20*	78.20*	76.68*	94.68*	90.43*	88.61*	49.43*	55.23*	38.12*
	Constrained	C	10k	90.89	81.73	81.47	52.45	46.57	33.79	88.21	77.63	76.38	94.90	90.42	89.04	45.52	44.44	30.10
	ConstraintMatch		10k	**93.17***	**85.88***	**85.92***	**57.15***	**54.37***	**40.59***	**90.08***	**80.57***	**79.81***	**95.68**	**92.09**	**90.70**	**50.73***	54.92*	**38.34***
Train	PICA† [17]	B	0	69.60	59.10	51.20	33.70	31.00	17.10	71.30	61.10	53.10	87.00	80.20	76.10	35.20	35.20	20.10
(+Test)	MICE† [38]	B	0	83.50	73.70	69.80	44.00	43.60	28.00	75.20	63.50	57.50	-	-	-	43.90	42.30	28.60
	CC† [24]	B	0	79.00	70.50	63.70	42.90	43.10	26.60	85.00	76.40	72.60	89.30	85.90	82.20	42.90	44.50	27.40
	TCC† [30]	B	0	90.60	79.00	73.30	49.10	47.90	31.20	81.40	73.20	68.90	89.70	84.80	82.50	**59.50**	**55.40**	**41.70**
	SCAN [40]	B	0	88.53	80.09	77.72	50.67	47.72	33.07	81.28	70.15	65.22	91.63	84.00	82.93	44.06	45.09	30.75
	Constrained	C	5k	91.14	82.30	82.05	51.63	46.58	33.37	80.51	68.38	63.68	95.09	88.42	89.49	43.25	38.82	28.93
	ConstraintMatch		5k	92.67*	85.12*	84.98*	54.16*	52.68*	37.79*	82.97*	71.13*	67.80*	95.61*	89.64*	90.59*	47.63*	47.82*	35.95*
	Constrained	C	10k	92.21	83.07	84.08	53.13	47.20	34.86	89.90	80.12	79.49	96.47	91.18	92.36	44.17	40.07	30.13
	ConstraintMatch		10k	**93.61***	**86.55***	**86.80***	**57.18***	**53.37***	**40.30***	**91.30***	**82.42***	**82.13***	**96.68***	**91.59***	**92.80***	49.34*	49.16*	37.37*

where $c_{ij} = 1$, if $y_i = y_j$ and $c_{ij} = 0$, if $y_i \neq y_j$. This results in a dataset $\mathcal{D}^c$ of n_c constrained samples (x_i^c, x_j^c, c_{ij}). To account for randomness in the constraint sampling process, we report performance averaged over five random sampling repetitions.

Implementation Details In accordance with prior work [40], we used a ResNet-18 backbone architecture [12] for the experiments with the Cifar10, Cifar100-20, and STL10 datasets and a ResNet-34 backbone [12] following [24] for the ImageNet datasets. We used model weights that were pretrained via SCAN [40] for the initialization of the model backbone as ConstraintMatch benefits from expressive feature representations as a warm start. Next to the model weights released by [40] for Cifar10, Cifar100-20, and STL10 we used the authors' codebase[2] to pretrain the ResNet-34 backbone via SCAN and then used these resulting model weights for model initialization. For model training, we used a standard SGD optimizer with momentum set to 0.9 and weight decay regularization [36] and all models were trained for a total of 20000 optimization steps unless noted otherwise. We used a cosine learning rate scheduler [26] which updates the learning rate at each update step to $\eta \cos\left(\frac{7\pi t}{16T}\right)$ with η being the initial learning rate, t the current training step and $T = 20000$ the total amount of training steps following [33]. Hyperparameters were tuned via a grid search on constraints mined from the validation datasets with hyperparameter ranges shown in Table IV and more details on the validation splits are given in Table V in the Appendix. The size of constrained/unconstrained batches was set to $200/600$ respectively for Cifar10 and Cifar100-20 and to $100/300$ for the other datasets.

[2]https://github.com/wvangansbeke/Unsupervised-Classification

Model Comparison We compare ConstraintMatch with different baselines (B), competitors (C), and upper bound models (U). This includes deep clustering models SCAN [40], TCC [30], CC [24], MICE [38] and PICA [17] as baselines and a constrained clustering model that was trained on D^c using the MCL [16] as competitor. As upper bounds, we compare with a fully constrained clustering model trained on a fully constraint version of training dataset D and a supervised baseline where the backbone was trained on the fully labeled training set D as reported by [40]. The authors of MICE [17], PICA [38], CC [24] and TCC [30] used a ResNet-34 backbone for the Cifar10, Cifar100-20 and STL10 datasets. We used a ResNet-18 backbone for these three datasets in adherence with SCAN [40]. A comparison with the semi-constrained approaches was not possible due to a lack of open-source code and performance metrics on established benchmarks.

B. Results

We summarize our main results in Table I measuring model performance in Accuracy (ACC), Normalized Mutual Information (NMI) [35] and the Adjusted Rand Index (ARI) [34], as standard in (constrained) clustering [47]. As established in the literature [14], [30], [40], we use the Hungarian Assignment method to optimally map the resulting cluster predictions to the true cluster labels [20]. As expected and shown in previous work [14], [16], we find training with pairwise constraints to be a valid option to train strong-performing clustering models. This benchmark is the first attempt to compare SCAN with subsequent deep clustering methods on the union of train and test datasets showing that SCAN is competitive with those

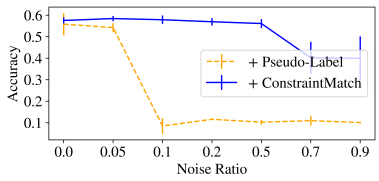


Fig. 3: Robustness of the pseudo-labeling baseline and ConstraintMatch towards pseudo-label noise.

methods (lower part of Table I). Further, fine-tuning SCAN via a subset of constraints improves model performance across all datasets but the fine-grained ImageNet-Dogs dataset. Overall, ConstraintMatch outperforms the (un-)constrained baselines and competitors across all datasets in both evaluation settings except ImageNet-Dogs, a task with semantically very similar classes, where it falls behind TCC in the Train(+Test) evaluation. This fine-grainedness makes training of constrained clustering models challenging, as the distinction between Must- and Cannot-link loses expressiveness which also explains the worse performance of the constrained competitor compared to SCAN. We interpret the finding that ConstraintMatch, in turn, outperforms both models by substantial margins in ACC and ARI as further proof of the effectiveness of pseudo-constraining. Relative (absolute) performance gains are the largest for Cifar100-20 with ConstraintMatch increasing model performance over the constrained baseline with 10k constraints by 8.96% (4.70 percentage points) Accuracy, 16.75% (7.80pp) NMI and 20.12% (6.80pp) ARI on the test dataset. We attribute these large performance gains to the complexity of the task and the efficient use of pseudo-constraints in this complex 20-cluster setting. Further, both the constrained competitor and ConstraintMatch benefit from more constraints n_c, with a larger relative performance increase for ConstraintMatch. We yield a statistically significant difference in model performance for ConstraintMatch and the constrained competitor across all datasets for all performance metrics using a Wilcoxon signed-rank test [7], [43] ($p < 0.05$) for $n_c \in \{5\text{k}, 10\text{k}\}$. Those empirical results confirm ConstraintMatch as a suitable method for semi-constrained clustering.

C. The Empirical Case for Pseudo-Constraints

We propose pseudo-constraining to overcome *confirmation bias*. To support this claim, we conducted a simulation experiment to evaluate the robustness of naive pseudo-labeling against confirmation bias in comparison to subsequent pseudo-constraining within ConstraintMatch. This naive pseudo-labeling baseline differs from ConstraintMatch in the handling of unconstrained samples, similar to the processing of unlabeled data in FixMatch [33]: weakly augmented, unconstrained samples are selected via a confidence threshold over their predicted cluster assignment and the major predicted cluster is chosen as pseudo-label. Predictions over strong augmented versions of these samples then serve as input for an auxiliary cross-entropy loss function (see Fig. 7 in the Appendix). We introduce a mode-flip function $m(\hat{y}_i)$ that swaps the position of the two largest predicted probabilities within the model prediction $\hat{y}_i$. This simulates a prediction error where the model "confuses" two cluster assignments within the pseudo-labeling of x_i. We randomly apply $m()$ to a noise fraction ρ of the unconstrained samples $x_i \in \mathcal{B}^u$ and train both models in this setting. The results in Fig. 3 confirm our intuition as naive pseudo-labeling already degrades at $\rho \geq 0.1$ while ConstraintMatch can cope with $\rho \leq 0.5$.

TABLE II: Ablation study on ConstraintMatch, results averaged over 5 folds with $n_c = 10000$.

Model	Test Performance		
	ACC	NMI	ARI
SCAN	45.90	46.80	30.10
+ Constrained	52.45	46.57	33.79
+ Pseudo-Labeling	55.38	53.49	39.98
+ Pseudo-Constraining	**57.15**	**54.37**	**40.59**

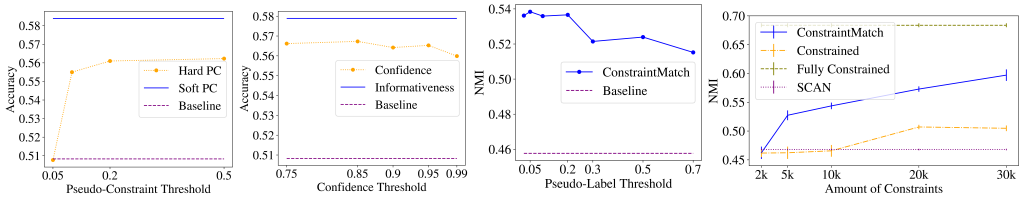
Pseudo-constraining further allows to use the same pairwise loss function as both the auxiliary and the initial objective for model training. We provide an ablation study on Cifar100-20 to quantify this benefit where we subsequently add constrained training, naive pseudo-labeling, and finally pseudo-constraining to the SCAN model, each with fine-tuned hyperparameters. The results in Table II show that while the use of naive pseudo-labeling leads to a substantial performance gain over the constrained baseline, the subsequent application of pseudo-constraining within ConstraintMatch enables further model improvements. We conclude its effectiveness is grounded in both the robustness w.r.t. the confirmation bias and the similarity in training objectives.

D. Additional Analyses

In this section, we analyze the several components of ConstraintMatch. Unless noted otherwise, these analyses and experiments were run on the Cifar100-20 dataset with $n_c = 10000$ using the optimal hyperparameters obtained for the main experiments and we report results on the test splits.

Pseudo-Constraining We use soft pseudo-constraints $\tilde{c}_{ij} \in [0, 1]$ in ConstraintMatch. One alternative would be the separation into hard pseudo-constraints $\tilde{c}^h_{ij} \in \{0, 1\}$ using a threshold μ such that $\hat{c}^h_{ij} = 1$ if $c_{ij} \geq \mu$ and $\hat{c}^h_{ij} = 0$ if $c_{ij} < \mu$. We find that while ConstraintMatch is still outperforming the constrained baseline using hard pseudo-constraints, it benefits further from the use of soft pseudo constraints as shown in Fig. 4a over different values of μ. Using soft constraints also eliminates the need to tune μ. We hypothesize that the model can effectively use the continuous information provided via the soft pseudo-constraints within the MCL loss, similar to the use of soft labels in supervised classification [27].

Pseudo-Label Selection We argue for the selection of *informative* samples as pseudo-labels over that of *confident* samples. Fig. 4b contrasts the use of both with a fixed



(a) $\mathcal{I}^{map}$: soft vs. hard pseudo-constraints.

(b) $\mathcal{I}^{sel}$: informativeness vs. confidence.

(c) $\mathcal{I}^{sel}$: impact of τ

(d) Effect of the amount of constraints n_c.

Fig. 4: Further analyses of ConstraintMatch.

value $\tau = 0.2$ for the informativeness criterion and with varying thresholds for confidence-based selection showing that informative samples enable ConstraintMatch to leverage unconstrained samples more effectively. Further, we provide a sensitivity analysis of the threshold τ within $\mathcal{I}^{sel}_{\tau}$ in Fig. 4c. This reveals that the sensitivity of ConstraintMatch towards τ lies within a reasonable margin and we recommend $\tau \in [0.025, 0.2]$ as a range for tuning.

Amount of Constraints Fig. 4d shows the effect of the amount of constraints n_c on the model performance as measured in NMI over five folds. The constrained clustering competitor model performs worse than the SCAN baseline for $n_c \leq 10000$ which might be due to the fact that $n_c = 10000$ results in 500 Must-Link constraints only in the Cifar100-20 scenario, allowing the MCL loss to overfit those few pairs quickly. In contrast to that, ConstraintMatch successfully overcomes this issue via its pseudo-constraining mechanism for $n_c \geq 5000$ with relative gains increasing for increasing n_c. The comparably low performance of ConstraintMatch for $n_c = 2000$ indicates that it still requires a certain degree of supervision to produce reliable pseudo-constraints.

Robustness w.r.t Noisy Constraints Next to the robustness of ConstraintMatch over noisy pseudo-labels, we further investigate its robustness towards noise in the annotation of the known ground truth constraints. This simulates the situation where the annotators might erroneously flip the constraint annotation, similar to the concept of label noise in supervised classification [13]. Therefore, we randomly flipped a varying percentage of the known constraint annotations and compared the effect of this annotation noise on model training. The results in Fig. 5 show that ConstraintMatch is more robust towards higher levels of annotation noise than the constrained competitor. We attribute this increased robustness to the stabilizing effect of the pseudo-constraining mechanism.

E. Overclustering

We further evaluate ConstraintMatch for overclustering, where the true amount of clusters K is unknown and the model can assign more clusters than inherently present in the data, $n_{out} \gg K$ [14]. Therefore, we compare ConstraintMatch with the constrained competitor and the SCAN baseline with $n_{out} = 5K$ resulting in 100 potential clusters for Cifar100-20 and 50 for Cifar10. Models were again evaluated using the Hungarian Assignment [20] with cluster predictions that do not match a corresponding ground truth cluster counting as an error. As shown in Table III, we find that the constrained competitor achieves strong performance gains over the unsupervised baseline despite the challenging learning task. Further, we find that the performance gains of ConstraintMatch translate well to this overclustering scenario yielding a relative (absolute) performance gain over the constrained competitor of 18.11% (8.03pp) NMI and 10.75% (4.24pp) Accuracy on Cifar100-20.

TABLE III: Overclustering results averaged over five folds.

Dataset	Cifar-10			Cifar 100-20		
n_{out}	50			100		
Metric (Test)	ACC	NMI	ARI	ACC	NMI	ARI
SCAN	34.68	61.56	34.52	29.88	47.35	23.23
Constrained	82.24	75.40	73.80	39.41	44.34	27.86
ConstraintMatch	**88.89**	**83.04**	**82.06**	**43.65**	**52.37**	**34.88**

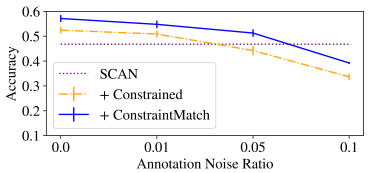


Fig. 5: Impact of ground truth constraint annotation noise.

V. Conclusion

ConstraintMatch is a novel method for training clustering models in a semi-constrained setting, using a combination of large amounts of unconstrained data and a limited number of constraint pairs. Therefore, it selects informative pseudo-labels processed within a pseudo-constraining mechanism that allows training the model on a unified loss function to overcome the limitations of naive pseudo-labeling in this setting. With empirical results across five benchmarks, we demonstrate ConstraintMatch's strong performance, outperforming baselines and competitors by substantial margins, even in challenging overclustering scenarios. We furthermore analyzed its several components, supporting our algorithmic choices with empirical evidence, and empirically showed that pseudo-constraining

leads to increased model robustness towards different sources of annotation noise.

While initially designed and evaluated on top of SCAN [40], ConstraintMatch is pre-training-agnostic, and hence alternative unsupervised pre-training methods would also be applicable. Further combining the pseudo-constraining mechanism with semi-supervised classification or object detection approaches would be an interesting avenue for future research.

References

[1] E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[2] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel. Mixmatch: A holistic approach to semi-supervised learning. *NeurIPS*, 32, 2019.

[3] M. Caron, P. Bojanowski, A. Joulin, and Ma. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[4] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan. Deep adaptive image clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5879–5887, 2017.

[5] 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, 2020.

[6] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[7] D. Janez. Statistical comparisons of classifiers over multiple data sets In *Journal of Machine Learning Research*, pages 1–30. JMLR, 2006.

[8] S. Fogel, H. Averbuch-Elor, D. Cohen-Or, and J. Goldberger. Clustering-driven deep embedding with pairwise constraints. *IEEE Computer Graphics and Applications*, 39(4):16–27, 2019.

[9] P. Gançarski, T. Dao, B. Crémilleux, G. Forestier, and T. Lampert. Constrained clustering: Current and new trends. In *A Guided Tour of Artificial Intelligence Research*, pages 447–484. Springer, 2020.

[10] X. Guo, L. Gao, X. Liu, and J. Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017.

[11] 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.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[13] M. Hedderich , D. Zhu, and D. Klakow. Analysing the noise model error for realistic noisy label data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7675–7684, 2021

[14] Y. Hsu and Z. Kira. Neural network-based clustering using pairwise constraints. *International Conference on Learning Representations Workshop*, 2016.

[15] Y. Hsu, Z. Lv, and Z. Kira. Learning to cluster in order to transfer across domains and tasks. *International Conference on Learning Representations*, 2018.

[16] Y. Hsu, Z. Lv, J. Schlosser, P. Odom, and Z. Kira. Multi-class classification without multi-class labels. *International Conference on Learning Representations*, 2019.

[17] J. Huang, S. Gong, and X. Zhu. Deep semantic clustering by partition confidence maximisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8849–8858, 2020.

[18] X. Ji, J. Henriques, and A. Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9865–9874, 2019.

[19] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Advances in Neural Information Processing Systems*, 2009.

[20] H. Kuhn. The Hungarian Method for the Assignment Problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[21] C. Kuo, C. Ma, J. Huang, and Z. Kira. Featmatch: Feature-based Augmentation for Semi-supervised Learning. In *Proceedings of the European conference on computer vision (ECCV)*. Springer, 2020.

[22] D. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, International Conference on Machine Learning*, page 896, 2013.

[23] F. Li, H. Qiao, and B. Zhang. Discriminatively boosted image clustering with fully convolutional auto-encoders. *Pattern Recognition*, 83:161–173, 2018.

[24] Y. Li, P. Hu, Z. Liu, D. Peng, J. Zhou, and X. Peng. Contrastive clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[25] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[26] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations*, 2017.

[27] C. Meister, E. Salesky, and R. Cotterell. Generalized entropy regularization or: There's nothing special about label smoothing. *ACL*, 2020.

[28] F. Nie, H. Zhang, R. Wang, and X. Li. Semi-supervised clustering via pairwise constrained optimal graph. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3160–3166, 2021.

[29] C. Niu and G. Wang. Spice: Semantic pseudo-labeling for image clustering. *arXiv preprint arXiv:2103.09382*, 2021.

[30] Y. Shen, Z. Shen, M. Wang, J. Qin, P. Torr, and L. Shao. You never cluster alone. *Advances in Neural Information Processing Systems*, 34, 2021.

[31] A. Shukla, G. Cheema, and S. Anand. Semi-supervised clustering with neural networks. In *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, pages 152–161. IEEE, 2020.

[32] M. Śmieja, L. Struski, and M. Figueiredo. A classification-based approach to semi-supervised clustering with pairwise constraints. *Neural Networks*, 127:193–203, 2020.

[33] K. Sohn, D. Berthelot, C. Li, Z. Zhang, N. Carlini, E. Cubuk, A. Kurakin, H. Zhang, and C. Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *CoRR*, abs/2001.07685, 2020.

[34] D. Steinley. Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3):386, 2004.

[35] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

[36] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, pages 1139–1147. PMLR, 2013.

[37] K. Tian, S. Zhou, and J. Guan. Deepcluster: A general clustering framework based on deep learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 809–825. Springer, 2017.

[38] T. Tsai, C. Li, and J. Zhu. Mice: Mixture of contrastive experts for unsupervised image clustering. In *International Conference on Learning Representations*, 2021.

[39] J. Van Engelen and H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

[40] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pages 268–285. Springer, 2020.

[41] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. *Proceedings of the AAAI Conference on Artificial Intelligence*, 1097:577–584, 2000.

[42] K. Wagstaff, C. Cardie, S. Rogers and S. Schrödl. Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning*, volume 1, pages 577–584, 2001.

[43] F. Wilcoxon. Individual comparisons by ranking methods. In *Biometrics*, volume 1, pages 80–83, 1945.

[44] J. Wu, K. Long, F. Wang, C. Qian, C. Li, Z. Lin, and H. Zha. Deep comprehensive correlation mining for image clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8150–8159, 2019.

[45] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016.

[46] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, volume 34, pages 18408–18419.

[47] H. Zhang, T. Zhan, S. Basu, and I. Davidson. A framework for deep constrained clustering. *Data Mining and Knowledge Discovery*, 35(2):593–620, 2021.

APPENDIX

A. Hyperparameter Tuning

We conducted a grid search over the validation splits detailed in Table V for one fold of sampled training constraints for hyperparameter tuning. We used the validation loss on the constraints from the validation splits to select the optimal hyperparameters for each dataset and model combination with the lowest final validation loss as performance criterion. Final models were then trained on these optimal hyperparameters on five repeated folds of the respective constrained and unconstrained training samples and final performance metrics were reported for both the "Test" and the "Train(+Test)" settings. The *shared* parameters were used in and tuned for all trained models and the specific hyperparameters for ConstraintMatch and the naive pseudo-labeling baseline were tuned over a grid of different values, see Table IV.

TABLE IV: Hyperparameters and their respective values considered in the grid search for the different models.

Parameter	Search Values
Shared	
Weight decay	$0.001, 0.0001, 0.00001$
Learning rate	$0.03, 0.01, 0.003, 0.001, 0.0001$
naive Pseudo-labeling	
λ	$1.0, 0.5, 0.1, 0.05$
τ	$0.7, 0.8, 0.9, 0.95, 0.99$
ConstraintMatch	
λ	$1.0, 0.5, 0.1, 0.05$
τ	$0.05, 0.1, 0.2, 0.3$

B. Data Augmentation

ConstraintMatch follows the rationale of consistency regularization via weak and strong augmentations $a()$ and $A()$. As weak augmentations $a()$, we used random cropping and horizontal flipping. For strong augmentations, $A()$, we used the RandAugment strategy with the data augmentation procedures used in FixMatch and described in Appendix D of [33].

C. Datasets

Table V provides an overview of the datasets used in the experimental section IV-A alongside their splits and sizes. The final column describes the exact dataset splits that were used in the Train(+Test) evaluation setting following [24].

