Semi Supervised Active Learning with Explicit Mislabel Modeling: An Application to Material Design

Hadjadj Lies

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Mathématiques et Informatique
Unité de recherche : Laboratoire d'Informatique de Grenoble

Apprentissage Actif semi-supervisé avec la Modélisation Explicite de Mislabel : Application à la Conception de Matériaux

Semi Supervised Active Learning with Explicit Mislabel Modeling: An Application to Material Design

Présentée par :

Lies HADJADJ

Direction de thèse :

Massih-Reza AMINI Professeur des Universités, Université Grenoble Alpes	Directeur de thèse
Alexis DESCHAMPS Professeur des Universités, GRENOBLE INP	Co-directeur de thèse
Sana LOUHICHI Professeur des Universités, Université Grenoble Alpes	Co-directrice de thèse

Rapporteurs :

MARIANNE CLAUSEL
Professeur des Universités, UNIVERSITE DE LORRAINE
FABIEN LAUER
Maître de conférences HDR, UNIVERSITE DE LORRAINE

Thèse soutenue publiquement le **6 mars 2023**, devant le jury composé de :

MASSIH-REZA AMINI Professeur des Universités, UNIVERSITE GRENOBLE ALPES	Directeur de thèse
SANA LOUHICHI Professeur des Universités, UNIVERSITE GRENOBLE ALPES	Co-directrice de thèse
MARIANNE CLAUSEL Professeur des Universités, UNIVERSITE DE LORRAINE	Rapporteure
FABIEN LAUER Maître de conférences HDR, UNIVERSITE DE LORRAINE	Rapporteur
NOEL JAKSE Professeur des Universités, GRENOBLE INP	Examinateur
LIONEL GERMAIN Professeur des Universités, UNIVERSITE DE LORRAINE	Examinateur

Invités :

ALEXIS DESCHAMPS
Professeur des Universités, GRENOBLE INP

Abstract

Machine Learning predictive models have been applied to many fields and applications so far. The majority of these learning algorithms rely on labeled training data which may be expensive to obtain as they require labeling by an expert. Additionally, with the new storage capabilities, large amounts of unlabeled data exist in abundance. In this context, the development of new frameworks to learn efficient models from a small set of labeled data, together with a large amount of unlabeled data, is a crucial emphasis of the current research community. Achieving this goal would significantly elevate the state-of-the-art machine intelligence to be comparable to or surpass the human capability of learning to generalize concepts from very few labeled examples. Semi-supervised learning and active learning are two ongoing active research subdomains that aim to achieve this goal.

In this thesis, we investigate two directions in machine learning theory for semi-supervised and active learning. First, We are interested in the generalization properties of a self-training algorithm using halfspaces with explicit mislabel modeling. We propose an iterative algorithm to learn a list of halfspaces from labeled and unlabeled training data, in which each iteration consists of two steps, exploration and pruning. We derive a generalization bound for the proposed algorithm under a Massart noise mislabeling model. Second, we propose a meta-approach for pool-based active learning strategies in the context of multi-class classification tasks, which relies on the proposed concept of learning on Proper Topological Regions (PTR) with an underlying smoothness assumption on the metric space. PTR allows the pool-based active learning strategies to obtain a better initial training set than random selection and increase the training sample size during the rounds while operating in a low-budget regime scenario. Experiments carried out on various benchmarks demonstrate the efficiency of our proposed approaches for semi-supervised and active learning compared to state-of-the-art methods.

A third contribution of the thesis concerns the development of practical deep-learning solutions in the challenging domain of Transmission Electron Microscopy

(TEM) for material design. In the context of orientation microscopy, ML-based approaches still need to catch up to traditional techniques, such as template matching or the Kikuchi technique, when it comes to generalization performance over unseen orientations and phases during training. This is due mainly to the limited experimental data about the studied phenomena for training the models. Nevertheless, it is a realistic and practical constraint, especially for narrow-domain applications where actual data are not widely available. Some successful attempts have been made to use unsupervised learning techniques to gain more insight into the data, but clustering information does not solve the orientation microscopy problem. To this end, we propose a multi-task learning framework based on neural architecture search for fast automation of phase and orientation determination in TEM images.

Résumé

Les modèles prédictifs d'apprentissage automatique ont été appliqués à de nombreux domaines et applications jusqu'à présent. La majorité de ces algorithmes d'apprentissage reposent sur des données d'apprentissage étiquetées qui peuvent être coûteuses à obtenir car elles nécessitent l'étiquetage par un expert. De plus, avec les nouvelles capacités de stockage, une grande quantité de données non étiquetées existe en abondance. Dans ce contexte, le développement de nouveaux cadres pour apprendre des modèles efficaces à partir d'un petit ensemble de données étiquetées, ainsi qu'une grande quantité de données non étiquetées est un accent crucial de la communauté de recherche actuelle. Atteindre cet objectif élèverait considérablement l'état de l'art de l'intelligence artificielle pour être comparable ou surpasser la capacité humaine sur comment apprendre à généraliser des concepts à partir de très peu d'exemples étiquetés. L'apprentissage semi-supervisé et l'apprentissage actif sont deux sous-domaines de recherche actifs en cours qui visent à atteindre cet objectif.

Dans cette thèse, nous étudions deux directions de la théorie de l'apprentissage automatique pour l'apprentissage semi-supervisé et actif. Premièrement, nous nous intéressons aux propriétés de généralisation d'un algorithme d'auto-apprentissage utilisant des demi-espaces avec une modélisation explicite des erreurs d'étiquetage. Nous proposons un algorithme itératif pour apprendre une liste de demi-espaces à partir de données d'apprentissage étiquetées et non étiquetées, dans lequel chaque itération consiste en deux étapes, l'exploration et l'élagage. Nous dérivons une borne de généralisation pour l'algorithme proposé sous un modèle d'étiquetage de bruit de Massart. Deuxièmement, nous proposons une méta-approche pour les stratégies d'apprentissage actif basées sur des pools dans le contexte de tâches de classification multi-classes, qui s'appuie sur le concept proposé d'apprentissage sur les régions topologiques propres (RTP) avec une hypothèse sous-jacente de lissage sur l'espace métrique. Le TRP permet aux stratégies d'apprentissage actif basées sur le pool d'obtenir un meilleur ensemble d'entraînement initial que la sélection aléatoire et d'augmenter la taille de l'échantillon d'entraînement pendant les tours tout en fonc-

tionnant dans un scénario de régime à petit budget. Des expérimentations menées sur différents benchmarks démontrent l'efficacité de nos approches proposées pour l'apprentissage semi-supervisé et actif par rapport aux méthodes de l'état de l'art.

Une troisième contribution de la thèse concerne le développement de solutions pratiques d'apprentissage en profondeur dans le domaine difficile de la microscopie électronique à transmission (TEM) pour la conception de matériaux. Dans le contexte de la microscopie d'orientation, les approches basées sur ML doivent encore rattraper les techniques traditionnelles, telles que l'appariement de modèles ou la technique de Kikuchi, en ce qui concerne les performances de généralisation sur des orientations et des phases inconnu lors de l'apprentissage. Cela est dû principalement au peu de données expérimentales sur les phénomènes étudiés pour l'entraînement des modèles. Néanmoins, il s'agit d'une contrainte réaliste et pratique, en particulier pour les applications à domaine étroit où les données réelles ne sont pas largement disponibles. Certaines tentatives réussies ont été faites pour utiliser des techniques d'apprentissage non supervisées pour mieux comprendre les données, mais le regroupement des informations ne résout pas le problème de la microscopie d'orientation. À cette fin, nous proposons un cadre d'apprentissage multi-tâches basé sur la recherche d'architecture neuronale pour l'automatisation rapide de la détermination de la phase et de l'orientation dans les images TEM.

Contents

List of Figures

Notations

$\mathbb{R}$	The set of real numbers
$\mathbb{Z}$	The set of relative integers
$\mathbb{N}$	Thr set of integers
$\mathbf{w}$	A parameter vector
$\mathbb{P}$	A probability distribution
$\mathbb{E}[X]$	The expected value of a random variable X
Cov	The covariance of random variables
$\perp\!\!\!\perp$	The independence operator
$\mathcal{X}$	The input space and subset of $\mathbb{R}^d$
$\mathbf{x}$	An observation from $\mathcal{X}$
$\mathcal{Y} = \{1, \ldots, c\}$	The output space of cardinal $c \in \mathbb{N}, c \geq 2$
$\mathcal{D}$	A joint probability distribution of the data
$h_{\mathbf{w}}$	A parametric model, with parameters $\mathbf{w}$
$\mathcal{H}_d = \{h_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}\}$	A hypothesis class
$\mathbf{S}$	A sample set
$\mathbf{S}_\ell$	The set of labeled examples of cardinal ℓ
$\mathbf{X}_u$	The set of unlabeled examples of cardinal u
arctan2	The arc tangent function
$\arg\max$	The argument of the maxima
$\arg\min$	The argument of the minima

Chapter 1

Introduction

In this chapter, we will introduce our study by first presenting its context in Section 1.1; then by describing the main application we considered in Material Science that is the Transmission Electron Microscopy in Section 1.2, and the motivation of using Machine Learning for this task in Section 1.3. We will finally present the structure of this thesis in Section 1.4 and give our personal publications and those that have been submitted for this work.

1.1 Context

This thesis is being written as part of the "Multidisciplinary Institute in Artificial Intelligence's" (MIAI)[1] *Magnet* chair. MIAI's mission is to establish a center of excellence in AI for research in Grenoble, where scientists from different research domains may meet and form new alliances. Magnet Chair's purpose is to create new learning frameworks that incorporate contextual knowledge in training models that explore the multidimensional design space of materials. The development of new materials is at the heart of any technological transition, many of which are on the horizon: lighter transportation (more efficient structures), energy production (renewable materials), circular economy (recyclable materials), resource crisis (substitution of critical chemical species), gas capture and release (CO2, toxic gases). The vastness of the materials design space presents a critical opportunity to combine high-throughput materials exploration tactics based on experimental and simulation strategies with powerful artificial intelligence capabilities. In this regard, the objective is to create novel materials with optimum functionality to address the industrial difficulties posed by future societal restrictions. The functions that are being sought include

[1] https://miai.univ-grenoble-alpes.fr

structural, chemical, and physical qualities, as well as those linked to safety, ecology, recycling, low cost, and accessibility. The quest for novel materials with specified qualities remains highly empirical, typically directed by intuition and trial and error. Beyond that, data-to-knowledge methodologies in materials research are particularly promising. The chair has used novel Machine Learning (ML) methodologies to produce genuine materials based on specified attributes (local structure, microstructure, thermodynamic and mechanical properties), anticipate new phases, and the atomic structure of materials.

The goal of my thesis is to use context to increase the effectiveness of learning models in two ways: first, by utilizing the structure of the data by using unlabeled instances in the training of the models, and secondly, by considering related tasks to the primary one in the training phase.

In this way the learning paradigms that we are taking into consideration; are self-training, active-learning, and multi-task learning. The next section gives a quick introduction to these frameworks.

1.1.1 Self-training

Self-training is a Semi-Supervised Learning (SSL) technique where we have a small set of labeled training data together with a large set of unlabeled set with a goal to

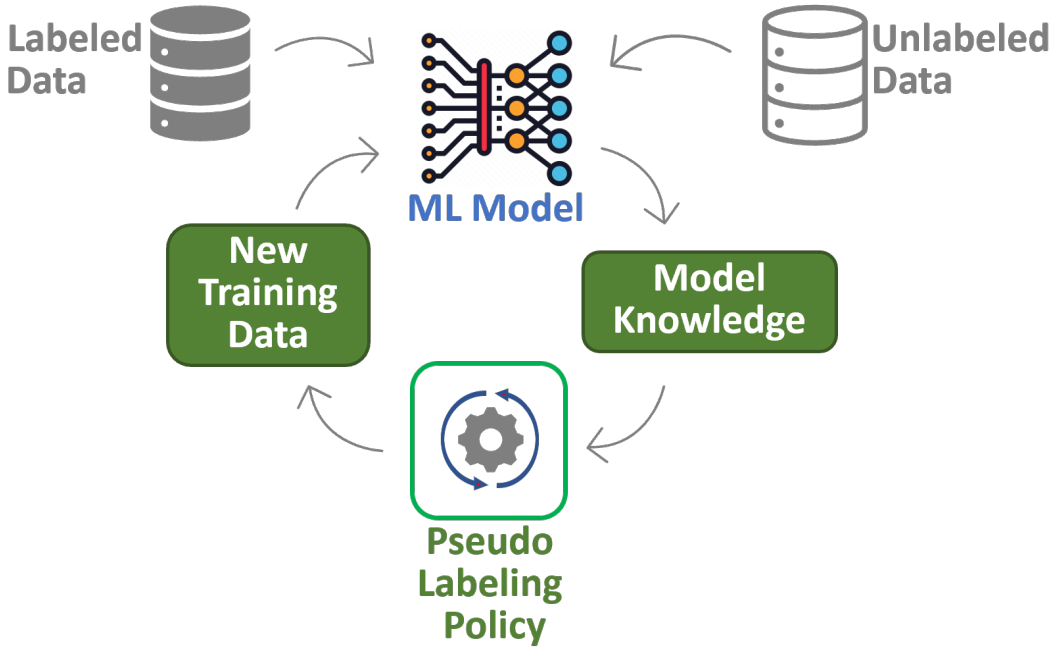


Figure 1.1: Self-learning algorithm.

learn a more efficient prediction model than by only utilizing the labeled training set. First introduced in [102], the self-training algorithm augments iteratively the training set with pseudo-labels obtained by labeling a selection of unlabeled examples with the current model iteration. Then retrain a new model on the augmented training set, and the iterations continue until a convergence criterion is verified. This criterion can be the use of all available unlabeled examples or derived from the current model's performance. In our work, we investigate in Chapter 2, the learnability of a self-training algorithm using halfspaces [72] for binary classification.

1.1.2 Active learning

In this framework, the learning algorithm is presented with unlabeled examples and an oracle, which can interactively be prompted to label unlabeled examples with the true outputs. The fundamental notion behind active learning is the idea of giving the ability to a learning algorithm to choose the data samples it desires to learn from, allowing it to possibly achieve better performance while requiring a fewer number of training examples than by training the same model in a supervised fashion. We identify in the literature two different active learning settings. In the pool-based

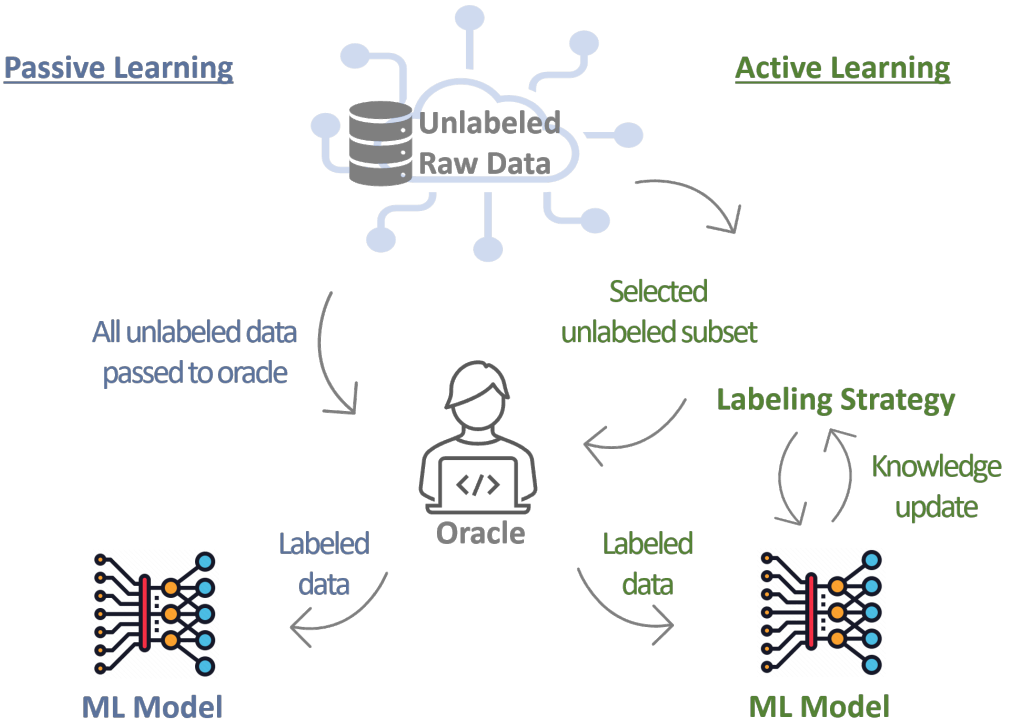


Figure 1.2: Active learning compared to traditional passive learning.

setting, the learning algorithm is presented with all the unlabeled examples at once in a pool. In the stream-based setting, the unlabelled examples are presented in a stream, where each example is sent individually to the learning algorithm [103]. In Chapter 3, we propose a study on pool-based active learning algorithms.

1.1.3 Transfer learning

In transfer learning, the knowledge of an already trained ML model is applied to a different but related problem. For example, suppose we trained a model to solve a classification problem on simulated data. In that case, we could use the knowledge gained during its training to solve a new classification problem on real data. The advantage of this technique is here to uses the knowledge a model has learned from a source task with a lot of available labeled training data in a new task that does not have much data. Instead of starting the learning process from scratch, we start with already some patterns learned from solving a related task [123]. In Chapter 4, we show how to use TL in our applicative contributions to material science.

1.1.4 Multi-task learning

Multi-task learning (MTL) is another subfield of ML in which multiple learning tasks are solved at the same time by the learning algorithm. It has many denominations, such as joint learning, learning to learn, and learning with auxiliary tasks, etc., but they all share the same principle, improving the performance of multiple tasks by learning them jointly rather than solving them separately [127]. In Chapter 4, we design a use case for MTL and show its benefits for our applicative contributions to material science.

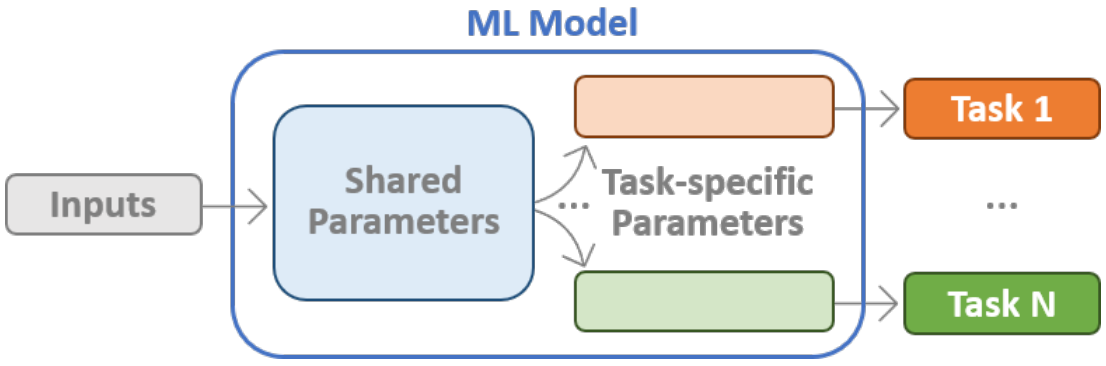


Figure 1.3: Multi-task learning framework in ML.

1.2 Application to Material Design: Transmission Electron Microscopy (TEM)

Transmission Electron Microscopy (TEM) is a particular type of microscopy that uses an electron beam traversing a thin sample (of the order of 100 nm) to characterize its microstructure (nature, orientation and spatial distribution of phases) at high magnification (down to the nanometer resolution or less). TEM has many applications and observation modes in a number of different fields, such as life sciences, nanotechnology, biological and material research, industry, etc. In Figure 1.4, we depict the general process of a TEM experiment under the particular observation mode considered in the present work, namely scanning transmission electron microscopy (STEM) accompanied by Automated Crystal Orientation Mapping (ACOM). In this mode, the sample is scanned (in 2D) by a small electron beam, generating a 2D map of typically 10×10 µm2 ; on each point of the map the diffraction diagram is acquired, which results in a 4D dataset [27]. Further analysis of the diffraction diagrams allows to generate many different information. For example, a virtual brightfield image (VBF) of the scanned area can be drawn by plotting the intensity of the transmitted spot (central spot in the diffraction diagram), see micrograph in Figure 1.4.

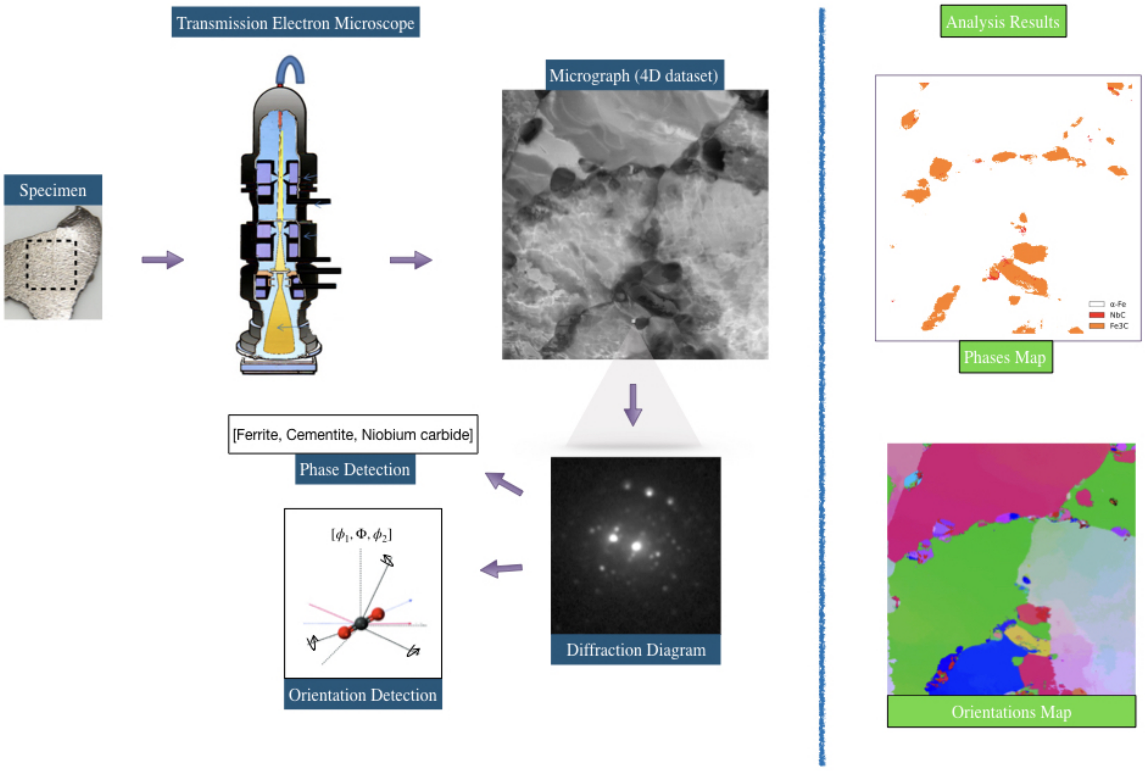


Figure 1.4: Transmission Electron Microscopy (TEM) analysis workflow.

More advanced interpretation requires the analysis of all the diffraction diagrams. Because it is resource intensive, this is performed in an offline fashion after the experiment's end. The algorithms designed to analyze these data are versions of the template matching algorithm [23], where the objective is to find the best match to each diffraction diagram from a predefined set of banks of different crystals (Section 4.1 of Chapter 4 details the literature). The resulting match provides information about the type of crystal (*phase detection*) and its orientation in the specimen (*orientation detection*) at the coordinate of the queried diffraction diagram. The analysis result is presented as two different maps, the phases map and the orientation map, shown on the right-hand side of Figure 1.4.

1.3 Motivation

As previously mentioned, effectively reducing the training cost of learning algorithms in terms of labeled examples for ML is of utmost importance. It will contribute directly to the spread of ML in narrow-domain applications where publicly available data collections are in their premise growth and other applications where the cost of labeling is such that having extensive training collections for Deep-Learning (DL) is simply unrealistic. Therefore, the main aspect of this thesis was not only to implement well-known strategies from the previously mentioned sub-fields of ML, which aim to solve this problem but also to contribute to these subdomains by proposing in Chapter 2, and Chapter 3 novel approaches and algorithms with solid theoretical foundations.

TEM is a good example of a narrow-domain application with relatively scarce publicly available datasets. Most of the matching algorithms designed to analyze TEM data, with millions of image to retrieve, have a high time complexity by definition, which implies that in the standard TEM workflow, the data analysis is performed offline after the experiment's end. ML approaches have a clear advantage over these algorithms because the model's prediction time is instantaneous, which enables online solutions during the TEM experiment. However, there are practical challenges and constraints to take into consideration for the development of ML solutions in this application:

- The limited amount of available data to train ML models.
- The algorithms should generalize to unseen orientations during training.
- The TEM data is dependent on the experiment's setting and microscope.

- TEM data exhibits a high frequency of duplicates, reducing the training data size even further.

Chapter 4 presents a detailed investigation of what DL has to offer in order to solve these challenges and proposes a DL model for the real-time analysis of TEM data.

1.4 Thesis structure

The rest of the thesis consists of the following:

- In **Chapter 2**, we investigate a new self-training algorithm for binary classification with halfspaces under the assumption the training set is corrupted by label noise. We use a Massart noise model to describe label corruption and examine the generalization properties of the classifier found by the self-training algorithm.

- **Chapter 3** uses topological clustering to provide a meta-approach for pool-based active learning algorithms in low-budget regime scenarios. We demonstrate how different active-learning algorithms might profit from this technique in order to operate on limited budgets and handle the cold-start problem in a cohesive manner.

- In **Chapter 4** shows how we can successfully use DL to automate the analysis of TEM data collections to achieve real-time prediction during the TEM experiment.

- Finally, **Chapter 5** concludes our study in general and offers some future prospects.

Personal References

[HALD22] Lies Hadjadj, Massih-Reza Amini, Sana Louhichi, and Alexis Deschamps. Auto-apprentissage de demi-espaces avec des garanties de généralisation sous le modèle de bruit de massart. In *Conférence sur l'Apprentissage Automatique (CAp)*, 2022.

[HDR+22] Lies Hadjadj, Alexis Deschamps, Edgar Rauch, Massih-Reza Amini, Muriel Veron, and Sana Louhichi. Neural architecture search for transmission electron microscopy: Rapid automation of phase and orientation determination in tem images. *Microscopy and Microanalysis*, 28(S1):3166–3169, 2022.

Under Review

[HDAL22] Lies Hadjadj, Alexis Deschamps, Massih-Reza Amini, and Sana Louichi. Deep learning for automated phase and orientation determination in tem maps with ceramic microstructures. *Materials Characterization Journal*, 2022.

[HDMA22] Lies Hadjadj, Emilie Devijver, Rémi Molinier, and Massih-Reza Amini. Pool-based active learning with topological clustering. *Machine Learning Journal*, 2022.

Chapter 2

Self-Training of Halfspaces with Generalization Guarantees under Massart Mislabeling Noise Model

We investigate the generalization properties of a self-training algorithm with halfspaces. The approach learns a list of halfspaces iteratively from labeled and unlabeled training data, in which each iteration consists of two steps: exploration and pruning. In the exploration phase, the halfspace is found sequentially by maximizing the unsigned-margin among unlabeled examples and then assigning pseudo-labels to those that have a distance higher than the current threshold. The pseudo-labeled examples are then added to the training set, and a new classifier is learned. This process is repeated until no more unlabeled examples remain for pseudo-labeling. In the pruning phase, pseudo-labeled samples that have a distance to the last halfspace greater than the associated unsigned-margin are then discarded. We prove that the misclassification error of the resulting sequence of classifiers is bounded and show that the resulting semi-supervised approach never degrades performance compared to the classifier learned using only the initial labeled training set. Experiments carried out on a variety of benchmarks demonstrate the efficiency of the proposed approach compared to state-of-the-art methods. This chapter is based on the following papers [HALD22**?**].

2.1 Introduction

In recent years, several attempts have been made to establish a theoretical foundation for semi-supervised learning. These studies are mainly interested in the generalization ability of semi-supervised learning techniques [99, 87] and the utility of unlabeled

data in the training process [28, 107, 81]. The majority of these works are based on the concept called *compatibility* in [12], and try to exploit the connection between the marginal data distribution and the target function to be learned. The common conclusion of these studies is that unlabeled data will only be useful for training if such a relationship exists.