D. Visualization of the Confirmation Bias

In Fig. 6, we visualized four samples from the unconstrained part of the ImageNet-10 dataset which suffer from the confirmation bias similar to Fig. 1, i.e. samples for which the model confidently predicted the wrong cluster assignment. These unconstrained samples were selected as high-confidence (max. predicted probability > 0.98) but wrongly predicted examples. We can observe that pseudo-labeling would lead to wrong prediction targets (e.g. cluster 'Airship' instead of the true cluster 'Soccer Ball' in the bottom left example) and hence confuse model training. On the other hand, pseudo-constraints generated on top of pairs of these wrongly assigned pseudo-labels still are semantically correct and can support model training on these unconstrained samples. This does not only hold for Cannot-Link (bottom) but also for Must-Link (top) pseudo-constraints where both samples with the same true cluster affiliation are assigned the same wrong cluster by the model. These samples were selected from a random batch of unconstrained samples $\mathcal{B}^u$ from the ImageNet-10 dataset from ConstraintMatch trained for 500 training steps with a ResNet-34 backbone.

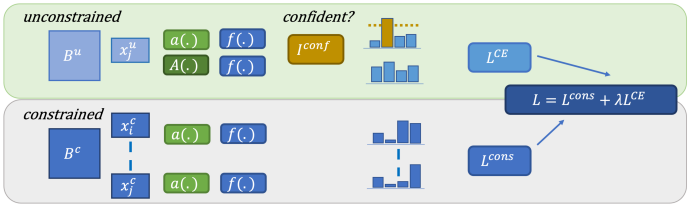
TABLE V: Datasets used in the experiments including the respective training, validation, and test splits. We also mention the evaluation dataset for the Train(+Test) setting in the last column following [24].

Dataset	K	Samples Train	Val	Test	Constraints Train	Val	Train(+Test)
Cifar10	10	45k	5k	10k	5/10k	10k	Train + Test
Cifar100-20	20	45k	5k	10k	5/10k	10k	Train + Test
STL10	10	4k	1k	8k	5/10k	5k	Train + Test
ImageNet-10	10	12k	1k	500	5/10k	1k	Train
ImageNet-Dogs	15	18.5k	1k	750	5/10k	1k	Train

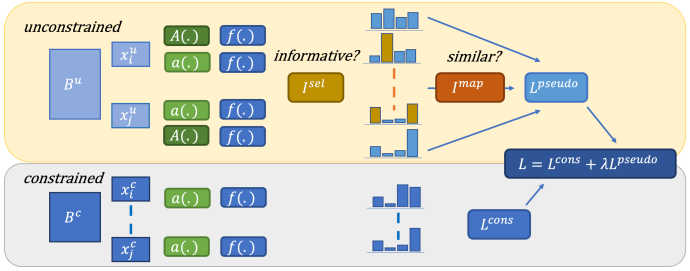
Fig. 6: Illustration of pseudo-labeling failure cases due to confirmation bias. Pseudo-constraints generated on top of these wrong pseudo-labels are still semantically correct.

E. Naive Pseudo-Labeling Baseline

In Fig. 7, we visualize the naive pseudo-labeling baseline, a simplified version of ConstraintMatch, with which we compared ConstraintMatch in the results Section IV-B. This baseline follows the use of unlabeled samples in FixMatch [33] and similarly leverages the weak-strong augmentation scheme for consistency regularization. Concretely, weakly augmented, unconstrained samples are selected via a confidence threshold over their predicted cluster assignments, and



(a) The naive pseudo-labeling baseline combines training on batches of pairwise constrained (gray) and individual unconstrained (green) samples leveraging weak and strong data augmentations a, A following the FixMatch approach [33].



(b) ConstraintMatch combines pairwise training on batches of constrained (gray) and unconstrained (yellow) samples leveraging weak and strong data augmentations a, A. It extends the naive pseudo-labeling baseline by the generation of pseudo-constraints from *informative* pseudo-labels to overcome the confirmation bias as detailed in the methods section of the paper.

Fig. 7: Illustration of a) the naive pseudo-labeling baseline and b) ConstraintMatch.

the predicted clusters with the highest assigned probability are subsequently chosen as pseudo-labels. This confidence-based selection criterion is depicted as $\mathcal{I}^{conf}$ in Fig. 7a and the associated threshold $\tau \in [0, 1]$ is a hyperparameter that we tuned on the validation set as described above and listed in Table IV. Predictions over strong augmented versions of these samples then serve as input for an auxiliary cross-entropy loss function, referred to as $\mathcal{L}^{CE}$ in Fig. 7a. Similar to ConstraintMatch, the constrained loss $\mathcal{L}^{cons}$ is calculated over model predictions on pairwise samples and their corresponding constraint annotations. The naive pseudo-labeling baseline is then trained via the final loss $\mathcal{L} = \mathcal{L}^{cons} + \lambda \mathcal{L}^{CE}$ as a weighted linear combination of both losses where hyperparameter λ controls the impact of the unconstrained samples.

4.3 Positive-unlabeled Learning with Uncertainty-aware Pseudo-Label Selection

Contributing article:

Emilio Dorigatti, Jann Goschenhofer, Benjamin Schubert, Mina Rezaei, and Bernd Bischl. 2022. Positive-unlabeled learning with uncertainty-aware pseudo-label selection. *arXiv preprint arXiv:2201.13192*

Author contributions:

Emilio Dorigatti was responsible for the conceptualization of the paper (i.e. idea, goal, and scope), supported by Jann Goschenhofer. The software accompanying this work was mainly developed and implemented by Emilio Dorigatti with support from Jann Goschenhofer who contributed to the general setup of the codebase, and the implementation of features as well as baseline models. The experimental design for the benchmark of the developed method (incl. hyperparameter tuning, data splits, choice of data sets) as well as the ablation studies was established with equal contributions from Emilio Dorigatti, Jann Goschenhofer, Benjamin Schubert, Mina Rezaei, and Bernd Bischl. The main experiments for the introduced method were run by Emilio Dorigatti while Jann Goschenhofer was responsible for the baseline experiments. Jann Goschenhofer and Emilio Dorigatti contributed equally to the analysis of the results and the story of the paper. Emilio led the writing of the manuscript with support from Jann Goschenhofer (i.e. editing, reviewing, and drafting). Emilio Dorigatti contributed the application example, with reviews and editing provided by Benjamin Schubert. Benjamin Schubert, Mina Rezaei, and Bernd Bischl contributed via proofreading and supervision and Bernd Bischl provided the computational resources for the experiments.

Copyright information:

Uncertainty-aware Pseudo-label Selection for Positive-Unlabeled Learning

Emilio Dorigatti[1,2,3*], Jann Goschenhofer[1,3,4], Benjamin Schubert[2,5], Mina Rezaei[1] and Bernd Bischl[1,3,4]

[1]Department of Statistics, Ludwig-Maximilians-Universität München, München, 80539, Germany.
[2]Institute of Computational Biology, Helmholtz Zentrum München—German Research Center for Environmental Health, Neuherberg, 85764, Germany.
[3]Munich Center for Machine Learning, München, Germany.
[4]Fraunhofer Institute for Integrated Circuits IIS, Erlangen, 91058, Germany.
[5]Department of Mathematics, Technical University of Munich, Garching bei München, 85748, Germany.

*Corresponding author(s). E-mail(s): edo@stat.uni-muenchen.de;
Contributing authors: jann.goschenhofer@stat.uni-muenchen.de;
benjamin.schubert@helmholtz-muenchen.de;
mina.rezaei@stat.uni-muenchen.de;
bernd.bischl@stat.uni-muenchen.de;

Abstract

Positive-unlabeled learning (PUL) aims at learning a binary classifier from only positive and unlabeled training data. Even though real-world applications often involve imbalanced datasets where the majority of examples belong to one class, most contemporary approaches to PUL do not investigate performance in this setting, thus severely limiting their applicability in practice. In this work, we thus propose to tackle the issues of imbalanced datasets and model calibration in a PUL setting through an uncertainty-aware pseudo-labeling procedure (*PUUPL*): by boosting the signal from the minority class, pseudo-labeling expands the labeled dataset with new samples from the unlabeled set, while explicit uncertainty quantification prevents

the emergence of harmful confirmation bias leading to increased predictive performance. Within a series of experiments, PUUPL yields substantial performance gains in highly imbalanced settings while also showing strong performance in balanced PU scenarios across recent baselines. We furthermore provide ablations and sensitivity analyses to shed light on PUUPL's several ingredients. Finally, a real-world application with an imbalanced dataset confirms the advantage of our approach.

Keywords: Uncertainty Quantification, Self-supervised Learning, Positive Unlabeled Learning, Imbalanced Data

1 Introduction

Many real-world applications involve positive-unlabeled (PU) datasets [1–3] in which only a few samples are labeled positive while the majority is unlabeled. PU learning (PUL) aims to learn a binary classifier in this challenging setting without any labeled negative examples, thus reducing the need for manual annotation and enabling entirely new applications where negative examples are costly or impossible to obtain [4]. Learning from PU data can reduce development costs in many deep learning applications that otherwise require costly annotations from experts or expensive experimental procedures such as medical image diagnosis [1] and protein function prediction [2]. PUL can even enable applications in settings where the measurement technology itself can not detect negative examples [3].

Many PUL applications share another intrinsic difficulty: class imbalance. Imbalanced settings arise when most samples in a dataset belong to the same class, and frequently the most interesting class happens to be the minority. In PUL, class imbalance refers specifically to a low class prior $\pi := p(y = 1)$ implying that the majority of the unlabeled samples are negatives. While this problem can be tackled in traditional (semi-)supervised learning by re-weighting the loss to increase the penalty of mis-classification of the minority class, a similar approach was introduced in PUL with some additional care in handling the unlabeled data points [5]. However, the issue remains in general under-studied in the literature and recent developments such as Self-PU [6] are solely targeted at balanced scenarios.

Motivated by this, we propose to tackle imbalancedness in PUL via pseudo-labeling [7], an iterative procedure that augments the labeled dataset with new samples from the unlabeled set, thus boosting the weak signal from the minority class. To prevent the emergence of harmful confirmation bias in this procedure, we propose to assign pseudo-labels based on likelihood-free uncertainty quantification via model ensembling [8]. By using soft targets we avoid artificially inflating the confidence of pseudo-labels and preserve the calibration signal for the ensemble in later training iterations, thus eventually obtaining a predictor that is both calibrated and well-performing. Another advantage

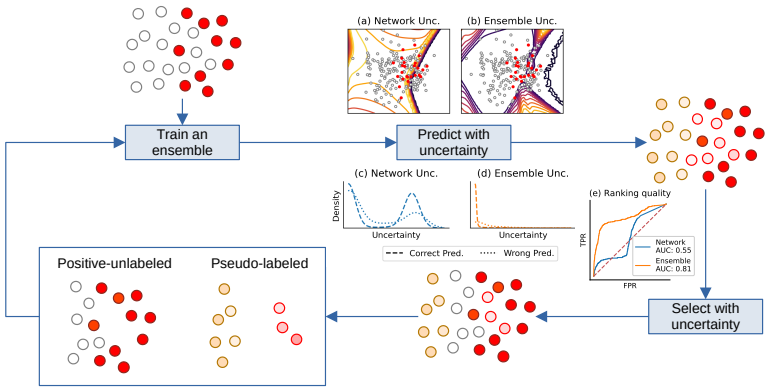


Fig. 1 *PUUPL* is a pseudo-labeling framework for PU learning that uses the epistemic uncertainty of an ensemble to select confident examples to pseudo-label. The ensemble can be trained with any PU loss for PU data while minimizing the cross-entropy loss on the previously assigned pseudo-labels. In a toy example, a single network is not very confident on most of the unlabeled data (a), resulting in many high-confidence incorrect predictions and many low-confidence correct ones (c). The epistemic uncertainty of an ensemble is, on the other hand, very low on most of the unlabeled data (b), resulting in most correct predictions having low uncertainty and most incorrect predictions having high uncertainty (d). Thus, the estimated uncertainty by ensemble can be used more reliably to rank predictions and select correct ones (e). Re-training the model with an increased number of labeled samples will result in a slightly more accurate model, than can be used to predict new pseudo-labels, which will further improve the model's performance, etc.

of pseudo-labeling is that it allows the model to harness the power of self-training without requiring modality-specific augmentations such as MixUp [9] that restrict most contemporary PUL methods [6, 10–13] to image data only.

To summarize, our contributions are:

1. We introduce *PUUPL* (Positive Unlabeled, Uncertainty aware Pseudo-Labeling), a novel framework that successfully overcomes the issue of imbalanced data distribution in PUL in a data-modality-agnostic framework while retaining competitive performance on balanced datasets.
2. We evaluate our methods on a wide range of benchmarks and PU datasets, achieving state-of-the-art results in self-training for PUL both with and without knowing the positive class prior π. Our results show that PUUPL is applicable to different data modalities such as images and text, can use any risk estimator for PUL and improve thereupon, and is robust to prior misspecification and class imbalance.
3. A real-world healthcare application confirms the advantage of PUUPL compared to other PUL methods as well as previous domain-specific state-of-the-art approaches.

These results demonstrate that our framework is highly reliable, extensible, and applicable in a variety of real-world scenarios.

2 Related work

Positive-unlabeled learning

PUL was introduced as a variant of binary classification [14] and is related to one-class learning [15, 16], multi-positive learning [17], multi-task learning [18], and semi-supervised learning [19]. Current existing methods for PUL can be divided into three branches: two-step techniques, class prior incorporation, and biased PUL [20]. In this work, we apply pseudo-labeling with biased PUL – also coined as reweighting methods – and refer to [20] for a comprehensive overview of the field. In this context, [21] introduced the unbiased risk estimator uPU. [4] showed that this loss function is prone to overfitting in deep learning contexts, as it lacks a lower bound, and proposed the non-negative risk estimator nnPU [4] as a remedy. Follow-up work on loss functions for PUL has focused on robustness w.r.t. biases in the sampling process [22–24] and handling of imbalanced datasets [5]. Further research in PUL focuses on estimating the class prior directly during training [10, 25, 26] or exploiting its knowledge to further improve the training process [6, 11–13].

Pseudo-labeling

Pseudo-labeling follows the rationale that the model leverages its own predictions on unlabeled data as pseudo-training targets to enable iterative semi-supervised model training. The major weakness of pseudo-labeling is that erroneously selected pseudo-labels can amplify errors during training, potentially leading to model degradation over time. This *confirmation bias* is grounded in poor model calibration which distorts the signal for the pseudo label selection [27]. Model calibration issues often occur in deep learning settings as deep neural networks are prone to over-confident predictions unless trained appropriately [28]. A variety of approaches were proposed for semi-supervised classification settings to mitigate this problem [29–34]. The commonality of these works is the explicit consideration of model uncertainty to improve pseudo-label selection, which motivates its application in the context of PUL. A first attempt to combine pseudo-labeling with PUL was made with Self-PU [6], where self-paced learning, a confidence-weighting scheme based on the model predictions, and a teacher-student distillation approach are combined. With *PUUPL*, we propose an alternative pseudo-labeling strategy for PUL that performs better in a simpler and more principled way using implicitly well-calibrated models to improve the pseudo-label selection. Moreover, uncertainty awareness allows *PUUPL* to work well in unbalanced data environments where Self-PU breaks down. To the best of our knowledge, we are the first to introduce an uncertainty-aware pseudo-labeling paradigm to PUL. Although our method shares the same motivation as that from [33] for semi-supervised classification with both positive and negative training samples, we differ in several important aspects dictated by the PUL setting: (1) we specifically target PU data with a PU loss, (2) we quantify uncertainty with an ensemble instead of Monte Carlo dropout, (3) we use epistemic uncertainty

instead of the predicted class probabilities for the selection, (4) we do not use temperature scaling and (5) use soft labels.

3 Method

PUUPL (Algorithm 1) separates the training set X^{tr} into the sets P, U, and L, which contain the initial positives, the currently unlabeled, and the pseudo-labeled samples, respectively. The set L is initially empty. At each pseudo-labeling iteration, we first train our model using all samples in P, U, and L until some convergence condition is met (Section 3). Then, model predictions over the samples in U are ranked w.r.t. their predictive uncertainty (Section 3), and samples with the most confident score are assigned the predicted label and moved into the set L (Section 3). Similarly, model predictions are derived for the samples in L, and the most uncertain samples are moved back to the unlabeled set U (Section 3). Next, the model is re-initialized to the same initial weights, and the next pseudo-labeling iteration starts.

Notation

Consider input samples X with label y and superscripts $\cdot^{tr}$, $\cdot^{va}$ and $\cdot^{te}$ for training, validation, and test data, respectively. The initial training labels y^{tr} are set to one for all samples in P and zero for all others in U. We group the indices of original positives, unlabeled, and pseudo-labeled samples in X^{tr} into the sets P, U, and L, respectively, where $y_i = 1$ for $i \in P$, $y_i = 0$ for $i \in U$, and y_i is the assigned pseudo-label for $i \in U$. Our proposed model is an ensemble of K deep neural networks whose random initial weights are collectively denoted as θ^0. The predictions of the k-th network for sample i are indicated with $\hat{p}_{ik} = \sigma(\hat{f}_{ik})$, with $\sigma(\cdot)$ as the logistic function and $\hat{f}_{ik}$ as the predicted logits. The logits and predictions for a sample averaged across the networks in the ensemble are denoted by $\hat{f}_i$ and $\hat{p}_i$, respectively. We subscript data and predictions with i to index individual samples, and use an index set in the subscript to index all samples in the set (e.g., $X^{tr}_U = \{X^{tr}_i | i \in U\}$ denotes the features of all unlabeled samples). We denote the total, epistemic, and aleatoric uncertainty of sample i as $\hat{u}^t_i$, $\hat{u}^e_i$, and $\hat{u}^a_i$, respectively.

Loss function

We train our proposed model with a loss function $\mathcal{L}$ that is a convex combination of a loss $\mathcal{L}_{PU}$ for the samples in the positive and unlabeled set $(P \cup U)$ and a loss $\mathcal{L}_L$ for the samples in the pseudo-labeled set (L):

$$\mathcal{L} = \lambda \cdot \mathcal{L}_L + (1 - \lambda) \cdot \mathcal{L}_{PU} \tag{1}$$

with $\lambda \in (0, 1)$. The loss $\mathcal{L}_L$ is the binary cross-entropy computed w.r.t. the assigned pseudo-labels y. Our method is agnostic to the specific PU loss $\mathcal{L}_{PU}$ used, allowing PUUPL to be easily adapted to provide further performance

Algorithm 1 The PUUPL Training Procedure

Hyperparameters:

- Loss mixing coefficient λ
- Number K of networks in the ensemble
- Maximum number T of pseudo-labels to assign at each round
- Maximum uncertainty threshold t_l to assign pseudo-labels
- Minimum uncertainty threshold t_u to remove pseudo-labels

Input: Train and validation $X^{tr}, y^{tr}, X^{va}, y^{va}$

$P \leftarrow$ indices of positive samples in X^{tr}
$U \leftarrow$ indices of unlabeled samples in X^{tr}
$L \leftarrow \emptyset$
$\theta^0 \leftarrow$ Random weight initialization
while not converged **do**
    Initialize model weights to θ^0
    Train an ensemble of K networks on X^{tr}, y^{tr}
    Update θ^* if performance on X^{va}, y^{va} improved
    $\hat{f} \leftarrow$ ensemble predictions for X^{tr}
    Compute epistemic uncertainty via Eq. 7
    $L^{\text{new}} \leftarrow$ Examples to pseudo-label via Eq. 8
    $U^{\text{new}} \leftarrow$ Examples to pseudo-unlabel
    $L \leftarrow L \cup L_b^{\text{new}} \setminus U^{\text{new}}$
    $U \leftarrow U \setminus L_b^{\text{new}} \cup U^{\text{new}}$
    $y_{L^{\text{new}}} \leftarrow \hat{p}_{L^{\text{new}}}$
    $y_{U^{\text{new}}} \leftarrow 0$
end while

increase in other scenarios for which a different PU loss might be more appropriate, e.g., when a set of biased negative samples is available [23], when coping with a selection bias in the positive examples [22] or an imbalanced class distribution [5] (see experiments). For the standard setting of imbalanced PUL, we use the imbnnPU loss [5]:

$$\mathcal{L}_{PU} = \pi' \ell(P, 1) + \max\left\{0, \frac{1-\pi'}{1-\pi}\ell(U, -1) - \frac{(1-\pi')\pi}{1-\pi}\ell(P, -1)\right\} \tag{2}$$

where $\pi = p(y = 1)$ is the prior probability that a sample is positive, π' the desired oversampled probability that we fix to $1/2$, and $\ell(S, y)$ the expected sigmoid loss of samples in the set S with label y:

$$\ell(S, y) = \frac{1}{\mathsf{S}} \sum_{i \in S} \frac{1}{1 + \exp(y \cdot \hat{p}_i)} \tag{3}$$

Similarly, we use use the non-negative correction *nnPU* of the PU loss [4] for the standard, balanced PU setting:

$$\mathcal{L}_{PU} = \pi \cdot \ell(P, 1) + \max\left\{0, \ell(U, -1) - \pi \cdot \ell(P, -1)\right\} \tag{4}$$

where $\pi = p(y = 1)$ is the prior probability that a sample is positive and $\ell(S, y)$ follows equation 3.

While π can be estimated from PU data [35], in our experimental results we treat π as a hyperparameter and optimize it without requiring negatively labeled samples via a PU validation set [36].

Model uncertainty

We utilize a deep ensemble with K networks with the same architecture, each trained on the full training dataset [8], to quantify the predictive uncertainty. Given the predictions $\hat{p}_{i1}, \ldots, \hat{p}_{iK}$ for a sample x_i, we associate three types of uncertainties to x_i's predictions [37]: the aleatoric uncertainty as the mean of the entropy of the predictions (Eq. 5), the total uncertainty as the entropy of the mean prediction (Eq. 6), and the epistemic uncertainty formulated as the difference between the two (Eq. 7).

$$\hat{u}_i^a = -\frac{1}{K} \sum_{k=1}^{K} [\hat{p}_{ik} \log \hat{p}_{ik} + (1 - \hat{p}_{ik}) \log(1 - \hat{p}_{ik})] \tag{5}$$

$$\hat{u}_i^t = -\hat{p}_i \log \hat{p}_i - (1 - \hat{p}_i) \log(1 - \hat{p}_i) \tag{6}$$

$$\hat{u}_i^e = \hat{u}_i^t - \hat{u}_i^a \tag{7}$$

where $\hat{p}_i = \sum_{k=1}^{K} \hat{p}_{ik}/K$. Epistemic uncertainty corresponds to the mutual information between the parameters of the model and the true label of the sample. Low epistemic uncertainty thus means that the model parameters would not change significantly if trained on the true label, suggesting that the prediction is indeed correct. The cumulative effect of many correct pseudo-labels added over time, however, provides a strong enough training signal to push the model towards better-performing parameters, as we show in the experimental results.