The three key types of relations considered in the literature are cluster assumption, manifold assumption, and low-density separation [130, 30]. The cluster assumption states that data contains homogeneous labeled clusters, and unlabeled training examples allow to recognize these clusters. In this case, the marginal distribution is viewed as a mixture of class conditional distributions, and semi-supervised learning has been shown to be superior to supervised learning in terms of achieving smaller finite-sample error bounds in some general cases, and in some others, it provides a faster rate of error convergence [28, 99, 87, 107]. In this line, [15] showed that the access to the marginal distribution over unlabeled training data would not provide sample size guarantees better than those obtained by supervised learning unless one assumes very strong assumptions about the conditional distribution over the class labels. Manifold assumption stipulates that the target function is in a low-dimensional manifold. [91] establishes a context through which such algorithms can be analyzed and potentially justified; the main result of this study is that unlabeled data may help the learning task in certain cases by defining the manifold. Finally, low-density separation states that the decision boundary lies in low-density regions. A principal way, in this case, is to employ a margin maximization strategy which results in pushing away the decision boundary from the unlabeled data [30, ch. 6]. Semi-supervised approaches based on this paradigm mainly assign pseudo-labels to high-confident unlabeled training examples with respect to the predictions and include these pseudo-labeled samples in the learning process [114]. However, [31] investigated empirically the problem of label noise bias introduced during the pseudo-labeling process in this case and showed that the use of unlabeled examples could have a minimal gain or even degraded performance, depending on the generalization ability of the initial classifier trained over the labeled training data.

In this chapter, we study the generalization ability of a self-training algorithm with halfspaces that operates in two steps. In the first step, halfspaces are found iteratively over the set of labeled and unlabeled training data by maximizing the unsigned margin of unlabeled examples and then assigning pseudo-labels to those with a distance greater than a found threshold. The pseudo-labeled unlabeled examples are then added to the training set, and a new classifier is learned. This process is

repeated until there are no more unlabeled examples to pseudo-label. In the second step, pseudo-labeled examples with an unsigned margin greater than the last found threshold are removed from the training set.

Our contribution is twofold: (a) we present a first generalization bound for self-training with halfspaces in the case where class labels of examples are supposed to be corrupted by a Massart noise model; (b) we show that the use of unlabeled data in the proposed self-training algorithm does not degrade the performance of the first halfspace trained over the labeled training data.

In the remainder of the chapter, Section 2.2 presents the definitions and the learning objective. In Section 2.3, we present in detail the adaptation of the self-training algorithm for halfspaces. Section 2.4 presents a bound over the misclassification error of the classifier outputted by the proposed algorithm and demonstrates that this misclassification error is upper-bounded by the misclassification error of the fully supervised halfspace. In Section 2.5, we present experimental results, and we conclude this work in Section 2.6.

2.2 Framework and Notations

We consider binary classification problems where the input space $\mathcal{X}$ is a subset of $\mathbb{R}^d$, and the output space is $\mathcal{Y} = \{-1, +1\}$. We study learning algorithms that operate in hypothesis space $\mathcal{H}_d = \{h_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}\}$ of centered halfspaces, where each $h_{\mathbf{w}} \in \mathcal{H}_d$ is a Boolean function of the form $h_{\mathbf{w}}(\mathbf{x}) = \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)$, with $\mathbf{w} \in \mathbb{R}^d$ such that $\|\mathbf{w}\|_2 \leq 1$.

Our analysis succeeds the recent theoretical advances in robust supervised learning of polynomial algorithms for training halfspaces under large margin assumption [41, 89, 42], where the label distribution has been corrupted with the Massart noise model [86]. These studies derive a PAC bound for generalization error for supervised classifiers that depends on the corruption rate of the labeled training set and shed light on a new perspective for analyzing the self-training algorithm. Similarly, in our analysis, we suppose that self-training can be seen as learning with an imperfect expert. Whereat at each iteration, labels of the pseudo-labeled set have been corrupted with a Massart noise [86] oracle defined as:

Definition 2.1 ([86] noise oracle)**.** Let $\mathcal{C} = \{f : \mathcal{X} \to \mathcal{Y}\}$ be a class of Boolean functions over $\mathcal{X} \subseteq \mathbb{R}^d$, with f an unknown target function in $\mathcal{C}$, and $0 \leq \boldsymbol{\eta} < 1/2$. Let η be an unknown parameter function such that $\mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\mathbf{x}}}[\eta(\mathbf{x})] \leq \boldsymbol{\eta}$, with $\mathcal{D}_{\mathbf{x}}$ any marginal distribution over $\mathcal{X}$. The corruption oracle $\mathcal{O}(f, \mathcal{D}_{\mathbf{x}}, \eta)$ works as follow: each

time $\mathcal{O}(f, \mathcal{D}_{\mathbf{x}}, \eta)$ is invoked, it returns a pair $(\mathbf{x}, y)$ where $\mathbf{x}$ is generated i.i.d. from $\mathcal{D}_{\mathbf{x}}$; $y = -f(\mathbf{x})$ with probability $\eta(\mathbf{x})$ and $y = f(\mathbf{x})$ with probability $1 - \eta(\mathbf{x})$.

Let $\mathcal{D}$ denote the joint distribution over $\mathcal{X} \times \mathcal{Y}$ generated by the above oracle with an unknown parameter function η such that $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta(\mathbf{x})] \leq \boldsymbol{\eta}$. We suppose that the training set is composed of ℓ labeled samples $\mathbf{S}_\ell = (\mathbf{x}_i, y_i)_{1\leq i\leq \ell} \in (\mathcal{X} \times \mathcal{Y})^\ell$ and u unlabeled samples $\mathbf{X}_u = (\mathbf{x}_i)_{\ell+1\leq i\leq \ell+u} \in \mathcal{X}^u$, where $\ell \ll u$. Furthermore, we suppose that each pair $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ is i.i.d. with respect to the probability distribution $\mathcal{D}$, we denote by $\mathcal{D}_{\mathbf{x}}$ the marginal of $\mathcal{D}$ on $\mathbf{x}$, and $\mathcal{D}_y(\mathbf{x})$ the distribution of y conditional on $\mathbf{x}$. Finally, for any integer d, let $[d] = \{0, ..., d\}$.

2.2.1 Learning objective

Given $\mathbf{S}_\ell$ and $\mathbf{X}_u$, our goal is to find a learning algorithm that outputs a hypothesis $h_{\mathbf{w}} \in \mathcal{H}_d$ such that with high probability, the misclassification error $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h_{\mathbf{w}}(\mathbf{x}) \neq y]$ is minimized and to show with high probability that the performance of such algorithm is better or equal to any hypothesis in $\mathcal{H}_d$ obtained from $\mathbf{S}_\ell$ only. Here we denote by $\eta_{\mathbf{w}}(\mathbf{x}) = \mathbb{P}_{y\sim\mathcal{D}_y(\mathbf{x})}[h_{\mathbf{w}}(\mathbf{x}) \neq y]$ the conditional misclassification error of a hypothesis $h_{\mathbf{w}} \in \mathcal{H}_d$ with respect to $\mathcal{D}$ and $\mathbf{w}^*$ the normal vector of $h_{\mathbf{w}^*} \in \mathcal{H}_d$ that achieves the optimal misclassification error; $\boldsymbol{\eta}^* = \min_{\mathbf{w},\|\mathbf{w}\|_2\leq 1} \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h_{\mathbf{w}}(\mathbf{x}) \neq y]$.

By considering the indicator function $\mathbb{1}_\pi$ defined as $\mathbb{1}_\pi = 1$ if the predicate π is true and 0 otherwise, we prove in the following lemma that the probability of misclassification of halfspaces over examples with an unsigned-margin greater than a threshold $\gamma > 0$ is bounded by the same quantity $1 > \boldsymbol{\eta} > 0$ that upper-bounds the misclassification error of these examples.

Lemma 2.1. *For all $h_{\mathbf{w}} \in \mathcal{H}_d$, if there exist $\boldsymbol{\eta} \in]0, 1[$ and $\gamma > 0$ such that $\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[|\langle \mathbf{w}, \mathbf{x}\rangle| \geq \gamma] > 0$ and that $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[(\eta_{\mathbf{w}}(\mathbf{x}) - \boldsymbol{\eta})\mathbb{1}_{|\langle \mathbf{w},\mathbf{x}\rangle|\geq\gamma}] \leq 0$, then*

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h_{\mathbf{w}}(\mathbf{x}) \neq y \big| |\langle \mathbf{w}, \mathbf{x}\rangle| \geq \gamma] \leq \boldsymbol{\eta}.$$

Proof. For all hypotheses $h_{\mathbf{w}}$ in $\mathcal{H}_d$, we know that the error achieved by $h_{\mathbf{w}}$ in the region of margin γ from $\mathbf{w}$ satisfies $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[(\eta_{\mathbf{w}}(\mathbf{x}) - \boldsymbol{\eta})\mathbb{1}_{|\langle \mathbf{w},\mathbf{x}\rangle|\geq\gamma}] \leq 0$; by rewriting the expectation, we obtain the following $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\mathbf{w}}(\mathbf{x})\mathbb{1}_{|\langle \mathbf{w},\mathbf{x}\rangle|\geq\gamma}] - \boldsymbol{\eta}\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[|\langle \mathbf{w}, \mathbf{x}\rangle| \geq \gamma] \leq 0$. We have then $\frac{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\mathbf{w}}(\mathbf{x})\mathbb{1}_{|\langle \mathbf{w},\mathbf{x}\rangle|\geq\gamma}]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[|\langle \mathbf{w},\mathbf{x}\rangle|\geq\gamma]} \leq \boldsymbol{\eta}$ and the result follows from the equality:

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h_{\mathbf{w}}(\mathbf{x}) \neq y \big| |\langle \mathbf{w}, \mathbf{x}\rangle| \geq \gamma] = \frac{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\mathbf{w}}(\mathbf{x})\mathbb{1}_{|\langle \mathbf{w},\mathbf{x}\rangle|\geq\gamma}]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[|\langle \mathbf{w}, \mathbf{x}\rangle| \geq \gamma]}.$$

□

Suppose that there exists a pair $(\widetilde{\mathbf{w}}, \widetilde{\gamma})$ minimizing:

$$(\widetilde{\mathbf{w}}, \widetilde{\gamma}) \in \underset{\mathbf{w}\in\mathbb{R}^d, \gamma\geq 0}{\arg\min} \frac{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta_\mathbf{w}(\mathbf{x})\mathbb{1}_{|\langle\mathbf{w},\mathbf{x}\rangle|\geq\gamma}]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[|\langle\mathbf{w},\mathbf{x}\rangle|\geq\gamma]}. \tag{2.1}$$

By defining $\widetilde{\eta}$ as:

$$\widetilde{\eta} = \inf_{\mathbf{w}\in\mathbb{R}^d, \gamma\geq 0} \frac{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta_\mathbf{w}(\mathbf{x})\mathbb{1}_{|\langle\mathbf{w},\mathbf{x}\rangle|\geq\gamma}]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[|\langle\mathbf{w},\mathbf{x}\rangle|\geq\gamma]}.$$

The following inequality holds:

$$\widetilde{\eta} \leq \inf_{\mathbf{w}\in\mathbb{R}^d} \frac{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta_\mathbf{w}(\mathbf{x})\mathbb{1}_{|\langle\mathbf{w},\mathbf{x}\rangle|\geq 0}]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[|\langle\mathbf{w},\mathbf{x}\rangle|\geq 0]} = \boldsymbol{\eta}^*.$$

This inequality paves the way for the following claim, which is central to the self-training strategy described in the next section.

Claim 2.2. *Suppose that there exists a pair $(\widetilde{\mathbf{w}}, \widetilde{\gamma})$ satisfying the minimization problem (2.1) with $\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[|\langle\widetilde{\mathbf{w}},\mathbf{x}\rangle|\geq\widetilde{\gamma}] > 0$, then $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h_{\widetilde{\mathbf{w}}}(\mathbf{x}) \neq y \big| |\langle\widetilde{\mathbf{w}},\mathbf{x}\rangle|\geq\widetilde{\gamma}] \leq \boldsymbol{\eta}^*$.*

Proof. The requirements of Lemma 2.1 are satisfied with $(\mathbf{w}, \gamma) = (\widetilde{\mathbf{w}}, \widetilde{\gamma})$ and $\eta = \widetilde{\eta}$. This claim is then proved using the conclusion of Lemma 2.1 together with the fact that $\widetilde{\eta} \leq \boldsymbol{\eta}^*$. □

The claim above demonstrates that for examples generated by the probability distribution $\mathcal{D}$, there exists a region in $\mathcal{X}$ on either side of a margin $\widetilde{\gamma}$ to the decision boundary defined by $\widetilde{\mathbf{w}}$ solution of (Eq. 2.1); where the probability of misclassification error of the corresponding halfspace in this region is upper-bounded by the optimal misclassification error $\boldsymbol{\eta}^*$. This result is consistent with semi-supervised learning studies that consider the margin as an indicator of confidence and search the decision boundary on low-density regions [69, 55, 9, 51, 112].

2.2.2 Problem resolution

We use a block coordinate minimization method for solving the optimization problem (2.1). This strategy consists in first finding a halfspace with parameters $\widetilde{\mathbf{w}}$ that minimizes Eq. (2.1) with a threshold $\gamma = 0$, and then by fixing $\widetilde{\mathbf{w}}$, finds the threshold $\widetilde{\gamma}$ for which Eq. (2.1) is minimum. We resolve this problem using the following claim, which links the misclassification error $\eta_\mathbf{w}$ and the perceptron loss $\ell_p(y, h_\mathbf{w}(\mathbf{x})) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+; \ell_p(y, h_\mathbf{w}(\mathbf{x})) = -y\langle\mathbf{w},\mathbf{x}\rangle\mathbb{1}_{y\langle\mathbf{w},\mathbf{x}\rangle\leq 0}$.

Claim 2.3. *For a given weight vector $\mathbf{w}$, we have:*

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[|\langle\mathbf{w},\mathbf{x}\rangle|\eta_\mathbf{w}(\mathbf{x})] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell_p(y, h_\mathbf{w}(\mathbf{x}))] \tag{2.2}$$

Proof. For a fixed weight vector $\mathbf{w}$, we have that:

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell_p(y, h_{\mathbf{w}}(\mathbf{x}))] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[-y\langle\mathbf{w},\mathbf{x}\rangle \mathbb{1}_{y\langle\mathbf{w},\mathbf{x}\rangle\leq 0}].$$

As we are considering misclassification errors, i.e., $-y\langle\mathbf{w},\mathbf{x}\rangle\mathbb{1}_{y\langle\mathbf{w},\mathbf{x}\rangle\leq 0} = \mathbb{1}_{y\langle\mathbf{w},\mathbf{x}\rangle\leq 0}|\langle\mathbf{w},\mathbf{x}\rangle|$, it comes that $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell_p(y, h_{\mathbf{w}}(\mathbf{x}))] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[|\langle\mathbf{w},\mathbf{x}\rangle|\mathbb{P}_{y\sim\mathcal{D}_{y(\mathbf{x})}}[-y\langle\mathbf{w},\mathbf{x}\rangle > 0]]$. The result then follows from the definition of the misclassification error, i.e., $\eta_{\mathbf{w}}(\mathbf{x}) = \mathbb{P}_{y\sim\mathcal{D}_{y(\mathbf{x})}}[-y\langle\mathbf{w},\mathbf{x}\rangle > 0]$. □

This claim shows that the minimization of the generalization error with ℓ_p is equivalent to minimizing $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[|\langle\mathbf{w},\mathbf{x}\rangle|\eta_{\mathbf{w}}(\mathbf{x})]$. Hence, the minimization of $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\ell_p(y, h_{\mathbf{w}}(\mathbf{x}))]$ cannot result in bounded misclassification error, as the distribution of margins $|\langle\mathbf{w},\mathbf{x}\rangle|$ might vary widely between samples in $\mathcal{X}$. In the following lemma, we show that it is possible to achieve bounded misclassification error under margin condition and L_2-norm constraint.

Lemma 2.4. *For a fixed distribution $\mathcal{D}$, let $R = \max_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}} \|\mathbf{x}\|_2$ and $\gamma > 0$, let $\widetilde{\mathbf{w}}$ and $\bar{\mathbf{w}}$ be defined as follows:*

$$\widetilde{\mathbf{w}} = \operatorname*{arg\,min}_{\mathbf{w},||\mathbf{w}||_2\leq 1} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[|\langle\mathbf{w},\mathbf{x}\rangle|\eta_{\mathbf{w}}(\mathbf{x})\big||\langle\mathbf{w},\mathbf{x}\rangle| \geq \gamma]$$
$$\overline{\mathbf{w}} = \operatorname*{arg\,min}_{\mathbf{w},||\mathbf{w}||_2\leq 1} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\mathbf{w}}(\mathbf{x})\big||\langle\mathbf{w},\mathbf{x}\rangle| \geq \gamma].$$

We then have:

$$\frac{\gamma}{R}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\widetilde{\mathbf{w}}}(\mathbf{x})\big||\langle\widetilde{\mathbf{w}},\mathbf{x}\rangle| \geq \gamma] \leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\overline{\mathbf{w}}}(\mathbf{x})\big||\langle\overline{\mathbf{w}},\mathbf{x}\rangle| \geq \gamma] \leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\widetilde{\mathbf{w}}}(\mathbf{x})\big||\langle\widetilde{\mathbf{w}},\mathbf{x}\rangle| \geq \gamma].$$

Proof. From the condition $|\langle\widetilde{\mathbf{w}},\mathbf{x}\rangle| \geq \gamma$ in the expectation, we have:

$$\gamma\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\widetilde{\mathbf{w}}}(\mathbf{x})\big||\langle\widetilde{\mathbf{w}},\mathbf{x}\rangle| \geq \gamma] \leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[|\langle\widetilde{\mathbf{w}},\mathbf{x}\rangle|\eta_{\widetilde{\mathbf{w}}}(\mathbf{x})\big||\langle\widetilde{\mathbf{w}},\mathbf{x}\rangle| \geq \gamma]$$

Applying the definition of $\widetilde{\mathbf{w}}$ to the right-hand side of the above inequality gives:

$$\gamma\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\widetilde{\mathbf{w}}}(\mathbf{x})\big||\langle\widetilde{\mathbf{w}},\mathbf{x}\rangle| \geq \gamma] \leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[|\langle\overline{\mathbf{w}},\mathbf{x}\rangle|\eta_{\overline{\mathbf{w}}}(\mathbf{x})\big||\langle\overline{\mathbf{w}},\mathbf{x}\rangle| \geq \gamma]$$

Using the Cauchy–Schwarz inequality and the definition of R, we get:

$$\gamma\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\widetilde{\mathbf{w}}}(\mathbf{x})\big||\langle\widetilde{\mathbf{w}},\mathbf{x}\rangle| \geq \gamma] \leq R\ \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\overline{\mathbf{w}}}(\mathbf{x})\big||\langle\overline{\mathbf{w}},\mathbf{x}\rangle| \geq \gamma]$$

Then from the definition of $\overline{\mathbf{w}}$, we know:

$$R\ \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\overline{\mathbf{w}}}(\mathbf{x})\big||\langle\overline{\mathbf{w}},\mathbf{x}\rangle| \geq \gamma] \leq R\ \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\widetilde{\mathbf{w}}}(\mathbf{x})\big||\langle\widetilde{\mathbf{w}},\mathbf{x}\rangle| \geq \gamma]$$

Dividing the two inequalities above by R gives the result. □

Algorithm 1 Self-Training with Halfspaces

Input : $\mathbf{S}_\ell = (\mathbf{x}_i, y_i)_{1 \le i \le l}$, $\mathbf{X}_u = (\mathbf{x}_i)_{l+1 \le i \le n}$, $p = 5$ number of threshold tests.
Set $k \leftarrow 0$, $\mathbf{S}^{(k)} = \mathbf{S}_\ell$, $\mathbf{U}^{(k)} = \mathbf{X}_u$, $w = \frac{|\mathbf{S}^{(k)}|}{p}$, $L = []$.
while $|\mathbf{S}^{(k)}| \geq \ell$ **do**
    Let $\hat{\mathcal{R}}_{\mathbf{S}^{(k)}}(\mathbf{w}) = \frac{1}{|\mathbf{S}^{(k)}|} \sum_{(\mathbf{x},y) \in \mathbf{S}^{(k)}} [\ell_p(y, h_{\mathbf{w}}(\mathbf{x}))]$
    Run projected SGD on $\hat{\mathcal{R}}_{\mathbf{S}^{(k)}}(\mathbf{w})$ to obtain $\mathbf{w}^{(k)}$ such that $\|\mathbf{w}^{(k)}\|_2 \leq 1$.
    Order $\mathbf{S}^{(k)}$ by decreasing order of margin from $\mathbf{w}^{(k)}$.
    Set a window of indices $I = [w, 2w, ..., pw]$,
    find $t = \arg\min_{i \in I} \frac{1}{|\mathbf{S}^{(k)}_{\geq i}|} \sum_{(\mathbf{x},y) \in \mathbf{S}^{(k)}_{\geq i}} \mathbb{1}_{h_{\mathbf{w}^{(k)}}(\mathbf{x}) \neq y}$.
    Set $\gamma^{(k)}$ to the margin of the sample at position $I[t]$.
    Let $\mathbf{U}^{(k)} = \{\mathbf{x} \in \mathbf{X}_u \big| |\langle \mathbf{w}^{(k)}, \mathbf{x} \rangle| \geq \gamma^{(k)}\}$.
    if $|\mathbf{U}^{(k)}| > 0$ **then**
        $\mathbf{S}^{(k)}_u = \{(\mathbf{x}, y) \big| \mathbf{x} \in \mathbf{U}^{(k)} \wedge y = \text{sign}(\langle \mathbf{w}^{(k)}, \mathbf{x} \rangle)\}$
        $\mathbf{S}^{(k+1)} \leftarrow \mathbf{S}^{(k)} \cup \mathbf{S}^{(k)}_u$
        $\mathbf{X}_u \leftarrow \mathbf{X}_u \setminus \mathbf{U}^{(k)}$
    else
        $L = L \cup [(\mathbf{w}^{(k)}, \gamma^{(k)})]$
        $\mathbf{S}^{(k+1)} = \{(\mathbf{x}, y) \in \mathbf{S}^{(k)} \big| |\langle \mathbf{w}^{(k)}, \mathbf{x} \rangle| < \gamma^{(k)}\}$
    end if
    Set $k \leftarrow k + 1, w = \frac{|\mathbf{S}^{(k)}|}{p}$
end while
Output : $L_m = [(\mathbf{w}^{(1)}, \gamma^{(1)}), ..., (\mathbf{w}^{(m)}, \gamma^{(m)})]$

Lemma 2.4 guarantees that the approximation of the perceptron loss to the misclassification error is more accurate for examples that have a comparable distance to the halfspace. This result paves the way for our implementation of the self-learning algorithm.

2.3 Self-Training with Halfspaces

Given $\mathbf{S}_\ell$ and $\mathbf{X}_u$ drawn i.i.d. from a distribution $\mathcal{D}$ corrupted with $\mathcal{O}(f, \mathcal{D}_\mathbf{x}, \eta^{(0)})$. Algorithm 1 learns iteratively a list of halfspaces $L_m = [(\mathbf{w}^{(1)}, \gamma^{(1)}), ..., (\mathbf{w}^{(m)}, \gamma^{(m)})]$ with each round consisting of *exploration* and *pruning* steps. The goal of the *exploration* phase is to discover the halfspace with the highest margin on the set of unlabeled samples that are not still pseudo-labeled. This is done by first, learning a halfspace that minimizes the empirical surrogate loss of $\mathcal{R}_\mathcal{D}(\mathbf{w}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell_p(y, h_\mathbf{w}(\mathbf{x}))]$ over a

set of labeled and already pseudo-labeled examples $\mathbf{S}^{(k)}$ from $\mathbf{S}_\ell$ and $\mathbf{X}_u$:

$$\min_{\mathbf{w}} \hat{\mathcal{R}}_{\mathbf{S}^{(k)}}(\mathbf{w}) = \frac{1}{|\mathbf{S}^{(k)}|} \sum_{(\mathbf{x},y)\in\mathbf{S}^{(k)}} \ell_p(y, h_{\mathbf{w}}(\mathbf{x})) \qquad (2.3)$$
$$\text{s.t.} \quad ||\mathbf{w}||_2 \leq 1$$

At round $k = 0$, we have $\mathbf{S}^{(0)} = \mathbf{S}_\ell$. Once the halfspace with parameters $\mathbf{w}^{(k)}$ is found, a threshold $\gamma^{(k)}$, defined as the highest unsigned margin in $\mathbf{S}^{(k)}$, is set such that the empirical loss over the set of examples in $\mathbf{S}^{(k)}$ with unsigned-margin above $\gamma^{(k)}$, is the lowest. In the pseudo-code of the algorithm, $\mathbf{S}^{(k)}_{\geq i}$ refers to the subset of examples in $\mathbf{S}^{(k)}$ having an unsigned margin greater or equal to $\omega \times i$. Unlabeled examples $\mathbf{x} \in \mathbf{X}_u$ that are not pseudo-labeled are assigned labels, i.e., $y = \text{sign}(\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle)$ iff $|\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \geq \gamma^{(k)}$. These pseudo-labeled examples are added to $\mathbf{S}^{(k)}$ and removed from $\mathbf{X}_u$, and a new halfspace minimizing Eq. (2.3) is found. Examples in $\mathbf{S}^{(k)}$ are supposed to be misclassified by the oracle $\mathcal{O}(f, \mathcal{D}_{\mathbf{x}}, \eta^{(k)})$ following Definition 2.1 with the parameter function $\eta^{(k)}$ that refers to the conditional probability of corruption in $\mathbf{S}^{(k)}$ defined as $\eta^{(k)}(\mathbf{x}) = \mathbb{P}_{y\sim\mathbf{S}^{(k)}_y(\mathbf{x})}[f(\mathbf{x}) \neq y] \leq \boldsymbol{\eta}^{(k)}$.

Once the halfspace with parameters $\mathbf{w}^{(k)}$ and threshold $\gamma^{(k)}$ are found such that there are no more unlabeled samples having an unsigned-margin larger than $\gamma^{(k)}$, the pair $(\mathbf{w}^{(k)}, \gamma^{(k)})$ is added to the list L_m, and samples from $\mathbf{S}^{(k)}$ having an unsigned-margin above $\gamma^{(k)}$ are removed (pruning phase). Remind that $\gamma^{(k)}$ is the largest threshold above which the misclassification error over $\mathbf{S}^{(k)}$ increases.

In detail. The self-training algorithm takes as input the labeled set $\mathbf{S}_\ell$, the unlabeled set $\mathbf{X}_u$ and p, which refers to the number of tests for threshold estimation, set to 5. After finding the weight vector $\mathbf{w}^{(k)}$ at round k, with projected SGD (step 5), we order the labeled set $\mathbf{S}^{(k)}$ (with $\mathbf{S}^{(0)} = \mathbf{S}_\ell$) by decreasing order of unsigned-margin to $\mathbf{w}^{(k)}$. The threshold $\gamma^{(k)}$ is defined as the largest margin such that the error of examples in $\mathbf{S}^{(k)}$ with an unsigned margin higher than $\gamma^{(k)}$ increases (step 9). At this stage, observations $\mathbf{x} \in \mathbf{X}_u$ with an unsigned margin greater than $\gamma^{(k)}$ (step $12 - 13$) are pseudo-labeled and added to the labeled set $\mathbf{S}^{(k)}$ and they are removed from the unlabeled set. This exploration phase of finding a halfspace with the largest threshold $\gamma^{(k)}$ is repeated until there are no more unlabeled samples with an unsigned margin larger than this threshold. After this phase, the pruning phase begins by removing examples in $\mathbf{S}^{(k)}$ with an unsigned margin strictly less than $\gamma^{(k)}$ (step 17). The parameters of the halfspace and the corresponding threshold are added to the list of selected classifiers L_m, and the procedure is repeated until the size of the labeled set becomes less than ℓ.

To classify an unknown example $\mathbf{x}$, the prediction of the first halfspace with normal vector $\mathbf{w}^{(i)}$ in the list L_m, such that the unsigned-margin $|\langle \mathbf{w}^{(i)}, \mathbf{x} \rangle|$ of $\mathbf{x}$ is higher or equal to the corresponding threshold $\gamma^{(i)}$, is returned. By abuse of notation, we note that the prediction for $\mathbf{x}$ is $L_m(\mathbf{x}) = h_{\mathbf{w}^{(i)}}(\mathbf{x})$. From Claim 2.2, we know that the misclassification error of this halfspace on the region where $\mathbf{x}$ lies is bounded by the optimal misclassification error η^*. If no such halfspace exists, the observation is classified using the prediction of the first classifier $h_{\mathbf{w}^{(1)}}$ that was trained over all the labeled and the pseudo-labeled samples without pruning; i.e., $L_m(\mathbf{x}) = h_{\mathbf{w}^{(1)}}(\mathbf{x})$.

$$L_m(\mathbf{x}) = \begin{cases} h_{\mathbf{w}^{(i)}}(\mathbf{x}) & \text{if } \exists i : i = \arg\min_{1 \le k \le m : |\langle \mathbf{w}^{(k)}, \mathbf{x} \rangle| \ge \gamma^{(k)}} \left(|\langle \mathbf{w}^{(k)}, \mathbf{x} \rangle| - \gamma^{(k)} \right), \\ h_{\mathbf{w}^{(1)}}(\mathbf{x}) & \text{otherwise.} \end{cases}$$

2.4 Corruption noise modeling and Generalization guarantees

In the following, we relate the process of pseudo-labeling to the corruption noise model $\mathcal{O}(f, \mathcal{D}_{\mathbf{x}}, \eta^{(k)})$ for all pseudo-labeling iterations k in Algorithm 1, then we present a bound over the misclassification error of the classifier L_m outputted by the algorithm and demonstrate that this misclassification error is upper-bounded by the misclassification error of the fully supervised halfspace.