Pseudo-labeling

The estimated epistemic uncertainty (Eq. 7) is used to rank and select unlabeled examples for pseudo-labeling. Let $\rho(i)$ denote the rank of sample i. Then, the set L^{new} of newly pseudo-labeled samples is formed by taking the T samples with lowest uncertainty from U, ensuring that it is lower than a threshold t_l:

$$L^{\text{new}} = \{i \in U | \rho(i) \leq T \wedge u_i^e \leq t_l\} \tag{8}$$

Previous works on semi-supervised classification have shown that balancing the pseudo-label selection between the two classes – i.e., ensuring that the ratio of newly labeled positives and negatives is close to a given target ratio r – is beneficial [33]. In this case, the set L^{new} is partitioned according to the model's predictions into a set L_+^{new} of predicted positives and L_-^{new} of predicted negatives, and the most uncertain samples in the larger set are discarded to reach the desired ratio r, which we fix to 1. We then assign soft pseudo-labels,

i.e., the average prediction in the open interval $(0, 1)$, to these samples:

$$y_i = \hat{p}_i \quad \forall i \in L_-^{\text{new}} \cup L_+^{\text{new}} \tag{9}$$

As discussed previously, low epistemic uncertainty signals likely correct predictions. Using such predictions as a target in the loss $\mathcal{L}_L$ provides a stronger, more explicit learning signal to the model, resulting in a larger decrease in risk compared to using the same example as unlabeled in $\mathcal{L}_{PU}$. At the same time, soft pseudo-labels provide an additional signal regarding the estimated aleatoric uncertainty of samples. Furthermore, they help reduce overfitting and the emergence of confirmation bias in case the assigned pseudo-label is wrong by acting as dynamically-smoothed labels [31, 38].

Pseudo-unlabeling

Similar to the way that low uncertainty on an unlabeled example indicates that the prediction can be trusted, high uncertainty on a pseudo-labeled example indicates that the assigned pseudo-label might not be correct after all. To avoid training on such possibly incorrect pseudo-labels, we move the pseudo-labeled examples with uncertainty above a threshold t_u back into the unlabeled set:

$$U^{\text{new}} = \{i \in L | \hat{u}_i^e \geq t_u\} \tag{10}$$

$$y_i = 0 \quad \forall i \in U^{\text{new}} \tag{11}$$

4 Experiments

To empirically compare our proposed framework to existing state-of-the-art losses and models, we followed standard protocols for PUL [4, 6, 10, 22] as described in Section 4.1. In Section 4.2, after presenting the main results, we empirically show the advantage of our framework in improving performance for both imbalanced and standard PU scenarios, being applicable to different data modalities, and using various losses for PU learning. Finally, in Section 4.3 we provide further analyses of PUUPL including an investigation of its sensitivity with respect to pseudo-labeling hyperparameters. The source code of the method and all the experiments are available at https://anonymous.4open.science/r/PUUPL-BE6E.

4.1 Experimental protocol

Datasets

We evaluated our method in the standard setting of MNIST [39] and CIFAR-10 [40] datasets, as well as Fashion MNIST (F-MNIST) [41], CIFAR-100-20 [40] and IMDb [42] to show the applicability to different data modalities. Similar to previous studies [4, 6, 10, 22], positives were defined as odd digits in MNIST and vehicles in CIFAR-10. For F-MNIST we used trousers, coats, and sneakers as positives, and for the experiments on CIFAR-100-20, we defined those 10

out of the 20 superclasses as Positives that correspond to living creatures (i.e., 'aquatic mammal', 'fish', 'insects', 'large carnivores', 'large omnivores', 'medium-sized mammals', 'non-insect invertebrates', 'people', 'reptiles', 'small mammals'). The number of training samples is reported in Supplementary Table A1.

For all datasets, we reserved a validation set of 5,000 samples and used all other samples for training, evaluating on the canonical test set. To simulate an imbalanced setting, we downsampled the positives in the training and validation sets to obtain $\pi = 0.1$ and labeled only 600 of them. We also report results with 1,000 and 3,000 randomly chosen labeled positives in the training set as is common in the literature. For the image datasets, we subtracted the mean pixel intensity in the training set and divided it by the standard deviation, while for IMDb we used pre-trained GloVe embeddings of size 200 on a corpus of six billion tokens.

Network architectures

To ensure a fair comparison with other works in PUL [4, 6, 10], we used the same architectures on the same datasets, namely a 13-layer convolutional neural network (CNN) for the experiments on CIFAR-10 and CIFAR-100-20 (Table A2) and a multi-layer perceptron (MLP) with four fully-connected hidden layers of 300 neurons each and ReLU activation for MNIST and F-MNIST. For IMDb, we used a bidirectional LSTM network with a MLP head whose number of units was optimized as part of the hyperparameter search (Table A3).

Training

We trained all models with the Adam optimizer [43] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and an exponential learning rate decay with $\gamma = 0.99$, while learning rate, batch size, and weight decay were tuned together with the other pseudo-labeling hyperparameters using the ranges in Table A4 in the Supplements. We provide experimental results using both a PU validation set, to provide a real-world performance estimate, as well as a fully-labeled (PN) validation set to compare against state-of-the-art PUL methods that used such a labeled validation set [6, 26] and to showcase the *potential* of our method. When using a PU validation set, we used the AUROC between positive and unlabeled samples as tuning criterion, as previous work [36, 44] has shown that higher AUROC on PU data directly translates to higher AUROC on fully labeled data.

Evaluation

We obtain the final results by training the model five times with random initialization and training/validation split while using the same canonical test set, reporting both the highest test accuracy obtained and the test accuracy when a PU validation set was used.

We compare PUUPL against VPU [10] and Self-PU [6] using the same network architecture and data splits. We consider the former as it does not require a known prior π and can use a PU validation set, and the latter as the state-of-the-art self-training method for PUL even though it requires a positive-negative (PN) labeled validation set. We additionally compare against a naive, uncertainty-unaware, pseudo-labeling baseline "+PL" that used the sigmoid outputs directly as a ranking measure for pseudo-labeling instead of the epistemic uncertainty, while still assigning soft pseudo-labels.

4.2 Main Results

Table 1 shows the performance of PUUPL and the other baselines in an imbalanced scenario with only 600 labeled positives and a true prior $\pi = 0.1$. PUUPL was the overall best performer in all comparisons except on the MNIST dataset with PU validation, where its performance was 0.31 percentage points (p.p.) lower than the imbnnPU baseline. PUUPL improved performance the most in the IMDb dataset, where accuracy was 2.2 and 3.5 p.p. higher with PN and PU validation sets, and the improvement in CIFAR-100-20 was similarly high with 2.0 and 2.8 p.p. respectively. Self-PU struggled in this setting, collapsing to negative predictions on CIFAR-100-20 and demonstrating unstable behavior on CIFAR-10, where the collapse only occurred in certain training/validation splits but not others. The naive pseudo-labeling baseline that did not use uncertainty worsened performance, compared to imbnnPU, in three datasets out of five, regardless of the method used for validation, hypothetically due to the emergence of the confirmation bias.

We performed the same comparison using 3,000 labeled training positives and the natural prior of each dataset, while using the nnPU loss as $\mathcal{L}_{PU}$ (Table 2). The results were qualitatively similar, with PUUPL providing the highest test accuracy except for Fashion-MNIST, and sometimes considerable performance increase, for example almost 5 p.p. more in the case of CIFAR-100-20 with 3,000 positives.

These findings substantiate the advantages of pseudo-labeling in PUL as well as the necessity of uncertainty quantification in this procedure and in particular the benefit that this brings in more imbalanced scenarios with few labeled positives or low prior.

4.3 Further analyses

In this section, we investigate the inner workings of PUUPL as well as the sensitivity of PUUPL with respect to pseudo-labeling hyperparameters. These are the class prior π, loss weighting parameter λ, and the number of training positives and we alter each of these parameters and compare the resulting test performance on CIFAR-10 in the general PUL setting. Further results regarding pseudo-labeling hyperparameters can be found in Supplementary Section B.

Valid.	Method	Dataset				
		MNIST	F-MN	C-100-20	CIF-10	IMDb
PN	Self-PU [6]	94.44±0.12	90.99±0.47	50.00±0.0	63.97±3.97	-
	imbnnPU [5]	95.65±0.11	91.54±0.18	71.61±0.73	87.59±0.26	74.44±0.61
	+ PL	95.19±0.20	91.26±0.22	71.80±0.93	85.82±0.50	75.94±0.61
	+ PUUPL	**96.09**±0.10	**91.93**±0.12	**73.79**±0.33	**88.93**±0.31	**78.16**±0.78
PU	VPU [10]	80.87±3.24	89.30±0.98	70.01±0.96	86.41±0.78	-
	imbnnPU [5]	**95.61**±0.05	89.88±0.51	67.13±1.26	87.61±0.25	74.32±0.58
	+PL	94.65±0.48	89.55±0.57	68.42±1.18	84.57±1.32	75.88±0.68
	+PUUPL	95.30±0.50	**91.86**±0.09	**72.80**±0.38	**87.97**±0.38	**77.78**±0.86

Table 1 Average test accuracy and its standard error over five repetitions where model training was performed with an imbalanced dataset with $\pi = 0.1$ and 600 labeled positives. The row "+PL" refers to an uncertainty-unaware pseudo-labeling baseline, while "+PUUPL" refers to our uncertainty-aware solution. The validation column refers to the use of a fully-labeled (PN) or PU validation set.

Method	Dataset					Valid.
	MNIST	F-MN	C-100-20	CIF-10	IMDb	
Self-PU [6]	95.64 ±0.13	91.55 ±0.18	75.41 ±0.44	90.56 ±0.09	-	PN
VPU [10]	93.84 ±0.88	91.90 ±0.22	72.12 ±1.05	87.50 ±1.05	-	PU
nnPU [4]	96.36±0.06	91.70±0.12	72.46±0.83	90.49±0.13	79.62±0.67	
+PL	96.22±0.13	92.09±0.12	74.07±0.71	90.56±0.11	79.04±0.39	PN
+ PUUPL	**97.02**±0.08	**92.13**±0.09	**77.39**±0.31	**91.12**±0.04	**80.43**±0.40	
nnPU [4]	95.70±0.11	90.93±0.26	72.48±0.83	90.49±0.15	79.62±0.65	
+PL	95.91±0.23	**91.36**±0.13	74.30±0.68	90.23±0.07	79.04±0.39	PU
+ PUUPL	**97.12**±0.07	91.26±0.26	**77.49**±0.33	**90.74**±0.15	**80.26**±0.51	

Table 2 Average test accuracy and its standard error over five repetitions on various datasets with 3,000 labeled training positives. The row "+PL" refers to an uncertainty-unaware pseudo-labeling baseline, while "+PUUPL" refers to our uncertainty-aware solution. The validation column refers to the use of a fully-labeled (PN) or PU validation set.

PUUPL is loss-agnostic

Our framework is uniquely positioned to take advantage of newly developed risk estimators for PU learning: as we showed above, PUUPL could make use of the imbnnPU loss [5] and the nnPU loss [4] to substantially improve on the state-of-the-art in the imbalanced and the general setting. Next to imbnnPU, there exists a variety of alternative PU losses exist for different scenarios. The nnPUSB loss [22] was developed to address the issue of labeling bias in the

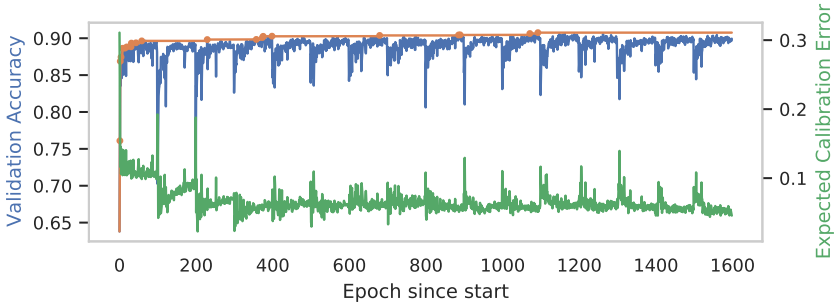


Fig. 2 Validation accuracy (left, blue) and expected calibration error (ECE, right, green) for a run on CIFAR-10 with 1,000 positives. Note the substantial reduction in ECE in the second and third pseudo-labeling iterations, when the ensemble is trained on soft labels. The orange line indicates the best validation accuracy at each epoch, with the new highest accuracy marked by orange dots. The overall highest was 90.76% at epoch 1092, corresponding to a test accuracy of 90.35%.

	PU loss	
	nnPU [4]	nnPUSB [22]
Only PU loss	87.05 ±0.14	87.31 ±0.12
PU loss+PUUPL	**87.70** ±0.14	**87.91** ±0.14

Table 3 Test accuracy of PUUPL on the CIFAR-10 dataset with a selection bias on the positive labels when using the nnPU and nnPUSB losses. Our framework improved over the base PU loss in both cases, and, in particular, PUUPL with nnPU loss achieved better performance than the nnPUSB loss alone.

training positives, a more general setting compared to the i.i.d. assumption of traditional PUL methods [20]. We tested PUUPL in such a biased setting where positives in the CIFAR-10 training and validation sets were with 50% chance an airplane, 30% chance an automobile, 15% chance a ship, and 5% chance a truck. The test distribution was instead balanced, meaning for instance that test samples were half as likely to be airplanes compared to the training set, and five times more likely to be truck images. We used the same hyperparameters as for the i.i.d. CIFAR-10 experiments except for the loss $\mathcal{L}_{PU}$ where we used the nnPUSB loss [22] to handle the positive bias. The baseline with nnPUSB loss performed better than the nnPU loss but worse than *PUUPL* with the nnPU loss, and the best performance was achieved with *PUUPL* on top of the nnPUSB loss (Table 3).

These results demonstrate that PUUPL can be applied even when a sampling bias is suspected by a practitioner and no *ad hoc* risk estimator is available, as our uncertainty-aware pseudo-labeling framework with the bias-oblivious nnPU loss obtained better results compared to a bias-aware risk estimator without pseudo-labeling.

	nnPU	+PL	+PUUPL
Test Expected Calibration Error (%)			
IMDb	25.94±0.78	25.23±0.11	**6.33**±0.17
CIFAR-10	10.89±0.10	9.24±0.15	**5.70**±0.72
CIFAR-100-20	31.62±0.29	27.51±0.59	**22.12**±0.39
Pseudo-labels Negative Log-Likelihood			
IMDb	-	2.74±0.19	0.61±0.02
CIFAR-10	-	0.66±0.05	**0.29**±0.03
CIFAR-100-20	-	3.76±0.29	**0.94**±0.05

Table 4 Expected calibration error (ECE) on the test set using 1,000 labeled positives for training (average and standard error over five runs), as well as negative log-likelihood of the assigned pseudo-labels against the true labels.

Uncertainty quantification improves pseudo-labels

According to the results in Table 1 and 2, naive pseudo-labeling frequently reduces performance, rather than improving it; it then follows that the performance improvement of PUUPL stems from the uncertainty ranking used to select and assign pseudo-labels (Eq. 8). We investigated this in a series of experiments with PUUPL, nnPU, and the naive pseudo-labeling baseline with 1,000 labeled positives, and we found that the improvement in expected calibration error (ECE) on the test set and negative log-likelihood (NLL) of the pseudo-labels assigned by PUUPL was at least 40% and often much larger (Table 4). As shown in Figure 2, the ECE decreased during the first few pseudo-labeling rounds, after which it stabilized while the accuracy continued improving. We also observed that a larger improvement in pseudo-label quality corresponds to a larger improvement in predictive performance.

Robustness of PUUPL

Towards prior misspecification: An important concern for practitioners is how to determine the prior π of a PU dataset, as in the case of sub-optimal estimation the performance of the PU classifier can be harmed considerably. Prior estimation constitutes a whole research branch in PUL [10, 25] and is a significant challenge in any practical PU application [20]. Some contemporary methods for PUL [6, 11–13] assume a known prior and do not discuss the practical consequences of not knowing such parameter, while other methods incorporate prior estimation directly into the training procedure [10, 25, 26]. We treated π as a hyperparameter optimized using the AUROC on a PU validation set as a criterion [36, 44], thus bridging the gap between estimating the prior during training and assuming it is known *a priori*.

Our experimental results show that optimizing the prior in such a way resulted in a consistent reduction in test accuracy between 0.8 and 1.2 percentage points for our framework, the PU loss, and the naive pseudo-labeling baseline (PU sections in Tables 1 and **??**). In a similar vein, methods such as

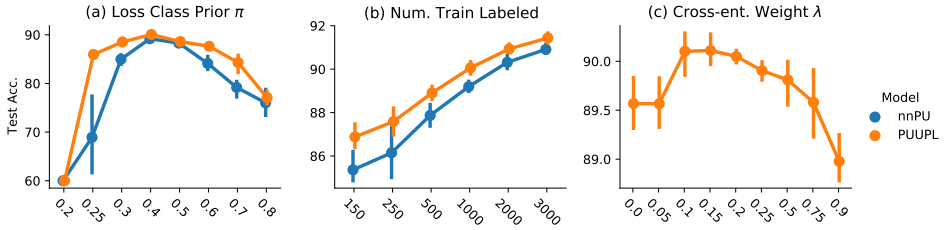


Fig. 3 Mean and standard deviation of the CIFAR-10 test accuracy obtained over five runs when training with wrong prior (a), number of training labeled positives (b) and different loss combination parameter λ (c). PUUPL proved to be more robust to prior misspecification (true $\pi = 0.4$), as the performance degradation was considerably reduced over a wide range of values. It was also more robust to the lower number of labeled samples, as the gap between our framework and nnPU widened when fewer labeled positives were available for training (note the different y-axes scales).

VPU that optimize the prior as part of the training procedure show a similar or larger difference compared to methods that use a PN labeled validation set such as Self-PU. However, in most cases, PUUPL remained the top performer in both settings.

Moreover, training using a wrong value for π was less harmful to PUUPL compared to nnPU only (Fig. 3a). For example, on CIFAR-10 the test accuracy showed a wide plateau around the true prior of 0.4 with a performance reduction of less than 2.5% in the range $[0.3, 0.6]$. With smaller priors, the nnPU loss collapsed to constantly predicting the majority class, and specialized oversampled risk estimators [5] were needed to cope with such a setting (we showed the effectiveness of PUUPL in imbalanced settings in the previous section). Furthermore, the performance gap between *PUUPL* and nnPU widened as π was more severely misspecified, indicating a higher degree of robustness.

Number of labeled training positives The performance of PUUPL steadily increased and seemed to plateau at 91.4% at 3,000 labeled positives (Fig. 3b). The gap between nnPU and PUUPL was largest in the low labeled data region with a 1.44% gap at 250 labels, where PUUPL achieved 87.59% accuracy, shrinking to a gap of 0.52% with 3,000 labels, where PUUPL's performance was 91.44%. This supports our intuition about the importance of accounting for prediction uncertainty because, as the amount of labeled data decreases, uncertainty becomes more important to detect overfitting and to prevent the model from assigning incorrect pseudo-labels.

Loss mixing parameter: As a loss, PUUPL uses a convex combination of a loss for the assigned pseudo-labels and the remaining PU data using a mixing coefficient λ (Eq.). The best performing combination used $\lambda = 0.1$, with modest performance reduction until $\lambda = 0.5$ (Fig. 3c), with too small values nullifying the effect of pseudo-labeling, and larger values harming performance. In general, when too few samples are pseudo-labeled, the loss $\mathcal{L}_L$ is a high variance estimator of the classification risk, and thus should not be weighted excessively. This effect may be reduced as more pseudo-labels are added, and

dynamic adaptation of λ over training could provide an additional performance improvement.

4.4 Real-world application

In this section we show the applicability and benefits of PUUPL to a real-world imbalanced dataset with applications related to healthcare, improving the predictive performance of previous methods developed *ad hoc*. Cancer is the result of malignant mutations that were not wiped out in time by the immune system. It is however possible to instruct the immune system to fight the tumor through specific vaccines that contain neo-epitopes that arose as a result of those mutations, i.e., short genomic regions surrounding the mutated sites that can trigger an immune response [45]. Such vaccines can be designed computationally by solving an optimization problem that chooses the most promising mutations to target while ensuring that the vaccine can be processed appropriately by the body [46–48]. One of the main steps of such processing [49] is the digestion of the vaccine by the proteasome, a tubular protein complex that degrades old or misfolded proteins into shorter fragments (Fig. 4). In order for the vaccine to be effective these pieces must correspond to the neoantigens originally contained in the vaccine. Therefore, accurately predicting proteasomal cleavage, i.e., the position where a sequence is cut, is very important to design more effective vaccines. Modern high-throughput pipelines [3] are able to detect MHC-presented epitopes on the cell surface (Fig. 44) which must have originated from proteasomal cleavage. While missed cleavage sites are never measured, not all presented epitopes are detected, and not all peptides resulting from proteasomal cleavage are presented. Thus, PUL is a natural abstraction of proteasomal cleavage prediction.

Dataset

We collected a dataset of 294,615 MHC-I epitopes from the IEDB [50] database and 89,853 from the Human MHC Ligand Atlas [51]. To identify the potential progenitor protein of each epitope, we used BLAST [52] and filtered for epitopes with a unique progenitor protein resulting in a total of 258,424 data points. Through the progenitor protein, we recovered the residues preceding the N-terminus and following the C-terminus of the epitope, thus providing context for the cleavage predictor. We generated two separate datasets based exclusively on N- or C-termini cleavage sites, as it is known that the biological signal differs in these two situations [53]. We generated "decoy" samples by considering cleavage sites located within three residues of the experimentally-determined terminus; as discussed previously, it is unknown whether cleavage could or could not have happened at those positions, hence we treat such decoys as unlabeled in our PUL training procedure. The final datasets were then composed of 1,285,659 samples with 229,163 positives for the N-terminus datasets and 1,277,344 samples with 222,181 positives for the C-terminus datasets.

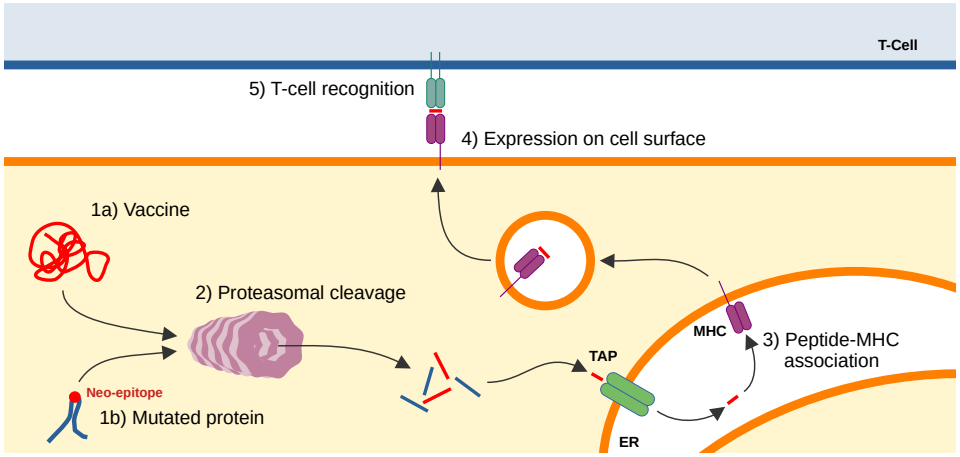


Fig. 4 Predicting the outcome of each event in the antigen processing pathway [49] is crucial to enable the design of epitope vaccines. Vaccines ingested by antigen presenting cells (1a) as well as mutated proteins produced by cancerous cells (1b) are cleaved in short fragments by the proteasome (2). Some of these fragments, or peptides, are then transported into the endoplasmic reticulum (ER) through the Transporter associated with Antigen Processing (TAP). A fraction of these peptides bind to the Major Histocompatibility Complex (MHC, 3) and the resulting construct is then expressed on the cell surface (4), where they can be inspected by passerby T-cells and possibly trigger an appropriate immune response (5).

Modeling, Training, and Evaluation

Each sequence contains ten amino acids, each of which was one-hot encoded and processed by a MLP. We used imbnnPU [5] as $\mathcal{L}_{PU}$. For imbnnPU and PUUPL we report the cross-validation scores and use the statistical test proposed by [54] to estimate the AUROC, its standard error, and confidence intervals. Note that, as we do not know the true negatives, traditional metrics to evaluate classification performance such as accuracy, F1, precision, recall, etc. are not applicable. As external baselines we consider NetChop [55] and NetCleave [56], evaluating their predictions on ten random bootstraps of our dataset. These baselines are based on MLPs and convolutional neural networks respectively and, importantly, they approach the problem as a supervised binary classification task, treating decoy samples as negatives rather than unlabeled. We also present evaluation scores for the imbnnPU loss [5], commonly used for PUL on imbalanced datasets.

Results

Both PUUPL and the imbnnPU loss achieved lower performance on the N-terminals dataset, confirming previous observations that this predictive task is harder due to the biological processes involved [53]. On the C-terminal dataset, the imbnnPU loss improved performance by 2.5 and 4.4 points compared to NetChop and NetCleave respectively, and PUUPL added a further 3.2 points reaching 87.2% AUROC (Table 5). In both datasets the difference

	AUROC	
	N-terminal	C-terminal
NetChop 20S	52.72 ±0.02	66.07 ±0.02
NetChop C term	50.99 ±0.02	81.53 ±0.01
NetCleave	49.27 ±0.02	79.61 ±0.01
imbnnPU	75.15 ±0.06	83.99 ±0.06
PUUPL	**78.00** ±0.06	**87.20** ±0.04

Table 5 Average and standard error of area under the ROC curve (AUROC) on both datasets for NetChop, NetCleave, the imbnnPU loss and PUUPL.

in AUROC between imbnnPU and PUUPL was statistically significant at a significance of 1%: the confidence intervals are $[74.99, 75.32]$ and $[77.85, 78.15]$ for N-terminals, and $[83.85, 84.14]$ and $[87.08, 87.32]$ for C-terminals. Note that both NetChop and NetCleave were only trained on C-terminals cleavage sites in the original publication, thus explaining their random predictions on the N-terminals dataset.

5 Discussion and conclusions

We introduced PUUPL, an uncertainty-aware pseudo-labeling framework for PUL that uses the epistemic uncertainty of an ensemble of networks to select which examples to pseudo-label. We conducted extensive experiments to demonstrate the benefits of our approach and show its reliability in settings that are likely to be encountered in the real world such as heavily imbalanced settings with small π and few labeled positives, a bias in the positive training data, the unavailability of labeled negatives for validation, and the misspecification of the class prior π. Unlike many alternative methods, PUUPL can be applied to learning problems in any domain out of the box as it does not rely on regularization methods that are restricted to a specific data modality, most frequently images, such as mixup [9] (used by [10, 13]) or contrastive representations (used by [12]). Furthermore, it is easy to adapt as it builds on standard methods (unlike [6]), and does not require pretrained representations to work (as [11] does). We further used our framework to advance the state-of-the-art on a real-world healthcare dataset with potential repercussions on efficacy and deployment cost of personalized epitope vaccines for cancer treatment.

Our choice of deep ensembles was rooted in their competitiveness in empirical benchmarks [57], however, *PUUPL* can easily be extended to take advantage of more accurate uncertainty quantification methods as they become available [58]. In fact, as the matter of uncertainty quantification in deep learning is far from settled, the performance and efficiency of our framework could be further improved by employing more accurate uncertainty quantification methods [58].

Limitations

We demonstrated robustness against biased positive labels and imbalanced datasets, however, it is the practitioners' responsibility to ensure that the obtained predictions are "fair", with "fairness" defined appropriately with respect to the target application, and do not systematically affect particular subsets of the population of interest. Ultimately, we can only leave it to practitioners to use their moral and ethical judgment as to whether all stakeholders and their interests are fairly represented in their application. While we have shown that PUUPL works across different modalities such as image, text, and epitope data it would be interesting to apply PUUPL on further modalities. Furthermore, we used deep ensembles as a strategy to obtain uncertainty which enables PUUPL's strong performance. Combining PUUPL with different alternative uncertainty quantification techniques would be an exciting avenue for further research.

6 Declarations

Funding Emilio Dorigatti was supported by the Helmholtz Association under the joint research school "Munich School for Data Science - MUDS" (Award Number HIDSS-0006). Jann Goschenhofer was supported by the Bavarian Ministry of Economic Affairs, Regional Development, and Energy through the Center for Analytics – Data – Applications (ADA-Center) within the framework of BAYERN DIGITAL II (20-3410-2-9-8). B. S. acknowledges financial support by the Postdoctoral Fellowship Program of the Helmholtz Zentrum München. Mina Rezaei and Bernd Bischl were supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A, Munich Center for Machine Learning (MCML).

Conflicts of interest/Competing interests The authors of the manuscript do not have a conflict of interest.

Ethics approval For this manuscript, no clinical data have been collected.

Consent for publication The author consent Journal of Machine Learning.

Availability of data and material The current manuscript examined and reported the results based on public datasets and benchmarks.

Authors' contributions The method was conceived by E.D. and finalized upon discussion with all other authors. All authors contributed to the experimental protocol, while the implementation was performed by E.D. and J.G., who also performed the experiments. All authors contributed to the interpretation of the results. The manuscript was written by E.D. and J.G. with feedback from all other authors. All authors read and approved the final manuscript.

Dataset	Train Pos.	Train Neg.	Test Size
MNIST	30,508	29,492	10,000
F-MNIST	30,000	30,000	10,000
CIFAR-10	20,000	30,000	10,000
CIFAR-100-20	25,000	25,000	10,000
IMDb	12,500	12,500	25,000
20 News	6,216	4,798	7,317

Table A1 Size of test set and number of positives and negatives in the training set for each dataset.

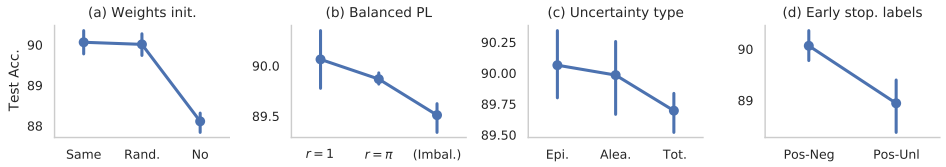


Fig. B1 Mean and standard deviation of the test accuracy obtained over five runs by different variations of our *PUUPL* algorithm: (a) different weight initialization at each iteration, (b) balanced or imbalanced PL selection, (c) type of uncertainty, (d) whether to use PN or PU validation set. Note the different scales on the y-axes.

Appendix A Network architecture, hyperparameters and datasets

Table A1 reports the number of samples in each dataset. Table A2 reports the network architecture used in the CIFAR-10 and CIFAR-100-20 experiments, while Table A3 reports the network used with IMDb. Table A4 reports the hyperparameters related to pseudo-labeling and their ranges.

Appendix B Further sensitivity analyses

We performed ablation studies on the CIFAR-10 dataset by changing one parameter at a time of the best configuration found by Hyperband, training and evaluating with five different splits, and reporting the test accuracy corresponding to the best validation score for each run. To limit the computational resources needed, we used at most 15 pseudo-labeling iterations.

Weight initialization: We confirmed the observation that it is beneficial to re-initialize the weights after each pseudo-labeling step [31], with slightly better performance (+0.052%) achieved when the weights are re-initialized to the same values before every pseudo-labeling iteration (Fig. B1a). We believe this encourages the model to be consistent across pseudo-labeling rounds.

Uncertainty: Ranking predictions by aleatoric performance was almost as good as ranking by epistemic uncertainty (−0.08%), while total uncertainty produced moderately worse rankings (−0.37%, Fig. B1c). An ensemble with

Layer type	**Layer parameters**
Conv. 2D	InC=3, OutC=96, k=3, s=1, p=1
Dropout	p=0.15
Batch Norm.	eps=1e-05, momentum=0.1
ReLU	
Conv. 2D	InC=96, OutC=96, k=3, s=1, p=1
Dropout	p=0.15
Batch Norm.	eps=1e-05, momentum=0.1
ReLU	
Conv. 2D	InC=96, OutC=96, k=3, s=2, p=1
Dropout	p=0.15
Batch Norm.	eps=1e-05, momentum=0.1
ReLU	
Conv. 2D	InC=96, OutC=192, k=3, s=1, p=1
Dropout	p=0.15
Batch Norm.	eps=1e-05, momentum=0.1
ReLU	
Conv. 2D	InC=192, OutC=192, k=3, s=1, p=1
Dropout	p=0.15
Batch Norm.	eps=1e-05, momentum=0.1
ReLU	
Conv. 2D	InC=192, OutC=192, k=3, s=2, p=1
Dropout	p=0.15
Batch Norm.	eps=1e-05, momentum=0.1
ReLU	
Conv. 2D	InC=192, OutC=192, k=3, s=1, p=1
Dropout	p=0.15
Batch Norm.	eps=1e-05, momentum=0.1
ReLU	
Conv. 2D	InC=192, OutC=192, k=1, s=1, p=0
Dropout	p=0.15
Batch Norm.	eps=1e-05, momentum=0.1
ReLU	
Conv. 2D	InC=192, OutC=10, k=1, s=1, p=0
Dropout	p=0.15
Batch Norm.	eps=1e-05, momentum=0.1
ReLU	
Flatten	
Linear	in features=640, out features=1000, bias=yes
ReLU	
Linear	in features=1000, out features=1000, bias=yes
ReLU	
Linear	in features=1000, out features=1, bias=yes

Table A2 Network architecture used for the CIFAR-10 experiments. InC: input channels, OutC: output channels, k: kernel size, s: stride, p: padding

Layer type	Layer parameters
LSTM	inpus size=200, hidden size=128, num layers=2, dropout=0.25, bidirectional=True
Dropout	p=0.2
Linear	in features=256, out features=196, bias=True
Batch Norm.	eps=1e-05, momentum=0.1
ReLU	
Dropout	p=0.2
Linear	in features=196, out features=196, bias=True
Batch Norm.	eps=1e-05, momentum=0.1
ReLU	
Linear	in features=196, out features=1, bias=True

Table A3 Network architecture used for the IMDb experiments

Hyper-parameter	Value range
Estimator	Ensemble or MC Dropout
Number of samples	$[2, 25]$
Uncertainty type	Aleatoric, epistemic, total
Max. new labels T	$[100, 5000]$
Max. new label uncertainty t_l	$[0, -\log 2]$
Min. unlabel uncertainty t_u	$[0, -\log 2]$
Reassign all pseudo-labels	Yes or no
Re-initialize to same weights	Yes or no
Cross-entropy weight λ	$[0, 1]$

Table A4 Pseudo-labeling hyperparameters

only two networks achieved the best performance, while larger ensembles performed worse, and Monte Carlo dropout (-0.85%) was better than ensembles of five (-1.00%) and ten networks (-1.58%).

Early stopping: Finally, performing early stopping on the validation PU loss resulted in worse accuracy (-1.12%) compared to using the accuracy on PN labels (Fig. B1d). Although considerable when compared to the impact of other algorithmic choices, such a performance drop indicates that *PUUPL* can be used effectively in real-world scenarios when no labeled validation data are available.

Pseudo-labeling hyperparameters: Our method was fairly robust to the maximum number T of assigned pseudo-labels and the maximum uncertainty threshold t_l for the pseudo-labels, with almost constant performance up to $T = 1000$ and $t_l = 0.1$. The best performance was achieved by the combination having $T = 1000$ and $t_l = 0.05$, but both of these experiments were performed while disabling the other constraint (i.e., setting $T = \inf$ when testing t_l and vice-versa). Using only a constraint on T resulted in a reduction of -0.11%, while constraining t_l alone resulted in a reduction of -1.04%. The results for t_u were less conclusive than for the general trend, possibly because

values lower than 0.35 require more than the 15 pseudo-labeling iterations we used for the experiment, and values above 0.4 did not show significant differences.

Moreover, soft pseudo-labels were preferred over hard ones (+0.75%). Contrary to expectation, however, re-assigning all pseudo-labels at every iteration slighly harmed performance (−0.12%); instead, pseudo-labels should be kept fixed after being assigned for the first time. A possible explanation is that fixed pseudo-labels prevent the model's predictions from drifting too far away from the initial pseudo-labeling towards an incorrect assignment, and thus contribute in mitigating the sort of confirmation bias that frequently plagues pseudo-labeling-based methods. It was also beneficial to assign the same number of positive and negative pseudo-labels compared to keeping the same ratio π of positives and negatives found in the whole dataset (−0.20%) or not balancing the selection at all (−0.55%). This prevents the pseudo-labeled set from becoming too imbalanced over time, a natural tendency deriving from the inherent imbalance between positive and unlabeled samples in the training set.

Appendix C Ethics statement and broader impact

Improving performance of PUL methods will catalyze research in areas where PU datasets are endemic and manual annotation is expensive or negative samples are impossible to obtain – for example, in bioinformatics and medical applications – which ultimately benefits human welfare and well-being. The explicit incorporation of uncertainty quantification further increases the trustworthiness and reliability of PUUPL's predictions. However, such advances in PUL could also reduce the resources required to create unwanted mass-surveillance systems by governments and/or private companies.

References

[1] Armenian, H.K., Lilienfeld, A.M.: The distribution of incubation periods of neoplastic diseases. American journal of epidemiology **99**(2), 92–100 (1974)

[2] Gligorijević, V., Renfrew, P.D., Kosciolek, T., Leman, J.K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B.C., Fisk, I.M., Vlamakis, H., *et al.*: Structure-based protein function prediction using graph convolutional networks. Nature communications **12**(1), 1–14 (2021)

[3] Purcell, A.W., Ramarathinam, S.H., Ternette, N.: Mass spectrometry–based identification of MHC-bound peptides for immunopeptidomics. Nat Protoc **14**(6), 1687–1707 (2019). https://doi.org/10.1038/s41596-019-0133-y

[4] Kiryo, R., Niu, G., du Plessis, M.C., Sugiyama, M.: Positive-unlabeled learning with non-negative risk estimator. Advances in Neural Information Processing Systems (2017)

[5] Su, G., Chen, W., Xu, M.: Positive-unlabeled learning from imbalanced data. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence Organization, ??? (2021). https://doi.org/10.24963/ijcai.2021/412

[6] Chen, X., Chen, W., Chen, T., Yuan, Y., Gong, C., Chen, K., Wang, Z.: Self-pu: Self boosted and calibrated positive-unlabeled training. In: International Conference on Machine Learning, pp. 1510–1519 (2020). PMLR

[7] Lee, D.-H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML, vol. 3, p. 896 (2013)

[8] Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc., ??? (2017)

[9] Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: Mixup: Beyond empirical risk minimization. International Conference on Learning Representations (2018)

[10] Chen, H., Liu, F., Wang, Y., Zhao, L., Wu, H.: A variational approach for learning from positive and unlabeled data. In: Advances in Neural Information Processing Systems, pp. 14844–14854 (2020)

[11] Hammoudeh, Z., Lowd, D.: Learning from positive and unlabeled data with arbitrary positive shift. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 13088–13099. Curran Associates, Inc., ??? (2020). https://proceedings.neurips.cc/paper/2020/file/98b297950041a42470269d56260243a1-Paper.pdf

[12] Acharya, A., Sanghavi, S., Jing, L., Bhushanam, B., Choudhary, D., Rabbat, M.G., Dhillon, I.S.: Positive unlabeled contrastive learning. ArXiv **abs/2206.01206** (2022)

[13] Zhao, Y., Xu, Q., Jiang, Y., Wen, P., Huang, Q.: Dist-pu: Positive-unlabeled learning from a label distribution perspective. In: Proceedings

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14461–14470 (2022)

[14] Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building text classifiers using positive and unlabeled examples. In: Third IEEE International Conference on Data Mining, pp. 179–186 (2003). IEEE

[15] Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: International Conference on Machine Learning, pp. 4393–4402 (2018). PMLR

[16] Li, W., Guo, Q., Elkan, C.: A positive and unlabeled learning algorithm for one-class classification of remote-sensing data. IEEE Transactions on geoscience and remote sensing **49**(2), 717–725 (2010)

[17] Xu, Y., Xu, C., Xu, C., Tao, D.: Multi-positive and unlabeled learning. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, pp. 3182–3188 (2017)

[18] Kaji, H., Yamaguchi, H., Sugiyama, M.: Multi task learning with positive and unlabeled data and its application to mental state prediction. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2301–2305 (2018). https://doi.org/10.1109/ICASSP.2018.8462108

[19] Chapelle, O., Scholkopf, B., Zien, A.: Semi-supervised learning. Cambridge, Massachusettes: The MIT Press View Article (2009)

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

[21] Du Plessis, M.C., Niu, G., Sugiyama, M.: Analysis of learning from positive and unlabeled data. Advances in neural information processing systems **27**, 703–711 (2014)

[22] Kato, M., Teshima, T., Honda, J.: Learning from positive and unlabeled data with a selection bias. In: International Conference on Learning Representations (2019)

[23] Hsieh, Y.-G., Niu, G., Sugiyama, M.: Classification from positive, unlabeled and biased negative data. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 2820–2829. PMLR, ??? (2019)

[24] Luo, C., Zhao, P., Chen, C., Qiao, B., Du, C., Zhang, H., Wu, W., Cai,

S., He, B., Rajmohan, S., *et al.*: Pulns: Positive-unlabeled learning with effective negative sample selector. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 8784–8792 (2021)

[25] Garg, S., Wu, Y., Smola, A.J., Balakrishnan, S., Lipton, Z.: Mixture proportion estimation and pu learning:a modern approach. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P.S., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems, vol. 34, pp. 8532–8544. Curran Associates, Inc., ??? (2021). https://proceedings.neurips.cc/paper/2021/file/47b4f1bfdf6d298682e610ad74b37dca-Paper.pdf

[26] Hu, W., Le, R., Liu, B., Ji, F., Ma, J., Zhao, D., Yan, R.: Predictive adversarial learning from positive and unlabeled data. Proceedings of the AAAI Conference on Artificial Intelligence **35**(9), 7806–7814 (2021)

[27] Van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Machine Learning **109**(2), 373–440 (2020)

[28] Gawlikowski, J., Tassi, C.R.N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., Shahzad, M., Yang, W., Bamler, R., Zhu, X.: A survey of uncertainty in deep neural networks. ArXiv **abs/2107.03342** (2021)

[29] Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Label propagation for deep semi-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5070–5079 (2019)

[30] Shi, W., Gong, Y., Ding, C., Tao, Z.M., Zheng, N.: Transductive semi-supervised deep learning using min-max features. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 299–315 (2018)

[31] Arazo, E., Ortego, D., Albert, P., O'Connor, N., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. 2020 International Joint Conference on Neural Networks (IJCNN), 1–8 (2020)

[32] Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint optimization framework for learning with noisy labels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5552–5560 (2018)

[33] Rizve, M.N., Duarte, K., Rawat, Y., Shah, M.: In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. International Conference on Learning Representations (2021)

[34] Beluch, W.H., Genewein, T., Nürnberger, A., Köhler, J.M.: The power

of ensembles for active learning in image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9368–9377 (2018)

[35] Christoffel, M., Niu, G., Sugiyama, M.: Class-prior estimation for learning from positive and unlabeled data. In: Asian Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 45, pp. 221–236 (2016)

[36] Menon, A., Rooyen, B.V., Ong, C.S., Williamson, B.: Learning from corrupted binary labels via class-probability estimation. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 125–134. PMLR, Lille, France (2015)

[37] Hüllermeier, E., Waegeman, W.: Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. Machine Learning **110**(3), 457–506 (2021)

[38] Müller, R., Kornblith, S., Hinton, G.E.: When does label smoothing help? In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc., ??? (2019). https://proceedings.neurips.cc/paper/2019/file/f1748d6b0fd9d439f71450117eba2725-Paper.pdf

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

[40] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Citeseer (2009)

[41] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)

[42] Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142–150. Association for Computational Linguistics, Portland, Oregon, USA (2011)

[43] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2015)

[44] Jain, S., White, M., Radivojac, P.: Recovering true classifier performance

in positive-unlabeled learning. Proceedings of the AAAI Conference on Artificial Intelligence **31**(1) (2017)

[45] Hu, Z., Ott, P.A., Wu, C.J.: Towards personalized, tumour-specific, therapeutic vaccines for cancer. Nature Reviews Immunology **18**(3), 168–182 (2017). https://doi.org/10.1038/nri.2017.131

[46] Dorigatti, E., Schubert, B.: Joint epitope selection and spacer design for string-of-beads vaccines. Bioinformatics **36**(Supplement_2), 643–650 (2020). https://doi.org/10.1093/bioinformatics/btaa790

[47] Dorigatti, E., Schubert, B.: Graph-theoretical formulation of the generalized epitope-based vaccine design problem. PLOS Computational Biology **16**(10), 1008237 (2020). https://doi.org/10.1371/journal.pcbi.1008237

[48] Toussaint, N.C., Maman, Y., Kohlbacher, O., Louzoun, Y.: Universal peptide vaccines – Optimal peptide vaccine design based on viral sequence conservation. Vaccine **29**(47), 8745–8753 (2011). https://doi.org/10.1016/j.vaccine.2011.07.132

[49] Blum, J.S., Wearsch, P.A., Cresswell, P.: Pathways of antigen processing. Annual Review of Immunology **31**(1), 443–473 (2013). https://doi.org/10.1146/annurev-immunol-032712-095910

[50] Vita, R., Mahajan, S., Overton, J.A., Dhanda, S.K., Martini, S., Cantrell, J.R., Wheeler, D.K., Sette, A., Peters, B.: The Immune Epitope Database (IEDB): 2018 update. Nucleic Acids Research **47**(D1), 339–343 (2018) https://academic.oup.com/nar/article-pdf/47/D1/D339/27436402/gky1006.pdf. https://doi.org/10.1093/nar/gky1006

[51] Marcu, A., Bichmann, L., Kuchenbecker, L., Kowalewski, D.J., Freudenmann, L.K., Backert, L., Mühlenbruch, L., Szolek, A., Lübke, M., Wagner, P., Engler, T., Matovina, S., Wang, J., Hauri-Hohl, M., Martin, R., Kapolou, K., Walz, J.S., Velz, J., Moch, H., Regli, L., Silginer, M., Weller, M., Löffler, M.W., Erhard, F., Schlosser, A., Kohlbacher, O., Stevanović, S., Rammensee, H.-G., Neidert, M.C.: The HLA ligand atlas - a resource of natural HLA ligands presented on benign tissues (2019). https://doi.org/10.1101/778944

[52] Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. Journal of Molecular Biology **215**(3), 403–410 (1990). https://doi.org/10.1016/s0022-2836(05)80360-2

[53] Schatz, M.M., Peters, B., Akkad, N., Ullrich, N., Martinez, A.N., Carroll, O., Bulik, S., Rammensee, H.-G., van Endert, P., Holzhütter, H.-G., Tenzer, S., Schild, H.: Characterizing the n-terminal processing motif of MHC

class i ligands. The Journal of Immunology **180**(5), 3210–3217 (2008). https://doi.org/10.4049/jimmunol.180.5.3210

[54] LeDell, E., Petersen, M.L., van der Laan, M.J.: Computationally efficient confidence intervals for cross-validated area under the roc curve estimates. Electronic journal of statistics **9 1**, 1583–1607 (2015)

[55] Nielsen, M., Lundegaard, C., Lund, O., Keşmir, C.: The role of the proteasome in generating cytotoxic t-cell epitopes: insights obtained from improved predictions of proteasomal cleavage. Immunogenetics **57**(1-2), 33–41 (2005). https://doi.org/10.1007/s00251-005-0781-7

[56] Amengual-Rigo, P., Guallar, V.: NetCleave: an open-source algorithm for predicting c-terminal antigen processing for MHC-i and MHC-II. Scientific Reports **11**(1) (2021). https://doi.org/10.1038/s41598-021-92632-y

[57] Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., Snoek, J.: Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. Advances in neural information processing systems **32** (2019)

[58] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U.R., *et al.*: A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information Fusion **76**, 243–297 (2021)

4.4 CC-Top: Constrained Clustering for Dynamic Topic Discovery

Contributing article:

Jann Goschenhofer, Pranav Ragupathy, Christian Heumann, Bernd Bischl, and Matthias Assenmacher. 2022. Cc-top: Constrained clustering for dynamic topic discovery. *Proceedings of the First Workshop on Ever Evolving NLP (EvoNLP)*, page 26–34

Author contributions:

This paper is based on Pranav Ragupathy's master's thesis which was supervised by Jann Goschenhofer and Matthias Aßenmacher. Jann Goschenhofer and Matthias Aßenmacher were responsible for the conceptualization of the idea (i.e. idea, goals, and transfer of the methods to the text domain). Pranav Ragupathy implemented the method on top of a codebase that was created by Jann Goschenhofer who also supported Pranav Ragupathy in the implementation. Pranav Ragupathy was responsible for the running of the experiments which were planned and designed by Jann Goschenhofer and Matthias Aßenmacher. The concept of overclustering was introduced by Jann Goschenhofer, and Matthias Aßenmacher introduced the idea to use it in the dynamic setting described in the paper. Jann Goschenhofer and Matthias Aßenmacher were responsible for writing the draft, reviewing, editing, and rebutting with the reviewers. Bernd Bischl and Christian Heuman contributed via supervision and by providing access to computing resources.