Claim 2.5. *Let $\mathbf{S}^{(0)} = \mathbf{S}_\ell$ be a labeled set drawn i.i.d. from $\mathcal{D} = \mathcal{O}(f, \mathcal{D}_{\mathbf{x}}, \eta^{(0)})$ and $\mathbf{U}^{(0)} = \mathbf{X}_u$ an initial unlabeled set drawn i.i.d. from $\mathcal{D}_{\mathbf{x}}$. For all iterations $k \in [K]$ of Algorithm 1; the active labeled set $\mathbf{S}^{(k)}$ is drawn i.i.d. from $\mathcal{D} = \mathcal{O}(f, \mathcal{D}_{\mathbf{x}}, \eta^{(k)})$ where the corruption noise distribution $\eta^{(k)}$ is bounded by:*

$$\forall k \in [K], \quad \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\mathbf{x}}}[\eta^{(k)}(\mathbf{x}) \big| \mathbf{x} \in \mathbf{S}^{(k)}] \le \max_{j \in [K]} \boldsymbol{\eta}^{(j)}$$

Proof. We know that $\forall k \in [K], \mathbf{S}^{(k)} \subseteq \mathbf{S}^{(0)} \cup \bigcup_{i=0}^{k-1} \mathbf{S}_u^{(i)}$, where $\mathbf{S}_u^{(i)}$ is the set of pseudo-labeled pairs of examples $\mathbf{x}$ from $\mathbf{U}^{(i)}$, $\mathbf{S}_u^{(i)} = \emptyset$ for the iterations $i \in [K]$ when no examples are pseudo-labeled. Then the noise distribution $\eta^{(k)}$ satisfies for all $k \in [K]$:

$$\mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\mathbf{x}}}[\eta^{(k)}(\mathbf{x}) \mathbb{1}_{\mathbf{x} \in \mathbf{S}^{(k)}}] = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\mathbf{x}}}[\eta^{(k)}(\mathbf{x}) \mathbb{1}_{\mathbf{x} \in \mathbf{S}^{(k)} \cap \mathbf{S}^{(0)}}] + \sum_{i=0}^{k-1} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\mathbf{x}}}[\eta^{(k)}(\mathbf{x}) \mathbb{1}_{\mathbf{x} \in \mathbf{S}^{(k)} \cap \mathbf{S}_u^{(i)}}]$$

If we condition on $\mathbf{x} \in \mathbf{S}^{(k)}$, we obtain for all $k \in [K]$:

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta^{(k)}(\mathbf{x})\big|\mathbf{x} \in \mathbf{S}^{(k)}] = \mathbb{P}[\mathbf{x} \in \mathbf{S}^{(0)}\big|\mathbf{x} \in \mathbf{S}^{(k)}]\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta^{(0)}(\mathbf{x})\big|\mathbf{x} \in \mathbf{S}^{(k)} \cap \mathbf{S}^{(0)}] + \sum_{i=0}^{k-1} \mathbb{P}[\mathbf{x} \in \mathbf{S}_u^{(i)}\big|\mathbf{x} \in \mathbf{S}^{(k)}]\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta_{\mathbf{w}^{(i)}}(\mathbf{x})\big|\mathbf{x} \in \mathbf{S}^{(k)} \cap \mathbf{S}_u^{(i)}],$$

this equation includes the initial corruption of the labeled set $\mathbf{S}^{(0)} = \mathbf{S}_\ell$ in addition to the noise injected by each classifier $h_{\mathbf{w}^{(k)}}$ at each round k when pseudo-labeling occurs. Now that we have modeled the process of pseudo-labeling, the result is straightforward considering the fact that $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta^{(0)}(\mathbf{x})] \leq \boldsymbol{\eta}^{(0)}; \forall k \in [K], \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta_{\mathbf{w}^{(k)}}(\mathbf{x})] \leq \boldsymbol{\eta}^{(k)}$; and,

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\mathbf{x} \in \mathbf{S}^{(0)}\big|\mathbf{x} \in \mathbf{S}^{(k)}] + \sum_{i=0}^{k-1} \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\mathbf{x} \in \mathbf{S}_u^{(i)}\big|\mathbf{x} \in \mathbf{S}^{(k)}] = \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\mathbf{x} \in \mathbf{S}^{(0)} \cup \bigcup_{i=0}^{k-1} \mathbf{S}_u^{(k)}\big|\mathbf{x} \in \mathbf{S}^{(k)}] \leq 1. \qquad \square$$

We can now state our main contribution that bounds the generalization error of the classifier L_m outputted by Algorithm 1 with respect to the optimal misclassification error η^* in the case where projected SGD is used for the minimization of Eq. (2.3). Note that in this case the time complexity of the algorithm is polynomial with respect to the dimension d, the upper bound on the bit complexity of examples, the total number of iterations, and the upper bound on SGD steps.

Theorem 2.6. *Let $\mathbf{S}_\ell$ be a set of i.i.d. samples of size ℓ drawn from a distribution $\mathcal{D} = \mathcal{O}(f, \mathcal{D}_\mathbf{x}, \eta^{(0)})$ on $\mathbb{R}^d \times \{-1, +1\}$, where f is an unknown concept function and $\eta^{(0)}$ an unknown parameter function bounded by $1/2$, let $\mathbf{X}_u$ be an unlabeled set of size u drawn i.i.d. from $\mathcal{D}_\mathbf{x}$. Algorithm 1 terminates after K iterations, and outputs a non-proper classifier L_m of m halfspaces such that with high probability:*

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[L_m(\mathbf{x}) \neq y] \leq \boldsymbol{\eta}^* + \max_{k\in I} \epsilon^{(k)} + \pi_{K+1},$$

where I is the set of rounds $k \in [K]$ at which the halfspaces were added to L_m, $\epsilon^{(k)}$ is the projected SGD convergence error rate at round k, and π_{K+1} a negligible not-accounted mass of $\mathcal{D}_\mathbf{x}$.

The proof of Theorem 2.6 is based on the following property of projected SGD.

Lemma 2.7 (From [45])**.** *Let $\hat{\mathcal{R}}$ be a convex function of any type. Consider the projected SGD iteration, which starts with $\mathbf{w}^{(0)}$ and computes for each step. $\mathbf{w}^{(t+\frac{1}{2})} =$*

$\mathbf{w}^{(t)} - \alpha^{(t)} g^{(t)}; \mathbf{w}^{(t+1)} = \arg\min_{\mathbf{w}:||\mathbf{w}||_2 \leq 1} ||\mathbf{w} - \mathbf{w}^{(t+\frac{1}{2})}||_2$. *Where* $g^{(t)}$ *is a stochastic sub-gradient such that* $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[g(\mathbf{w},\mathbf{x})] \in \partial\hat{\mathcal{R}}(\mathbf{w}) = \{g : \hat{\mathcal{R}}(\mathbf{w}') \geq \hat{\mathcal{R}}(\mathbf{w}) + \langle \mathbb{E}[g], \mathbf{w}' - \mathbf{w}\rangle$ *for all* $\mathbf{w}'\}$ *and* $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[||g(\mathbf{w},\mathbf{x})||_2^2] \leq M^2$. *For any* $\epsilon, \delta > 0$*; if the projected SGD is executed* $T = \Omega(\log(1/\delta)/\epsilon^2)$ *times with a step size* $\alpha^{(t)} = \frac{1}{M\sqrt{t}}$*, then for* $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}^{(t)}$*, we have with probability at least* $1-\delta$ *that:*

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\hat{\mathcal{R}}(\bar{\mathbf{w}})] \leq \min_{\mathbf{w},\|\mathbf{w}\|_2\leq 1} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\hat{\mathcal{R}}(\mathbf{w})] + \epsilon.$$

Proof of Theorem 2.6. We consider the steps of Algorithm 1. At iteration k of the while loop, we consider the active training set $\mathbf{S}^{(k)}$ consisting of examples not handled in previous iterations.

We first note that the algorithm terminates after at most K iterations. From the fact that at every iteration k, we discard a non-empty set from $\mathbf{S}^{(k)}$ when we do not pseudo-label or from $\mathbf{U}^{(k)}$ when we pseudo-label, and that the empirical distributions $\mathbf{S}_\ell$ and $\mathbf{X}_u$ are finite sets. By the guarantees of Lemma 2.7, running SGD (step 4) on $\hat{\mathcal{R}}_{\mathbf{S}^{(k)}}$ for $T = \Omega(\log(1/\delta)/\epsilon^2)$ steps, we obtain a weight vector $\mathbf{w}^{(k)}$ such that with probability at least $1-\delta$:

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\hat{\mathcal{R}}_{\mathbf{S}^{(k)}}(\mathbf{w}^{(k)})] \leq \min_{\mathbf{w},\|\mathbf{w}\|_2\leq 1} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\hat{\mathcal{R}}_{\mathbf{S}^{(k)}}(\mathbf{w})] + \epsilon^{(k)},$$

from Claim 2.3, we derive with high probability:

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[|\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \eta_{\mathbf{w}^{(k)}}(\mathbf{x})] \leq \min_{\mathbf{w},\|\mathbf{w}\|_2\leq 1} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[|\langle \mathbf{w}, \mathbf{x}\rangle| \eta_{\mathbf{w}}(\mathbf{x})] + \epsilon^{(k)}.$$

Then the margin $\gamma^{(k)}$ is estimated minimizing Eq. (2.1) given $\mathbf{w}^{(k)}$, following Lemma 2.4 with $R^{(k)} = \max_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}} \|\mathbf{x}\|_2$ the radius of the truncated support of the marginal distribution $\mathcal{D}_\mathbf{x}$ at iteration k, we can assume that $\frac{\gamma^{(k)}}{R^{(k)}} \approx 1, \forall k \in [K]$, one may argue that the assumption is unrealistic knowing that the sequence of $(\gamma^{(k)})_{k=1}^m$ decreases overall, but as we show in the supplementary, we prove in Theorem B.1 that under some convergence guarantees of the pairs $\{(\mathbf{w}^{(k)}, \mathbf{w}^{(k+1)})\}_{k=1}^{m-1}$, one can show that the sequence $\{R^{(k)}\}_{k=1}^m$ decreases as a function of $\gamma^{(k)}$ respectively to k. As a result, we can derive with high probability:

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta_{\mathbf{w}^{(k)}}(\mathbf{x}) \big| |\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \geq \gamma^{(k)}] \leq \min_{\mathbf{w},\|\mathbf{w}\|_2\leq 1} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta_{\mathbf{w}}(\mathbf{x}) \big| |\langle \mathbf{w}, \mathbf{x}\rangle| \geq \gamma^{(k)}] + \epsilon^{(k)}.$$

and the fact that for each round k only points with comparable large margins are considered, we can assume that the conditional covariance for these examples

with an unsigned margin greater than $\gamma^{(k)}$ satisfy $\mathrm{Cov}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}(|\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle|, \eta_{\mathbf{w}^{(k)}}(\mathbf{x})) \approx \mathrm{Cov}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}(|\langle \mathbf{w}^{*}, \mathbf{x}\rangle|, \eta_{\mathbf{w}^{*}}(\mathbf{x}))$, which implies that at round k:

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\mathbf{w}^{(k)}}(\mathbf{x})\big||\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \geq \gamma^{(k)}] \;-\; \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[\eta_{\mathbf{w}^{*}}(\mathbf{x})\big||\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \geq \gamma^{(k)}] \quad \leq \quad \epsilon^{(k)}$$

where $\epsilon^{(k)} = \frac{3R^{(k)}}{2\sqrt{P^{(k)}}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[|\langle \mathbf{w}^{*}, \mathbf{x}\rangle|\big||\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle|\geq\gamma^{(k)}]}$, $\forall k \in [K]$.

From the statement of Claim 2.2 and giving the pair $(\mathbf{w}^{(k)}, \gamma^{(k)})$, we obtain with high probability that at round k:

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h_{\mathbf{w}^{(k)}}(\mathbf{x}) \neq y\big||\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \geq \gamma^{(k)}] \leq \boldsymbol{\eta}^{*} + \epsilon^{(k)}. \tag{2.4}$$

When the while loop terminates, we have accounted $m \leq K$ halfspaces in the list L_m satisfying Eq. (2.4). For all $k \in I$, every classifier $h_{\mathbf{w}^{(k)}}$ in L_m has guarantees on an empirical distribution mass of at least $\widetilde{\kappa} = \min_{k\in I} \mathbb{P}_{\mathbf{x}\sim\mathbf{S}^{(k)}}[|\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \geq \gamma^{(k)}]$; the DKW (Dvoretzky-Kiefer-Wolfowitz) inequality [46] implies that the true probability mass $\kappa = \min_{k\in I} \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[|\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \geq \gamma^{(k)}]$ of this region is at least $\widetilde{\kappa} - \sqrt{\frac{\log\frac{2}{\delta}}{2|\mathbf{S}^{(n)}|}}$ with probability $1-\delta$, where $n = \arg\min_{k\in I} \mathbb{P}_{\mathbf{x}\sim\mathbf{S}^{(k)}}[|\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \geq \gamma^{(k)}]$.
The pruning phase in the algorithm ensures that these regions are disjoint for all halfspaces in L_m, it follows that using the Boole–Fréchet inequality [21] on the conjunctions of Eq. (2.4) overall rounds $k \in [I]$, implies that L_m classifies at least a $(1 - m\kappa)$-fraction of the total probability mass of $\mathcal{D}$ with guarantees of Eq. (2.4) with high probability, let $\pi_{K+1} = \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}[x \in \mathbf{S}^{(K+1)}]$ be the probability mass of the region not accounted by L_m. We argue that this region is negligible from the fact that $|\mathbf{S}^{(K+1)}| < \ell$ and $\ell \ll u$, such that setting $\boldsymbol{\epsilon} = \max_{k\in I} \epsilon^{(k)} + \pi_{K+1}$ provides the result. □

In the following, we show that the misclassification error of the classifier L_m output of Algorithm 1 is at most equal to the error of the supervised classifier obtained over the labeled training set $\mathbf{S}_\ell$, when using the same learning procedure. This result suggests that the use of unlabeled data in Algorithm 1 does not degrade the performance of the initially supervised classifier.

Theorem 2.8. *Let $\mathbf{S}_\ell$ be a set of i.i.d. samples of size ℓ drawn from a distribution $\mathcal{D} = \mathcal{O}(f, \mathcal{D}_{\mathbf{x}}, \eta^{(0)})$ on $\mathbb{R}^d \times \{-1, +1\}$, where f is an unknown concept function and $\eta^{(0)}$ an unknown parameter function bounded by $1/2$, let $\mathbf{X}_u$ be an unlabeled set of size u drawn i.i.d. from $\mathcal{D}_{\mathbf{x}}$. Let L_m be the output of Algorithm 1 on input $\mathbf{S}_\ell$ and $\mathbf{X}_u$, and let $h_{\mathbf{w}^{(0)}}$ be the halfspace of the first iteration obtained from the empirical distribution $\mathbf{S}^{(0)} = \mathbf{S}_\ell$, there is a high probability that:*

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[L_m(\mathbf{x}) \neq y] \leq \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h_{\mathbf{w}^{(0)}}(\mathbf{x}) \neq y]$$

Proof. By the guarantees of Lemma 2.7, the classifier $h_{\mathbf{w}^{(0)}}$ obtained on running SGD on $\hat{\mathcal{R}}_{\mathbf{S}^{(0)}}$ with projection to the unit l_2-ball for $P^{(0)}$ steps satisfies :

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\text{Relu}(-y\langle\mathbf{w}^{(0)},\mathbf{x}\rangle)] \;-\; \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\text{Relu}(-y\langle\mathbf{w}^{*},\mathbf{x}\rangle)] \quad\leq\quad \frac{3\max_{\mathbf{x}\in\mathbf{S}_\ell}\|\mathbf{x}\|}{2\sqrt{P^{(0)}}}$$

Let k be the iteration at which the first pair $(\mathbf{w}^{(1)},\gamma^{(1)})$ is added to L_m. The first *pruning* phase in Algorithm 1 results in a set $\mathbf{S}^{(k)} \subseteq \mathbf{S}_\ell \cup \bigcup_{i=1}^{k-1}\mathbf{S}_u^{(i)}$. Claim 2.5 ensures that the probability of corruption in the pseudo-labeled set $\bigcup_{i=1}^{k-1}\mathbf{S}_u^{(i)}$ is bounded by $\max_{j\in[k]}\boldsymbol{\eta}^{(j)} \leq \boldsymbol{\eta}^* + \boldsymbol{\epsilon}$.
In other words, the weight vector $\mathbf{w}^{(1)}$ is obtained from an empirical distribution that includes both the initial labeled set $\mathbf{S}_\ell$ and a pseudo-labeled set from $\mathbf{X}_u$. Particularly, if this pseudo-labeled set is not empty, then its pseudo-labeling error is nearly optimal, which implies that $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h_{\mathbf{w}^{(1)}}(\mathbf{x}) \neq y] \leq \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h_{\mathbf{w}^{(0)}}(\mathbf{x}) \neq y]$.

Ultimately, L_m classifies a large fraction of the probability mass of $\mathcal{D}$ with nearly optimal guarantees (e.i., Eq. (2.4) in proof of Theorem 2.6) and the rest using $h_{\mathbf{w}^{(1)}}$ with an error of misclassification at most equal to $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h_{\mathbf{w}^{(0)}}(\mathbf{x}) \neq y]$. □

We show the assumption admitted for the proof of Theorem 4.5 in the following Theorem.

Theorem 2.9. *Let $\mathbf{S}_\ell$ be a set of i.i.d. samples of size ℓ drawn from a distribution $\mathcal{D} = \mathcal{O}(f,\mathcal{D}_\mathbf{x},\eta^{(0)})$ on $\mathbb{B}^d\times\{-1,+1\}$, where f is an unknown concept function and $\eta^{(0)}$ an unknown parameter function bounded by $1/2$, let $\mathbf{X}_u$ be an unlabeled set of size u drawn i.i.d. from $\mathcal{D}_\mathbf{x}$, let $L_m = [(\mathbf{w}^{(i)},\gamma^{(i)})]_{i=1}^m$ be the outputted list by Algorithm 1 on input $\mathbf{S}_\ell$ and $\mathbf{X}_u$, and let $\alpha^{(k)}$ be the smallest angle between two consecutive halfspaces $(\mathbf{w}^{(k)},\mathbf{w}^{(k+1)})$ in L_m for all $k\in[m-1]$. We define $(R^{(k)})_{i=1}^m$ the sequence of bounds where each $R^{(k)}$ is the bound of the margin $|\langle\mathbf{w}^{(k)},\mathbf{x}\rangle|$ over $\mathcal{D}_\mathbf{x}$ when the pair $(\mathbf{w}^{(k)},\gamma^{(k)})$ is obtained, we have for all k in $[m]$:*

$$R^{(1)} = 1\,; \quad R^{(k+1)} = \begin{cases} \sin\left(\alpha^{(k)} + \arcsin\left(\gamma^{(k)}\right)\right) & \text{if } \alpha^{(k)} \in \left[0, \arccos\left(\gamma^{(k)}\right)\right], \\ 1 & \text{otherwise.} \end{cases}$$

Proof. For $k=1$, it is trivial to say that $R^{(1)}$ is the upper-bound for the margin distribution of $\mathbf{w}^{(1)}$ using Cauchy-Schwarz inequality and given $\|\mathbf{w}^{(1)}\|_2 \leq 1$. Next, we suppose that the definition is true for k, and we will show in the following that the definition holds for $k+1$, for that we will distinguish two different cases:

- Case when $\alpha^{(k)} \in \left[0, \arccos\left(\gamma^{(k)}\right)\right]$
  In Figure 2.1, left, we show that if the halfspace with weight vector $\mathbf{w}^{(k+1)}$ does not extremely deviates from $\mathbf{w}^{(k)}$, then we can express its upper-bound $R^{(k+1)}$ of the margin distribution on the truncated space of $\mathcal{D}_{\mathbf{x}}$ as a function of $\gamma^{(k)}$ and the angle deviation $\alpha^{(k)}$. Note that if $\alpha^{(k)} = 0$, then we have that $R^{(k+1)} = \gamma^{(k)}$.

- Case when $\alpha^{(k)} \in \left]\arccos\left(\gamma^{(k)}\right), \frac{\pi}{2}\right[$
  In Figure 2.1, right, we show that if the halfspace with weight vector $\mathbf{w}^{(k+1)}$ extremely deviates from $\mathbf{w}^{(k)}$, then its upper-bound $R^{(k+1)}$ of the margin distribution on the truncated space of $\mathcal{D}_{\mathbf{x}}$ is equal to the radius of the unit ball.

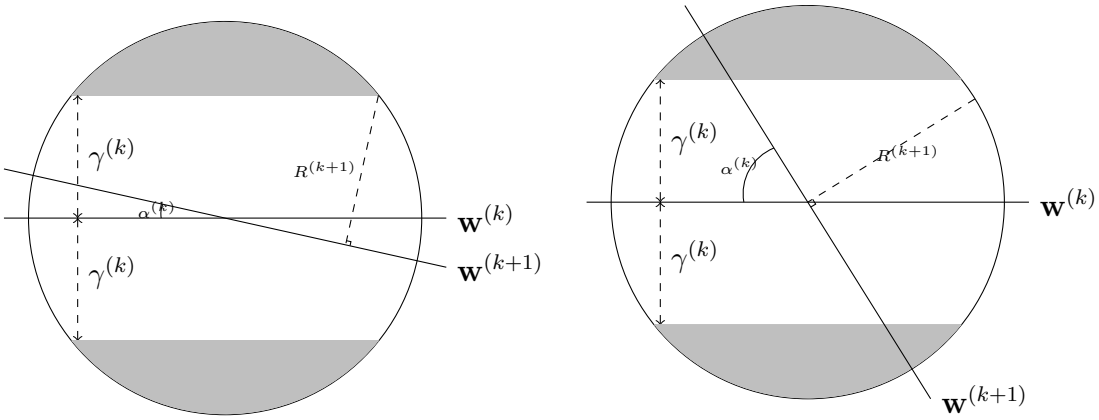


Figure 2.1: Case when $\alpha^{(k)} \in \left[0, \arccos\left(\gamma^{(k)}\right)\right]$ (left) and $\alpha^{(k)} \in \left]\arccos\left(\gamma^{(k)}\right), \frac{\pi}{2}\right[$ (right) for a pair $\left(\mathbf{w}^{(k)}, \mathbf{w}^{(k+1)}\right)$ in L_m.

□

For a pair $(\mathbf{w}^{(k)}, \gamma^{(k)})$ estimated in Algorithm 1 at iteration $k \in [K]$, the conditional covariance $\mathrm{Cov}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}(|\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle|, \eta_{\mathbf{w}^{(k)}}(\mathbf{x}) \mid |\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \geq \gamma^{(k)})$ for examples in $\mathcal{D}$ with an unsigned margin greater or equal than $\gamma^{(k)}$ to $\mathbf{w}^{(k)}$ is *comparable* to the conditional covariance $\mathrm{Cov}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}(|\langle \mathbf{w}^{*}, \mathbf{x}\rangle|, \eta_{\mathbf{w}^{*}}(\mathbf{x}) \mid |\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \geq \gamma^{(k)})$ for these same examples. Similarly, the unsigned margin average of these examples to $\mathbf{w}^{(k)}$ denoted as $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}(|\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \mid |\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \geq \gamma^{(k)})$ is also *comparable* to the unsigned margin average to $\mathbf{w}^{*}$ denoted as $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}(|\langle \mathbf{w}^{*}, \mathbf{x}\rangle| \mid |\langle \mathbf{w}^{(k)}, \mathbf{x}\rangle| \geq \gamma^{(k)})$.

In the following, we relate the process of pseudo-labeling to the corruption noise model $\mathcal{O}(f, \mathcal{D}_{\mathbf{x}}, \eta^{(k)})$ for all pseudo-labeling iterations k in Algorithm 1, then we present a bound over the misclassification error of the classifier L_m outputted by the algorithm and demonstrate that this misclassification error is upper-bounded by the misclassification error of the fully supervised halfspace.

Claim 2.10. *Let* $\mathbf{S}^{(0)} = \mathbf{S}_\ell$ *be a labeled set drawn i.i.d. from* $\mathcal{D} = \mathcal{O}(f, \mathcal{D}_\mathbf{x}, \eta^{(0)})$ *and* $\mathbf{U}^{(0)} = \mathbf{X}_u$ *an initial unlabeled set drawn i.i.d. from* $\mathcal{D}_\mathbf{x}$*. For all iterations* $k \in [K]$ *of Algorithm 1; the active labeled set* $\mathbf{S}^{(k)}$ *is drawn i.i.d. from* $\mathcal{D} = \mathcal{O}(f, \mathcal{D}_\mathbf{x}, \eta^{(k)})$ *where the corruption noise distribution* $\eta^{(k)}$ *is bounded by:*

$$\forall k \in [K], \quad \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta^{(k)}(\mathbf{x})\big|\mathbf{x} \in \mathbf{S}^{(k)}] \leq \max_{j\in[K]} \boldsymbol{\eta}^{(j)}$$

Proof. We know that $\forall k \in [K], \mathbf{S}^{(k)} \subseteq \mathbf{S}^{(0)} \cup \bigcup_{i=0}^{k-1} \mathbf{S}_u^{(i)}$, where $\mathbf{S}_u^{(i)}$ is the set of pseudo-labeled pairs of examples $\mathbf{x}$ from $\mathbf{U}^{(i)}$, $\mathbf{S}_u^{(i)} = \emptyset$ for the iterations $i \in [K]$ when no examples are pseudo-labeled. Then the noise distribution $\eta^{(k)}$ satisfies for all $k \in [K]$:

$$\begin{aligned}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta^{(k)}(\mathbf{x})\mathbb{1}_{\mathbf{x}\in\mathbf{S}^{(k)}}] &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta^{(k)}(\mathbf{x})\mathbb{1}_{\mathbf{x}\in\mathbf{S}^{(k)}\cap\mathbf{S}^{(0)}}] + \sum_{i=0}^{k-1} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta^{(k)}(\mathbf{x})\mathbb{1}_{\mathbf{x}\in\mathbf{S}^{(k)}\cap\mathbf{S}_u^{(i)}}] \\ &= \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\mathbf{x} \in \mathbf{S}^{(k)} \cap \mathbf{S}^{(0)}]\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta^{(0)}(\mathbf{x})\big|\mathbf{x} \in \mathbf{S}^{(k)} \cap \mathbf{S}^{(0)}]+ \\ &\sum_{i=1}^{k-1} \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\mathbf{x} \in \mathbf{S}^{(k)} \cap \mathbf{S}_u^{(i)}]\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta_{\mathbf{w}^{(i)}}(\mathbf{x})\big|\mathbf{x} \in \mathbf{S}^{(k)} \cap \mathbf{S}_u^{(i)}]\end{aligned}$$

If we condition on $\mathbf{x} \in \mathbf{S}^{(k)}$, we obtain for all $k \in [K]$:

$$\begin{aligned}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta^{(k)}(\mathbf{x})\big|\mathbf{x} \in \mathbf{S}^{(k)}] &= \mathbb{P}[\mathbf{x} \in \mathbf{S}^{(0)}\big|\mathbf{x} \in \mathbf{S}^{(k)}]\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta^{(0)}(\mathbf{x})\big|\mathbf{x} \in \mathbf{S}^{(k)} \cap \mathbf{S}^{(0)}]+ \\ &\sum_{i=0}^{k-1} \mathbb{P}[\mathbf{x} \in \mathbf{S}_u^{(i)}\big|\mathbf{x} \in \mathbf{S}^{(k)}]\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta_{\mathbf{w}^{(i)}}(\mathbf{x})\big|\mathbf{x} \in \mathbf{S}^{(k)} \cap \mathbf{S}_u^{(i)}],\end{aligned}$$

this equation includes the initial corruption of the labeled set $\mathbf{S}^{(0)} = \mathbf{S}_\ell$ in addition to the noise injected by each classifier $h_{\mathbf{w}^{(k)}}$ at each round k when pseudo-labeling occurs. Now that we have modeled the process of pseudo-labeling, the result is straightforward considering the fact that $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta^{(0)}(\mathbf{x})] \leq \boldsymbol{\eta}^{(0)}; \forall k \in [K], \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\eta_{\mathbf{w}^{(k)}}(\mathbf{x})] \leq \boldsymbol{\eta}^{(k)}$; and,

$$\begin{aligned}&\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\mathbf{x} \in \mathbf{S}^{(0)}\big|\mathbf{x} \in \mathbf{S}^{(k)}]+ \sum_{i=0}^{k-1} \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\mathbf{x} \in \mathbf{S}_u^{(i)}\big|\mathbf{x} \in \mathbf{S}^{(k)}] \\ &\qquad = \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_\mathbf{x}}[\mathbf{x} \in \mathbf{S}^{(0)} \cup \bigcup_{i=0}^{k-1} \mathbf{S}_u^{(k)}\big|\mathbf{x} \in \mathbf{S}^{(k)}] \leq 1. \qquad \square\end{aligned}$$

2.5 Empirical Results

We compare the proposed approach to state-of-the-art strategies developed over the three fundamental working assumptions in semi-supervised learning over ten publicly available datasets. We shall now describe the corpora and methodology.