Copyright information:

CC-Top: Constrained Clustering for Dynamic Topic Discovery

Jann Goschenhofer[1,2♠] **Pranav Ragupathy**[1♣] **Christian Heumann**[1♠] **Bernd Bischl**[1,2,3♠]
Matthias Aßenmacher[1♠]
[1] Department of Statistics, LMU, Munich, Germany
[2] Fraunhofer IIS, Erlangen, Germany
[3] Munich Center for Machine Learning (MCML), LMU, Munich, Germany
♠{jann.goschenhofer,chris,bernd.bischl,matthias}@stat.uni-muenchen.de
♣p.ragupathy@campus.lmu.de

Abstract

Research on multi-class text classification of short texts mainly focuses on supervised (transfer) learning approaches, requiring a finite set of pre-defined classes which is constant over time. This work explores deep constrained clustering (CC) as an alternative to supervised learning approaches in a setting with a dynamically changing number of classes, a task we introduce as *dynamic topic discovery* (DTD). We do so by using pairwise similarity constraints instead of instance-level class labels which allow for a flexible number of classes while exhibiting a competitive performance compared to supervised approaches. First, we substantiate this through a series of experiments and show that CC algorithms exhibit a predictive performance similar to state-of-the-art supervised learning algorithms while requiring less annotation effort. Second, we demonstrate the overclustering capabilities of deep CC for detecting topics in short text data sets in the absence of the ground truth class cardinality during model training. Third, we showcase how these capabilities can be leveraged for the DTD setting as a step towards dynamic learning over time. Finally, we release our codebase to nurture further research in this area.

1 Introduction

There has been substantial research on methods for the classification of short user-generated texts such as customer reviews, search queries, tweets, or articles (Mohammad et al., 2016; Sun et al., 2019; Barbieri et al., 2020). Often, despite being handled differently in supervised frameworks, one does not know *a-priori* what these classes are, how many there are at time point t, or how many there will be at a future time point $t+1$. In existing benchmark data sets from the natural language processing (NLP) research community (e.g. Lang, 1995; Lehmann et al., 2015), this potential issue is largely ignored, since only one training set is provided alongside one test set. Performance can thus only be measured in a static fashion, i.e. for one fixed time point. While this problem of an unknown number of classes is often tackled using unsupervised learning techniques (Deerwester et al., 1990; Blei et al., 2003), these algorithms come with an array of limitations and are not able to (automatically) adapt to a changing number of classes. We formally introduce this novel problem setting with dynamically changing topics as DTD and explore the potential of deep constrained clustering (CC; Hsu et al., 2019) algorithms coupled with pre-trained language models (BERT; Devlin et al., 2019) for text classification in this setting.

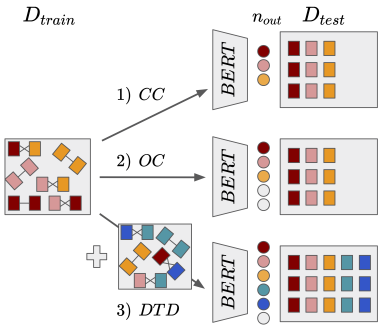


Figure 1: Illustration of CC-Top and the training paradigms 1) constrained clustering (CC), 2) overclustering (OC) and 3) dynamic topic discovery (DTD). Crosses and lines represent Cannot- and Must-Link pairwise relations, respectively.

Various approaches have been developed to combine CC (Wagstaff and Cardie, 2000) with neural networks, mainly for image datasets (Hsu and Kira, 2015; Hsu et al., 2019). In addition to strong predictive clustering performance, these methods are able to recover the number of distinct clusters in the data without access to instance-level class labels during training. Hence, they can be used for category detection, a capability that we leverage for the detection of dynamically changing topics.

Proceedings of the The First Workshop on Ever Evolving NLP (EvoNLP), pages 26 - 34

Moreover, they address and alleviate the problem of label annotation: Human annotators only need to annotate pairs of samples indicating whether they belong to a similar topic or not instead of annotating one distinct class label per sample. We argue that for short texts this is easier and more efficient than annotating individual samples.

We propose the use of **C**onstrained **C**lustering for **Top**ic classification (CC-Top, cf. Fig. 1): We 1) leverage pairwise constraint annotations for topic classification of short texts in a weakly supervised manner, we 2) demonstrate its topic discovery capabilities and 3) introduce a new problem setting with dynamically changing topics. In a series of experiments, we substantiate these findings and publish our codebase[1] to nurture further research on constrained clustering in the NLP community.

2 Related Work

With the advent of supervised fine-tuning of pre-trained models, text clustering performance further increased (Huang et al., 2020; Schopf et al., 2021). One main limitation of these models is their dependence on a given amount of clusters as input for model training, which limits their use for the detection of clusters, i.e., topics/classes. Unsupervised topic modeling algorithms (e.g. Blei et al., 2003; Grootendorst, 2022) are no real alternative here, since we focus on topic *classification* and not on topic *modeling*. Note, that we make a clear distinction between these two approaches here: Topic modeling aims at uncovering latent structures in the data and puts a large emphasis on explaining and interpreting the detected clusters. Further, as opposed to *Topic classification*, it does not assume the cluster assignment to be mutually exclusive, i.e. a document is regarded as a (potential) mixture of multiple topics. Since this is in sharp contrast to the setting we are investigating, we do not consider such approaches as potential unsupervised baselines.

In turn, CC allows this detection of the number of clusters using binary pairwise constraint annotations. The introduction of pairwise constraints for clustering (Wagstaff and Cardie, 2000) led to the adaptation of existing clustering methods towards the use of constraints (Basu et al., 2004) (see Gançarski et al. (2020) for an overview). With the proposal of the KCL loss based on the Kullback-Leibler divergence, Hsu and Kira (2016) introduced CC to deep learning settings. They further showed its applicability to transfer learning (Hsu et al., 2018), introduced the MCL as an alternative loss (Hsu et al., 2019), and showed its applicability for cluster detection, i.e., overclustering. We use these two pairwise loss functions.

3 Materials and Methods

3.1 Method

We consider a dataset $\mathcal{D}$ that contains n_c constraint pairs of the form $x_{ij} = (x_i, x_j, c_{ij}) \in \mathcal{D}^c$, where x_i, x_j are two input samples and $c_{ij} \in \{0, 1\}$ is the associated binary constraint describing whether the samples are in the same ($c_{ij} = 1$, *Must-Link*) or different clusters ($c_{ij} = 0$, *Cannot-Link*). We refer to true class labels as $y_i \in \mathcal{Y}$, where $K = |\mathcal{Y}|$ describes the number of true underlying classes K in the data set. When K is not known, the model's number of output neurons n_{out} may differ from K. We train a deep CC model f with its final head consisting of a softmax layer i.e., the model predicts a probability distribution over cluster assignments $\hat{y}_i = f(x_i)$, where $\hat{y}_{il}$ denotes the predicted probability of x_i belonging to cluster $l \in 1, ..., n_{out}$.

We follow Hsu and Kira (2016); Hsu et al. (2019) for the training of the CC model: the model predictions $\hat{y}_i, \hat{y}_j$ for text samples x_i, x_j are fed into a pairwise loss function with their associated constraint c_{ij}. There exists a variety of loss functions that can deal with pairwise constraints (Zhang et al., 2021b), with the KCL (Hsu and Kira, 2016) and the MCL (Hsu et al., 2019) being the most prominent ones. The KCL is a pairwise loss function based on the Kullback-Leibler divergence between the pairwise model assignments $\hat{y}_i, \hat{y}_j$. Similarly, the MCL loss is aligned on the binary cross entropy loss and reportedly enables smoother model training. Following prior work (Lin et al., 2020; Zhang et al., 2021a), we use BERT (Devlin et al., 2019) as a language model backbone for f.[2] Note that throughout our experiments we randomly subsample a training dataset of $20,000$ pairwise constraints from the original fully labeled dataset.

Next to the application in settings where the true number of clusters K is known a-priori, CC models can also be used when this information is absent during model training. This is also referred to as overclustering (OC) where the model can

[1] https://github.com/rpranav22/cc-top

[2] Note that any (pre-trained) architecture can be used as a backbone in conjunction with these loss functions. All configurations can be found in Table 5 in Appendix A.

assign more clusters than present in the data, i.e. $n_{out} > K$. This capability to learn the number of clusters in the data from constraint annotations differentiates CC from clustering methods such as k-means, where K needs to be provided as a hyperparameter to the model, or supervised approaches.

3.2 Baselines

As a lower, unsupervised baseline, we use BERT embeddings combined with K-MEANS++ (Arthur and Vassilvitskii, 2006). For the fully supervised upper bound trained via instance-level class labels, we finetune the BERT-BASE-UNCASED architecture from huggingface (Wolf et al., 2020), following the standard pretrain-finetune paradigm. Both baselines are trained on the entire training dataset.

3.3 Dynamic Topic Discovery (DTD)

We now consider the scenario, where the set of classes is not fixed and known *a-priori* at time point t but is dynamically changing over time ($t+1, t+2, \dots$): First, at t, we have pairwise annotations for samples that belong to K_t distinct classes. Second, we train a CC model f_t to assign any new data point to one of the discovered clusters. Third, at $t+1$, we obtain new samples that could either belong to one of the initial K_t classes or to new, unseen classes and the model fails to classify the new samples accurately.

If our model was fully supervised (i.e., trained on instance-level class labels), we would have to reconsider the entire labeling scheme (i.e., produce the new classes and revisit all existing labeled samples from t) and re-train the entire model. However, in the case of CC, we can continue annotating the data using pairwise constraints and continue to train the existing model (i.e., let the model determine (i) if there are new classes and (ii) how many of them). We construct the following scenario to investigate the model's capability to adapt to a changing number of classes over time: First, we fix the architectural setup to CC-KCL on DBpedia and use $n_{out} = 30$ to provide the model with enough overclustering flexibility. Second, for $t = 1$, we take a subset of the training set, consisting of samples from 10 classes only, and sample $n_c = 20,000$ constraints from this subset, resulting in $38,056$ samples from 10 classes for training ($D_{train,t=1}$). Third, for $t = 2$, we select $18,000$ samples from the remainder of the training set ($D_{train,t=2}$) controlling for the ratio of the classes that the samples belong to $x\%$ from the 'old' 10 classes at $t = 1$ and $(100 - x)\%$ from the 'new' 4 classes at $t = 2$, which were withheld from $D_{train,t=1}$. The DBpedia test set is also split into two distinct parts: $D_{test,1}$ contains only samples from the 10 'old' classes, and $D_{test,2}$ contains only samples from the 4 'new' ones. During the DTD experiments, we denote the entire test set as $D_{test,combined}$.

3.4 Datasets

We run experiments on three English datasets of short texts with associated instance-level class labels. An overview of the analyzed data sets AG News (Zhang et al., 2015), TREC coarse (Li and Roth, 2002), and DBpedia (Lehmann et al., 2015) is provided in Table 1. We did not perform any further special preprocessing. We used only DBPedia for further experiments with respect to DTD, since the number of classes in the other two data sets was too small to construct a meaningful DTD scenario.

Name	K	$\#Train$	$\#Val$	$\#Test$	Avg. Length
AG News	4	120,000	8,000	7,600	40
TREC coarse	6	4,952	500	500	10
DBpedia	14	560,000	35,000	35,000	50

Table 1: Overview of the data sets used for evaluation.

3.5 Performance Metrics

Following prior work (Hsu et al., 2019; Lin et al., 2020), we report model performance as measured in Accuracy (ACC), Normalized Mutual Information (NMI; Strehl and Ghosh, 2002) and the Adjusted Rand Index (ARI; Steinley, 2004). For more in-depth explanations and for the formulas of all three metrics, please refer to Appendix C. All three metrics are normalized to $[0, 1]$, where higher values indicate better performance. Similarly, we use the Hungarian algorithm (Kuhn, 1955) to optimally map predicted labels to the true cluster assignments before calculating the performance metrics.

4 Experiments

In Table 2, we compare the CC models trained via both the MCL and the KCL loss with the lower and upper baselines. These results confirm that CC is a suitable method to train weakly supervised models for the detection of topics in short texts, reaching almost full supervision performance.

Furthermore, we investigated the capabilities of these models in the OC scenario, where the ground truth number of classes is unknown during training and the model can potentially assign $n_{out} = 30 >$

		Lower Baseline			CC-KCL			CC-MCL			Upper Baseline		
Data set	K	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
AG News	4	0.830	0.577	0.605	0.870	0.714	0.739	**0.917**	**0.755**	**0.795**	0.919	0.759	0.800
TREC-coarse	6	0.542	0.299	0.302	0.953	0.890	0.900	**0.967**	**0.908**	**0.923**	0.962	0.897	0.917
DBPedia	14	0.631	0.726	0.494	**0.982**	**0.963**	**0.967**	0.661	0.805	0.653	0.989	0.974	0.977

Table 2: Averaged results for the baselines on all available training samples as well as for CC-MCL and CC-KCL trained with $20,000$ constraints each. The better CC model (between KCL and MCL) is marked in bold and CC models almost reach full supervision level performance (upper baseline). Refer to a larger version of this table including standard deviations across runs in Appendix B, Table 6.

Dataset	ACC	NMI	ARI
AG News	0.821 ± 0.068	0.670 ± 0.033	0.677 ± 0.067
TREC coarse	0.912 ± 0.070	0.892 ± 0.057	0.882 ± 0.075
DBpedia	0.986 ± 0.002	0.966 ± 0.003	0.969 ± 0.003

Table 3: Mean results $\pm$ std. deviations over 5 repetitions for overclustering with $n_{out} = 30$. The model performs well despite the absence of the true K.

K potential clusters. From the results in Table 3, we observe that CC copes very well with this challenging scenario. This motivates the extension towards DTD.

Following Section 3.3, we train five *Phase 1* models $f_{i,t=1}$ on $D_{train,t=1}$ and evaluate their performance on the three different test sets using the DBPedia data set. We use the KCL loss due to its superior performance in the previous experiments. We observe a decent performance on $D_{test,1}$ along with a correctly detected number of classes in Table 4. Note, that we consider a class as 'detected' if the model assigns at least one percent of the respective test set to the specific cluster. We acknowledge that this is a rather heuristic choice. For $D_{test,2}$ and $D_{test,combined}$, the models perform substantially worse and are not able to detect the correct number of classes. Still, it is noteworthy, that the model is able to detect that the four novel classes in $D_{test,2}$ are distinct as it assigns them different clusters and does not simply assign them one 'outlier' cluster. From the observation that the model detects a total of ten clusters, as opposed to the correct $K = 14$ for $D_{test,combined}$, we infer that while it realizes these four new clusters are distinct, it assigns them to the clusters present in $D_{train,t=1}$. However, the *Phase 2* model $f_{t=2}$ obtained by fine-tuning the best performing *Phase 1* for 200 epochs on $10,000$ constraints sampled from $D_{train,t=2}$ (50% new vs. 50% old) performs very well on all three test sets and is able to detect the correct overall number of classes. Refer to the confusion matrices in Figure 2 for further illustration of these results. When $D_{train,t=2}$ contains more samples from the 'old' classes (25% new vs. 75% old), overall model performance still improves compared to *Phase 1*, but substantially less compared to when there is more information about the 'new' classes. These results imply that the algorithm shows considerable sensitivity to the degree of novelty present in the new training data, which has to be investigated further in future research. This experiment shows how an OC-KCL model can easily be adapted to a dynamically changing number of clusters via continued training on pairwise annotations from newly incoming training data.

5 Discussion and Conclusion

In this work, we connected two branches of research: contemporary NLP research and weakly supervised learning approaches. While the usefulness of CC-KCL (and MCL) had already been shown for computer vision settings (Hsu and Kira, 2016; Hsu et al., 2019), we extended it towards NLP. Based on this, we showcased how existing shortcomings of ordinary supervised approaches – the requirement of fixed, static label sets – could be regarded as a new type of learning task which we introduced as *dynamic topic discovery*. Within DTD, we subsume a dynamic setting where an initial, weakly annotated training data set at time $t = 1$ is accompanied by a second data set at time $t = 2$ which contains novel classes unseen at $t = 1$. We proposed a potential solution for such DTD settings via an alternative training scheme leveraging the overclustering and category detection capabilities of CC models. We acknowledge that there are still numerous unsolved problems such as the application on *very* short texts, *very* large label sets with large class cardinality, or multi-label scenarios. Nevertheless, we hope that our experimental results can serve as a foundation for further research toward tackling these increasingly complex problems to ultimately reduce manual labeling efforts in NLP.

	Test set	ACC	NMI	Predicted K
Phase 1	$D_{test,1}$	**0.988** / 0.982 ± 0.009	**0.969** / 0.964 ± 0.005	10 (Range: $[10-10]$)
(**Best** / Mean $\pm$ Std. Dev)	$D_{test,2}$	**0.616** / 0.570 ± 0.043	**0.409** / 0.410 ± 0.048	4 (Range: $[4-5]$)
	$D_{test,combined}$	**0.717** / 0.710 ± 0.011	**0.809** / 0.808 ± 0.015	10 (Range: $[10-11]$)

		50% new – 50% old			25% new – 75% old		
		ACC	NMI	Predicted K	ACC	NMI	Predicted K
Phase 2	$D_{test,1}$	0.980	0.951	10	0.880	0.895	9
	$D_{test,2}$	0.971	0.929	4	0.951	0.866	4
	$D_{test,combined}$	0.978	0.953	14	0.832	0.887	12

Table 4: DTD (with KCL) on DBpedia for different ratios of new versus old classes in $D_{train,t=2}$, from which we sample the 10,000 constraints for Phase 2, controlling the degree of novelty. Phase 1 is based on five different models on $D_{train,t=1}$. For Phase 2, we pick the best Phase 1 model and continue training on the constraints from $D_{train,t=2}$ (no standard deviations, since no random initialization of any model weights for Phase 2).

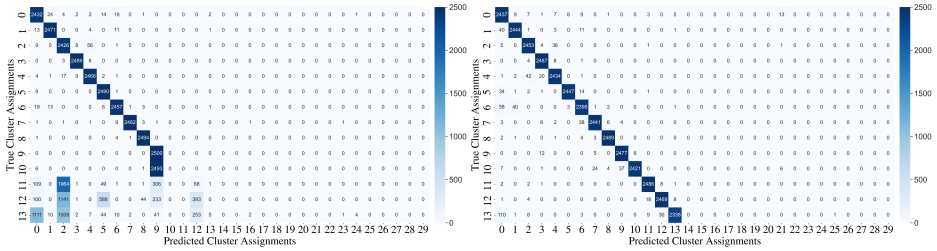


Figure 2: Confusion matrices for the two DTD phases on the $D_{test,combined}$. Phase 2 results (right) from the 50% new - 50% old setting illustrate a clear improvement over the results from Phase 1 (left). This shows that the Phase 2 model is able to cluster both the new and old data correctly.

Further, we believe that there is a high necessity for investigating DTD more in-depth. We believe it is important to design appropriate benchmarks and to investigate their relations to other dynamic paradigms, such as e.g. online learning or novel category discovery, and we hope this work can serve as a step in that direction.

Limitations

While we hope that this work provides valuable insights, there are still a couple of issues we did not yet address. First, we observed considerable instability during model training, especially for a lower number of constraints. Second, we found KCL to work better for DBpedia than MCL, which is surprising given the findings of Hsu et al. (2019). Finally, we (i) only evaluated DTD for one fixed set of constraints, (ii) only used the DBPedia dataset (due to the low number of classes in the other two datasets), and (iii) used a rather heuristic rule for determining the number of detected classes.

Acknowledgements

This work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of BERD@NFDI - grant number 460037581. This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy through the Center for Analytics – Data – Applications (ADACenter) within the framework of BAYERN DIGITAL II (20-3410-2-9-8).

References

David Arthur and Sergei Vassilvitskii. 2006. How slow is the k-means method? In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 144–153.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. 2020. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*.

Sugato Basu, Arindam Banerjee, and Raymond J Mooney. 2004. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004*

SIAM international conference on data mining, pages 333–344. SIAM.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Pierre Gançarski, Thi-Bich-Hanh Dao, Bruno Crémilleux, Germain Forestier, and Thomas Lampert. 2020. Constrained clustering: Current and new trends. In *A Guided Tour of Artificial Intelligence Research*, pages 447–484. Springer.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Yen-Chang Hsu and Zsolt Kira. 2015. Neural network-based clustering using pairwise constraints. *arXiv preprint arXiv:1511.06321*.

Yen-Chang Hsu and Zsolt Kira. 2016. Neural network-based clustering using pairwise constraints. *ICLR Workshop*.

Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. 2018. Learning to cluster in order to transfer across domains and tasks. *ICLR*.

Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. 2019. Multi-class classification without multi-class labels. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Shaohan Huang, Furu Wei, Lei Cui, Xingxing Zhang, and Ming Zhou. 2020. Unsupervised fine-tuning for text clustering. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5530–5534.

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

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8360–8367.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *International Conference on Learning Representations*.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Workshops*, Long Beach, CA, USA.