Datasets. We mainly consider benchmark data sets from [30]. Some of these collections such as *baseball-hockey*, *pc-mac* and *religion-atheism* are binary classification tasks extracted from the 20-newsgroups data set.

data set	d	-1	$+1$	$\ell + u$	test
one-two	64	177	182	251	108
banknote	4	762	610	919	453
odd-even	64	906	891	1257	540
pc-mac	3868	982	963	1361	584
baseball-hockey	5724	994	999	1395	598
religion-atheism	7829	1796	628	1696	728
spambase	57	2788	1813	3082	1519
weather	17	43993	12427	37801	18619
delicious2	500	9610	6495	12920	3185
mediamill2	120	15969	27938	30993	12914

Table 2.1: data set statistics, -1 and $+1$ refer to the size of negative and positive class respectively, and test is the size of test set.

We used tf-idf representation for all textual data sets above. *spambase* is a collection of spam e-mails from the UCI repository [43]. *one-two*, *odd-even* are handwritten digits recognition tasks originally from optical recognition of handwritten digits database also from UCI repository, *one-two* is digits "1" versus "2"; *odd-even* is the artificial task of classifying odd "1, 3, 5, 7, 9" versus even "0, 2, 4, 6, 8" digits. *weather* is a data set from Kaggle which contains about ten years of daily weather observations from many locations across Australia, and the objective is to classify the next-day rain target variable. We have also included data sets from extreme classification repository [18] *mediamill2* and *delicious2* by selecting the label which gives the best ratio in class distribution. The statistics of these data sets are given in Table 2.1.

Baseline methods. We implemented the halfspace or Linear Threshold Function (LTF) using TensorFlow 2.0 in python aside from Algorithm 1[1] (L_m), we ran a Support Vector Machine (SVM) [39] with a linear kernel from the LIBLINEAR library [49] as another supervised classifier. We compared results with a semi-supervised Gaussian naive Bayes model (GM) [30] from the scikit-learn library. The working hypothesis behind (GM) is the cluster assumption stipulating that data contains homogeneous labeled clusters, which can be detected using unlabeled training samples.

[1]For research purposes, the code will be freely available.

We also compared results with label propagation (LP) [131] which is a semi-supervised graph-based technique. We used the implementation of LP from the scikit-learn library. This approach follows the manifold assumption that the decision boundary is located on a low-dimensional manifold and that unlabeled data may be utilized to identify it. We also included entropy regularized logistic regression (ERLR) proposed by [55] from [76]. This approach is based on low-density separation that stipulates that the decision boundary lies on low-density regions. In the implementation of [76], the initially supervised classifier is a logistic regression that has a similar performance to the SVM classifier. We tested these approaches with relatively small labeled training sets $\ell \in \{10, 50, 100\}$, and because labeled information is scarce, we used the default hyper-parameters for all approaches.

Experimental setup. In our experiments, we have randomly chosen 70% of each data collection for training and the remaining 30% for testing. We randomly selected sets of different sizes (i.e., $\ell \in \{10, 50, 100\}$) from the training set as labeled examples; the remaining was considered as unlabeled training samples. Results are evaluated over the test set using the accuracy measure. Each reported performance value is the average over the 20 random (labeled/unlabeled/test) sets of the initial collection. All experiments are carried out on a machine with an Intel Core i7 processor, 2.2GhZ quad-core, and 16Go 1600 MHz of RAM memory.

Analysis of results. Table 2.2 summarizes the results. We used boldface (resp. underline) to indicate the highest (resp. the second-highest) performance rate, and the symbol $^{\downarrow}$ indicates that performance is significantly worse than the best result, according to a Wilcoxon rank-sum test with a p-value threshold of 0.01 [117]. From these results, it comes out that the proposed approach (L_m) consistently outperforms the supervised halfspace (LTF). This finding is in line with the result of Theorem 2.8. Furthermore, compared to other techniques, L_m generally performs the best or the second best. We also notice that in some cases, LP, GM, and ERLR outperform the supervised approaches, SVM and LTF (i.e., GM on *spambase* for $\ell \in \{10, 50\}$), but in other cases, they are outperformed by both SVM and LTF (i.e., GM on *religion-atheism*). These results suggest that unlabeled data contain useful information for classification and that existing semi-supervised techniques may use it to some extent. They also highlight that developing semi-supervised algorithms following the given assumptions are necessary for learning with labeled and unlabeled training data but not sufficient. The importance of developing theoretically founded semi-supervised

Table 2.2: Mean and standard deviations of accuracy on test sets over the 20 trials for each data set. The best and the second-best performance are respectively in bold and underlined. $^{\downarrow}$ indicates statistically significantly worse performance than the best result, according to a Wilcoxon rank-sum test ($p < 0.01$) [117].

Dataset	ℓ	SVM	LTF	LP	GM	ERLR	L_m
one-two	10	$61.38 \pm 13.71^{\downarrow}$	$70.87 \pm 13.24^{\downarrow}$	$48.61 \pm 3.98^{\downarrow}$	$\underline{75.09 \pm 1.30}$	$53.65 \pm 10.65^{\downarrow}$	$\mathbf{77.77 \pm 1.75}$
	50	$\mathbf{92.77 \pm 3.05}$	$88.00 \pm 3.24^{\downarrow}$	$49.35 \pm 4.20^{\downarrow}$	$84.67 \pm 4.98^{\downarrow}$	$75.78 \pm 8.74^{\downarrow}$	$\underline{91.34 \pm 3.21}$
	100	$\mathbf{96.15 \pm 1.38}$	$92.50 \pm 1.43^{\downarrow}$	$67.82 \pm 12.99^{\downarrow}$	$86.52 \pm 3.26^{\downarrow}$	$79.25 \pm 6.87^{\downarrow}$	$\underline{94.62 \pm 2.46}$
banknote	10	$57.50 \pm 7.21^{\downarrow}$	$\underline{69.40 \pm 5.53}^{\downarrow}$	$55.98 \pm 2.00^{\downarrow}$	$69.04 \pm 4.60^{\downarrow}$	$56.71 \pm 4.53^{\downarrow}$	$\mathbf{77.24 \pm 3.81}$
	50	$61.67 \pm 4.86^{\downarrow}$	$\underline{82.31 \pm 2.13}^{\downarrow}$	$56.28 \pm 1.89^{\downarrow}$	$75.48 \pm 5.30^{\downarrow}$	$65.95 \pm 2.01^{\downarrow}$	$\mathbf{85.64 \pm 5.36}$
	100	$71.65 \pm 6.24^{\downarrow}$	$\underline{89.38 \pm 3.24}$	$57.20 \pm 2.19^{\downarrow}$	$77.56 \pm 4.34^{\downarrow}$	$70.95 \pm 3.24^{\downarrow}$	$\mathbf{90.82 \pm 3.31}$
odd-even	10	$53.45 \pm 4.80^{\downarrow}$	$58.20 \pm 4.71^{\downarrow}$	$50.37 \pm 1.95^{\downarrow}$	$\underline{60.69 \pm 7.48}$	$50.40 \pm 2.21^{\downarrow}$	$\mathbf{63.21 \pm 7.51}$
	50	$64.75 \pm 5.65^{\downarrow}$	$\underline{76.84 \pm 2.99}^{\downarrow}$	$50.37 \pm 1.95^{\downarrow}$	$62.67 \pm 5.82^{\downarrow}$	$53.17 \pm 4.80^{\downarrow}$	$\mathbf{80.61 \pm 3.10}$
	100	$75.89 \pm 6.25^{\downarrow}$	$\underline{77.68 \pm 4.56}^{\downarrow}$	$53.37 \pm 1.95^{\downarrow}$	$64.25 \pm 8.18^{\downarrow}$	$59.23 \pm 6.28^{\downarrow}$	$\mathbf{84.58 \pm 2.12}$
pc-mac	10	$51.00 \pm 3.22^{\downarrow}$	$\underline{54.92 \pm 2.00}^{\downarrow}$	$50.93 \pm 1.59^{\downarrow}$	$54.76 \pm 3.42^{\downarrow}$	$50.14 \pm 2.06^{\downarrow}$	$\mathbf{57.75 \pm 3.19}$
	50	$58.85 \pm 5.09^{\downarrow}$	$\underline{61.78 \pm 2.86}^{\downarrow}$	$50.83 \pm 2.08^{\downarrow}$	$58.78 \pm 4.31^{\downarrow}$	$49.71 \pm 1.99^{\downarrow}$	$\mathbf{64.31 \pm 3.55}$
	100	$64.57 \pm 4.42^{\downarrow}$	$\underline{67.98 \pm 2.37}$	$50.76 \pm 2.26^{\downarrow}$	$62.49 \pm 1.88^{\downarrow}$	$50.36 \pm 2.19^{\downarrow}$	$\mathbf{68.15 \pm 5.66}$
baseball-hockey	10	$51.57 \pm 2.98^{\downarrow}$	$55.41 \pm 3.16^{\downarrow}$	$\mathbf{56.53 \pm 5.18}$	$49.86 \pm 1.77^{\downarrow}$	$49.88 \pm 1.89^{\downarrow}$	$\underline{56.47 \pm 5.50}$
	50	$58.66 \pm 6.90^{\downarrow}$	$\underline{69.29 \pm 4.32}$	$50.11 \pm 1.84^{\downarrow}$	$66.76 \pm 5.40^{\downarrow}$	$50.16 \pm 1.90^{\downarrow}$	$\mathbf{72.85 \pm 6.52}$
	100	$68.40 \pm 4.65^{\downarrow}$	$\underline{76.25 \pm 2.41}^{\downarrow}$	$49.97 \pm 1.82^{\downarrow}$	$71.12 \pm 5.06^{\downarrow}$	$50.35 \pm 1.89^{\downarrow}$	$\mathbf{79.48 \pm 4.36}$
religion-atheism	10	67.30 ± 6.95	$57.30 \pm 4.89^{\downarrow}$	$\underline{67.59 \pm 6.36}$	$60.67 \pm 16.37^{\downarrow}$	$\mathbf{71.95 \pm 5.03}$	$64.25 \pm 7.24^{\downarrow}$
	50	$\mathbf{74.61 \pm 1.62}$	$71.79 \pm 1.98^{\downarrow}$	$67.43 \pm 6.05^{\downarrow}$	69.16 ± 7.88	$\underline{74.16 \pm 1.88}$	72.47 ± 2.00
	100	$\mathbf{74.66 \pm 1.59}$	73.67 ± 1.76	$62.84 \pm 19.33^{\downarrow}$	$70.45 \pm 4.39^{\downarrow}$	73.21 ± 1.75	$\underline{73.77 \pm 1.82}$
spambase	10	$61.20 \pm 5.15^{\downarrow}$	$57.80 \pm 5.29^{\downarrow}$	$60.82 \pm 0.84^{\downarrow}$	$\mathbf{74.41 \pm 6.64}$	$53.38 \pm 11.23^{\downarrow}$	$\underline{68.92 \pm 5.83}^{\downarrow}$
	50	$62.59 \pm 9.42^{\downarrow}$	74.99 ± 6.04	$61.15 \pm 0.86^{\downarrow}$	$\mathbf{78.25 \pm 2.62}$	$53.63 \pm 9.86^{\downarrow}$	$\underline{76.13 \pm 3.08}$
	100	$69.43 \pm 10.19^{\downarrow}$	$\underline{80.07 \pm 4.08}$	$61.24 \pm 10.26^{\downarrow}$	$79.08 \pm 2.83^{\downarrow}$	$58.21 \pm 6.34^{\downarrow}$	$\mathbf{81.93 \pm 2.46}$
weather	10	74.85 ± 0.51	$68.09 \pm 1.73^{\downarrow}$	$\mathbf{75.49 \pm 0.34}$	75.02 ± 2.79	$40.35 \pm 17.29^{\downarrow}$	$\underline{75.08 \pm 4.18}$
	50	$\underline{75.79 \pm 0.28}$	75.30 ± 3.85	$\mathbf{77.99 \pm 0.31}$	75.68 ± 2.78	$41.55 \pm 27.39^{\downarrow}$	75.34 ± 3.80
	100	$\mathbf{77.99 \pm 0.25}$	76.27 ± 3.64	$\mathbf{77.99 \pm 0.25}$	74.92 ± 1.92	$46.00 \pm 24.87^{\downarrow}$	$\underline{77.28 \pm 2.99}$
delicious2	10	$\underline{51.83 \pm 9.88}$	$50.59 \pm 2.65^{\downarrow}$	$\mathbf{60.02 \pm 0.61}$	$49.41 \pm 3.83^{\downarrow}$	$51.83 \pm 10.42^{\downarrow}$	$51.08 \pm 1.80^{\downarrow}$
	50	$\mathbf{60.04 \pm 0.62}$	$54.78 \pm 2.57^{\downarrow}$	$\underline{60.00 \pm 0.59}$	$48.35 \pm 1.31^{\downarrow}$	$53.48 \pm 8.66^{\downarrow}$	$55.37 \pm 3.33^{\downarrow}$
	100	$\underline{58.88 \pm 3.70}$	$56.04 \pm 1.83^{\downarrow}$	$\mathbf{59.87 \pm 0.67}$	$48.92 \pm 0.94^{\downarrow}$	$54.43 \pm 7.27^{\downarrow}$	$56.54 \pm 1.87^{\downarrow}$
mediamill2	10	$62.54 \pm 2.62^{\downarrow}$	$60.98 \pm 6.85^{\downarrow}$	$36.35 \pm 0.15^{\downarrow}$	$\underline{63.92 \pm 1.71}$	$47.24 \pm 14.08^{\downarrow}$	$\mathbf{64.31 \pm 3.14}$
	50	$63.64 \pm 0.15^{\downarrow}$	$60.88 \pm 7.45^{\downarrow}$	$36.36 \pm 0.15^{\downarrow}$	$\mathbf{65.98 \pm 3.32}$	$58.58 \pm 11.88^{\downarrow}$	$\underline{65.41 \pm 4.83}$
	100	$63.64 \pm 0.15^{\downarrow}$	64.26 ± 4.79	$36.37 \pm 0.15^{\downarrow}$	$\underline{67.34 \pm 0.73}$	$63.64 \pm 0.16^{\downarrow}$	$\mathbf{67.80 \pm 2.21}$

algorithms exhibiting the generalization ability of the method can provide a better understanding of the usefulness of unlabeled training data in the learning process.

2.6 Conclusion

In this chapter, we presented a first bound over the misclassification error of a self-training algorithm that iteratively finds a list of halfspaces from partially labeled training data. Each round consists of two steps: exploration and pruning. The exploration phase aims to determine the halfspace with the largest margin and assign pseudo-labels to unlabeled observations with an unsigned margin larger than the discovered threshold. The pseudo-labeled instances are then added to the training set, and the procedure is repeated until there are no more unlabeled instances to pseudo-label. In the pruning phase, the last halfspace with the largest threshold is preserved, ensuring that there are no more unlabeled samples with an unsigned margin greater

than this threshold and pseudo-labeled samples with an unsigned margin greater than the specified threshold are removed. Our findings are based on recent theoretical advances in robust supervised learning of polynomial algorithms for training halfspaces under large margin assumptions with a corrupted label distribution using the Massart noise model. We ultimately show that the use of unlabeled data in the proposed self-training algorithm does not degrade the performance of the initially supervised classifier. As future work, we are interested in quantifying the real gain of learning with unlabeled and labeled training data compared to a fully supervised scheme.

Chapter 3

Pool-Based Active Learning with Proper Topological Regions

Pool-based active learning methods are one of the most promising paradigms of active learning in solving the problem of annotation efficiency. One of the main criticism of these approaches is that they are supposed to operate under low-budget regimes where they can have an advantage over semi-supervised or self-supervised methods. However, most of these approaches rely on the underlying trained estimator accuracy, which often has a higher sample complexity than the low-budget regime scenario. Secondly, they commonly share the initial training set selection drawback, namely the cold-start problem. This chapter presents a meta-approach for pool-based active learning strategies in the context of multi-class classification tasks. Our approach relies on the proposed concept of learning on proper topological regions with an underlying smoothness assumption on the metric space. This allows us to increase the training sample size during the rounds while operating in a low-budget regime scenario. We show empirically on various real benchmark datasets that our approach dramatically improves the performance of uncertainty-based sampling strategies over the random selection, not only for the cold-start problem but overall the iterations in a low-budget regime. Furthermore, comparisons on the same benchmarks show that the performance of our approach is competitive to the state-of-the-art methods from the literature that address the cold-start problem in active learning. This chapter is based on the following paper [HDMA22].

3.1 Introduction

In recent years, machine learning has found gainful application in diverse domains. However, it still has a heavy dependence on expensive labeled data: advances in

cheap computing and storage have made it easier to store and process large amounts of unlabeled data, but labeling needs often to be done by humans or using costly tools. Therefore, there is a need to develop general domain-independent methods to learn models effectively from a large amount of unlabeled data at the disposal, along with a minimal amount of labeled data. Active learning specifically aims to detect the observations to be labeled to optimize the learning process and efficiently reduce the labeling cost. In this setting, learning occurs iteratively. At each round, the algorithms can interactively query a ground truth oracle to label unlabeled examples. Then, after training, the algorithms proactively select the subset of examples to be labeled next from the pool of unlabeled data. The primary assumption behind the active learner algorithm concept is that machine learning algorithms could reach a higher level of performance while using a smaller number of training labels if they were allowed to choose the training data set [103]. Most common and straightforward active learning approaches are iterative, also known as pool-based methods [80, 38], where we first derive a model trained on a small random labeled subsample. Then, at each iteration, we choose unlabeled examples to query based on the predictions of the current model and a predefined priority score. These approaches show their limitations in low-budget regime scenarios from their need for a sufficient budget to learn a weak model [95]. The literature has shown that for active learning to operate in a low-budget regime successfully, we need to introduce a form of regularization in training [56] usually found in other sub-domains, such as semi-supervised learning or self-learning [29]. Another line of work shows that the choice of the initial seed set in these approaches significantly impacts the end performance of their models [63, 34], also known as the cold-start problem in active learning. Our work is a step further in this direction. We propose a unified meta-approach for pool-based active learning methods to efficiently resolve these previously mentioned drawbacks and to enhance even further the performance of these methods while reducing the amount of queried examples.

Topological data analysis (TDA) [48, 6] has been successful in various fields [118, 98, 67, 77], including machine learning. A critical insight in TDA, a widely accepted assumption, is that data sets often have nontrivial topologies that should be exploited in their analysis [25]. TDA provides mathematically well-founded and flexible tools based on the algebraic topology to recover topological information from data to get insights from this hidden information. Many of these tools use persistent homology which allows studying the underlying topological information of a wide variety data types, even in high-dimension. To understand the underlying topology, we

can construct Vietoris-Rips complexes [58] from the data which are then inspected through persistent homology, topological information is then encoded with persistence modules and diagrams [47]. These topological insights can then be exploited to enhance the study of the structure of the data [108, 84, 26].

3.2 Related literature

Different attempts have been made to reduce the annotation burden of machine learning algorithms. We can refer to the remarkable advances made in semi-supervised learning [131, 55, 14, 10, 126, 17], these methods take as input a small set of labeled training data together with a large number of unlabeled examples. They introduce a form of consistency regularization to the supervised loss function by applying data augmentation using unlabeled observations [29]. Similarly, pool-based active learning methods also take as input a large number of unlabeled examples together with an expert in which they iteratively query to annotate data samples in order to maximize the model knowledge while minimizing the number of queries. Most commonly known pool-based strategies are uncertainty sampling [80, 129], margin sampling and entropy sampling strategies [103]. Some proposed strategies rely on the query-by-committee approach [128, 120, 78], which learns an ensemble of models at each round. Query by bagging and query by boosting are two practical implementations of this approach that use bagging and boosting to build the committees [5]. There has been exhaustive research on how to derive efficient disagreement measures and query strategies from a committee, including vote entropy, consensus entropy, and maximum disagreement [103], whereas [8] introduces model selection for a committee. Some research focuses on solving a derived optimization problem for optimal query selection, in [101] they use Monte Carlo estimation of the expected error reduction on test examples. In contrast, other strategies employ Bayesian optimization on acquisition functions such as the probability of improvement or the expected improvement [53], and in [11], the authors propose to cast the problem of selecting the most relevant active learning criterion as an instance of the multi-armed bandit problem. Aside from the pool-based setting, we also find the stream-based setting for active learning in the literature [83, 13]. In this case, the learner has no access to any unlabeled examples. Instead, each unlabeled sample is given to him individually, and he queries its label if he finds it helpful. For instance, an example can be marked as valuable if the prediction is uncertain, and acquiring its label would remove this uncertainty.

Recent advances in active learning propose enhancing the pool-based methods by extracting knowledge from the distribution of unlabeled examples [20]. [93] propose to use clustering of unlabeled examples to boost the performance of pool-based active learners, with the expert annotating at each iteration cluster rather than single examples. Such strategy allows to reduce the annotation effort under the assumption that the cost of cluster annotation is comparable to single example labeling. Similarly, [74, 75] propose to combine clustering with Bayesian optimization in the stream-based setting. In [111]authors propose a procedure for binary domain feature sets to recover the labeling of a set of examples while minimizing the number of queries. They show that this routine reduces label complexity for training learners, [124] propose a two-stage clustering constraint in the active learning algorithm, a first exploration phase to discover representative clusters of all classes, and a post-clustering reassignment phase where the learner is constrained on the initial clusters found at the first stage. Finally, unsupervised algorithms such as clustering showed promising results for addressing the cold-start problem in pool-based active learning strategies [63, 34].

The meta-method that we propose for pool-based active learning relies on notions from topological data analysis (TDA), which has recently brought exciting new ideas to the machine learning community. Primarily, topological clustering has been used in unsupervised learning [19, 24]. Among these studies, `ToMATo` [52] is a mode-seeking clustering algorithm with a cluster merging phase guided by topological persistence [94].

3.3 Framework and topological considerations

3.3.1 Framework and notations

We consider multi-class classification problems such that the input space $\mathcal{X}$ is a subset of $\mathbb{R}^m$, the output space $\mathcal{Y} = \{1, \ldots, c\}$ is a set of unknown classes of cardinal $c \in \mathbb{N}, c \geq 2$, the pair $(\mathcal{X}, d)$ is a metric space, and $d\colon \mathcal{X} \times \mathcal{X} \to [0, \infty)$ is a fixed and known distance metric. Let $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be the data sample of size n generated by some unknown distribution over $\mathcal{X} \times \mathcal{Y}$ with unknown labels y_i, and let $\mathbb{P}$ be the unknown marginal distribution over $\mathcal{X}$. An active learner will have access to the unlabeled sample set $\mathcal{S}_\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^n$ and the conditional concept function $\mathcal{O} : \mathcal{X} \to \mathcal{Y}$. We denote $C_R\colon \{1, \ldots, n\} \to \{1, \ldots, k\}$ the partition function induced by a graph $R = (\mathcal{S}_\mathbf{x}, E)$ of k connected components with $k \in \mathbb{N}^*$, where $\mathcal{C}_R(\mathbf{x}_i) = \{\forall \mathbf{x}_j \in \mathcal{S}_\mathbf{x}, C_R(i) = C_R(j)\}$ is the connected component of graph R that includes the node $\mathbf{x}_i$. For all $i \in \{1, \ldots, n\}$,

we define $C_R^{\mathbb{P}}\colon \mathcal{S}_{\mathbf{x}} \to \mathcal{Y}$ as the labeling function that propagates the label of the sample with the highest density in the connected component $\mathcal{C}_R(\mathbf{x}_i)$ to all the examples of the same connected component in graph R :

$$C_R^{\mathbb{P}}(\mathbf{x}_i) = \mathcal{O}\left(\underset{\mathbf{x}_j \in \mathcal{C}_R(\mathbf{x}_i)}{\arg\max}\, \mathbb{P}(\mathbf{x}_j)\right), \forall i \in \{1, \ldots, n\},$$

note that $C_R^{\mathbb{P}}(.)$ is a crucial notion in our proposed meta-approach to allow pool-based active learning strategies to operate in a low-budget regime.

Definition 3.1 (Rips graph)**.** Given a finite point cloud $\mathcal{S}_{\mathbf{x}} = \{\mathbf{x}_i\}_{i=1}^n$ from a metric space $(\mathcal{X}, d)$ and a real number $\delta \geq 0$, the *Rips graph* $R_\delta(\mathcal{S}_{\mathbf{x}})$ is the graph of vertex set $\mathcal{S}_{\mathbf{x}}$ whose edges correspond to the pairs of points $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{S}_{\mathbf{x}}$ such that $d(\mathbf{x}_i, \mathbf{x}_j) \leq \delta$:

$$R_\delta(\mathcal{S}_{\mathbf{x}}) = (V, E) : V = \mathcal{S}_{\mathbf{x}}, E = \{(\mathbf{x}_i, \mathbf{x}_j) \in V^2, i \neq j, d(\mathbf{x}_i, \mathbf{x}_j) < \delta\},$$

let the hypothesis class of Rips graphs over $\mathcal{S}_{\mathbf{x}}$ be $\mathcal{H}_R = \{R_\delta(\mathcal{S}_{\mathbf{x}}), \forall \delta \in \mathbb{R}^+\}$.

Definition 3.2 (σ-Rips graph)**.** Given a finite point cloud $\mathcal{S}_{\mathbf{x}} = \{\mathbf{x}_i\}_{i=1}^n$ from a metric space $(\mathcal{X}, d)$, a real-valued function $\sigma_\delta\colon \mathcal{X}^2 \to \mathbb{R}_+^*$ with parameters δ, the *σ-Rips graph* $R_\delta^\sigma(\mathcal{S}_{\mathbf{x}})$ is the graph of vertex set $\mathcal{S}_{\mathbf{x}}$ whose edges correspond to the pairs of points $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{S}_{\mathbf{x}}$ such that $d(\mathbf{x}_i, \mathbf{x}_j) \leq \sigma_\delta(\mathbf{x}_i, \mathbf{x}_j)$:

$$R_\delta^\sigma(\mathcal{S}_{\mathbf{x}}) = (V, E) : V = \mathcal{S}_{\mathbf{x}}, E = \{(\mathbf{x}_i, \mathbf{x}_j) \in V^2, i \neq j, d(\mathbf{x}_i, \mathbf{x}_j) < \sigma_\delta(\mathbf{x}_i, \mathbf{x}_j)\},$$

let the hypothesis class of σ-Rips graphs over $\mathcal{S}_{\mathbf{x}}$ be $\mathcal{H}_{R^\sigma} = \{R_\delta^\sigma(\mathcal{S}_{\mathbf{x}}), \forall \delta \in dom(\sigma)\}$.

Classification algorithms in machine learning generally assume (often implicitly) that close samples in the considered metric space $(\mathcal{X}, d)$ are associated with similar labels, also known as the *smoothness assumption*. Given a data sample $\mathcal{S}$, the Rips graph $R_\delta(\mathcal{S}_{\mathbf{x}})$ encodes this notion to some extent with an appropriate threshold δ. However, class similarity might be different over the metric space. For example, samples in high-density regions should be closer to being associated with similar labels than those in low-density regions. Consequently, we need to generalize the definition of the Rips graph to account for such cases, namely the σ-Rips graph $R_\delta^\sigma(\mathcal{S}_{\mathbf{x}})$ with an appropriate threshold function σ_δ, and parameters δ. In this work, we choose the following threshold function:

$$\begin{aligned} \sigma_{(a,r,t)}\colon \mathcal{X} \times \mathcal{X} &\longrightarrow \mathbb{R}_+^* \\ (\mathbf{x}, \mathbf{x}') &\longrightarrow a(r - \max\left(\mathbb{P}(\mathbf{x}), \mathbb{P}(\mathbf{x}')\right))^{\frac{1}{t}}, \end{aligned} \tag{3.1}$$

with $t \in (0, 1]$, and $a, r \in \mathbb{R}_+^*$ such that $r > \max_{\mathbf{x}} \mathbb{P}(\mathbf{x})$, note that the Rips graph is a special case of the σ-Rips graph when σ_δ is a constant function. Next, we compare the persistence between the Rips and the σ-Rips graphs.

3.3.2 Persistence on superlevel sets

We refer to [48] for an introduction to topological persistence and its applications. Let $\mathcal{X}$ be a Riemannian manifold, and $\mathbb{P}$ be a K-Lipschitz-continuous probability density function with respect to the Hausdorff measure for a real constant $K \geq 0$.

A *persistence module* is a sequence of vector spaces $\mathbf{X} = (X_\alpha)_{\alpha \in \overline{\mathbb{R}}}$ where $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ together with linear maps $X_\beta \to X_\alpha$ whenever $\alpha \leq \beta$ (setting $X_\alpha \to X_\alpha$ as the identity). The *persistence diagram* $D\mathbf{X}$ of this persistence module is then a multi-set of points in $\overline{\mathbb{R}}^2$ containing the diagonal $\Delta = \{(x, x) \mid x \in \overline{\mathbb{R}}\}$ and points (i, j) corresponding to the lifespan of some generator appearing at time i and dying at time $j < i$ (here, according to the way our spaces are connected, we see alpha starting from $+\infty$ and at $-\infty$). The multiplicity of a point of $D_0\mathbf{X}$ is $+\infty$ for the points of Δ, and a finite alternating sums of ranks of composed linear maps (see for example [133] for more details).