Tim Schopf, Daniel Braun, and Florian Matthes. 2021. Lbl2vec: An embedding-based approach for unsupervised document retrieval on predefined topics. In *WEBIST*, pages 124–132.

Douglas Steinley. 2004. Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3):386.

Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.

Kiri Wagstaff and Claire Cardie. 2000. Clustering with instance-level constraints. *AAAI/IAAI*, 1097:577–584.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021a. Discovering new intents with deep aligned clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14365–14373.

Hongjing Zhang, Tianyang Zhan, Sugato Basu, and Ian Davidson. 2021b. A framework for deep constrained clustering. *Data Mining and Knowledge Discovery*, 35(2):593–620.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Appendix

A Training Model Configurations

In Table 5 we list the specifications of the BERT-based language model that we use as architectural backbone which we obtained via huggingface (Wolf et al., 2020). We implemented our models and data loading logic in PyTorch (Paszke et al., 2017). Model training for the constrained clustering and the overclustering experiments was done on an NVIDIA A100-SXM4-40GB GPU with a batch size of 256 for 200 epochs. The models for the DTD part were trained on an NVIDIA Tesla-V100-16GB GPU with a batch size of 196 for 100 training epochs for phase 1 and for 200 training epochs for phase 2.

Parameter	Value
Base model	BERT-BASE-UNCASED
Learning rate	1×10^{-5}
Optimizer	AdamW (Loshchilov and Hutter, 2019)
Adam Epsilon	1×10^{-8}

Table 5: BERT configurations for all experiments.

B Detailed Results

In Table 6, we show results for the constrained clustering experiments with $n_{out} = K$ and a total of 20.000 constraint annotations for model training for the three datasets. This table includes mean $\pm$ standard deviations for the performance metrics across 5 repeated training runs to account for randomness in the training process. The results show that constrained clustering offers a viable alternative to supervised learning, almost reaching the upper baseline performance for the three datasets. Further, the MCL loss works best for the AGNews and the TREC-coarse datasets whereas the KCL loss is more suitable for the DBPedia dataset. Hence, we used the KCL loss in the experiments on DBPedia for the dynamic topic discovery experiments in Section 3.3.

C Performance metrics

Normalized Mutual Information (NMI) NMI is generally used to measure the tightness of the cluster formations. In other words, it quantifies if all the clusters are mutually exclusive without outliers (Strehl and Ghosh, 2002). Mathematically,

		Lower Baseline		
Data set	K	ACC	NMI	ARI
AG News	4	0.830	0.577	0.605
TREC-coarse	6	0.542	0.299	0.302
DBPedia	14	0.631	0.726	0.494
		CC-KCL		
AG News	4	0.870 ± 0.088	0.714 ± 0.059	0.739 ± 0.087
TREC-coarse	6	0.953 ± 0.007	0.890 ± 0.010	0.900 ± 0.012
DBPedia	14	$\mathbf{0.982} \pm 0.005$	$\mathbf{0.963} \pm 0.005$	$\mathbf{0.967} \pm 0.009$
		CC-MCL		
AG News	4	$\mathbf{0.917} \pm 0.003$	$\mathbf{0.755} \pm 0.004$	$\mathbf{0.795} \pm 0.006$
TREC-coarse	6	$\mathbf{0.967} \pm 0.004$	$\mathbf{0.908} \pm 0.009$	$\mathbf{0.923} \pm 0.009$
DBPedia	14	0.661 ± 0.057	0.805 ± 0.038	0.653 ± 0.055
		Upper Baseline		
AG News	4	0.919 ± 0.001	0.759 ± 0.005	0.800 ± 0.003
TREC-coarse	6	0.962 ± 0.002	0.897 ± 0.006	0.917 ± 0.005
DBPedia	14	0.989 ± 0.001	0.974 ± 0.001	0.977 ± 0.001

Table 6: Results for the baselines on all available training samples for all of the analyzed data sets as well as for CC-MCL and CC-KCL on 20,000 constraints each. The better CC model (between KCL and MCL) is marked in bold. Mean and standard deviations of the metrics over five runs.

NMI describes the change in entropy of class labels given the true cluster labels:

$$NMI = \frac{2 \cdot I(Y, \hat{Y})}{H(Y) + H(\hat{Y})}$$

where $I(Y, \hat{Y}) = H(Y) - H(Y|\hat{Y})$ is the mutual information. $H(Y)$ and $H(\hat{Y})$ are the entropy of the ground truth class label Y distribution and the entropy of the predicted cluster label distribution $\hat{Y}$, respectively. The NMI is bound to $[0, 1]$ where a higher score implies better clustering performance.

Accuracy (ACC) Accuracy measures the similarity of predicted results with the respective ground truth. For clustering accuracy, we use the Hungarian algorithm (Kuhn, 1955) to assign predicted clusters with associated class labels. Given ground truth classes Y and predicted clusters $\hat{Y}$ we calculate accuracy as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

Adjusted Rand Index (ARI) The ARI is used to measure the similarity between two clustering outputs (Steinley, 2004). Here, the actual class labels

are compared to predicted cluster labels to measure the clustering performance. When comparing Y and $\hat{Y}$, the ARI is calculated as follows:

$$R = \frac{a+b}{\binom{n}{2}}$$

where a is the number of times, pairs of elements are in the same cluster for Y and $\hat{Y}$, b is the number of times a pair of elements is not in the same cluster for Y and $\hat{Y}$ and n is the total number of samples in the batch.

4.5 Wearable-based Parkinson's Disease Severity Monitoring using Deep Learning

Contributing article:

Jann Goschenhofer, Franz MJ Pfister, Kamer Ali Yuksel, Bernd Bischl, Urban Fietzek, and Janek Thomas. 2019. Wearable-based parkinson's disease severity monitoring using deep learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 400–415. Springer

Author contributions:

Jann Goschenhofer was responsible for the conceptualization of this paper, supported by Kamer Yuksel, Janek Thomas, and Franz Pfister. Franz Pfister and Urban Fietzek provided the medical background knowledge for this project and were also responsible for the data collection and data curation process. Jann Goschenhofer was responsible for the implementation of the different modeling approaches, supported by Kamer Yuksel. The design for the experiments as well as the evaluation strategy was developed by Jann Goschenhofer, Kamer Yuksel, Janek Thomas, and Bernd Bischl. Jann Goschenhofer was responsible for the running of the experiments, the tuning, and the evaluation of the different modeling approaches. Jann Goschenhofer was responsible for the drafting of the manuscript, Urban Fietzek contributed by placing the topic in the correct medical context. Janek Thomas, Kamer Yuksel, Franz Pfister, and Bernd Bischl contributed to reviewing and proofreading the manuscript. Bernd Bischl also contributed by supervision.

Statement on difference between paper and master's thesis: This paper is based on the master's thesis written by Jann Goschenhofer with the title "Wearable-based Severity Detection in the Context of Parkinson's Disease using Deep Learning Techniques" at the LMU Munich submitted on January 5, 2019. The master's thesis was supervised by Janek Thomas, Kamer Yuksel, Franz Pfister, and Bernd Bischl. The goal of this master thesis was to explore machine learning techniques to predict clinical disease scores from movement data for patients with Parkinson's disease. In the master thesis, different ways to frame this problem as well as different machine learning and deep learning approaches were compared using a tailored evaluation strategy. Also, a customized loss that reflects the requirements of the involved medical staff, as well as the high class imbalancedness in the data, was developed. Furthermore, different approaches for uncertainty quantification were compared in the original master thesis. For the paper version, all figures were re-created and refined, the results were condensed into a more concise format for presentation. The paper version was written from scratch and extensive additional literature research was performed. Furthermore, the focus was put on the modeling task and the comparison of the different modeling paradigms, including a section on transfer learning from a weakly supervised task.

Copyright information:

Wearable-Based Parkinson's Disease Severity Monitoring Using Deep Learning

Jann Goschenhofer[1,2( )], Franz M. J. Pfister[1,2], Kamer Ali Yuksel[2], Bernd Bischl[1], Urban Fietzek[3,4], and Janek Thomas[1]

[1] Department of Statistics, Ludwig-Maximilians University, Munich, Germany
jann.goschenhofer@stat.uni-muenchen.de
[2] ConnectedLife GmbH, Munich, Germany
[3] Department of Neurology, Ludwig-Maximilians University, Munich, Germany
[4] Department of Neurology and Clinical Neurophysiology, Schoen Clinic Schwabing, Munich, Germany

Abstract. One major challenge in the medication of Parkinson's disease is that the severity of the disease, reflected in the patients' motor state, cannot be measured using accessible biomarkers. Therefore, we develop and examine a variety of statistical models to detect the motor state of such patients based on sensor data from a wearable device. We find that deep learning models consistently outperform a classical machine learning model applied on hand-crafted features in this time series classification task. Furthermore, our results suggest that treating this problem as a regression instead of an ordinal regression or a classification task is most appropriate. For consistent model evaluation and training, we adopt the leave-one-subject-out validation scheme to the training of deep learning models. We also employ a class-weighting scheme to successfully mitigate the problem of high multi-class imbalances in this domain. In addition, we propose a customized performance measure that reflects the requirements of the involved medical staff on the model. To solve the problem of limited availability of high quality training data, we propose a transfer learning technique which helps to improve model performance substantially. Our results suggest that deep learning techniques offer a high potential to autonomously detect motor states of patients with Parkinson's disease.

Keywords: Motor state detection · Sensor data · Time series classification · Deep learning · Personalized medicine · Transfer learning

1 Introduction

Parkinson's disease (PD) is one of the most common diseases of the elderly and the second most common neurodegenerative disease in general after Alzheimer's [38]. Two million Europeans are affected and 1% of the population over the age of 60 in industrial nations are estimated to suffer from PD [1,36]. Fortunately, the disease can be managed by applying the correct personalized dosage

U. Brefeld et al. (Eds.): ECML PKDD 2019, LNAI 11908, pp. 400–415, 2020.
https://doi.org/10.1007/978-3-030-46133-1_24

and schedule of medication, which has to be continuously adapted regarding the progress of this neurodegenerative disease. Crucial for the optimal medication is knowledge about the current latent motor state of the patients, which can not yet be measured effortlessly, autonomously and continuously. The motoric capabilities of the patients are distinguishable into three different motor states which can vary substantially over the course of a day within hours. The most prominent symptom is the tremor but the disease defining symptom is the loss of amplitude and slowness of movement, also referred as bradykinesia [35]. In contrast to bradykinesia, an overpresence of dopaminergic medication can make affected patients execute involuntary excessive movement patterns which may remind an untrained observer of a bizarre dance. This hyperkinetic motor state is termed dyskinesia [40]. In a very basic approximation, people with Parkinson's disease (PwP) can be in three motor states: (1) the bradykinetic state (OFF), (2) a state without appearant symptoms (ON), and (3) the dyskinetic state (DYS) [31]. If the true motor state of PwP was known at all times, the medication dose could be optimized in such a way, that the patient has an improved chance to spend the entirety of his waking day in the ON state. An example for such a closed-loop approach can be found in Diabetes therapy, where the blood sugar level serves as a biomarker for the disease severity. Patients suffering from Diabetes can continuously measure their blood sugar level and apply the individual, correct medication dose of insulin in order to balance the disease. Analogously, an inexpensive, autonomous and precise method to assess the motor state might allow for major improvements in personalized, individual medication of PwP.

Advancements in both wearable devices equipped with motion sensors and statistical modeling tools accelerated the scientific community in researching solutions for motor state detection of PwP since the early 2000s. In 1993, Ghika et al. did pioneering work in this field by proposing a first computer-based system for tremor measurement [14]. A comprehensive overview on the use of machine learning and wearable devices in a variety of PD related problems was recently provided by Ahlrichs et al. [1]. A variety of studies compare machine learning approaches applied on hand-crafted features with deep learning techniques where the latter show the strongest performance [9,20,24–27,38,40,41]. In the present setting, a leave-one-subject-out (LOSO) validation is necessary to yield unbiased performance estimates of the models [37]. Thus, it is surprising that only a subset of the reviewed literature deploys a valid LOSO validation scheme [9,24,25,40,41]. It is noteworthy that one work proposes modeling approaches with a continuous response [26], while the rest of the literature tackles this problem as a classification task to distinguish between the different motor states. Amongst the deep learning approaches, it is surprising that none of the related investigations describe their method to tune the optimal amount of training epochs for the model, which is not a trivial problem as discussed in Sect. 3.3. A strutured overview on the related literature is given in Table 1.

Contributions. This paper closes the main literature gaps in machine learning based monitoring of PD: the optimal problem setting for this task is discussed,

Table 1. Overview on results from the literature on Motor State detection for PwP. In the method column, the MLP refers to a Multi-layer Perceptron, FE to manual feature extraction, SVM to a Support Vector Machine, RF to a Random Forest and LSTM for Long-short-term-memory network. In the label column, the names of the class labels are depicted. From this column, one can infer that only two authors used continuous labels and thus regression models for their task. Generally, a comparison of the reviewed approaches is difficult due to high variation in the data sets, methods and evaluation criteria.

Author	Method	Validation	Subjects	Sensors	Position	Setting	Labels	Results
[25]	FE, SVM	LOSO	19	6	Arist, Ankle	Lab	ON, OFF	Acc.: 90.5%
[41]	CNN	LOSO	30	1	Wrist	Free	OFF, ON, DYS	Acc.: 63.1%
[38]	FE, SVM	Holdout Patients	20	1	Belt	Lab	ON, OFF	Acc.: 94.0%
[24]	LSTM	LOSO	12	1	Ankle	Free	ON, OFF	Acc.: 73.9%
	FE, SVM	LOSO	12	1	Ankle	Free	ON, OFF	Acc.: 65.7%
[9]	CNN	LOSO	10	2	Wrist	Lab	ON, OFF	Acc.: 90.9%
[19]	FE, MLP	Leave-one-day-out	34	2	Wrist	Free	OFF, ON, DYS, Sleep	F1: 55%
[20]	FE, MLP	7-fold CV	34	2	Wrist	Lab	OFF, ON, DYS, Sleep	F1: 76%
[27]	FE, MLP	Train set	23	6	Trunk, wrist, leg	Lab	ON, OFF	F1: 97%
[40]	FE, MLP	LOSO	29	6	Wrist, leg, chest, waist	Lab	DYS Y/N	Acc.: 84.3%
[26]	FE, MLP	80/20 Split	13	6	Trunk, wrist, leg	Free	Continuous	Acc.: 77%
[30]	FE, RF	LOSO	20	1	Wrist	Lab	ON, OFF	AUC: 0.73
	FE, RF	LOSO	20	1	Wrist	Lab	Tremor Y/N	AUC: 0.79
Our approach[a]	CNN	LOSO	28	1	Wrist	Free	Continuous	MAE: 0.77
	CNN	LOSO	28	1	Wrist	Free	9-class	± 1 Acc.: 86.95%

[a] Performance measures are detailed in Sect. 5

a customized performance measure is introduced and a valid LOSO validation strategy is applied to compare time series classification (TSC) deep learning and classical machine learning approaches. Furthermore, the application of a transfer learning strategy in this domain is investigated.

This paper is structured as follows: The used data sets are described in Sect. 2. In Sect. 3, peculiarities of the problem as well as the transfer learning strategy are discussed. Furthermore, in Sect. 4 model architectures and problem settings are proposed and their results are discussed in Sect. 5.

2 Data

Data was collected from PwP to model the relation between raw movement sensor data and motor states. The acceleration and rotation of patient's wrists was measured via inertial measurement units (IMUs) integrated in the Microsoft band 2 fitness tracker [32] with a standard frequency of 62.5 Hz. The wrist was chosen as sensor location as it is the most comfortable location for a wearable device to be used in the patients' daily lifes and was shown to be sufficient for the detection of Parkinson-related symptoms [7,30]. The raw sensor data was downsampled to a frequency of 20 Hz as PD related patterns do not exceed this frequency [20]. A standard procedure in human activity recognition is the segmentation of continuous sensor data streams into smaller windows. As the data in this study was annotated by a medical doctor on a minute-level, the window length was set to one minute. To increase the amount of training data, the windows were segmented with an overlap of 80% which is in line with related literature [9,19,44]. To neutralize any direction-specific information, the L_2-norms of the accelerometer and gyroscope measurements are used as model input, leading to two time series per window. Finally, the data was normalized to a $[0, 1]$ range via quantile transformation.

We consider the machine learning problem of the feature space $\mathcal{X} \subset \mathbb{R}^p$, with $p = 1200 \cdot 2$, a target space $\mathcal{Y}$ described below and a performance measure $\mathcal{P} : \mathcal{Y} \times f(\mathcal{X}) \to \mathbb{R}$ measuring the prediction quality of a model $f : \mathcal{X} \to \mathcal{Y}$, trained on the data set $\mathcal{D} = \left\{(x^{(1)}, y^{(1)}), ..., (x^{(n)}, y^{(n)})\right\}$ where a tuple $\left(x^{(i)}, y^{(i)}\right) \in \mathcal{X} \times \mathcal{Y}, i = 1, ..., n$ refers to a single labeled one minute window with a frequency of 20 Hz.

The disease severity $\mathcal{Y}$ is measured on a combined version of the UPDRS [16] and the mAIMS scale [29]. The UPDRS scale is based on a diagnostic questionnaire for physicians to rate the severity of the bradykinesia of PwP on a scale with 0 representing the ON state to 4, the severly bradykinetic state. The mAIMS scale is analogue to the UPDRS, but in contrast used for the clinical evaluation of dyskinetic symptoms. Both scales were combined and the UPDRS scale was flipped to cover the whole disease spectrum. The resulting label scale takes values in $\mathcal{Y} = \{-4, ..., 4\}$ where $y^{(i)} = -4$ means a patient is in a severely bradykinetic state, $y^{(i)} = 0$ is assigned to a patient in the ON state and $y^{(i)} = 4$ resembles a severely dyskinetic motor state. The sensor data was labeled by a medical doctor who shadowed the PwP during one day in a free living setting. Thus, the rater monitored each patient, equipped with an IMU, while they

performed regular daily activities and the rater clinically evaluated the patients' motor state at each minute. In total, 9356 windows were extracted from the data of 28 PwP. By applying the above described preprocessing steps, the amount of windows was increased to 45944.

3 Challenges

3.1 Class Imbalance

The labeled data set suffers from high label imbalance towards the center of the scale as shown in Fig. 1. Thus, machine learning models will be biased towards predicting the majority classes [21].

A straightforward way of dealing with this problem is to reweight the loss contribution of different training data samples. This way, the algorithm incurs heavier loss for errors on samples from minority classes than for those of majority classes, putting more focus on the minority classes during training. The weights for the classes $j \in \mathcal{Y} = \{-4, ..., 4\}$ are calculated as follows:

$$c_j = \frac{n}{n_j}; \quad \tilde{c}_j = |\mathcal{Y}| \cdot \frac{c_j}{\sum_{j \in \mathcal{Y}} c_j} \qquad (1)$$

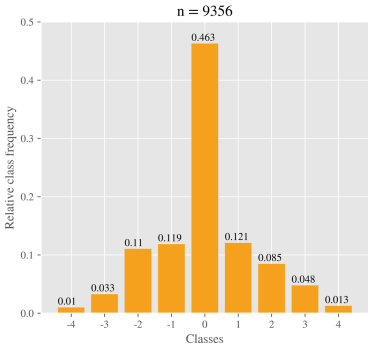


Fig. 1. Label distribution of the data which is highly centered around $y = 0$.

where $|\mathcal{Y}|$ describes the amount of classes, n is the total amount of samples, n_j the amount of samples for class j and thus c_j is the inverse relative frequency of class j in the data. Further, the weights $c_j, j \in \mathcal{Y}$ are normalized such that the sum of the weights is equal to the amount of classes. The individual weight of one sample is referred to as $\omega^{(i)}$ which is the normalized weight $\tilde{c}_j$ associated with the label $y^{(i)}$ of this sample i such that $y^{(i)} = j$.

3.2 Custom Performance Measure

It is crucial for the practical application of the final model to select an adequate performance measure which reflects the practical requirements on the model. Based on discussions with involved medical doctors, we found that larger errors should be penalized heavier which implies a quadratic error. Additionally, errors in the wrong direction of the scale, e.g. $\hat{y}^{(i)} = -1, y^{(i)} = 1$, should have a higher negative impact than errors with the same absolute distance in the correct direction, e.g. $\hat{y}^{(i)} = 3, y^{(i)} = 1$. The rationale behind this is that an exaggerated diagnostic evaluation which follows the true pathological scenario harms the patient less than an opposing one. Furthermore, the cost of predicting the wrong pathological direction increases with the severity of the disease: diagnostic errors

weigh heavier on patients with strong symptoms compared to patients that are only mildly affected by the disease. In summary, three main requirements on the custom performance measure were identified: non-linearity, asymmetry and not being translation invariant.

Inspired by econometric forecasting [8], the following asymmetric performance measure, which satisfies the first two previous requirements, is introduced:

$$P_\alpha(\mathcal{D}, f) = \frac{1}{|\mathcal{D}|} \sum_{x^{(i)}, y^{(i)} \in \mathcal{D}} \left[\alpha + sign\left(y^{(i)} - f(x^{(i)})\right)\right]^2 \left(f(x^{(i)}) - y^{(i)}\right)^2 \quad (2)$$

where $\alpha \in [-1, 1]$ controls the asymmetry such that:

$$\alpha \begin{cases} \in [-1, 0[, & \text{penalization of underestimation,} \\ = 0, & \text{symmetric loss,} \\ \in]0, 1], & \text{penalization of overestimation.} \end{cases} \quad (3)$$

This performance measure is the squared error multiplied by a factor that depends on the parameter α and on the over- or underestimation of the true label via the *sign* function. As motivated in the third requirement, the asymmetry should depend on the true label values. Therefore, y is connected with α by introducing α^* such that $\alpha = \frac{y^{(i)}}{4}\alpha^*$ where $y^{(i)} \in \mathcal{Y} = \{-4, ..., 4\}$, hence $\alpha^* \in [0, 1]$. The constant denominator 4 is used to link α and α^* in such a way that the sign of α that governs the direction of the asymmetric penalization is controlled by the true labels y. This leads to the formalization:

$$P_{\alpha^*}(\mathcal{D}, \hat{f}) = \frac{1}{|\mathcal{D}|} \sum_{x^{(i)}, y^{(i)} \in \mathcal{D}} \left[\frac{y^{(i)}}{4}\alpha^* + sign\left(y^{(i)} - \hat{f}(x^{(i)})\right)\right]^2 \left(\hat{f}(x^{(i)}) - y^{(i)}\right)^2 \quad (4)$$

The parameter $\alpha^* = 0.25$ was set based on the feedback of the involved medical experts[1]. The model will be heavily penalized for the overestimation of negative labels and for the underestimation of positive labels. For instance, the performance measure for $y^{(i)} = 2$ and prediction $\hat{y}^{(i)} = 1$ is higher (1.265) than for $\hat{y}^{(i)} = 3$ (0.765). The asymmetry of the measure is reciprocally connected to the magnitude of the label y in both, the negative as well as the positive direction, e.g. for $y^{(i)} = 1$ it is more symmetric than for $y^{(i)} = 3$. Furthermore, P_{α^*} collapses to a regular quadratic error for $y^{(i)} = 0$. The behavior of the measure is further illustrated in Fig. 2.