Persistence is often used with homology, and we refer the reader to [57] for more details. Here we will only use the 0-dimensional homology, which detects connected components. More precisely, if X is a topological space or a graph, $H_0(X)$ will be the vector space spanned by the (path) connected components of X. Moreover, if $g\colon X \to Y$ is a continuous map between spaces or a graph homomorphism between graphs, it induces a natural linear application $f_*\colon H_0(X) \to H_0(Y)$. Let us now look at meaningful examples for the rest of the chapter. The first is the 0-dimensional persistence homology induced by $\mathbb{P}$. More precisely, for $\alpha \in \mathbb{R}$, we set $F^\alpha = \mathbb{P}^{-1}([\alpha, +\infty])$. If $\alpha \leq \beta$ are two reals, then there is an inclusion $F^\beta \subseteq F^\alpha$, and this induces linear maps $H_0(F^\beta) \to H_0(F^\alpha)$. We will denote by $D_0\mathbb{P}$ the corresponding persistence diagram (the 0 is there to remind us that we are working with the 0-dimensional homology). Another persistence diagram we will consider is the one induced by the Rips graph.

Definition 3.3 (upper-star Rips filtration)**.** Given a finite point cloud $\mathcal{S}_\mathbf{x}$ from a metric space $(\mathcal{X}, d)$ with the probability density function $\mathbb{P}$, a real value $\delta \in \mathbb{R}^+$, The *upper-star Rips filtration* of $\mathbb{P}$, denoted $\mathcal{R}_\delta(\mathcal{S}_\mathbf{x}, \mathbb{P})$, is the nested family of subgraphs of the Rips graph $R_\delta(\mathcal{S}_\mathbf{x})$ defined as $\mathcal{R}_\delta(\mathcal{S}_\mathbf{x}, \mathbb{P}) = (R_\delta(\mathcal{S}_\mathbf{x} \cap \mathbb{P}^{-1}([\alpha, +\infty]))_{\alpha \in \overline{\mathbb{R}}}$.

Such a nested family of graphs give rise to, after applying the 0-dimensional homology, a persistent module $\mathbb{R}_\delta(\mathcal{S}_\mathbf{x}, \mathbb{P})$ and a persistence diagram $D_0\mathbb{R}_\delta(\mathcal{S}_\mathbf{x}, \mathbb{P})$. A word of caution might be necessary. Technical difficulties can occur when there are infinitely many points (counted with multiplicity) away from the diagonal in a persistent diagram. Fortunately, with an upper-star Rips filtration, it is not an issue

if the point cloud $\mathcal{S}_{\mathbf{x}}$ is finite since only a finite number of changes appear in the nested family of graphs. When considering the persistence diagram $D_0\mathbb{P}$ induced by a function $\mathbb{P}$ which is continuous, one can for example require that $\mathbb{P}$ has finitely many critical points to avoid problems. The bottleneck distance is an effective and natural proximity measure for these objects to compare persistence diagrams:

Definition 3.4 (bottleneck distance)**.** Given two multi-subsets A_1, A_2 of $\bar{\mathbb{R}}^2$, a multi-bijection γ between A_1 and A_2 is a bijection:

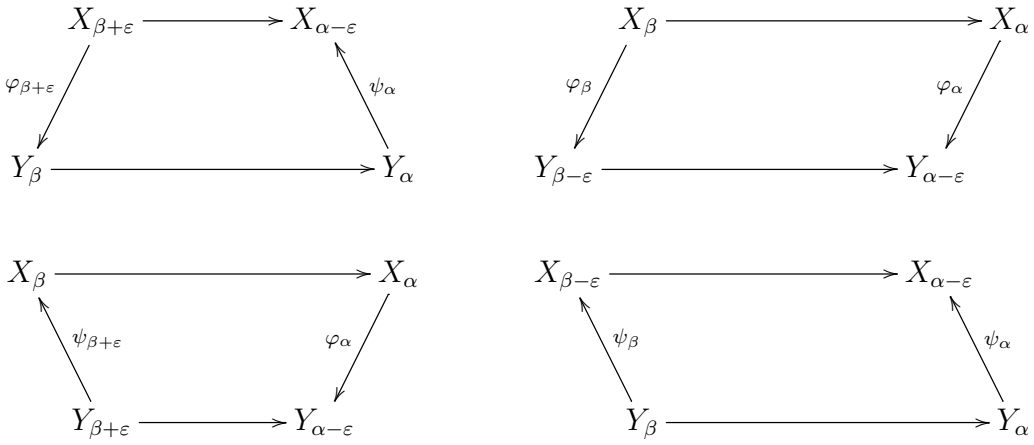
$$\gamma : \bigcup_{p\in|A_1|} \amalg_{i=1}^{\mu(p)} p \to \bigcup_{q\in|A_2|} \amalg_{i=1}^{\mu(q)} q,$$

where, for $i \in \{1, 2\}$, $|A_i|$ denotes the support of A_i and $\mu(p)$ denotes the multiplicities of point $p \in |A_i|$. The bottleneck distance $d_B^\infty(A_1, A_2)$ between A_1 and A_2 is the quantity:

$$d_B^\infty(A_1, A_2) = \min_{\gamma} \max_{p\in A_1} \|p - \gamma(p)\|_\infty$$

To show that two persistence diagrams are close to one another with respect to the bottleneck distance, one can use the following notion introduced in [32].

Definition 3.5 (ε-interleaved)**.** Let $\mathbf{X} = (X_\alpha)_{\alpha\in\overline{\mathbb{R}}}$ and $\mathbf{Y} = (Y_\alpha)_{\alpha\in\overline{\mathbb{R}}}$ be two persistence modules and let $D_0\mathbf{X}$ and $D_0\mathbf{Y}$ be there associated persistence diagrams. We say that $\mathbf{X}$ and $\mathbf{Y}$ are *strongly ε-interleaved* if there exists two families of linear application $\{\varphi_\alpha \colon X_\alpha \to Y_{\alpha-\varepsilon}\}_{\alpha\in\mathbb{R}}$ and $\{\psi_\alpha \colon Y_\alpha \to X_{\alpha-\varepsilon}\}_{\alpha\in\mathbb{R}}$, such that for all $\alpha, \beta \in \mathbb{R}$, if $\alpha \leq \beta$, then the following diagrams, whenever they make sens, are commutative:



The idea behind these diagrams is that every component appearing (resp. dying) in $\mathbf{X}$ at some time α must appear (resp. die) in $\mathbf{Y}$ within $[\alpha-\varepsilon, \alpha+\varepsilon]$, and vice-versa. The following lemma highlights how important this notion is.

Lemma 3.1. *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be two persistence modules such that* $D_0\mathbf{X}$ *and* $D_0\mathbf{Y}$ *have only finitely many points away from the diagonal, and let* $\varepsilon > 0$*. If* $\mathbf{X}$ *and* $\mathbf{Y}$ *are strongly* ε*-interleaved, then* $D_0\mathbf{X}$ *and* $D_0\mathbf{Y}$ *are at a distance at most* ε *with respect to the bottleneck distance.*

Proof. This is a direct consequence of [32, Theorem 4.4] where the result is proven for every homological dimension. □

For example, in [33, Theorem 5], it is proven that given the density function $\mathbb{P}$ on a point cloud $\mathcal{S}_\mathbf{x}$ with sufficient sampling density, the persistence diagram $D_0\mathbb{R}_\delta(\mathcal{S}_\mathbf{x}, \mathbb{P})$ built upon the Rips graph $R_\delta(\mathcal{S}_\mathbf{x})$ with an appropriate δ is a good approximation of $D_0\mathbb{P}$ the persistence diagram of $\mathbb{P}$. Consequently, $D_0\mathbb{R}_\delta(\mathcal{S}_\mathbf{x}, \mathbb{P})$ encodes the *0th homology groups* of the underlying space of $\mathcal{S}_\mathbf{x}$, this is a crucial ingredient in the proof of the theoretical guarantees of `ToMATo`.

3.3.3 Persistence of Rips graph and σ-Rips graph

`ToMATo` is a clustering method that uses the hill climbing algorithm on the Rips graph along with a merging rule on the Rips graph's persistence, it comes with theoretical guarantees under the *manifold assumption*, we would like to derive similar guarantees for our proposed approach. As previously mentioned, `ToMATo` guarantees are deduced from a careful comparison (with respect to the bottleneck distance) between $D_0\mathbb{P}$ and $D_0\mathbb{R}_\delta(\mathcal{S}_\mathbf{x}, \mathbb{P})$, one way to get similar guarantees for our procedure is to control the bottleneck distance between $D_0\mathbb{R}_\delta(\mathcal{S}_\mathbf{x}, \mathbb{P})$ and the persistence diagram of the σ-Rips graph. To do so, we need to introduce the following.

Lemma 3.2. *Given a finite point cloud* $\mathcal{S}_\mathbf{x}$ *from a metric space* $(\mathcal{X}, d)$*, for all* σ*-Rips graphs* $R^\sigma_\delta(\mathcal{S}_\mathbf{x})$ *with real-valued function* $\sigma_\delta \colon \mathcal{X}^2 \to \mathbb{R}^*_+$ *and parameters* δ*, there exist a non-metric space* $(\mathcal{X}, \hat{d})$ *such that* $R^\sigma_\delta(\mathcal{S}_\mathbf{x})$ *is the Rips graph* $R_1(\mathcal{S}_\mathbf{x})$*.*

Proof. The proof is trivial by the following definition of $\hat{d}$:

$$\begin{aligned} \hat{d} \colon \mathcal{X} \times \mathcal{X} &\longrightarrow \mathbb{R}^+ \\ (\mathbf{x}, \mathbf{x}') &\longrightarrow \frac{d(\mathbf{x}, \mathbf{x}')}{\sigma_\delta(\mathbf{x}, \mathbf{x}')}. \end{aligned}$$

□

Given a graph $R(\mathcal{S}_\mathbf{x})$, we will denote by $\mathcal{P}_R(\mathbf{x}_i, \mathbf{x}_j)$ the set of all paths in $R(\mathcal{S}_\mathbf{x})$ from the vertex $\mathbf{x}_i$ to the vertex $\mathbf{x}_j$, where a path p is a sequence of vertices of $R(\mathcal{S}_\mathbf{x})$ where two consecutive vertices of p are adjacent in $R(\mathcal{S}_\mathbf{x})$.

Definition 3.6 (appearance level)**.** Given a finite point cloud $\mathcal{S}_{\mathbf{x}} = \{\mathbf{x}_i\}_{i=1}^n$ from a metric space $(\mathcal{X}, d)$ with the probability density function $\mathbb{P}$, and a graph $R(\mathcal{S}_{\mathbf{x}})$. We define $f_R(\mathbf{x}_i, \mathbf{x}_j)$ as the level α at which the first connection between the vertices $\mathbf{x}_i, \mathbf{x}_j$ appearances in the upper-star Rips filtration $\mathcal{R}_\delta(\mathcal{S}_{\mathbf{x}}, \mathbb{P})$, $\forall i, j \in \{1, \ldots, n\}, i \neq j$:

$$f_R(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \max_{p \in \mathcal{P}_R(\mathbf{x}_i, \mathbf{x}_j)} \min_{\mathbf{x} \in p} \mathbb{P}(\mathbf{x}) & \text{if } \mathcal{P}_R(\mathbf{x}_i, \mathbf{x}_j) \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

Now we can for example prove that the persistent module $\mathbb{R}_\delta(\mathcal{S}_{\mathbf{x}}, \mathbb{P})$ of the Rips graph $R_\delta(\mathcal{S}_{\mathbf{x}})$ over the metric space $(\mathcal{X}, d)$ and the persistent module $\mathbb{R}_1(\mathcal{S}_{\mathbf{x}}, \mathbb{P})$ of the σ-Rips graph $R_\delta^\sigma(\mathcal{S}_{\mathbf{x}})$ over the non-metric space $(\mathcal{X}, \hat{d})$ of Lemma 3.2 are ϵ-interleaved.

Theorem 3.3. *Given a finite point cloud $\mathcal{S}_{\mathbf{x}} = \{\mathbf{x}_i\}_{i=1}^n$ from a metric space $(\mathcal{X}, d)$ with the probability density function $\mathbb{P}$. Let the Rips graph $R_{\delta'}(\mathcal{S}_{\mathbf{x}})$ with parameter δ', and the σ-Rips graph $R_\delta^\sigma(\mathcal{S}_{\mathbf{x}})$ with a threshold function σ of parameter set δ. Let denote by $\mathbf{R} = \mathbb{R}_{\delta'}(\mathcal{S}_{\mathbf{x}}, \mathbb{P})$, $\mathbf{R}^\sigma = \mathbb{R}_1(\mathcal{S}_{\mathbf{x}}, \mathbb{P})$, $R = R_{\delta'}(\mathcal{S}_{\mathbf{x}})$, and $R^\sigma = R_\delta^\sigma(\mathcal{S}_{\mathbf{x}})$. For $\alpha \in \mathbb{R}$, we set:*

$$R_\alpha = R_{\delta'}\left(\mathcal{S}_{\mathbf{x}} \cap \mathbb{P}^{-1}([\alpha, +\infty])\right) \qquad \text{and} \qquad R_\alpha^\sigma = R_\delta^\sigma\left(\mathcal{S}_{\mathbf{x}} \cap \mathbb{P}^{-1}([\alpha, +\infty])\right).$$

Concretely, for all $i, j \in \{1, \ldots, n\}$ such that $i \neq j$, $\mathbf{x}_i$ and $\mathbf{x}_j$ appear in $\mathbf{R}$ and $\mathbf{R}^\sigma$ at $\alpha = \mathbb{P}(\mathbf{x}_i)$ and $\mathbb{P}(\mathbf{x}_j)$, respectively. They are then merged in $\mathbf{R}$, resp. $\mathbf{R}^\sigma$, at $\alpha = f_R(\mathbf{x}_i, \mathbf{x}_j)$, resp. $\alpha = f_{R^\sigma}(\mathbf{x}_i, \mathbf{x}_j)$. Finally, by choosing ε as $\varepsilon = \max_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{S}_{\mathbf{x}} i \neq j} |f_R(\mathbf{x}_i, \mathbf{x}_j) - f_{R^\sigma}(\mathbf{x}_i, \mathbf{x}_j)|$, we have that $\mathbf{R}$ and $\mathbf{R}^\sigma$ are strongly ε-interleaved.

Proof. For $\alpha \in \mathbb{R}$, let $\mathcal{C}_1, \ldots, \mathcal{C}_k$ be the connected components of R_α. For every $i \in \{1, \ldots, k\}$, and each vertices $\mathbf{x}, \mathbf{x}' \in \mathcal{C}_i$, we have that $f_{R_\alpha}(\mathbf{x}, \mathbf{x}') \geq \alpha$ and thus, by definition of ε, $f_{R_\alpha^\sigma}(\mathbf{x}, \mathbf{x}') \geq \alpha - \varepsilon$. Hence $\mathcal{C}_i$ is contained in a connected component of $R_{\alpha-\varepsilon}^\sigma$. This gives a linear map:

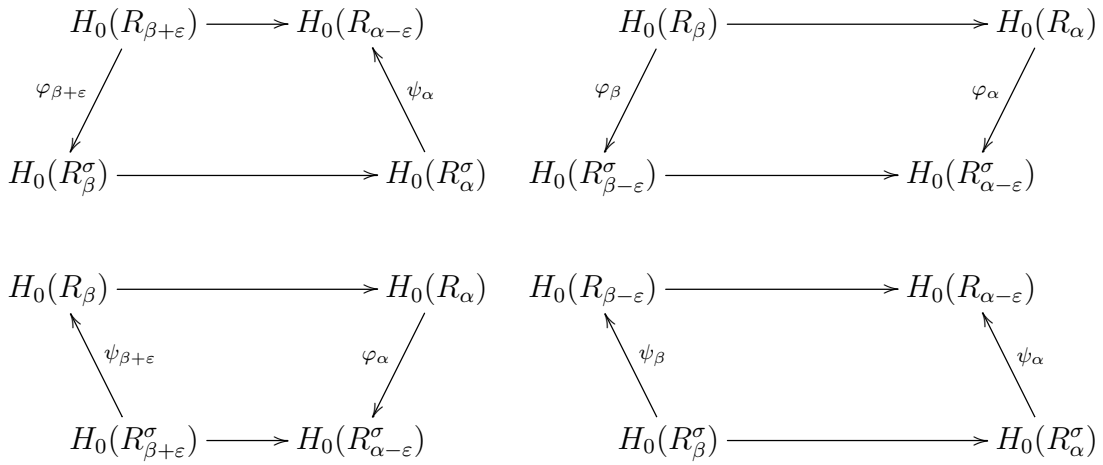
$$\varphi_\alpha \colon H_0(R_\alpha) \to H_0(R_{\alpha-\varepsilon}^\sigma).$$

By a similar argument, we get a linear map:

$$\psi_\alpha \colon H_0(R_\alpha^\sigma) \to H_0(R_{\alpha-\varepsilon}^\sigma).$$

By construction, the following diagrams are commutative (inclusions on connected

components induce the linear maps involved).



Consequently, $\mathbf{R}$ and $\mathbf{R}^\sigma$ are strongly ε-interleaved. □

In other words, when switching from the metric distance d to the non-metric distance $\hat{d}$, if the dendrogram induced by the upper star Rips graph is mostly the same during the persistence process, our procedure enjoys the same theoretical guarantees as the `ToMATo` method.

3.4 Learning with proper topological regions

In the following, we will first clarify our proposed notion of *proper topological regions* in the context of our framework, and then derive our meta-approach for pool-based active learning strategies.

3.4.1 Proper topological regions

The proper topological regions of a sample set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is the σ-Rips graph $R^\sigma_{\delta^*}(\mathcal{S}_\mathbf{x})$, with an appropriate threshold function σ, resulting from solving the following optimization problem:

$$\begin{aligned} &\underset{R \in \mathcal{H}_{R^\sigma}}{\text{minimize}} \quad \mathit{PuritySize}(R^\sigma_\delta(\mathcal{S}_\mathbf{x})) = \left[\frac{k}{n} + \frac{1}{n}\sum_{i=1}^{n} 1_{C^{\mathbb{P}}_R(\mathbf{x}_i) \neq y_i}\right] \in [0,1] \\ &\text{subject to} \quad d_B^\infty\left(\mathbf{R}^\sigma, D_0\mathbb{P}\right) \leq \varepsilon. \end{aligned} \tag{3.2}$$

Similarly, we can define the optimal Rips graph $R_{\delta^*}(\mathcal{S}_\mathbf{x})$ that encodes the proper topological regions of $\mathcal{S}$. The *PuritySize* is an objective function that penalizes the labeling error when propagating the labels inside the connected components of the graph with $C^{\mathbb{P}}_R$, and the coverage with k, the number of connected components in the

graph R. However, there is a need to derive an unsupervised objective function in the context of our approach. To this end, we empirically investigated the following score functions, typically used to assess clustering quality.

For a given graph $R(\mathcal{S}_{\mathbf{x}})$, let $\mathcal{C}_1, \ldots, \mathcal{C}_k$ be the connected components of this graph, $\mu_i = \frac{1}{|\mathcal{C}_i|} \sum_{\mathbf{x} \in \mathcal{C}_i} \mathbf{x}, \forall i \in \{1, \ldots, k\}$, and $\mu = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$ are the mean-sample per connected component $\mathcal{C}_i$, and the mean-sample of $\mathcal{S}_{\mathbf{x}}$, respectively:

- Calinski-Harabasz score

$$S_{ch}(R(\mathcal{S}_{\mathbf{x}})) = \left[\frac{(n-k)B}{(k-1)\sum_{i=1}^{k} W_i} \right] \in [0, +\infty),$$

  with $B = \sum_{i=1}^{k} |\mathcal{C}_i| \|\mu_i - \mu\|^2$ is the inter-group variance, and $W_i = \sum_{\mathbf{x} \in \mathcal{C}_i} \|\mathbf{x} - \mu_i\|^2$ is the intra-group variance, for all $i \in \{1, \ldots, k\}$. It translates that good partitioning should maximize the average inter-group variance and minimize the average intra-group variance; some well known clustering algorithms, such as K-means [82], maximize this criterion by construction.

- Davies-Bouldin score

$$S_{db}(R(\mathcal{S}_{\mathbf{x}})) = \left[\frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left(\frac{\bar{\delta}_i + \bar{\delta}_j}{d(\mu_i, \mu_j)} \right) \right] \in (+\infty, 0],$$

  with $\bar{\delta}_i = \frac{1}{|\mathcal{C}_i|} \sum_{\mathbf{x} \in \mathcal{C}_i} d(\mathbf{x}, \mu_i)$ is the average distance of all samples in the group to their mean-sample group, for all $i \in \{1, \ldots, n\}$.

- Dunn score

$$S_d(R(\mathcal{S}_{\mathbf{x}})) = \left[\frac{\min_{i,j} d(\mu_i, \mu_j)}{\max_i \Delta_i} \right] \in [0, +\infty),$$

  with $\Delta_i = \max_{\mathbf{x}, \mathbf{x}' \in \mathcal{C}_i} d(\mathbf{x}, \mathbf{x}')$ being the diameter of group $\mathcal{C}_i$, similar to the Calinski-Harbasz score, we aim to maximize the minimum distance between the mean-sample groups and minimize the maximum group diameter.

- Silhouette score

$$S_{sil}(R(\mathcal{S}_{\mathbf{x}})) = \left[\frac{1}{k} \sum_{i=1}^{k} \frac{1}{|\mathcal{C}_i|} \sum_{\mathbf{x} \in \mathcal{C}_i} s_{il}(\mathbf{x}) \right] \in [-1, 1],$$

this score affects $s_{il}(\mathbf{x}) = \frac{b(\mathbf{x})-a(\mathbf{x})}{\max(a(\mathbf{x}),b(\mathbf{x}))}$, the silhouette coefficient to every sample, with $a(\mathbf{x}) = \frac{1}{|\mathcal{C}_R(\mathbf{x})|-1}\sum_{\mathbf{x}'\in\mathcal{C}_R(\mathbf{x}),\mathbf{x}\neq\mathbf{x}'} d(\mathbf{x},\mathbf{x}')$ being the average distance of sample $\mathbf{x}$ to his group and $b(\mathbf{x}) = \min_{j\neq i,\mathbf{x}\in\mathcal{C}_i}\frac{1}{|\mathcal{C}_j|}\sum_{\mathbf{x}'\in\mathcal{C}_j} d(\mathbf{x},\mathbf{x}')$ being the average distance of sample $\mathbf{x}$ to his neighbor group.

We observed promising empirical evidence that the Silhouette score is the best candidate among the above scores to substitute the propagation error term in the *PuritySize* cost function in (3.2). Therefore, we define the unsupervised cost function $SilSize(R,\mathcal{S}_\mathbf{x})$ that we use in solving the unsupervised setting of problem (3.2).

$$SilSize_\alpha(R(\mathcal{S}_\mathbf{x})) = S_{Sil}(R(\mathcal{S}_\mathbf{x})) - \alpha\frac{k}{n},\ \text{with}\ \alpha\in\mathbb{R}^+. \tag{3.3}$$

Given our choice of the threshold function σ_δ in (3.1), and the new objective function (3.3), our problem (3.2) becomes:

$$\begin{aligned}&\underset{\delta=(a,r,t)\in dom(\sigma)}{\text{maximize}} \quad SilSize_\alpha(R^\sigma_\delta(\mathcal{S}_\mathbf{x})) = \left[S_{Sil}(R^\sigma_\delta(\mathcal{S}_\mathbf{x})) - \alpha\frac{k}{n}\right] \in \left[-1-\alpha, 1-\frac{\alpha}{n}\right]\\ &\text{subject to}\quad d_B^\infty\left(\mathbf{R}^\sigma, D_0\mathbb{P}\right) \leq \varepsilon.\end{aligned} \tag{3.4}$$

Theorem 3.3 tells us that one way to ensure the bottleneck constraint in (3.4) is to apply the same post-processing phase used in the `ToMATo` algorithm on the σ-Rips graph. It consists of applying a merging rule along the hill-climbing method on the graph with $\mathbb{P}$. This merging rule compares the *topological persistence* of groups to an additional merging parameter $\tau\in[0,\max_{\mathbf{x}\in\mathcal{S}_\mathbf{x}}\mathbb{P}(\mathbf{x})]$ [52]. We will refer to this procedure as $\texttt{ToMATo}_\tau(R(\mathcal{S}_\mathbf{x}),\mathbb{P})$, which returns the partition function C_R that encodes the proper topological regions of $\mathcal{S}_\mathbf{x}$. We describe in Algorithm 2 a two-stage black-box optimization scheme to uncover the σ-graph parameters δ, and the merging parameter τ, solution to our optimization problem 3.4, for the proper topological regions of $\mathcal{S}$.

Algorithm 2 Optimization procedure for PTR

Require: $\mathcal{S}_{\mathbf{x}} := \{\mathbf{x}_i\}_{i=1}^n$, $d : \mathcal{X} \times \mathcal{X} \to [0, \infty)$, s the step size for the linear search, and l the number of trials for the optimization strategy.

1: Set $\alpha = s$.
2: Compute density estimator $\widetilde{\mathbb{P}}$ with (3.5).
3: Optimize the unconstrained problem (3.4) for l trials, and return $\tilde{\delta} = (\tilde{a}, \tilde{r}, \tilde{t})$.
4: Build the σ-Rips graph $R_{\tilde{\delta}}^{\sigma}(\mathcal{S}_{\mathbf{x}})$.
5: **while** $R_{\tilde{\delta}}^{\sigma}(\mathcal{S}_{\mathbf{x}})$ is not a *degenerate graph* **do**
6: Save the current graph parameters $\tilde{\delta}$.
7: Update $\alpha \longleftarrow \alpha + s$.
8: Optimize the unconstrained problem (3.4) for l trials.
9: Build the σ-Rips graph $R_{\tilde{\delta}}^{\sigma}(\mathcal{S}_{\mathbf{x}})$, with new parameters $\tilde{\delta}$.
10: **end while**
11: Update $\alpha \longleftarrow \alpha - s$.
12: Optimize problem (3.4) with $\texttt{ToMATo}_{\tau}(R_{\tilde{\delta}}^{\sigma}(\mathcal{S}_{\mathbf{x}}), \widetilde{\mathbb{P}})$ for l trials, on merging parameter τ, given the fixed parameters $\tilde{\delta}$ of line 6, and return $\tilde{\tau}$.
13: **Output :** parameters $\tilde{\delta} = (\tilde{a}, \tilde{r}, \tilde{t})$, and $\tilde{\tau}$.

Remark 1. *Note that the trade-off parameter α in (3.3) is key in uncovering the proper topological regions of the sample set $\mathcal{S}$. Higher α values penalize the coverage compactness, resulting in partitions with a high degree of agglomeration, meaning fewer groups with large cardinals, which is typically the case in clustering methods, for example. However, an additional way to control the labeling propagation error term of the PuritySize objective in an unsupervised setting is to control the size distribution of groups in the resulting partition. Put differently, minimizing the group size is an excellent proxy for minimizing the labeling propagation error. Inversely, lower α values results in highly fragmented partitions with many groups with small cardinals, such setting is sub-optimal to our purpose of increasing the training sample size, additionally, the Silhouette is a score metric used to evaluate clustering quality and defined for non-singleton groups. Using it solely as an objective function in (3.4) often converges to graphs with a single non-singleton connected component. Accordingly, there should be a middle ground for α values that we find with a line search method and a stopping criterion on the size distribution of groups. We will clarify other technical details and choices in the next subsection.*

Algorithm 3 Pool-based active learning on PTR

Require: $\mathcal{S}_{\mathbf{x}} := \{\mathbf{x}_i\}_{i=1}^{n}$, oracle $\mathcal{O}$, budget $\mathcal{B}$ and $d : \mathcal{X} \times \mathcal{X} \to [0, \infty)$, graph parameters δ and merging parameter τ, active learner agent $h_{st}(\mathcal{S}_{\mathbf{x}}, \mathcal{B})$ with an underlying pool-based strategy st, and r the active training rounds.