3.3 Leave-One-Subject-Out Validation

Proposed models are expected to perform well on data from patients not seen before. Using regular cross validation (CV) strategies, subject-specific information could be exploited resulting in an overly optimistic estimate of the generalization performance [37]. Consequently, a leave-one-subject-out (LOSO) validation scheme is often applied in settings where much data are gathered from few

[1] Feedback was collected by comparing multiple cost matrices as shown in Fig. 3.

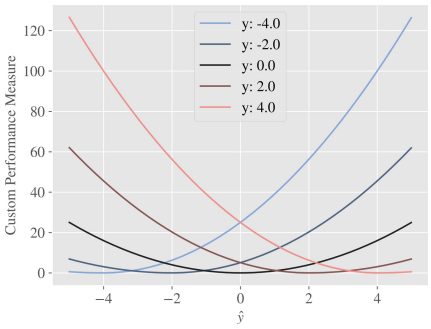


Fig. 2. Behavior of the performance measure $P_{\alpha^*=0.25}$ on the y-axis for different labels y and the corresponding predictions $\hat{y}$ on the x-axis.

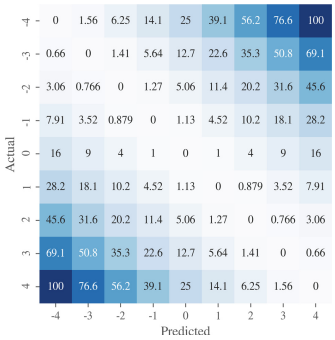


Fig. 3. Cost factors resulting from $P_{\alpha^*=0.25}$ that are associated with each combination of actual and predicted values.

subjects [2,9,12]. Thereby, a model is trained on all except one subject and then tested on the left out subject, yielding an unbiased performance estimate. This is repeated for each individual subject and all resulting estimates are averaged.

The usage of early stopping [17] requires the introduction of a tuning step to determine the optimal amount of training epochs e^* in each of the LOSO folds, which in turn requires a second inner split of the data set. In a setting with unlimited computational resources, one would run a proper LOSO validation in the inner folds, determine e^*, train the model on the whole data except the left out subject and evaluate the trained model on that subject. With a total amount of 28 patients, this would result in the training of $28 \cdot 27 = 756$ models for the validation of one specific architecture. As a cheaper solution, the first 80% one minute windows per patient are used for training and the last 20% for early stopping.

3.4 Transfer Learning

One of the most important requirements for the successful training of deep neural networks with strong generalization performance is the availability of a large amount of train data. Next to strong regularization and data set augmentation, one prominent method to fight overfitting and improve the model's generalization performance is transfer learning [43]. A model architecture is first trained on source task $\mathcal{D}_A$. The learned knowledge, manifested in the model's weights, is used to initialize a model that should be trained on the target task $\mathcal{D}_B$. The model is then fine-tuned on $\mathcal{D}_B$ which often leads to faster model convergence and, dependent on the similarity of the tasks, to an improvement in model performance. Though TSC is still an emerging topic in the deep learning community, first investigations into the adoption of transfer learning to time series data have been made [11].

As a source task for the motor state detection, we train the model to classify between one-minute windows that were either gathered from PwP or from healthy patients. Therefore, we use a weakly labeled data set that contains 70175 one-minute windows of sensor data along with the binary target if the corresponding patient suffers from Parkinson's disease or not. Among those patients, 50% were healthy and 50% suffered from PD. The proposed deep learning models were trained on this task and their weights were used for initialization of the models which were then fine-tuned on the actual data as described in Sect. 5.

4 Problem Setting and Models

4.1 Problem Setting

As explained in Sect. 2, the target was measured on a discrete scale $y \in \mathcal{Y} = \{-4, ..., 4\}$ where $y = -4$ represents severe bradykinesia, $y = 0$ the ON state and $y = 4$ severe dyskinesia. This gives rise to the question whether the problem should be modeled as a classification, an ordinal regression or a regression task. The majority of previous research in this domain treats the problem as binary sub-problems with the goal to just detect whether the PwP experience symptoms, regardless of their severity. The granular labeling scheme used here follows an ordinal structure. For instance, a patient with $y = -4$ suffers from more severe bradykinesia than one with $y = -3$. In contrast, simple multi-class classification treats all class labels as if they were unordered. A simple way of including this ordinal information is to treat the labels as if they were on a metric scale and apply standard regression methods. However, this implies a linear relationship between the levels of the labels. For example, a change in the motor state from $y = -4$ to $y = -3$, $\delta_{-4,-3}$, could have a different meaning than $\delta_{-2,-1}$, though they would be equivalent on a metric scale. The formally correct framing of such problems is ordinal regression which takes into account the ordered structure of the target but does not make the strong linearity assumption [18]. This model class is methodologically located at the intersection of classification and metric regression. All three problem settings are compared in Sect. 5.

4.2 Models

Random Forest. A Random Forest [3] was trained on manually extracted features from the raw sensor data, similar to related literature [9,20,24,38]. From each sample window of both signal norms, a total of 34 features such as mean, variance and energy were extracted (a complete list can be found in the digital Appendix). This is a standard procedure in TSC [4,6]. The Random Forest was specifically chosen as a machine learning baseline due to its low dependency on hyperparameter settings and its strong performance in general.

FCN. The Fully Convolutional Net (FCN) was introduced as a strong baseline deep learning architecture for TSC [42]. The implementation resembles that of Wang et al. except that the final layer consists of $|\mathcal{Y}| = 9$ or 1 neuron(s) for classification and regression, respectively.

FCN Inception. Inception modules led to substantial performance increases in computer vision and are motivated by the observation that the kernel size of the convolutional layers are often chosen rather arbitrarily by the deep learning practitioner [39]. The rationale is to give the model the opportunity to choose from different kernel sizes for each convolutional block and distribute the amount of propagated information amongst the different kernels. One inception module consists of branches with with kernel sizes $1, 5, 7$ and 13 respectively and a depth of 64 each, plus one additional max-pooling branch with a kernel size of 3, followed by a convolution block with depth 64 and a kernel size 1. The final FCN Inception architecture essentially follows the original FCN with simple convolutional layers being replaced by 1D inception modules.

FCN ResNet. Similar to the inception modules, the introduction of residual learning has met with great enthusiasm in the deep learning community [22]. The main advantage of such Residual Networks (ResNet) over regular CNNs is the usage of skip-connections between subsequent layers. These allow the information to flow around layers and skip them in case they do not contribute to the model performance, which makes it possible to train much deeper networks. Unlike inception modules, this model class was already adapted for TSC and proven to be a strong competitor for the original FCN [42]. The FCN ResNet was shown to outperform the standard FCN especially in multivariate TSC problems [10]. Others argue that the ResNet is prone to overfitting and thus found it to perform worse than the FCN [42]. For the comparison in Sect. 5, three residual modules are stacked where each of the modules is identical to the standard FCN in order to provide comparability among architectures. The module depths were chosen as proposed by Wang et al. [42].

FCN Broad. Pathologically, the disease severity changes rather slowly over time. Thus, it can be hypothesized that additional input information and a broader view on the data could be beneficial for the model. This model is referred to as FCN Broad and includes the following extension: the raw input data from the previous sample window x_{t-1} and the following sample window x_{t+1} are padded to the initial sample window x_t, which results in a channel depth of 6 for the input layer.

FCN Multioutput. A broad variety of techniques for ordered regression exist [5,13,23,33]. As a neural network based approach for ordered regression is required, a simple architecture is to create a single CNN, which is trained jointly on a variety of binary ranking-based sub-tasks [33]. A key element to allow the network to exploit the ordinal structure in the data is a rank-based transformation of labels. The categorical labels $y \in \mathcal{Y}$ are transformed into $K = |\mathcal{Y}| - 1$ rank-based labels by:

$$y_k^{(i)} = \begin{cases} 1, & \text{if } y^{(i)} > r_k \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where r_k is the rank for the k-th sub-problem for $k \in \{1, ..., K\}$. Following this label transformation, a multi-output CNN architecture is proposed where each of the K outputs refers to one binary ranking-based sub-task. These are optimized jointly on a single CNN corpus. Thus, the sub-task k is trained on a binary classification problem minimizing the binary cross entropy loss. The total model output consists of K probability outputs for each input sample. In order to train the CNN jointly on those sub-tasks, the individual losses are combined to one cumulative loss:

$$L^{\text{ranks}}(y^{(i)}, f(x^{(i)})) = \sum_{k=1}^{K} L_k^b(y_k^{(i)}, \hat{y}_k^{(i)}) \tag{6}$$

where L_k^b is the binary cross-entropy loss for sub-task output $\hat{y}_k^{(i)}$. For inference, the K outputs are summed up such that $\hat{y}^{(i)} = \sum_{k=1}^{K} \hat{y}_k^{(i)} - 4$, where the scalar 4 is subtracted from the sum over all probability outputs to map the predictions back to the initial label scale, yielding a continuous output.

FCN Ordinal. A second ordinal regression model was created by training a regular FCN with an additional distance-based weighting factor in the multi-class cross entropy loss L^{m}:

$$L^{\text{ordinal}}(y^{(i)}, f(x^{(i)})) = \left| y^{(i)} - \hat{y}^{(i)} \right| \cdot L^{\text{m}}(y^{(i)}, \hat{y}^{(i)}) \tag{7}$$

This way, the model is forced to learn the inherent ordinal structure of the data as it is penalized higher for predictions that are very distant to the true labels.

5 Results

The models described in Sect. 4 were implemented in pytorch [34]. Model weights were initialized by Xavier-uniform initialization [15] and ADAM [28] (learning rate = 0.00005, $\beta_1 = 0.9$, $\beta_2 = 0.99$) was used for training with a weight decay of 10^{-6}. The performances of the models were compared in a LOSO evaluation as discussed in Sect. 3.3, using the performance measure $P_{\alpha^*=0.25}$ as introduced in Sect. 3.2. Finally, the sequence of motor state predictions is smoothed via a Gaussian filter whose μ and σ parameters were optimized using the same LOSO scheme that was used for model training. The results are summarized in Table 2. An additional majority voting model which constantly predicts $\hat{y} = 0$ is added as a naive baseline.

The FCN was applied in all three problem settings. From Table 2, one can observe that regression performs better than ordered regression and classification. Similar results were obtained for the Random Forest baseline, where regression is superior to classification. It seems that the simple assumption of linearity

Table 2. Results for different models in multiple problem settings, measured using the performance measure introduced in Sect. 3.2 evaluated by LOSO validation. Additional commonly used performance measures are shown for completeness where the MAE is reported in a class-weighted (MAE w.) and a regular version and Acc. ±1 refers to accuracy relaxed by one class level.

Frame	Model	$P_{\alpha^*} = 0.25$	F1	Acc.	Acc. ±1	MAE w.	MAE
Baseline	Majority vote	2.900	0.293	0.702	0.463	0.661	0.960
Classification	FCN	0.800	0.366	0.809	0.340	0.312	0.890
	Random Forest	1.542	0.394	0.802	0.459	0.465	0.802
Ordinal	FCN	0.752	0.321	0.767	0.302	0.311	0.985
	Multioutput FCN	0.922	0.361	0.820	0.352	0.344	0.873
Regression	FCN	**0.635**	0.346	0.843	0.338	0.293	0.836
	FCN Inception	0.726	0.380	0.841	0.370	0.304	0.842
	FCN ResNet	0.841	0.334	0.809	0.309	0.336	0.924
	FCN Broad	0.673	0.347	0.835	0.339	0.294	0.852
	Random Forest	1.310	0.411	0.848	0.436	0.423	0.760

between labels does not have a derogatory effect and a simpler model architecture as well as training process is of larger importance.

The comparison of the deep learning models with the Random Forest offers another interesting finding. For both, regression and classification, all deep learning models outperform the classic machine learning models. This finding justifies the focus on deep learning approaches and is in line with previous research discussed in the Introduction.

Niu et al. [33] claim that the Multioutput CNN architecture outperforms regular regression models in ordinal regression tasks. This can not be supported by the current results as the Multioutput FCN shows weaker performance than each of the deep learning architectures in the regression frame.

Looking at the results from the regression setting, one can observe that the simple FCN manages to outperform all more complex architectures as well as the Random Forest baseline. This could be explained by the increased complexity of these models: the FCN consists of $283,145$ weights, while the FCN Inception contains $514,809$ and the FCN ResNet $512,385$ weights. This problem is aggravated by the limited amount of training data.

As shown in Table 3, the transfer learning approach consistently improved the performance of all tested FCN architectures. This strategy also helped to further push the best achieved performance by the regression FCN, making it the overall best performing model. Transfer learning has the biggest effect on the performance of the Multioutput FCN, which indicates that this model requires a higher amount of training data. This is reasonable as it is arguably the most complex model considered. Further increasing the amount of training data might improve these complex models even more.

Table 3. Performance of the transfer learning approaches compared to their non-pretrained counterparts. Transfer learning consistently improves model performances. Additional commonly used measures are shown for the pretrained models only where the MAE is reported in a class-weighted (MAE w.) and a regular version and ±1 Acc. refers to accuracy relaxed by one class level.

Frame	Model	$P_{\alpha^*=0.25}$		Gain	F1	Acc.	Acc. ±1	MAE w.	MAE
		Regular	Transfer						
Classification	FCN	0.800	0.771	0.029	0.375	0.361	0.813	0.318	0.897
Ordinal	FCN	0.752	0.616	0.136	0.350	0.326	0.802	0.295	0.921
	Multioutput FCN	0.922	0.657	0.265	0.367	0.360	0.829	0.301	0.857
Regression	FCN	0.635	**0.600**	0.035	0.407	0.388	0.870	0.273	0.772

Some resulting predictions[2] from the best performing model are illustrated in Fig. 5 and a confusion matrix of the model predictions is shown in Fig. 4. It is noteworthy that despite the class weighting scheme and the transfer learning efforts, the final model fails in correctly predicting the most extreme class labels.

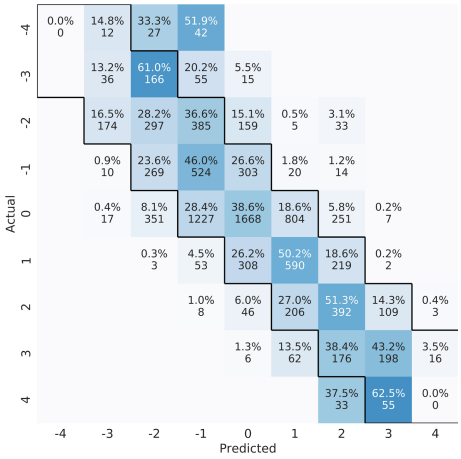


Fig. 4. Row-normalized confusion matrix for predictions from the pretrained regression FCN. Predicted continuous scores were rounded to integers. Allowing for deviations of ±1 (framed diagonal region) yields a relaxed accuracy of 86.96%.

[2] Results on all patients can be found here: https://doi.org/10.6084/m9.figshare.8313149.v1.

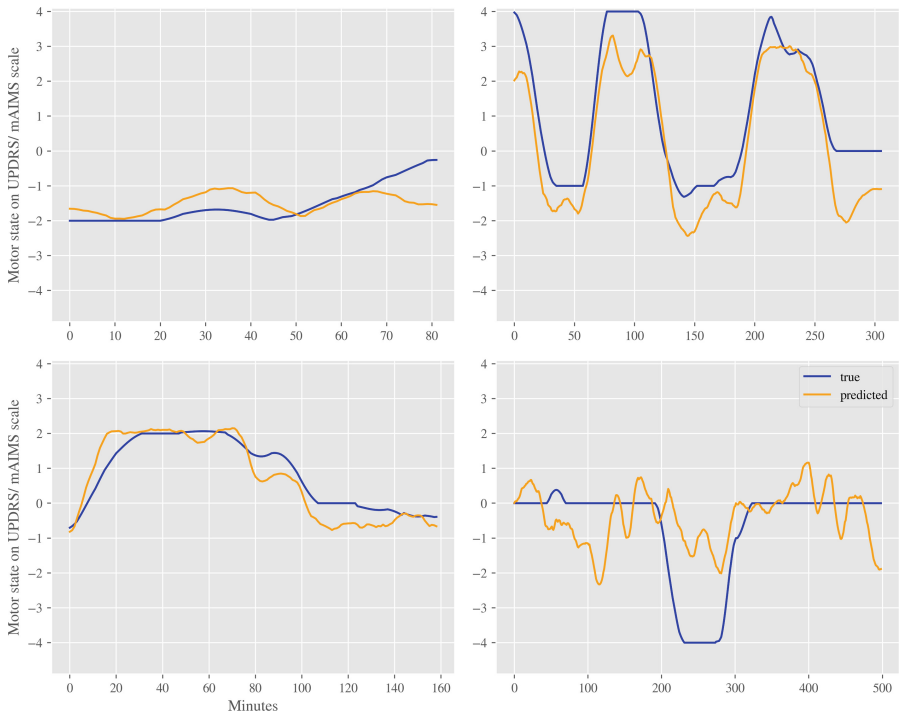


Fig. 5. Comparison of true (blue) and predicted (orange) motor state sequences of four exemplary patients. The label scores are depicted on the y-axis and the minutes on the x-axis. The final model is able to capture the intra-day motor state regime changes of the PwP as shown on the top right plot. Still, the model fails to correctly detect the motor states in some patients e.g. the bottom right one. (Color figure online)

6 Conclusion

Different machine learning and deep learning approaches were evaluated on the task to detect motor states of PwP based on wearable sensor data. While the majority of related literature handles the problem as a classification task, the high quality and resolution of the provided data allows evaluation in different problem settings. Framing the problem as a regression task was shown to result in better performance than ordered regression and classification. Evaluation was done using a leave-one-patient-out validation strategy on 28 PwP using a customized performance measure, developed in cooperation with medical experts in the PD domain. The deep learning approaches outperformed the classic machine learning approach. Furthermore, the comparatively simple FCN offered the most promising results. A possible explanation would be that these intricate models call for more available data for successful training. Since high quality labeled data are scarce and costly in the medical domain, this is not easily achievable.

First investigations into transfer learning approaches were successfully employed and showed model improvements for the deep learning approaches.

There exists a plethora of future work to investigate. Computational limitations made it impossible to evaluate all possible models in all problem settings as well as investigate recurrent neural network approaches. The successful usage of a weakly labeled data set for transfer learning suggests further research on the application of semi-supervised learning strategies. This work clearly shows the difficulty in fairly and accurately comparing existing approaches, as available data, problem setting and evaluation criteria differ widely between publications. The introduced performance measure could be a step into the right direction and can hopefully become a reasonable standard for the comparison of such models. In future work, one could directly use this performance measure as a loss function to train deep neural networks instead of using it for evaluation only.

Acknowledgements. This work was financially supported by ConnectedLife GmbH and we thank the Schoen Klinik Muenchen Schwabing for the invaluable access to medical expert knowledge and the collection of the data set. This work has been partially supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A, and by an unrestricted grant from the Deutsche Parkinson Vereinigung (DPV) and the Deutsche Stiftung Neurologie.

References

1. Ahlrichs, C., Lawo, M.: Parkinson's disease motor symptoms in machine learning: a review. Health Informatics **2**, (2013)
2. Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. In: Ferscha, A., Mattern, F. (eds.) Pervasive 2004. LNCS, vol. 3001, pp. 1–17. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24646-6_1
3. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
4. Casale, P., Pujol, O., Radeva, P.: Human activity recognition from accelerometer data using a wearable device. In: Vitrià, J., Sanches, J.M., Hernández, M. (eds.) IbPRIA 2011. LNCS, vol. 6669, pp. 289–296. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21257-4_36
5. Chen, S., Zhang, C., Dong, M., Le, J., Rao, M.: Using ranking-CNN for age estimation. In: The IEEE Conference on Computer Vision and Pattern Recognition (2017)
6. Christ, M., Kempa-Liehr, A.W., Feindt, M.: Distributed and parallel time series feature extraction for industrial big data applications. arXiv preprint arXiv:1610.07717 (2016)
7. Curtze, C., Nutt, J.G., Carlson-Kuhta, P., Mancini, M., Horak, F.B.: Levodopa is a double edged sword for balance and gait in people with parkinson's disease. Mov. Disord. **30**(10), 1361–1370 (2015)
8. Elliott, G., Timmermann, A., Komunjer, I.: Estimation and testing of forecast rationality under flexible loss. Rev. Econ. Stud. **72**(4), 1107–1125 (2005)
9. Eskofier, B.M., et al.: Recent machine learning advancements in sensor-based mobility analysis: deep learning for parkinson's disease assessment. In: Engineering in Medicine and Biology Society, pp. 655–658 (2016)

10. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. arXiv preprint arXiv:1809.04356 (2018)
11. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Transfer learning for time series classification. arXiv preprint arXiv:1811.01533 (2018)
12. Fisher, J.M., Hammerla, N.Y., Ploetz, T., Andras, P., Rochester, L., Walker, R.W.: Unsupervised home monitoring of parkinson's disease motor symptoms using body-worn accelerometers. Parkinsonism Relat. Disord. **33**, 44–50 (2016)
13. Frank, E., Hall, M.: A simple approach to ordinal classification. In: De Raedt, L., Flach, P. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 145–156. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44795-4_13
14. Ghika, J., Wiegner, A.W., Fang, J.J., Davies, L., Young, R.R., Growdon, J.H.: Portable system for quantifying motor abnormalities in parkinson's disease. IEEE Trans. Biomed. Eng. **40**(3), 276–283 (1993)
15. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks, pp. 249–256 (2010)
16. Goetz, C.G., et al.: Movement disorder society-sponsored revision of the unified parkinson's disease rating scale (mds-updrs): scale presentation and clinimetric testing results. Mov. Disord. **23**(15), 2129–2170 (2008)
17. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning (2016). http://www.deeplearningbook.org, book in preparation for MIT Press
18. Gutierrez, P.A., Perez-Ortiz, M., Sanchez-Monedero, J., Fernandez-Navarro, F., Hervas-Martinez, C.: Ordinal regression methods: survey and experimental study. IEEE Trans. Knowl. Data Eng. **28**(1), 127–146 (2016)
19. Hammerla, N.Y., Halloran, S., Ploetz, T.: Deep, convolutional, and recurrent models for human activity recognition using wearables. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (2016)
20. Hammerla, N.Y., Fisher, J., Andras, P., Rochester, L., Walker, R., Plötz, T.: PD disease state assessment in naturalistic environments using deep learning, pp. 1742–1748 (2015)
21. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **9**, 1263–1284 (2009)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
23. Herbrich, R., Graepel, T., Obermayer, K.: Support vector learning for ordinal regression. In: International Conference on Artificial Neural Networks (1999)
24. Hssayeni, M.D., Burack, M.A., Ghoraani, B.: Automatic assessment of medication states of patients with parkinson's disease using wearable sensors, pp. 6082–6085 (2016)
25. Hssayeni, M.D., Burack, M.A., Jimenez-Shahed, J., Ghoraani, B., et al.: Wearable-based medication state detection in individuals with parkinson's disease. arXiv preprint arXiv:1809.06973 (2018)
26. Keijsers, N.L., Horstink, M.W., Gielen, S.C.: Automatic assessment of levodopa-induced dyskinesias in daily life by neural networks. Mov. Disord. **18**, 70–80 (2003)
27. Keijsers, N.L., Horstink, M.W., Gielen, S.C.: Ambulatory motor assessment in parkinson's disease. Mov. Disord. **21**, 34–44 (2006)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014). http://arxiv.org/abs/1412.6980
29. Lane, R.D., Glazer, W.M., Hansen, T.E., Berman, W.H., Kramer, S.I.: Assessment of tardive dyskinesia using the abnormal involuntary movement scale. J. Nerv. Ment. Dis. **173**, 353–357 (1985)