Compute density estimator $\widetilde{\mathbb{P}}$ with (3.5).
Build the σ-Rips graph $R_\delta^\sigma(\mathcal{S}_{\mathbf{x}})$.
Apply $\texttt{ToMATo}_\tau(R_\delta^\sigma(\mathcal{S}_{\mathbf{x}}), \widetilde{\mathbb{P}})$ to get $C_{R_\delta^\sigma}$, the partition function that encodes the proper topological regions of $\mathcal{S}_{\mathbf{x}}$.
$\mathcal{S}_0 = \{\mathcal{B}$ largest connected components (ccs) of $R_\delta^\sigma(\mathcal{S}_{\mathbf{x}})$ labeled with $C_{R_\delta^\sigma}^{\widetilde{\mathbb{P}}}\}$.
for $i = 0, \ldots, r$ **do**
    Train the learning agent $h_{st}(\mathcal{S}_i, \mathcal{B})$.
    Ask a set $\mathcal{S}_r$ from h_{st} of size $\mathcal{B}$.
    $S_{i+1} = S_i \cup \{$label ccs containing $\mathcal{S}_r$ with $C_{R_\delta^\sigma}^{\widetilde{\mathbb{P}}}\}$.
    if extra budget **then**
        $S_{i+1} = S_{i+1} \cup \{$label largest ccs of $R_\delta^\sigma(\mathcal{S}_{\mathbf{x}})$ with $C_{R_\delta^\sigma}^{\widetilde{\mathbb{P}}}\}$.
    end if
end for
Output : trained agent h_{st}

Algorithm 3 presents the main contribution of this work, our meta-approach for training pool-based active learning strategies on proper topological regions of a sample set $\mathcal{S}_{\mathbf{x}}$. In addition to standard inputs in active learning methods such as the unlabeled sample set $\mathcal{S}_{\mathbf{x}}$, the oracle $\mathcal{O}$, the budget $\mathcal{B}$, and the number of rounds r. It also takes as input the metric distance d, a given pool-based active learning method $h_{st}(\mathcal{S}_{\mathbf{x}}, \mathcal{B})$, the parameters δ and τ tuned by Algorithm 2. The proper topological regions of $\mathcal{S}_{\mathbf{x}}$ are encoded in the partition function $C_{R_\delta^\sigma}$, we typically obtain a much higher number of regions than the number of classes c, or what we expect to have with clustering methods. Initially, we aim to maximize the initial training set size by labeling the largest regions. Then the underlying active learner strategy consumes the rest of the regions during the rounds of active training. We might receive queries during these rounds which contain samples from the same region; in that case, we use the extra budget to label the largest regions. After r rounds, we return the trained estimator of the strategy. Note that we also can have cases where all the regions are consumed before the end of rounds of active training. Next, we shall discuss practical considerations and technical details related to implementing this approach.

3.4.2 Practical considerations

We consider for d, the Euclidean distance overall experiments. Furthermore, we use l-nearest neighbors for the estimation of the distance matrix D as follows: Let $D = (d_{i,j}) \in \mathbb{R}^{n \times n}$ be a sparse distance matrix, with only ℓ non zero values in each row:

$$d_{i,j} = \begin{cases} d(\mathbf{x}_i, \mathbf{x}_j) & \text{if } \mathbf{x}_j \text{ is one of the } \ell\text{-nearest neighbors of } \mathbf{x}_i, \\ 0 & \text{otherwise.} \end{cases}$$

The density estimation $\widetilde{\mathbb{P}}$: $\mathcal{S}_{\mathbf{x}} \to \mathbb{R}^+$ is calculated as follows:

$$\widetilde{\mathbb{P}}(\mathbf{x}_i) = \left(\frac{1}{l} \sum_{j=1}^{n} m_{i,j}^2 \right)^{-1/2}, \qquad \text{for all } i \in \{1, \ldots, n\}. \tag{3.5}$$

We consider $\ell = 2000$, for all datasets of greater size. We choose the Tree-structured Parzen Estimator (TPE) [16] for the optimization procedure of Algorithm 2, with a number of trials $l = 1000$, and a step size s of 0.1 for the line search procedure.

3.5 Empirical results

In this section, we will describe the corpora and methodology.

Datasets We conduct experiments on benchmark datasets for classification problems also often used in active learning: coil-20 [121], isolet [50], protein [60], banknote, pendigits, nursery, and adult [100]. Table 3.1 presents statistics of the datasets.

dataset	n	p	c	imbalance	test
protein	756	77	8	0.70	324
banknote	943	4	2	0.83	405
coil-20	1008	1024	20	1.00	432
isolet	4366	617	26	0.99	1872
pendigits	7694	16	10	0.92	3298
nursery	9070	8	4	0.09	3888
adult	34.2k	14	2	0.31	14.7k

Table 3.1: dataset statistics, n is the size of training set, test is the size of test set and imbalance corresponds to class imbalance ratio. The third column corresponds to the dimension of the feature space $\mathbb{R}^p$.

Baseline methods For the cold-start experiments, we consider the following approaches from the literature:

- K-Means clustering
  The K-Means algorithm [82] partitions a collection of examples into K clusters by minimizing the sum of squared distances to the cluster centers. It has been used for active learning in [129], to generate the initial training set by labeling the closest sample to each centroid. Another variation proposed in [70] adds artificial samples from the centroids, named *model examples*, to the initial training set. This approach is named K-Means+ME and leads to an initial training set twice as large as the one created using K-Means.

- K-Medoids clustering
  The K-Medoids algorithm [71] is very similar to K-Means except that it uses the actual samples for centers, namely medoids, as the center of each cluster. These medoids are then used to form the initial training set in active learning.

- Agglomerative Hierarchical Clustering (AHC)
  Agglomerative hierarchical clustering [116] is a bottom-up clustering approach that builds a hierarchy of clusters. Initially, each sample represents a singleton cluster. Then the algorithm recursively merges the closest clusters using a *linkage function* until until only one cluster is left. This process is usually presented in a dendogram, where each level refers to a merge in the algorithm. AHC has been used for active learning in [40] by pruning the dendogram at a certain level to obtain clusters, then similar to K-Means, selecting the closest samples to the centroid clusters to generate the initial training set.

- Furthest-First-Traversal (FFT)
  The farthest-first traversal of a sample set is a sequence of a selected sample, where the first sample in the sequence is selected arbitrarily, and each successive sample in the sequence is located the furthest away from the set of previously-selected samples. The resulting sequence is then used as the initial training set for active learning. The FFT algorithm has been used for active learning in [13].

- Affinity Propagation Clustering (APC)
  Affinity propagation is a clustering algorithm designed to find *exemplars* of the sample set which are representative of clusters. It simultaneously considers all

the sample set as possible *exemplars* and uses the message-passing procedure to converge to a relevant set of *exemplars*. The *exemplars* found are then used as an initial training set for active learning [63].

For the active learning experiments, and following results from [106], we only consider the comparison against the random labeling strategy, as it outperforms many recent strategies in active learning in small budget scenarios.

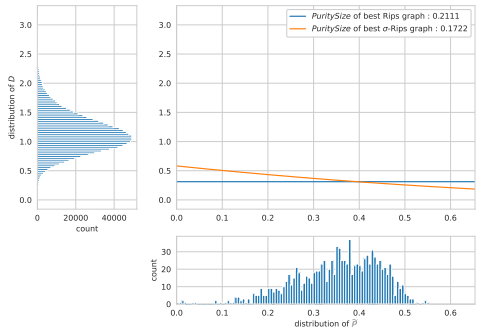
In all our experiments, we use the random forest classifier [61] as base estimator for the different strategies with default parameters, we also consider several budgets $\mathcal{B} \in [3, 10, 20]$, and 20 stratified random splits, with 70% of the data in the training set and 30% in the test set.

3.5.1 Rips graph vs σ-Rips graph

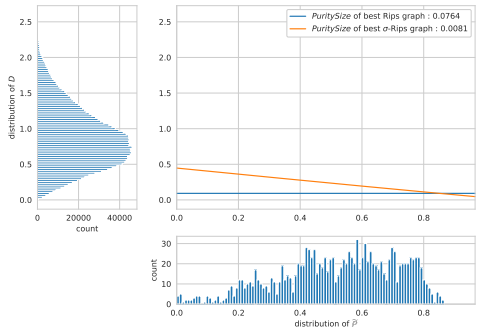
To validate our hypothesis of a density-aware threshold (3.1) for class similarity, and to motivate our generalization of the Rips graph to express this notion, we present a comparison study in Figure 3.1 between the Rips and the σ-Rips graphs overall considered datasets. Each plot presents the threshold of the best Rips graph and σ-Rips graph in minimizing the *PuritySize* cost function on all the datasets with the same practical considerations of Subsection 3.4.2. Note that the Rips graph's threshold is a constant presented as a horizontal line in the plots. We also show two additional side plots. The x-axis plot shows the distribution of the density estimation $\widetilde{\mathbb{P}}$ in the dataset. In contrast, the y-axis plot shows the distribution of the Euclidean distances in the dataset's distance matrix D. We are interested in comparing the threshold rules within these intervals because, from Definition 3.1 of the Rips graph, and Definition 3.2 of the σ-Rips graph, threshold values greater than the maximum distance will result in a clique graph.

For all the datasets, we observe that the optimal threshold rule's values found in the hypothesis class of the σ-Rips graph with our proposed threshold function σ_δ in (3.1) are negatively correlated to the estimation density $\widetilde{\mathbb{P}}$. We also observe that the best σ-Rips graph achieves better *PuritySize* scores overall datasets, except for coil-20 and nursery datasets, where it has similar scores to the best Rips graph found. Particularly for the coil-20 dataset where the optimal threshold seems to be a constant, we notice that, in this case, both graphs converge to the same threshold function, this shows that our proposed threshold function (3.1) can effectively approximate constant threshold functions. These findings confirm our hypothesis that class similarity is a

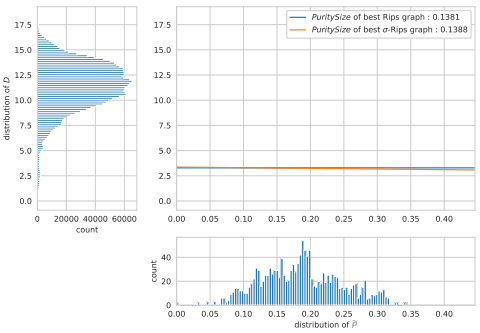
density-aware measure. It also supports our choice of σ_δ in (3.1) as an appropriate threshold function to generalize the Rips graph.



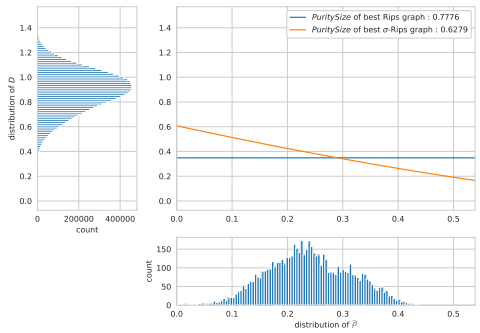
(a) protein



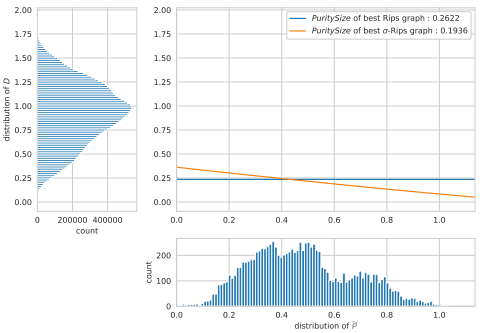
(b) banknote



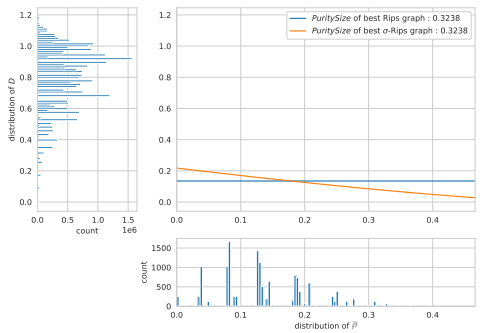
(c) coil-20



(d) isolet



(e) pendigits



(f) nursery

Figure 3.1: Comparison study between Rips graph and σ-Rips graph overall datasets, the *PuritySize* score is reported for each minimizer.

3.5.2 Cold-start results

Table 3.2 presents the cold-start results of our meta-approach for pool-based active learning strategies, denoted TPR(USRG) where USRG is refering to the unsupervised setting of our optimization procedure for solving Problem (3.4) using the σ-Rips graph, next to the Random Selection (RS), and other considered baseline methods overall the datasets. In all cases, TPR(USRG) provides significantly better results than the random selection approach except for the imbalanced dataset nursery with the largest budget. This shows that using the largest proper topological regions found by Algorithm 2 as an initial training set (Line 4 of Algorithm 3) provides a better starting point for pool-based active learning strategies than random selection. Furthermore, our meta-approach shows very competitive results overall datasets, when compared to the baseline methods that are solely designed to tackle the cold-start problem in active learning.

Table 3.2: Average balanced classification accuracy (in %) and standard deviation of random forest classifier with the initial training set obtained from different methods over 20 stratified random splits for different budgets $\mathcal{B}$. $^{\uparrow}/^{\downarrow}$ indicate statistically significantly better/worse performance than Random Selection RS, according to a Wilcoxon rank sum test ($p < 0.05$) [117].

Dataset	$\mathcal{B}$	RS	K-Means	K-Means+ME	K-Medoids	AHC	FFT	APC	TPR(USRG)
protein	3	16.91 ± 3.98	$21.21 \pm 1.80^{\uparrow}$	$\mathbf{23.87 \pm 2.16}^{\uparrow}$	$21.24 \pm 4.36^{\uparrow}$	$22.74 \pm 2.54^{\uparrow}$	17.35 ± 3.33	16.72 ± 3.32	$22.07 \pm 6.20^{\uparrow}$
	10	28.20 ± 3.21	$30.61 \pm 4.56^{\uparrow}$	$31.40 \pm 4.53^{\uparrow}$	29.28 ± 4.35	$31.58 \pm 3.67^{\uparrow}$	$21.78 \pm 3.80^{\downarrow}$	28.79 ± 3.43	$\mathbf{40.49 \pm 3.92}^{\uparrow}$
	20	36.42 ± 3.76	$42.07 \pm 3.90^{\uparrow}$	$45.53 \pm 2.52^{\uparrow}$	39.16 ± 4.84	$43.43 \pm 3.40^{\uparrow}$	$26.11 \pm 3.37^{\downarrow}$	39.18 ± 3.70	$\mathbf{53.99 \pm 3.39}^{\uparrow}$
banknote	3	55.48 ± 7.22	$73.98 \pm 4.59^{\uparrow}$	$\mathbf{84.32 \pm 5.58}^{\uparrow}$	$62.48 \pm 3.32^{\uparrow}$	$63.69 \pm 4.49^{\uparrow}$	$58.22 \pm 7.30^{\uparrow}$	58.74 ± 7.96	$70.21 \pm 14.73^{\uparrow}$
	10	79.88 ± 9.91	$85.23 \pm 5.68^{\uparrow}$	$86.80 \pm 4.75^{\uparrow}$	$87.59 \pm 3.33^{\uparrow}$	$85.59 \pm 5.08^{\uparrow}$	$70.58 \pm 5.30^{\downarrow}$	82.40 ± 6.92	$\mathbf{88.68 \pm 4.43}^{\uparrow}$
	20	87.58 ± 2.91	$90.74 \pm 2.40^{\uparrow}$	$92.43 \pm 2.00^{\uparrow}$	$92.34 \pm 2.44^{\uparrow}$	$92.58 \pm 2.86^{\uparrow}$	$71.89 \pm 7.16^{\downarrow}$	$90.92 \pm 3.18^{\uparrow}$	$\mathbf{93.88 \pm 3.44}^{\uparrow}$
coil-20	3	12.35 ± 2.62	$\mathbf{14.99 \pm 0.00}^{\uparrow}$	$\mathbf{14.99 \pm 0.00}^{\uparrow}$	$\mathbf{14.99 \pm .00}^{\uparrow}$	$\mathbf{14.99 \pm 0.00}^{\uparrow}$	$10.83 \pm 1.98^{\downarrow}$	11.70 ± 2.26	13.59 ± 1.66
	10	28.97 ± 5.66	$36.68 \pm 4.24^{\uparrow}$	$38.15 \pm 2.70^{\uparrow}$	$32.85 \pm 5.14^{\uparrow}$	$36.00 \pm 3.66^{\uparrow}$	$18.56 \pm 3.38^{\downarrow}$	27.21 ± 4.80	$\mathbf{44.18 \pm 2.43}^{\uparrow}$
	20	42.02 ± 5.83	$56.69 \pm 3.74^{\uparrow}$	$62.99 \pm 2.78^{\uparrow}$	42.30 ± 3.47	$58.10 \pm 4.07^{\uparrow}$	$25.61 \pm 2.46^{\downarrow}$	41.40 ± 4.74	$\mathbf{71.05 \pm 3.78}^{\uparrow}$
isolet	3	07.64 ± 1.54	$08.69 \pm 0.85^{\uparrow}$	$09.68 \pm 0.63^{\uparrow}$	07.81 ± 1.63	$09.05 \pm 1.86^{\uparrow}$	$09.24 \pm 0.96^{\uparrow}$	07.51 ± 1.84	$\mathbf{10.82 \pm 1.05}^{\uparrow}$
	10	13.78 ± 2.97	$22.30 \pm 1.55^{\uparrow}$	$\mathbf{27.61 \pm 1.59}^{\uparrow}$	$07.06 \pm 1.85^{\downarrow}$	$23.26 \pm 1.76^{\uparrow}$	$16.47 \pm 1.67^{\uparrow}$	15.41 ± 3.18	$27.53 \pm 2.84^{\uparrow}$
	20	19.20 ± 2.69	$27.88 \pm 2.46^{\uparrow}$	$\mathbf{40.37 \pm 3.22}^{\uparrow}$	$10.73 \pm 1.98^{\downarrow}$	$28.19 \pm 2.14^{\uparrow}$	18.79 ± 2.42	$21.14 \pm 3.13^{\uparrow}$	$38.63 \pm 3.17^{\uparrow}$
pendigits	3	21.45 ± 3.49	21.27 ± 1.93	22.53 ± 2.09	$26.57 \pm 2.55^{\uparrow}$	$19.42 \pm 1.79^{\downarrow}$	$17.26 \pm 3.72^{\downarrow}$	$17.81 \pm 4.86^{\downarrow}$	$\mathbf{29.89 \pm 0.04}^{\uparrow}$
	10	37.35 ± 7.19	$62.54 \pm 3.46^{\uparrow}$	$65.63 \pm 2.25^{\uparrow}$	$53.90 \pm 5.21^{\uparrow}$	$61.39 \pm 1.86^{\uparrow}$	$27.17 \pm 4.87^{\downarrow}$	38.33 ± 8.20	$\mathbf{80.11 \pm 2.60}^{\uparrow}$
	20	54.25 ± 5.86	$72.26 \pm 2.72^{\uparrow}$	$75.80 \pm 2.31^{\uparrow}$	$64.00 \pm 3.64^{\uparrow}$	$72.34 \pm 2.45^{\uparrow}$	$34.78 \pm 4.49^{\downarrow}$	52.24 ± 5.93	$\mathbf{87.68 \pm 4.10}^{\uparrow}$
nursery	3	30.71 ± 4.00	29.20 ± 5.19	30.23 ± 6.50	$25.04 \pm 0.15^{\downarrow}$	$28.33 \pm 3.86^{\downarrow}$	29.97 ± 3.24	29.99 ± 3.74	$\mathbf{35.07 \pm 5.57}^{\uparrow}$
	10	42.74 ± 7.20	44.45 ± 5.71	$\mathbf{49.26 \pm 4.01}^{\uparrow}$	$28.43 \pm 1.30^{\downarrow}$	44.92 ± 7.22	39.06 ± 3.52	45.12 ± 6.70	46.47 ± 5.98
	20	$\mathbf{55.33 \pm 2.77}$	$52.78 \pm 3.27^{\downarrow}$	54.42 ± 2.99	$32.91 \pm 1.10^{\downarrow}$	53.82 ± 2.71	$39.79 \pm 1.07^{\downarrow}$	$52.52 \pm 4.90^{\downarrow}$	54.08 ± 4.51

3.5.3 Active learning results

Lastly, we present the results of our meta-approach for pool-based active learning strategies. We consider the strategies mentioned above, namely the uncertainty sampling query strategy, the entropy sampling query strategy, and the margin sampling query strategy. When using our meta-approach presented in Algorithm 3 with USRG

as described in the previous subsection. Compared to RS, the vanilla training approach in pool-based active learning, when using random selection to initiate the active learning strategies. We also show SSRG to illustrate the difference of performance, when we minimize the supervised setting for TPR (Problem(3.2)), compared to the unsupervised setting (Problem(3.4)). We present one figure per each dataset, and each figure is constituted of six error bars plots, referring to the average balanced accuracy and standard deviation over the splits, for all the active learning rounds, plus the initial round. The plots are indexed to show a specific budget per row, and a specific active learning strategy per column.

We can see from the results that all the considered pool-based active learning strategies clearly benefit from our approach in comparison to the vanilla setting. The only instance where we don't see such advantage is for nursery dataset. The reason being that nursery has a high class imbalance ratio. We choose in Algorithm 3 to prioritize the gain in the training sample size, regardless of class discovery, or class ratio, which may empathizes the class imbalance in such case. This shows that we need to introduce other sampling criterion of TPR in Algorithm 3, than simply selecting the largest ones, when training with highly class-imbalanced datasets.

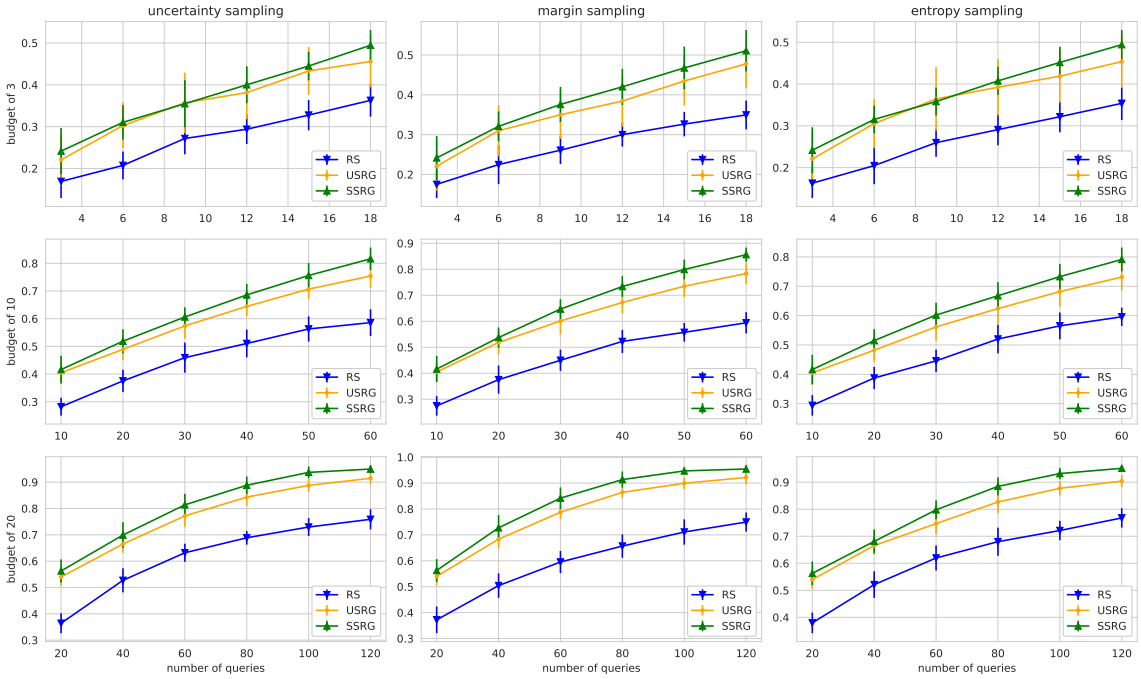


Figure 3.2: Average balanced classification accuracy and standard deviation of different pool-based active learning strategies and budgets on protein dataset, using random forest estimator over 20 stratified random splits.

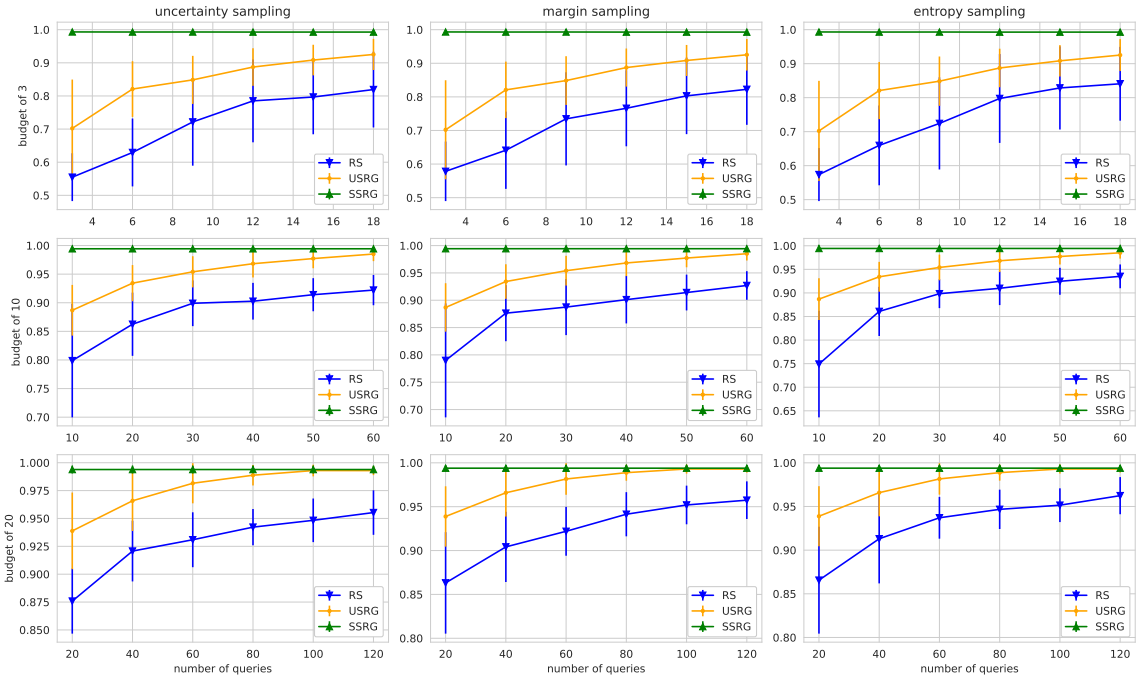


Figure 3.3: Average balanced classification accuracy and standard deviation of different pool-based active learning strategies and budgets on banknote dataset, using random forest estimator over 20 stratified random splits.

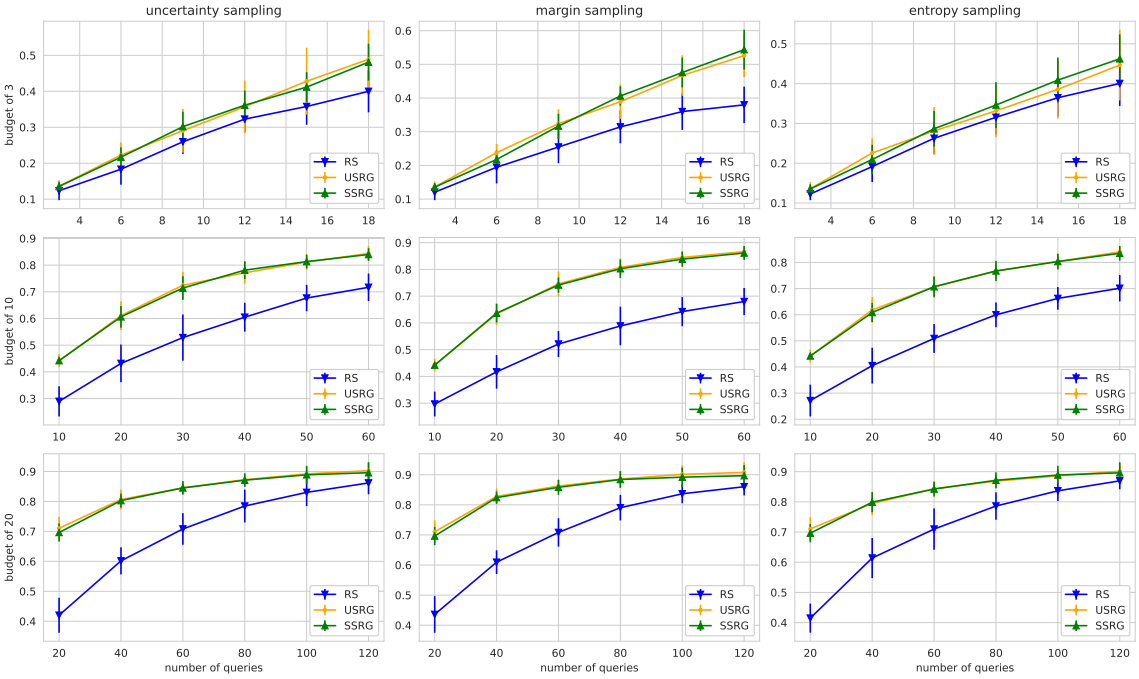


Figure 3.4: Average balanced classification accuracy and standard deviation of different pool-based active learning strategies and budgets on coil-20 dataset, using random forest estimator over 20 stratified random splits.

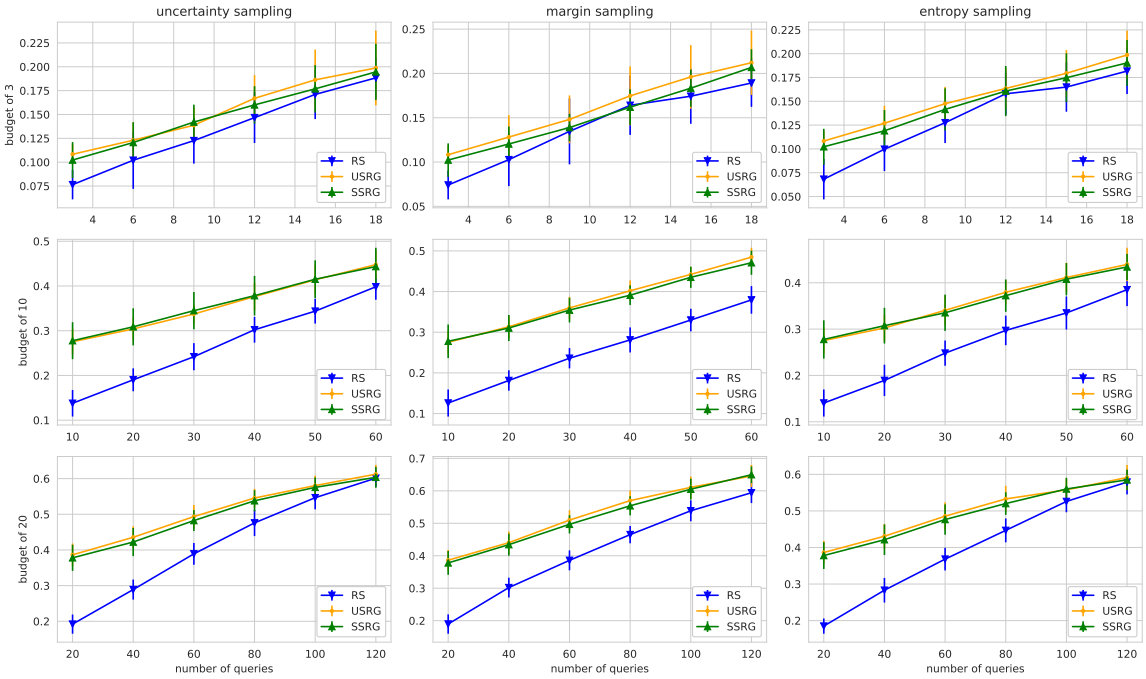


Figure 3.5: Average balanced classification accuracy and standard deviation of different pool-based active learning strategies and budgets on isolet dataset, using random forest estimator over 20 stratified random splits.

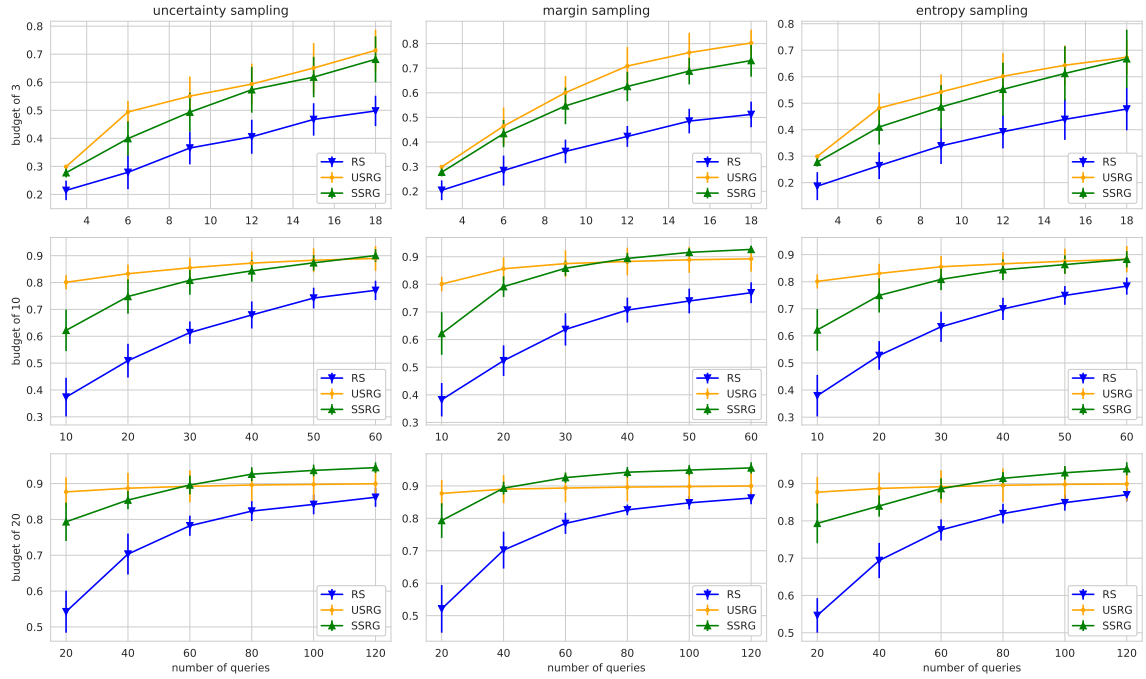


Figure 3.6: Average balanced classification accuracy and standard deviation of different pool-based active learning strategies and budgets on pendigits dataset, using random forest estimator over 20 stratified random splits.

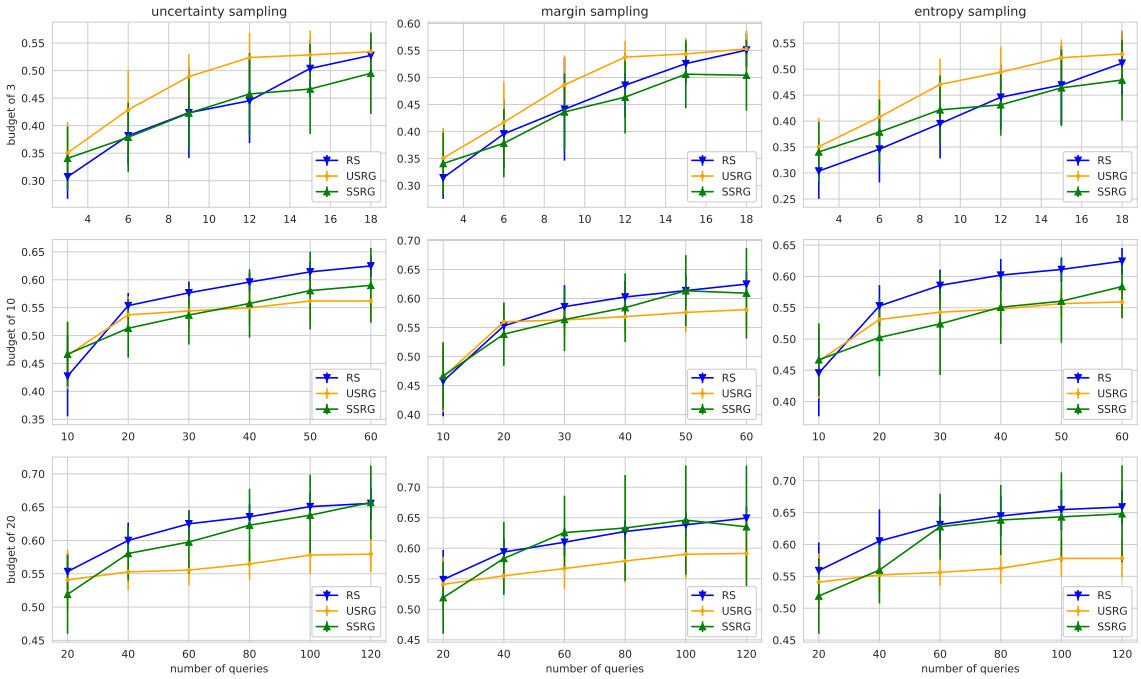


Figure 3.7: Average balanced classification accuracy and standard deviation of different pool-based active learning strategies and budgets on nursery dataset, using random forest estimator over 20 stratified random splits.

3.6 Conclusion

In this chapter, we proposed a data driven meta-approach for pool-based active learning strategies for multi-class classification problems. Our approach is based on an introduced notion of Topological Proper Regions (TPR) of a given sample set, we showed the theoretical foundations of this notion, and derived a black-box optimization problem to uncover the TPR. Our empirical study validates our meta-approach on different benchmarks, in low budget scenario, for various pool-based active learning strategies. Challenging open questions are left, a theoretical analysis that guarantee good performance in active learning, such as, generalization bounds. The use of semi-supervised approaches to conclude the analysis with a model dependent approach, by having a regularization term derived from the TPR.

Chapter 4

Deep Learning for Rapid Automation of Transmission Electron Microscopy Analysis

Deep learning is revolutionizing many areas of science and technology, including the analysis of data obtained by Transmission Electron Microscopy (TEM). This chapter presents our practical contributions aimed at automating the analysis of phase and orientation mapping from scanning diffraction data obtained during TEM analysis. More precisely, we aim to derive a DL approach capable of accurately predicting a crystal's phase and orientation, given its diffraction diagram. These contributions aim to achieve real-time orientation and phase determination maps during the acquisition experiments. This chapter is based on the following paper [HDR+22, HDAL22].

4.1 Introduction

Transmission Electron Microscopy (TEM) has expanded the type of information obtained on nanocrystalline microstructures, such as phase and orientation [27]. Orientation microscopy is a technique that enables spatially resolved measurements of crystal phases and orientations in a sample and reconstructs the microstructure from this information. Using a scanning mode and acquiring on each scanning data point a full diffraction diagram, orientation mapping TEM experiments, alternatively called ACOM (Automated Crystal Orientation Mapping) or SPED (Scanning Precession Electron Diffraction) are producing large collections of datasets, which are often impossible to process manually. As a result, extensive research [96, 125] has proposed semi-automated approaches to analyze these datasets. These deterministic methods rely on classical computer vision techniques (e.g., Hough transform, Fourier filtering,

segmentation, and cross-correlation for similarity measure), which typically require manual hyperparameter tunning and a computation cost for each experiment. Deep neural networks (DNNs) have shown superior performance compared to classical computer vision techniques in most benchmark tasks. This led to the emergence of fully automated approaches [132, 7] and tools [119, 90] for various TEM tasks. In the context of orientation microscopy, ML-based approaches are still falling behind traditional techniques such as template matching [96] or Kikuchi technique [73] when it comes to generalization performance to unseen crystal orientations and phases during training. This is due mainly to the limited experimental data about the studied phenomena for training the models. It is a realistic and practical constraint, especially for narrow-domain applications where real data is not widely available. Some successful attempts have been made to use unsupervised learning techniques to gain more insight into the data [85, 104], but clustering information does not directly solve the orientation microscopy problem.

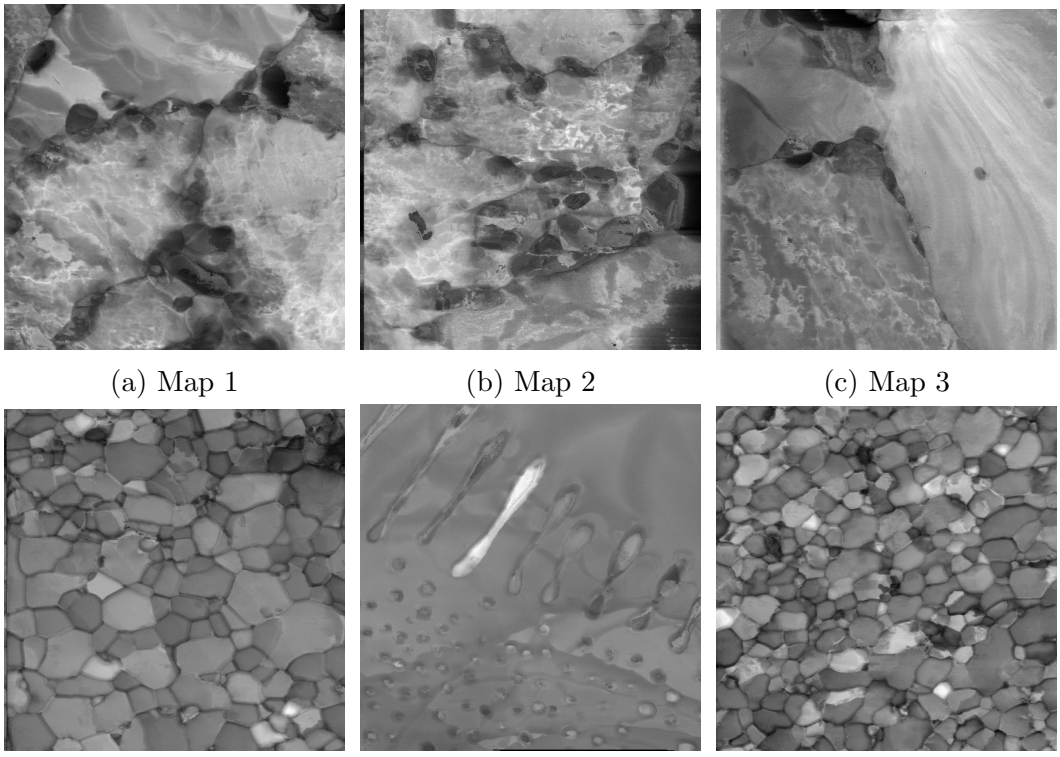
Early Deep learning breakthroughs were primarily in the computer vision domain, mainly due to the increased availability of new big data benchmarks and organized competitions such as ImageNet [66] since 2009. This dynamic resulted in many sophisticated models for image classification [59, 105, 64, 36]. There is a clear potential for automated image analysis tools using state- of-the-art machine learning techniques for phase and orientation determination to complement the existing relatively slow, complex, and hyperparameter-dependent approaches. To this end, we investigate multi-task DL solutions with the purpose of boosting the existing slow phase and orientation determination techniques by replacing them with DL models for fast real-time prediction during data acquisition but with lower generalization accuracy to be used as a less accurate but real-time analysis for TEM experiments.

4.2 Experimental

We conduct our experiments using two batches of maps from two different studies. The first batch is constituted of three different experimental Micrographs presented in (a),(b), and (c) of Figure 4.1. Each collected map consists of 500×500 diffraction images of size 144×144 pixels each. The three micrographs contain four phases, namely the alpha iron (α-Fe), the gamma iron (γ-Fe), the niobium carbide (NbC), and the cementite (Fe_3C).

The second batch of TEM data was collected in a study investigating the structural and mechanical properties of different ultrafine-grained (UFG) structures obtained

from aluminum alloys [44].The study investigated a specimen of an Al-2wt%Fe alloy after severe plastic deformation (SPD) by high-pressure torsion (HPT) at strain levels (10 turns 10T and 100 turns 100T) resulting in sub-micrometer grain refinement. These specimens correspond to the micrographs (d) and (f) presented in Figure 4.1 respectively and contain the phases α-Al and Al_6Fe. The last micrograph (e) in the figure corresponds to the as-cast material microstructure used in the study, meaning the microstructure of the studied alloy in the as-cast conditions with no further SPD.



(a) Map 1 (b) Map 2 (c) Map 3

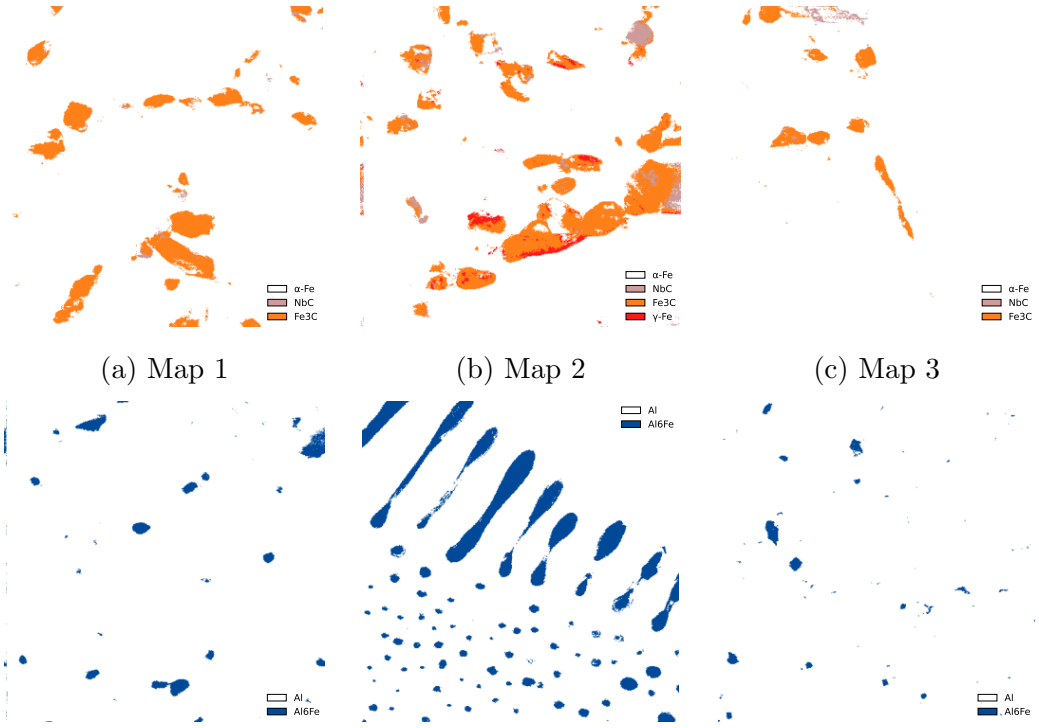
(d) Map 10T (e) Map as-cast (f) Map 100T

Figure 4.1: Collected micrographs of size 500×500.

4.2.1 Labeling strategy

Transmission Electron Microscopy (TEM) was performed using a JEOL 2100F microscope using a Stingray camera recording the phosphorus screen and equipped with automated crystal orientation mapping using the (ASTAR) package [96, 97, 113]. The ASTAR template matching package was used to analyze the phase and orientation distribution maps, which will be considered as the ground truth for the following.

Figure 4.2 shows the phase determination maps provided by ASTAR analysis of TEM data. We have the following microstructure composition:



(a) Map 1 (b) Map 2 (c) Map 3

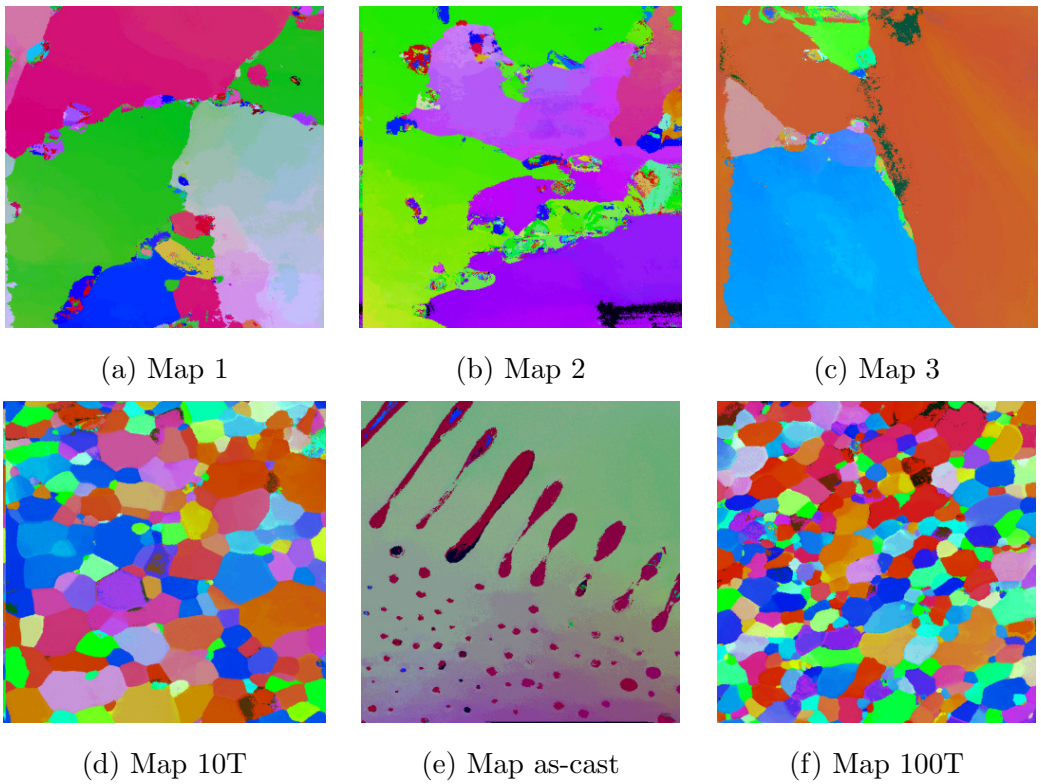
(d) Map 10T (e) Map as-cast (f) Map 100T

Figure 4.2: ASTAR phase determination maps for all considered micrographs.

- (a) Map 1 : 93.0% alpha iron (α-Fe), 0.2% niobium carbide (NbC), and 6.8% cementite (Fe_3C).
- (b) Map 2 : 86.5% alpha iron (α-Fe), 1.1% gamma iron (γ-Fe), 1.4% niobium carbide (NbC), and 11.0% cementite (Fe_3C).
- (c) Map 3 : 97.4% alpha iron (α-Fe), 0.3% niobium carbide (NbC), and 2.3% cementite (Fe_3C).
- (d) Map 10T : 98.0% aluminium (Al), and 2.0% aluminium-iron (Al_6Fe).
- (e) Map as-cast : 89.6% aluminium (Al), and 10.4% aluminium-iron (Al_6Fe).
- (f) Map 100T : 99.1% aluminium (Al), and 0.9% aluminium-iron (Al_6Fe).

The crystal orientations in the specimen are also provided from the template matching analysis. The Euler angles [54] usually denoted as (ϕ_1, Φ, ϕ_2) can fully describe these crystal orientations. A compact approach for representing these orientations is to map them to the RGB channels to create an image showing a given micrograph's orientation map. Figure 4.3 presents the resulting Euler's orientation

images of this type of mapping. For better visual quality, we apply a brightness adjustment to the resulting mappings for each orientation image.



(a) Map 1 (b) Map 2 (c) Map 3

(d) Map 10T (e) Map as-cast (f) Map 100T

Figure 4.3: ASTAR Euler's orientation determination maps for all considered micrographs.

Additionally, we have access to template diffraction diagrams where the diffraction patterns are simulated using the crystal information and the TEM experimental settings, Figure 4.4 shows a simulated diffraction pattern of alpha iron at a given orientation with Euler's angles. Each simulated template is provided as a set of coordinates with the corresponding intensities which is the radius at these spots. We draw simulated DPs using OpenCV [65], and we cover the Euler space of each crystal by simulating the fundamental zone corresponding to the class symmetry of each considered phase.

As mentioned previously, the objective of this study is to retrieve information about the crystal phase and orientation from the collected data during TEM experiments. Ultimately, we want to investigate Deep learning methods to infer Euler's orientation and phase determination maps from real diffraction images. From the descriptions provided in this section, we can already identify important challenges to take into consideration when designing such approaches:

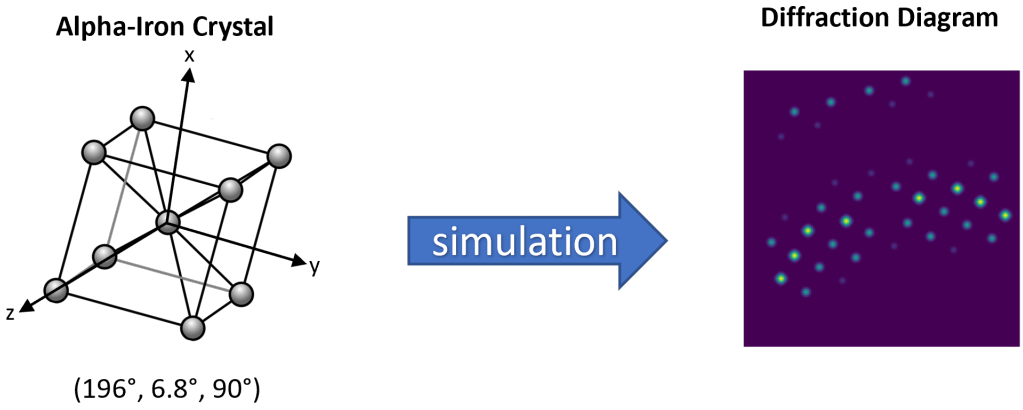


Figure 4.4: Simulation of diffraction patterns for TEM experiments.

- TEM data exhibits a high frequency of duplicates: the lack of diversification drastically reduces the relevant sample size for training DL models, directly impacting the quality of these models on analyzing new micrographs.
- TEM data is highly heterogeneous: the experimental settings in which the data is collected are often different from one TEM experiment to another for practical reasons. This adds another difficulty for DL approaches to adapt to new experimental settings when analyzing new micrographs.

4.2.2 Data preprocessing for ML

A lot of effort has been made into researching and developing new ways of preprocessing real diffraction patterns or DPs and simulated DPs to improve the prediction accuracy of ML-based approaches [132, 90]. Nowadays, computer vision models rely on the End-to-End approach with minimal feature engineering. Nonetheless, such an approach has some underlying assumptions that cannot be met in our case, namely a highly favorable signal-to-noise ratio which is usually the case in standard high-resolution RGB images in datasets such as ImageNet [66], and a tremendous amount of relevant training data to allow the underlying algorithm to model and adapt to the noise distribution. Initial End-to-End experiments for phase classification with out-of-the-shelf DL models showed that these models indeed fail to generalize to DPs from unseen micrographs, confirming our belief in the need for an appropriate preprocessing to alleviate this phenomenon. Additionally, traditional normalization techniques are not adapted to our use case. In figure 4.5, we present the mean experimental diffraction diagram for each map (calculated on all map's datapoints). These spots are present in the mean images because of the high frequency of duplicates in TEM

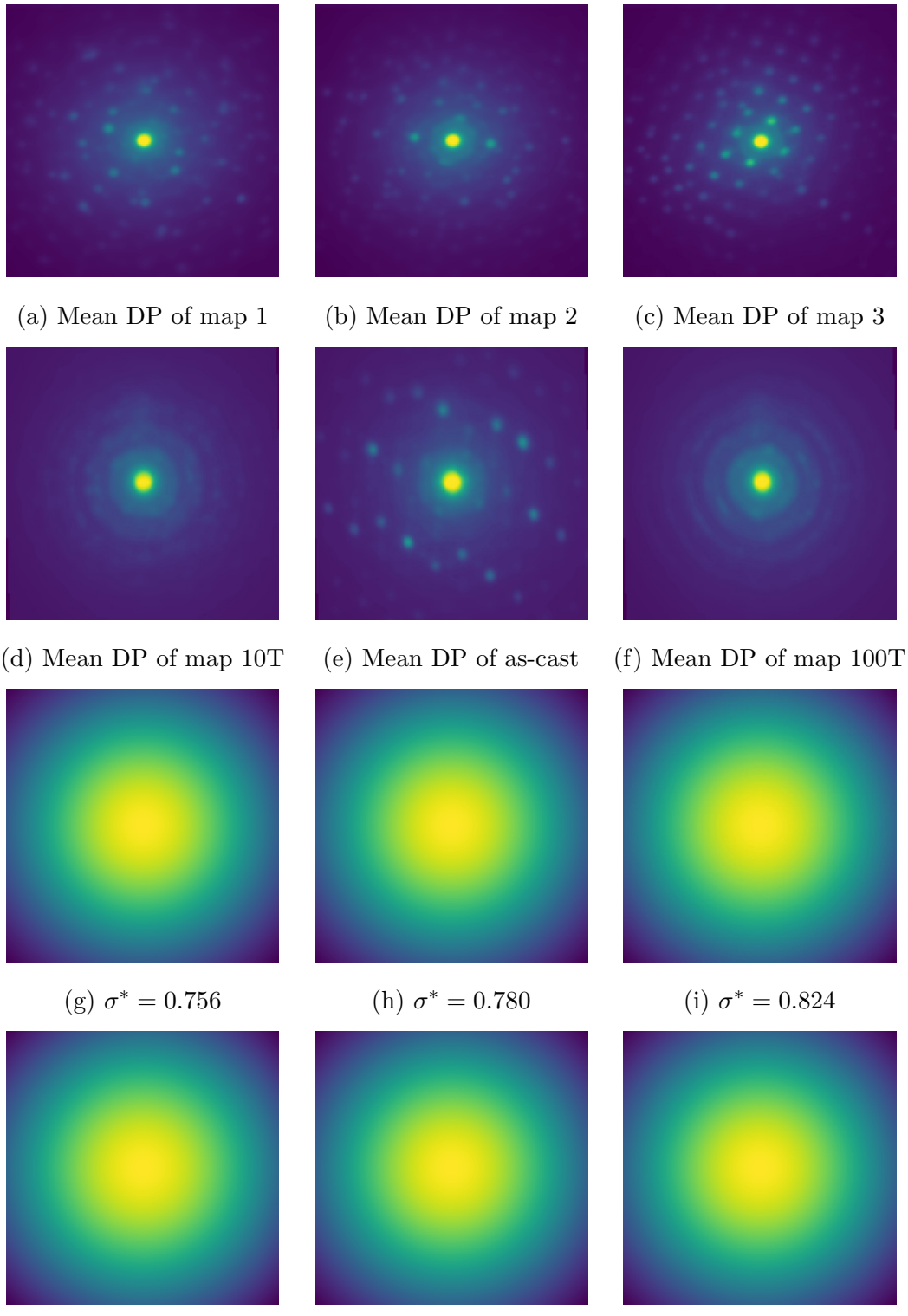
data from the major grains. It is relevant information in frequent DPs, and subtracting it in a normalization scheme would result in a signal loss. However, the mean images present statistical noise intrinsic to the TEM experiment, which should be subtracted.

$$g_\sigma(i,j) = \frac{1}{2\pi\sigma^2} e^{-(i^2+j^2)/(2\sigma^2)} \tag{4.1}$$

In our study, after standardizing the images, we propose to filter out the statistical noise by fitting a centered Gaussian filter in Eq (4.1), where i is the distance from the center in the horizontal axis, j is the distance from the center in the vertical axis, and σ is the standard deviation, to the mean images of each micrograph (map) and subtracting the resulted filter from all the DPs of that map. After subtraction, all negative points on the image are assigned a value of 0. Concretely, For each mean image m of a given micrograph, we solve:

$$\sigma^* = \arg\min_\sigma \left[\sum_{i=1}^{144} \sum_{j=1}^{144} |m(i,j) - g_\sigma(i,j)| \right]$$

Figure 4.5 shows the gaussian filters g_{σ^*} estimated from each micrograph, and Figure 4.6 shows the effect of our approach on random DPs from each micrograph before and after applying our preprocessing. On the other hand, we employ a similar preprocessing introduced in [125] to preprocess simulated DPs or templates. The simulation is parametrized by the number of spot diffractions and their size tuned to best fit the real DPs. Next we apply a blurring filter to smooth the intensities, then we extract intensity descriptors from their polar representation to obtain similar descriptors to those extracted from real diffraction diagrams.