30. Lonini, L., et al.: Wearable sensors for parkinson's disease: which data are worth collecting for training symptom detection models. NPJ Digit. Med. **1**, 1–8 (2018)
31. Marsden, C.D., Parkes, J.: "On-off" effects in patients with parkinson's disease on chronic levodopa therapy. Lancet **307**(7954), 292–296 (1976)
32. Microsoft: Microsoft band 2 wearable device (2018). https://www.microsoft.com/en-us/band
33. Niu, Z., Zhou, M., Wang, L., Gao, X., Hua, G.: Ordinal regression with multiple output cnn for age estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4920–4928 (2016)
34. Paszke, A., et al.: Automatic differentiation in pytorch (2017)
35. Postuma, R.B., et al.: MDS clinical diagnostic criteria for parkinson's disease. Mov. Disord. **30**(12), 1591–1601 (2015)
36. Pringsheim, T., Jette, N., Frolkis, A., Steeves, T.D.: The prevalence of parkinson's disease: a systematic review and meta-analysis. Mov. Disord. **29**(13), 1583–1590 (2014)
37. Saeb, S., Lonini, L., Jayaraman, A., Mohr, D.C., Kording, K.P.: The need to approximate the use-case in clinical machine learning. Gigascience **6**(5), 1–9 (2017)
38. Sama, A., et al.: Dyskinesia and motor state detection in parkinson's disease patients with a single movement sensor. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1194–1197 (2012)
39. Szegedy, C., et al.: Going deeper with convolutions. In: Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
40. Tsipouras, M.G., Tzallas, A.T., Fotiadis, D.I., Konitsiotis, S.: On automated assessment of levodopa-induced dyskinesia in parkinson's disease. In: Engineering in Medicine and Biology Society, pp. 2679–2682 (2011)
41. Um, T.T., et al.: Parkinson's disease assessment from a wrist-worn wearable sensor in free-living conditions: deep ensemble learning and visualization. arXiv preprint arXiv:1808.02870 (2018)
42. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: a strong baseline. In: International Joint Conference on Neural Networks, pp. 1578–1585 (2017)
43. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems, pp. 3320–3328 (2014)
44. Zeng, M., et al.: Convolutional neural networks for human activity recognition using mobile sensors. In: 2014 6th International Conference on Mobile Computing, Applications and Services (MobiCASE), pp. 197–205 (2014)

4.6 Robust Colon Tissue Cartography with Semi-supervision

Contributing article:

Jakob Dexl, Michaela Benz, Petr Kuritcyn, Thomas Wittenberg, Volker Bruns, Carol Geppert, Arndt Hartmann, Bernd Bischl, and Jann Goschenhofer. 2022. Robust colon tissue cartography with semi-supervision. *Current Directions in Biomedical Engineering*, 8(2):344–347

Author contributions:

Jann Goschenhofer and Jakob Dexl were responsible for the conceptualization of this paper, supported by Petr Kuritcyn and Michaela Benz. The analyzed data were collected within a collaboration of Michaela Benz, Petr Kuritcyn, Volker Bruns, Carol Geppert, and Arndt Hartmann. Jakob Dexl implemented the methods with support from Jann Goschenhofer on top of a codebase established by Jakob Dexl and Petr Kuritcyn. Jann Goschenhofer, Jakob Dexl, and Michaela Benz were responsible for the design of the experiments which were executed by Jakob Dexl. Jann Goschenhofer and Jakob Dexl were responsible for the writing and editing of the manuscript with support from Michaela Benz and Thomas Wittenberg. Bernd Bischl contributed via supervision.

Copyright information:

DE GRUYTER Current Directions in Biomedical Engineering 2022;8(2): 344-347

Jakob Dexl, Michaela Benz, Petr Kuritcyn, Thomas Wittenberg*, Volker Bruns, Carol Geppert, Arndt Hartmann, Bernd Bischl, and Jann Goschenhofer

Robust Colon Tissue Cartography with Semi-Supervision

https://doi.org/10.1515/cdbme-2022-1088

Abstract: We explore the task of tissue classification for colon cancer histology in a low label regime comparing a semi-supervised and a supervised learning strategy in a series of experiments. Further, we investigate the model robustness w.r.t. distribution shifts in the unlabeled data and domain shifts across different scanners to prove their practicality in a histology context. By utilizing unlabeled data in addition to $n_l = 1000$ labeled tiles per class, we yield a substantial increase in accuracy from 89.9% to 91.4%.

Keywords: Computational Pathology, Semi-Supervised Learning, Colon Cancer, Model Robustness

1 Introduction

Deep learning based approaches have been successfully applied in computational histology. In this context, supervised training is the most common learning paradigm [1]. While large amounts of data are often available and easy to gather, the lack of experienced experts for data annotation is often a bottleneck for successful model training in this context. Therefore, learning approaches that enable robust model training even with a small amount of annotated data are desirable. Semi-supervised learning has shown promising results in other image-based domains to improve predictive algorithms with few labeled data incorporating large amounts of unlabeled data [2]. In this work, we investigate the applicability of this learning paradigm in histopathological tissue cartography in a realistic setting. To this end, we compare supervised and semi-supervised models trained in the low-data regime with colon tissue images. In addition, we explore the robustness of these models with respect to distribution shifts in the unlabeled data and to scanner domain shifts, i.e. the generalization performance across different scanners.

Jakob Dexl, Michaela Benz, Petr Kuritcyn, Thomas Wittenberg*, Volker Bruns, Fraunhofer Institute for Integrated Circuits IIS, Am Wolfsmantel 33, Erlangen, Germany, e-mail: Thomas.Wittenberg@iis.fraunhofer.de
Carol Geppert, Arndt Hartmann, Institut für Pathologie, Universitätsklinikum Erlangen; Friedrich-Alexander-Universität (FAU), Erlangen-Nürnberg, Germany
Bernd Bischl, Jann Goschenhofer, Department of Statistics, Ludwig-Maximilians-Universität, Munich, Germany

2 Related Work

The goal of semi-supervised learning (SSL) is to increase the performance of prediction algorithms in settings where only a few samples can be annotated by utilizing vast amounts of unlabeled data. Data annotation is especially tedious and expensive in medical imaging, rendering semi-supervised learning an interesting modeling approach in this setting. Methods such as entropy minimization, pseudo-labeling or consistency regularization are commonly referred to in the literature. Modern SSL approaches, such as MixMatch [3] or FixMatch [4], make use of a combination of these methods. In the field of computational pathology, SSL approaches have recently gained attention. For example, Jaiswal et. al [5] use an iterative process to pseudo-label data and retrain models using additional unlabeled data reporting significant performance gains in the classification of lymph node sections. Other approaches make use of Mean-Teacher [6] or Student-Teacher [7] approaches to detect colorectal cancer. Some works explicitly investigated the use of FixMatch in histopathology. For instance, Pulido et al. [8] compare the performance of MixMatch and FixMatch trained on histology data with three classes (Squamous, Barrett's, Dysplasia). They further investigate the influence of strong noise and class imbalance and found MixMatch to be superior. Unfortunately, they do not provide the baseline performance of a purely supervised model, making model comparison hard. Schmidt et al. [9] combine multiple instance learning (MIL) with FixMatch [4] in order to mitigate the lack of labeled data and propose an efficient labeling strategy. Both works showed that strong performing models could be trained with only a few labels.

3 Materials and Methods

We use multiple subsets of colon cancer histopathology data to train a realistic supervised baseline model and compare its performance with a FixMatch-based semi-supervised approach, which leverages a larger unlabeled dataset.

3.1 Data

The used database consists of 152 annotated hematoxylin and eosin (H&E) stained colon tissue sections acquired with a 3DHISTECH MIDI scanner and approved by an experienced pathologist. The dataset contains seven tissue classes, namely tumor, necrosis, inflammation, connective tissue combined with adipose tissue, muscle tissue, mucosa, and mucus. The data was tiled into patches of 224 x 224 pixels, and background images were detected and removed. We uniformly sampled a fixed set of 100k patches, or the maximum number of patches per class, from 92 whole slide images (WSIs) to construct the training dataset D^{train}. From this set, we then randomly sampled labeled subsets D_l^{train} of 100, 500, 1k, and 10k patches per class, with each larger set containing the patches from the smaller sets. This procedure was repeated three times to account for sampling-dependent variances of the small sets during different training runs. The set D^{train} is treated as unlabeled dataset D_u^{train} and used within semi-supervised model training, ignoring the labels in this dataset. For experimentation, we explore two data settings. The first uses the four classes tumor, connective tissue, muscle, and mucosa only, where at least 100k samples are available, resulting in a balanced D^{train} of 400k training patches, which we refer to as the balanced "controlled setting". This is a rather unrealistic setting as it assumes an equally balanced class distribution. Naturally, the classes occur imbalanced in this domain. For the labeled sets, we argue that it is realistic to annotate up to 10k tiles per class with considerable effort, and hence assume it is possible to balance small sets of D_l^{train}. In the second setting, we add the three underrepresented classes inflammation, necrosis and mucus. Since these classes occur less frequently in our data, the resulting unlabeled set D_u^{train} with a total of 510k patches is imbalanced. Hence, we face a class distribution shift between D_l^{train} and D_u^{train} in this scenario, which we refer to as the "realistic setting". For the validation set D^{val}, we sampled 10k patches per class from 30 WSIs. Our test set D^{test} consists of approximately 1.4 million patches from 30 additional WSIs. This set is highly imbalanced since entire WSIs were used. The class distributions of D_u^{train} and D^{test} are shown in Table 1.

Tab. 1: Support in D^{train} and D^{test} and averaged F_1 scores per class for the supervised (Sup) and semi-supervised (SSL) approach with 1000 labeled samples per class.

	Sup	SSL	$\Delta_{\text{SSL - Sup}}$	$\lvert D_u^{train}\rvert$	$\lvert D^{test}\rvert$
Tumor	0.9072	0.9302	0.0230	100,000	287,750
Conn.	0.8954	0.8998	0.0044	100,000	440,142
Muscle	0.8841	0.8959	0.0118	100,000	381,209
Mucosa	0.8863	0.9181	0.0319	100,000	217,236
Infl.	0.5042	0.5428	0.0386	38,115	9,427
Mucus	0.7777	0.7366	-0.0412	38,218	16,441
Necrosis	0.7698	0.7097	-0.0602	30,376	29,111
Macro avg.	0.8035	0.8047	0.0012		

3.2 Method

We use an EfficientNetB0 [10] as backbone architecture due to its good trade-off between model performance and required inference runtime as demonstrated by [11]. We initialized it with weights trained on ImageNet and employ simple flips of the patches as an augmentation strategy. Similar to Pulido et al. [8] and Schmidt et al. [9] we use FixMatch as semi-supervised learning approach. In contrast to Pulido et al. [8], which show MixMatch to outperform FixMatch in their application, the initial tests with our database were in favor of FixMatch. FixMatch is an SSL approach based on consistency regularization via strong data augmentation and pseudo-labeling. Therefore, a batch of images from an unlabeled data pool D_u^{train} is transformed into a weakly augmented batch and into a strongly augmented batch. In our specific approach, we use flips as soft augmentation and a cascade of flips, hue, saturation, Gaussian blur, a H&E stain augmentation [14], occlusion and Gaussian noise as strong augmentation. During training, a supervised batch and the two augmented unlabeled batches are fed into the model. If the softmax scores of the weakly augmented data lie above a certain threshold τ, the samples are considered to be a pseudo-label. The error between the prediction of a strongly augmented counterpart and the pseudo-label is then used as a regularization term. The influence of this unsupervised loss is weighted by a factor λ. Following [4], we use weight exponential moving averaging (EMA) with $\alpha = 0.999$ and $\alpha = 0.995$ for the supervised and SSL model, respectively. Models were trained via stochastic gradient descent (SGD) with momentum and weight decay. Further, we use the one cycle learning rate scheme [12] with a cosine functional term and a maximal learning rate after 10% of the cycle. For comparability, we fix the optimization steps to 8000 for both modeling paradigms and increase the batch size for larger D_l^{train} while ensuring that the unsupervised batch size is five times larger than the supervised one in FixMatch. Hyperparameters were optimized via a grid search over learning rates $lr = \{0.01, 0.001, 0.0001\}$, $\lambda = \{1, 2, 5\}$ and $\tau = \{0.85, 0.9, 0.95\}$ for models trained on data with 1k samples per class. We use minority oversampling in the labeled batches while unlabeled data is shuffled randomly. The models were implemented using Tensorflow 2.6 and trained on either an Nvidia P100 or an Nvidia RTX 3080 GPU with the Tensorflow mixed-precision 16-bit float option.

4 Experiments and Results

We split our experiments into the balanced controlled setting and a seven class problem with a distribution shift in the unsupervised dataset D_u as explained above. Accuracy is used as the main performance criterion, while we also report the F_1 score for the second experiment to account for the class imbalances. Metrics are averaged over three different folds.

4.1 Controlled Setting

With the supervised model trained on all 400k samples, we reach an upper bound of 93.6% accuracy. As shown in Figure 1, the FixMatch approach consistently outperforms the supervised baseline over different amounts of labels by a small margin. The biggest gain is yielded with 1k samples per class, where the hyperparameters have been optimized on. At 10k samples per class, both methods closely reach the upper bound. Furthermore, variance over folds decreases in the lower data regime, which is opposite to the supervised baseline.

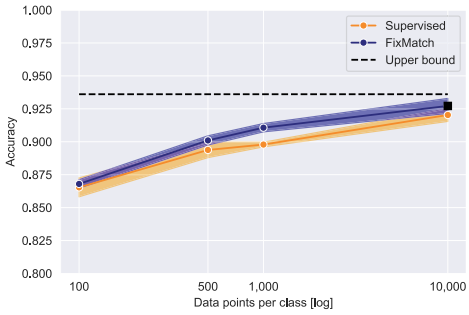


Fig. 1: Model comparison in the controlled setting. The upper bound reflects a supervised model trained on the whole D^{train}. Visible is a small improvement provided by the semi-supervised model. The black square indicates that the FixMatch models needed 16k optimization steps to converge.

4.2 Realistic Setting: Distribution Shift

The introduction of the three minor classes leads to a distribution shift between D_l^{train} and D_u^{train} as these minor classes are underrepresented in D_u^{train}. As shown in Figure 2, the behavior is similar to the balanced case with a small increase in accuracy of FixMatch over the supervised baseline. Inspecting the class-specific scores for $n_l = 1000$ in Table 1, we find this is mainly driven by a performance boost in the majority classes in the semi-supervised case. We hypothesize that this is due to the fact that the semi-supervised model is able to use a large number of unlabelled samples referring to the four majority classes to learn sharper decision boundaries around these classes at the cost of performance loss in the minority classes that are substantially less present in D_u^{train}. The intra-fold variance increases towards $n_l = \{100, 10000\}$ indicating a stronger dependence on the hyperparameters, which were tuned on the $n_l = 1000$ dataset.

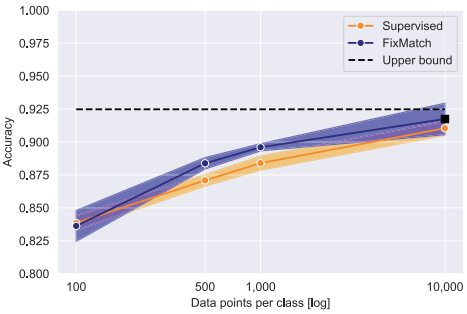


Fig. 2: Comparison of test accuracy for the seven class problem with a distribution shift in D_u^{train}. We observe an improvement provided by the semi-supervised model and the black square indicates that the FixMatch models trained on more data required more optimizer steps to converge.

4.3 Realistic Setting: Domain Shift

One important property of classification algorithms is their ability to generalize to unseen domains. In computational pathology, a key challenge is to overcome domain shifts introduced by unstandardized staining protocols and the use of different scanners. Since consistency-regularization-based SSL algorithms are known to improve model robustness, we investigate their generalization performance on data from a multi-scanner database. Therefore, we evaluate the models trained on $n_l = 1000$ samples per class on data obtained from five additional scanners. In addition, we add an augmentation strategy to the supervised batch of the two approaches in order to increase domain generalization across multiple scanners [13]. It consists of hue, saturation, Gaussian blur and HE [14] and we refer to these results as "aug" in Figure 3. The consistency regularization within FixMatch is partly based on the same augmentation functions and we refer to [13] for a detailed description. From Figure 3, we observe that the standard supervised baseline has a minimal generalization capability. In contrast to that, semi-supervised model training greatly improves the model robustness but also struggles with the different scan-

ner data and has a high variance between different runs and folds. This performance loss over the alternative scanners vanishes for both model paradigms when using the tailored data augmentation strategy. Still, we find the semi-supervised modeling approach to yield a small but substantial performance gain over the supervised model across scanners using this advanced data augmentation scheme.

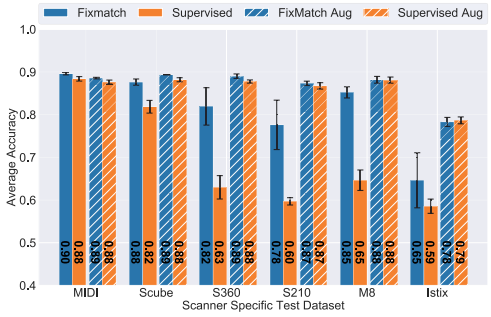


Fig. 3: Performance of the (semi-)supervised models on test data collected from different scanners, introducing a domain shift. Models are trained on data from the MIDI scanner using a simple and a more robust data augmentation strategy ("aug").

5 Discussion and Conclusion

Overall, we are able to yield strong model performance with relatively few annotated samples across both semi- and supervised model training. For instance, the SSL (supervised) model trained on $n_l = 1000$ samples per class yields a test accuracy of 91.37% (89.89%) closing in on the fully supervised baseline trained on a total of 400k labeled patches which yields a test accuracy of 93.76%. Still, one should keep in mind that these relatively few annotated tiles were sampled from 92 different slides, covering a relatively broad part of the overall data distribution. Comparing both learning paradigms, for this application we are able to yield a considerable performance gain using semi-supervision, though this is smaller than suggested by results from other domains [4]. In terms of model robustness w.r.t distribution shifts in the unlabeled data, we find the semi-supervised model to be prone to overfit on the unlabeled data distribution. This stresses the need to develop semi-supervised learning models that are able to cope with such more realistic scenarios. Further, we confirm that semi-supervision can lead to increased model robustness w.r.t. to domain shifts across different scanners in our specific application. Though it is important to mention that similar robustness can be achieved using a well-tailored data augmentation strategy. In conclusion, we were able to successfully employ SSL algorithms in the field of histopathology. Performance improvements provided by the unlabeled data were limited and simple supervised models with tailored augmentations were competitive in terms of performance and domain robustness. Further research is needed to leverage unlabeled data more effectively in the described setting.

Author Statement

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development & Energy through the Center for Analytics - Data - Applications (ADA-Center) within "Bayern Digital II" and by the BMBF (16FMD01K, 16FMD02, 16FMD03).

References

[1] Srinidhi et al: Deep neural network models for computational histopathology: A survey. Medical Image Analysis 2021; 67:101813.
[2] Van Engelen & Hoos: A survey on semi-supervised learning. Machine Learning 2020; 109:373–440.
[3] Berthelot et al.: MixMatch: A holistic approach to semi-supervised learning. Adv. Neural Inf. Proc. Syst. 32, 2019.
[4] Sohn et al.: FixMatch: Simplifying semi-supervised learning with consistency & confidence. Adv in Neural Inf. Proc. Systems. 2020:596–608.
[5] Jaiswal et al.: Semi-supervised learning for cancer detection of lymph node metastases. arXiv, 2019.
[6] Yu et al.: Accurate recognition of colorectal cancer with semi-supervised deep learning on pathological images. Nature Comm. 2021; 12:6311.
[7] Shaw et al.: Teacher-student chain for efficient semi-supervised histology image classification. arXiv, 2020.
[8] Pulido et al.: Semi-supervised classification of noisy, gigapixel histology images. Proc's Int. Symp. Bioinformatics & Bioeng. 2020:563–8.
[9] Schmidt et al.: Efficient cancer classification by coupling semi supervised and multiple instance learning. IEEE Access, 2022; 10:9763–73.
[10] Tan & Le: EfficientNet: Rethinking Model Scaling for convolutional neural networks. Proc's 36th Int. Conf. on Machine Learning, 2019:6105–14.
[11] Kuritcyn et al.: Comparison of CNN models on a multi-scanner database in colon cancer histology. In: Medical Imaging with Deep Learning, 2021.
[12] Smith & Topin: Super-convergence: very fast training of neural networks using large learning rates. Artif. Intell. & Machine Learning for Multi-Domain Operations Applications. 11006. SPIE, 2019:369–86.
[13] Kuritcyn et al.: Robust slide cartography in colon cancer histology. In: Bildverarbeitung f. d. Medizin 2021. 2021:229–34.
[14] Tellez et al.: Whole-slide mitosis detection in HE breast histology using PHH3 as a reference to train distilled stain-invariant convolutional networks. Trans. Med. Imaging 2018; 37:2126–36.

Publication List

Jann Goschenhofer, Pranav Ragupathy, Christian Heumann, Bernd Bischl, and Matthias Assenmacher. 2022. Cc-top: Constrained clustering for dynamic topic discovery. *Proceedings of the First Workshop on Ever Evolving NLP (EvoNLP)*, page 26–34

Jann Goschenhofer, Bernd Bischl, and Zsolt Kira. 2023. Constraintmatch for semi-constrained clustering. *International Joint Conference on Neural Networks (IJCNN)*

Jakob Dexl, Michaela Benz, Petr Kuritcyn, Thomas Wittenberg, Volker Bruns, Carol Geppert, Arndt Hartmann, Bernd Bischl, and Jann Goschenhofer. 2022. Robust colon tissue cartography with semi-supervision. *Current Directions in Biomedical Engineering*, 8(2):344–347

Emilio Dorigatti, Jann Goschenhofer, Benjamin Schubert, Mina Rezaei, and Bernd Bischl. 2022. Positive-unlabeled learning with uncertainty-aware pseudo-label selection. *arXiv preprint arXiv:2201.13192*

Jann Goschenhofer, Rasmus Hvingelby, David Rügamer, Janek Thomas, Moritz Wagner, and Bernd Bischl. 2021. Deep semi-supervised learning for time series classification. In *20th IEEE International Conference on Machine Learning and Applications (ICMLA)*

Jann Goschenhofer, Franz MJ Pfister, Kamer Ali Yuksel, Bernd Bischl, Urban Fietzek, and Janek Thomas. 2019. Wearable-based parkinson's disease severity monitoring using deep learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 400–415. Springer

5 Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12. Juli 2011, § 8 Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, den 10.03.2023 Jann Goschenhofer