(a) Mean DP of map 1 (b) Mean DP of map 2 (c) Mean DP of map 3

(d) Mean DP of map 10T (e) Mean DP of as-cast (f) Mean DP of map 100T

(g) $\sigma^* = 0.756$ (h) $\sigma^* = 0.780$ (i) $\sigma^* = 0.824$

(j) $\sigma^* = 0.863$ (k) $\sigma^* = 0.824$ (l) $\sigma^* = 0.809$

Figure 4.5: The two first rows coorespond to the mean diffraction diagram of DPs from each micrograph, and thier resulting Gaussian filters in the next rows.

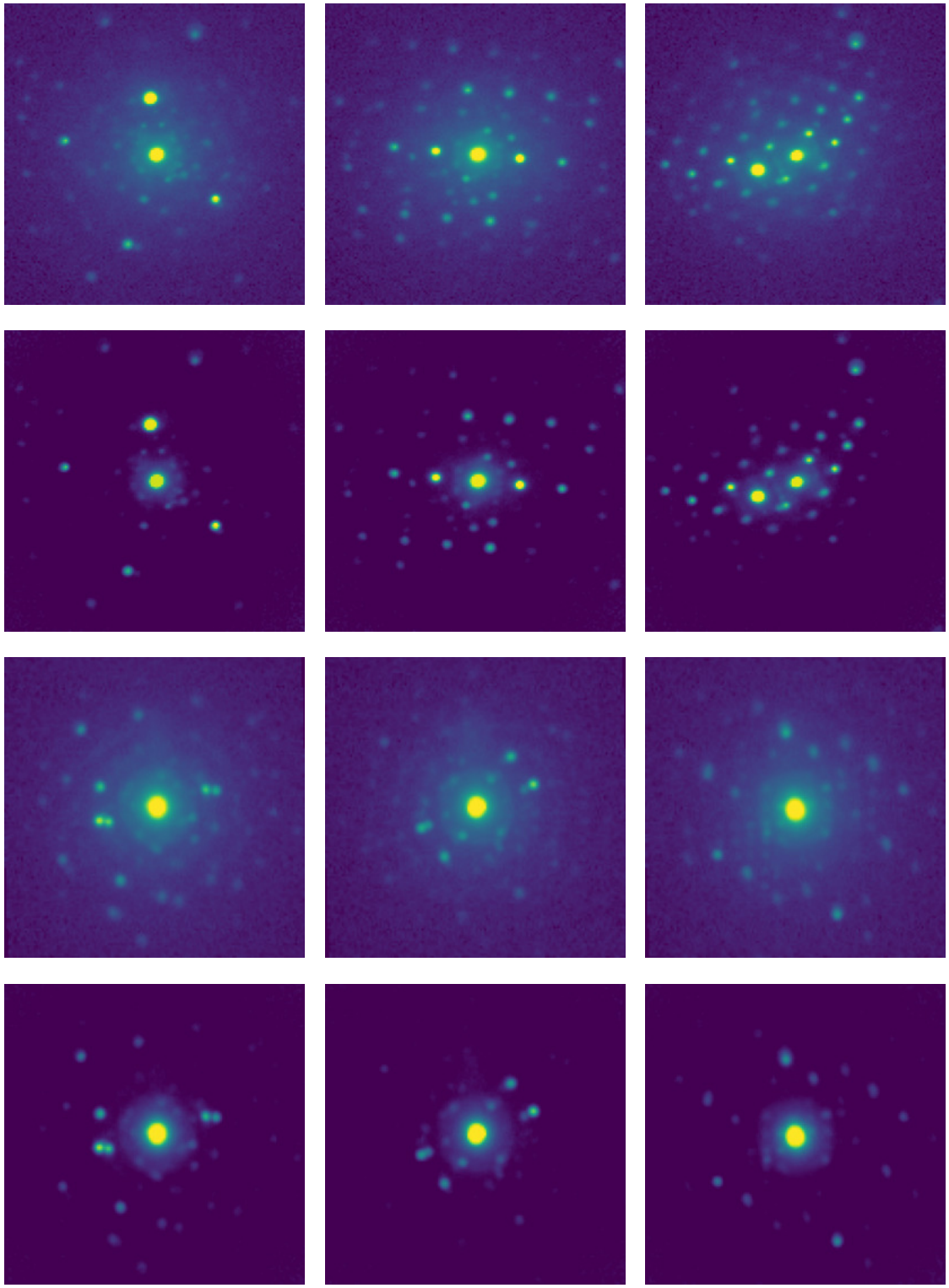


Figure 4.6: Preprocessing result on random sampled DPs. The first and second rows from the top are the raw DPs from each specimen, whereas the second and last rows are the resulting DPs after preprocessing.

Descriptors are often used for template matching techniques, and their compact size allows us to efficiently simulate the Euler's orientation space of a given phase. Once the Euler's orientation space is simulated for a given phase, the last step of preprocessing is to filter out the symmetries and keep only the descriptors of angles in the fundamental zone of each phase. This last step is dependent on the symmetry class of each phase since for different classes' symmetries, different fundamental zones need to be considered [92]). Figure 4.7 illustrates the overall preprocessing for simulated DPs and real diffraction diagrams.

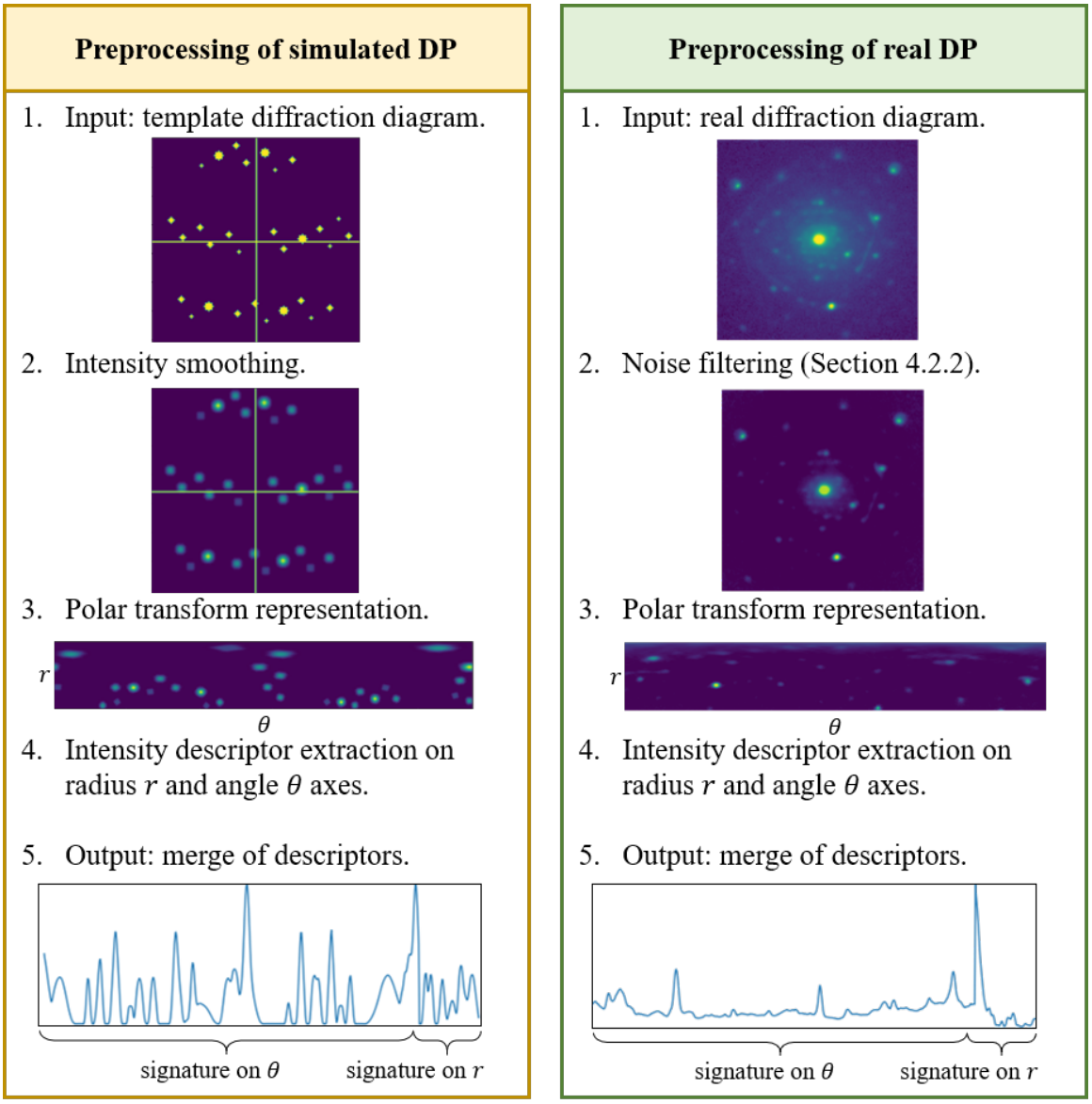


Figure 4.7: Preprocessing steps of real diffraction diagrams vs template simulated real diffraction diagrams.

4.3 Deep learning for TEM data analysis

This section considers two different research directions, the first direction was to investigate the capability of DL approaches to train with simulated TEM data, and to evaluate how well it translates to our problem of real TEM data analysis. The second direction of this work was to focus on the performance of SOTA DL models in predicting accurate unseen maps while relying directly on real TEM data for training.

4.3.1 Training on simulated TEM data

To solve our objective of analyzing TEM data using DL models, we address two known tasks in ML, a classification task to identify phases and be able to predict the phase determination map of a micrograph (see Figure 4.2) given its diffraction data, and a regression task to predict the Euler's orientation map given the diffraction data of the considered datapoint (see Figure 4.3). In the following, we will only consider the first batch of micrographs, namely, the maps 1, 2, and 3 of Figure 4.1. Additionally, we consider the desciptors extracted following the procedure detailed in the preprocessing section, of the fundamental zone of the phases alpha iron, niobium carbide, and cementite described in Table 4.1.

phase	signature size	nbr of signatures	proportion	label
α-Fe	667	898904	27.5%	1
NbC	667	898901	27.5%	2
Fe_3C	667	1465334	45.0%	3

Table 4.1: Statistics of signatures extracted from simulated DPs of three phases: the alpha iron, the niobium carbide, and the cementite.

We consider the task of classifying the considered phases with their code label described in Table 4.1, and the regression task of Euler's angle prediction, such that for the angles of all signatures we apply the following transformation in order to map the cyclical angle space $[0, 2\phi]$ to a continuous space:

$$\begin{aligned} f(\phi_1, \Phi, \phi_2) &= (\sin(\phi_1), \cos(\phi_1), \sin(\Phi), \cos(\Phi), \sin(\phi_2), \cos(\phi_2)) \\ f^{-1}(x_1, x_2, x_3, x_4, x_5, x_6) &= (\text{arctan2}(x_1, x_2), \text{arctan2}(x_3, x_4), \text{arctan2}(x_5, x_6)) \end{aligned} \tag{4.2}$$

We reserved 20% of the total simulated data as a separate test set to estimate the performance of the trained DL models using two metrics: the balanced accuracy

score for the classification task evaluation, and to evaluate the quality of the regression predictions we introduce the mean absolute angle error defined between a 1D vector A of true angles and 1D vector $\hat{A}$ of angle predictions both of size n as:

$$maae(A, \hat{A}) = \frac{1}{n} \sum_{i=1}^{n} \min(|A_i - \hat{A}_i|, 360 - |A_i - \hat{A}_i|) \tag{4.3}$$

The rest of the simulated data is used for training and validating the considered DL models, in a $80\%, 20\%$ split sizes, we consider three DL architectures in this experiment:

- MLP: a full Multi-Layer Perceptron [115], with three first dense layers of size 500 with gelu activations, and two last dense layers with sizes 100, 50 with also gelu activations.
- CNN: a Convolutional Neural Network [79], with four consecutive 1D convolutions of size 128, kernel size 7, and relu activations, followed by a 1d max pooling layer of kernel size 3, a flatten operation and a final dense layer of size 64 with relu activation.
- LSTM: Long Short-Term Memory model [62], with a first CNN layer of size 128, kernel size 7, and relu activation, followed by a 1d max pooling layer of kernel size 3, the LSTM layer with 256 units, a flatten layer output the full sequence of the LSTM layer.

The CNN and MLP models were retrieved using a Neural Architecture Search (NAS) procedure with AutoKeras [68], the LSTM model was manually handcrafted based on the architecture presented in [88]. In this experiment, we compare these models in two different settings. The first setting where a first instance of a model is trained to classify the signatures, and three other dedicated instances for the regression task for each class. The second setting is the multi-task approach (MT) where we trained a single instance of each model with two heads one for the regression task and the other for the classification task, the classification head has an output of size 3 as the number of classes, and the regression head has an output of size 6 to predict the angle transformation f of Eq (4.2) of the signature's angles. All the models were trained with a batch size of 1024 for 500 epochs, and with an early stopping on the validation loss. The nuances between these two settings of predicting phase and angle information from DP descriptors are depicted in Figure 4.8.

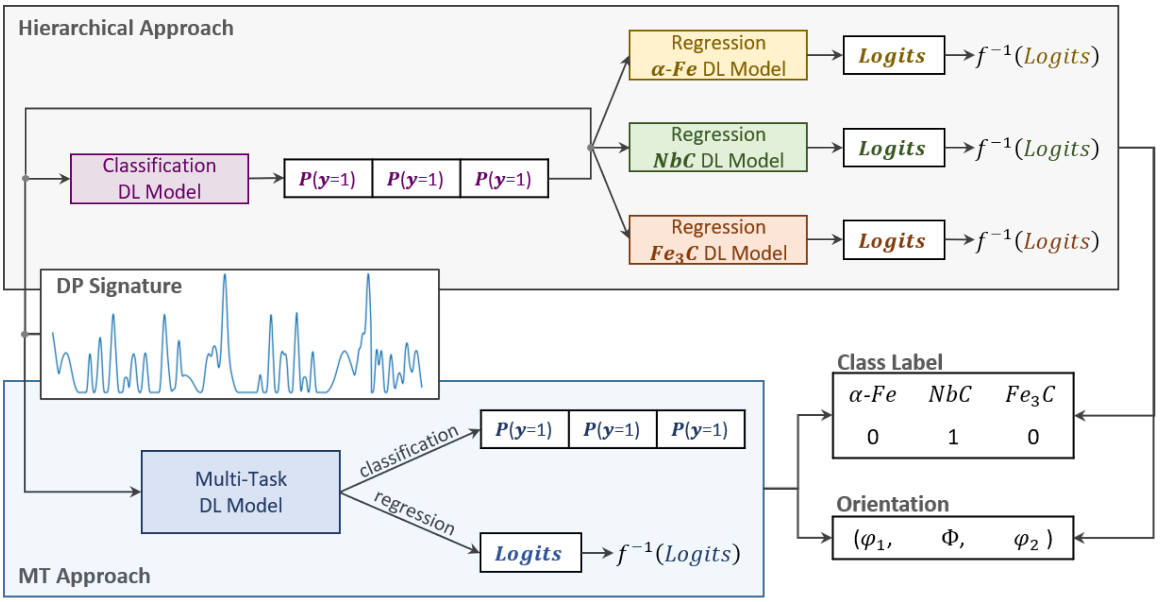


Figure 4.8: Multi-Task Learning approach compared to hierarchical learning approach for training DL models in analyzing TEM data.

From Table 4.1, we show that the classification task on the descriptors extracted from simulated DPs is a relatively simple task compared to angle prediction, even during training, all the models reached a perfect accuracy score in training and validation after a few epochs. Each model is compared in the two settings described above in Multi Task MT, or in a hierarchical way with a model instance per each class for the regression and another anstance for classification only, the MT setting has the advantage of being less complex, having a single model handling both predictions simplify greatly the approach, and as we can see in Table 4.1 it can provide

models	balanced accuracy	maae (degrees)		
		ϕ_1	Φ	ϕ_2
LSTM	100%	1.178	0.776	1.226
MT/LSTM	100%	**0.735**	**0.503**	**0.820**
MLP	100%	**1.030**	**0.140**	**0.424**
MT/MLP	100%	1.842	0.263	0.765
CNN	100%	2.933	0.525	1.595
MT/CNN	100%	**2.543**	**0.370**	**1.089**

Table 4.2: Balanced accuracy score and mean absolute angle error of the models LSTM, CNN, and MLP trained on hierarchy or in MT with simulated DPs.

the models with a better accuracy by letting a single model ingest all available

training data and take advantage of the implicit knowledge between the two tasks to enhance the accuracy of each one of them.

Next, we investigate the capabilities of these models to adapt from simulated descriptors to the descriptors extract from real DPs, transfert learning has become a common technique used in ML, from NLP to computer vision, pretrained models are widely used in various applications to enhance model performance [122]. In the next set of experiments, we will investigate if the trained models from the previous step can transfer their knowledge from simulated TEM data to successfully analyze real TEM diffraction data. For this experiment, we consider the first batch of maps, where map 2 and map 3 are used for fine-tuning the models, and the map 1 for the performance evaluation. Table 4.3 presents the results of our experiments, for each MT model we report performance results for the prediction of angles with maae metric, and phases with accuracy and balanced accuracy metrics. The results are reported with and without fine-tuning for the models MT/LSTM, MT/MLP, and MT/CNN, fine-tuned models are prefixed with FT, best performance are in bold. * refers to the best overall performance for each metric.

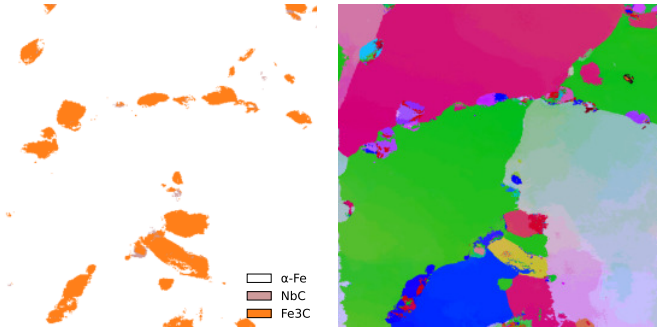
models	accuracy	balanced accuracy	maae (degrees)		
			ϕ_1	Φ	ϕ_2
MT/LSTM	**98.6**%*	34.5%	92.6	8.9	18.8
FT/MT/LSTM	98.1%	**37.8**%*	**46.0**	**6.2**	**7.5***
MT/MLP	83.7%	31.5%	101.4	6.5	17.1
FT/MT/MLP	**97.8**%	**34.4**%	**39.9***	**5.9***	**8.9**
MT/CNN	**93.9**%	35.0%	78.0	9.6	26.0
FT/MT/CNN	93.2%	**36.1**%	**40.6**	**6.37**	**8.2**

Table 4.3: Balanced accuracy score and mean absolute angle error of the pretrained models MT/LSTM, MT/CNN, and MT/MLP on map 1, each model is compared with and without fine-tuning using maps 2 and 3.

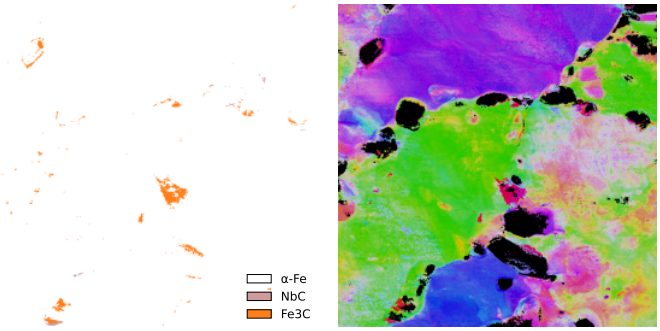
From Table 4.3, we note that the classification accuracy of all pre-trained models is high. It indicates a benefit in transferring from training on simulated DP signatures. All the models can successfully apply the learned knowledge of classifying extracted signatures from simulated diffractions to classify extracted signatures from real diffraction images correctly. The preprocessing adopted in the previous section has played a critical role in facilitating this transfer. Secondly, fine-tuning shows superior performance compared to using the pre-trained models without fine-tuning, especially for the angle's predictions. This result is expected, there are an intrinsic

difference between real TEM data, and simulations. Allowing DL models to fine-tune thier acquired knowledge on real data will naturally lead to better predictions. We also note that the balanced accuracy is relatively low, it is a consequence of the highly class imbalanced distribution in TEM micrographs (we refer the reader to Table 4.4). Lastly, fine-tuning increased the overall prediction accuracy of Euler's angles for map 1. However, the accuracy is still not comparable to the level of precision that we obtained with simulated TEM data. In Figure 4.9, we depict the predicted maps for the orientation determination and the phases determination of map 1, by all the fine-tunned models. As expected from the classification accuracy, the phase maps are well predicted by the predictor. Concerning the orientation map, the situation is of course less favorable. However, regardless of the relatively high maae error reported in the previous table by these models on map 1, the predicted Euler's orientation map highlights the main grain boundary in the micrograph of map 1, although some other boundaries are not so clearly visible. the FT/MT/LSTM model achieved the best balanced accuracy metric, and it is visually validated by its predicted phases determination map, compared to the two other maps, from models FT/MT/CNN, and FT/MT/MLP. Finally, we can notice in the predicted orientation maps the presence of noise in the predictions compared to the ground truth map provided by ASTAR. The lack of relevant real DPs data covering the Euler space has a consequence in the robustness of these estimators.

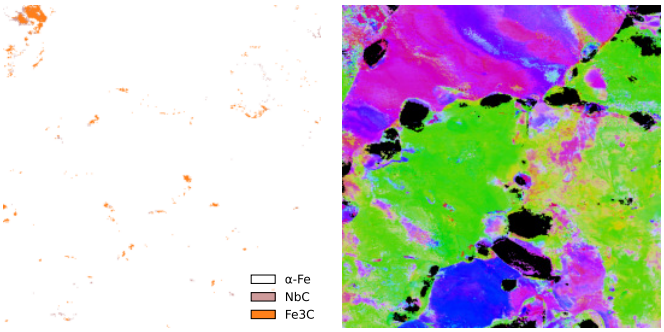
In summary, the quality of the similarity between the simulated and real descriptors heavily depends on the preprocessing steps, which has a direct implication on the final model performance. Fine-tuning, multi-task learning, and an efficient preprocessing procedure allow us to reduce this bias when training with real descriptors. We successfully analyzed TEM experimental data by combining all these tools. The qualitative and quantitative promising results show that DL approaches can successfully be used for real-time TEM data analysis, especially for phase identification.



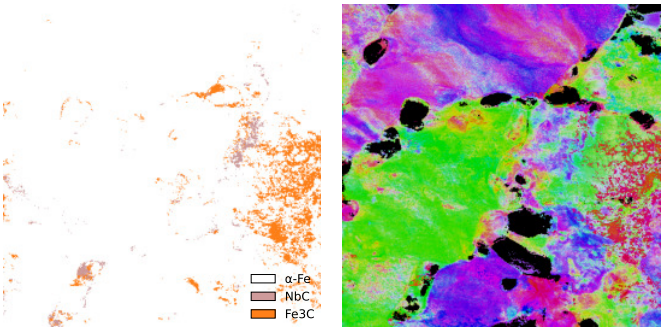
(a) Phases and Euler's orientation maps of micrograph of map 1.



(b) Phases and Euler's orientation maps predicted by FT/MT/LSTM model.



(c) Phases and Euler's orientation maps predicted by FT/MT/MLP model.



(d) Phases and Euler's orientation maps predicted by FT/MT/CNN model.

Figure 4.9: Predicted Euler's orientation maps, and phase determination maps of all fine-tunned DL models for the micrograph of map 1.

The approach presented in this section has however some limitations. First, it requires the simulation of all DPs in the fundamental zone of each considered phase. This simulation might be expensive depending on the symmetry class of the considered crystals. Our choice of using compact descriptors rather than raw 2D images reduce this cost significantly at the expense of an information loss regarding the experimental DPs. Finally, even if the orientation maps present relevant information, the high noise rate in the predictions made by this initial DL analysis of the data much less efficient than the deterministic algorithms such as template matching.

Alternatively, in the next section, we will investigate the potential of the state-of-the-art DL models in analyzing experimental TEM data, relying solely on real diffraction diagrams, and using the images as inputs.

4.3.2 Training on real TEM data

We are interested in the following in the task of analyzing TEM data by relying solely on experimental DPs. For this purpose, we will use all the diffraction data available to us from the considered micrographs of Figure 4.1 to train and evaluate SOTA DL models from computer vision, next we will first show the potential of these models and their limitations in solving the TEM data analysis task. The aggregated diffraction data is presented in Table 4.4.

phase	size	proportion	label
Al	818260	56.1%	1
Al_6Fe	35246	2.4%	2
α-Fe	557322	38.2%	3
NbC	3943	0.3%	4
Fe_3C	40390	2.8%	5
γ-Fe	2252	0.2%	6

Table 4.4: Raw TEM data statistics overall micrographs.

The majority of traditional ML experimental protocols include the filtering of sample duplicates in their feature engineering steps. Sample duplicates are, in our case, all DP images that share the same class label and the same Euler's angles. Thus, dropping sample duplicates is a crucial step in learning efficient models. Yet, when analyzing TEM data, sample duplicates may be essential for training DL models. First, as previously mentioned, TEM data exhibits a high volume of duplicates.

Therefore, discarding the duplicates will drastically reduce the data for training DL models. Additionally, for experimental TEM data, DP duplicates do not contain the same signal, depending on the detailed electron beam diffraction, the superposition of grains, each diffraction pattern is slightly different and results in a different signal in the duplicates. Hence, trained models should be able to correctly predict these DPs in experimental conditions, to consider a realistic evaluation for TEM data analysis by DL approaches.

To this end, we design the following experimental protocol to evaluate the potential of DL models to analyze TEM data. We used all available diffraction data from Table 4.4, to constitute two datasets, the first dataset contains only unique DPs, and the second dataset, contains DPs dupliactes up to 100 duplicates. To sample this dataset, we undersampled all DPs images that have more than 100 duplicates, Table 4.5 resumes the statistics of this two sampled datasets.

	unique DPs dataset		100 dup. DPs dataset	
phase	size	proportion	size	proportion
Al	13669	59.8%	254642	66.0%
Al_6Fe	2839	12.5%	24838	6.5%
α-Fe	2271	10.0%	72024	18.6%
NbC	611	2.7%	3447	0.9%
Fe_3C	3075	13.5%	28682	7.4%
γ-Fe	351	1.5%	2166	0.6%

Table 4.5: Sampled TEM data statistics from all available diffraction data.

We consider for this set of experiments the SOTA models in computer vision tasks, the ResNet model [59] and its variations, ResNet50, ResNet50V2, ResNet101, InceptionResNetV2, and ResNet152. The EfficientNet model [110] and its variations from B0 to B7. The DenseNet architecture [64] and its variations, the Xception model [36], the Inception model [109], VGG16 and VGG19 [105]. The barplot in Figure 4.10 presents the results for this SOTA models in phases classification with the unique DPs dataset in blue, and the 100 duplicate DPs dataset in orange. All models were trained on diffraction data from all the maps except for map 1 (from steel sample) and 100T (from Aluminum sample) which were kept for test set evaluation.

From the results depicted in Figure 4.10, we first notice that most of the DL models achieve a high classification accuracy in the unique DPs dataset. Conversely, we

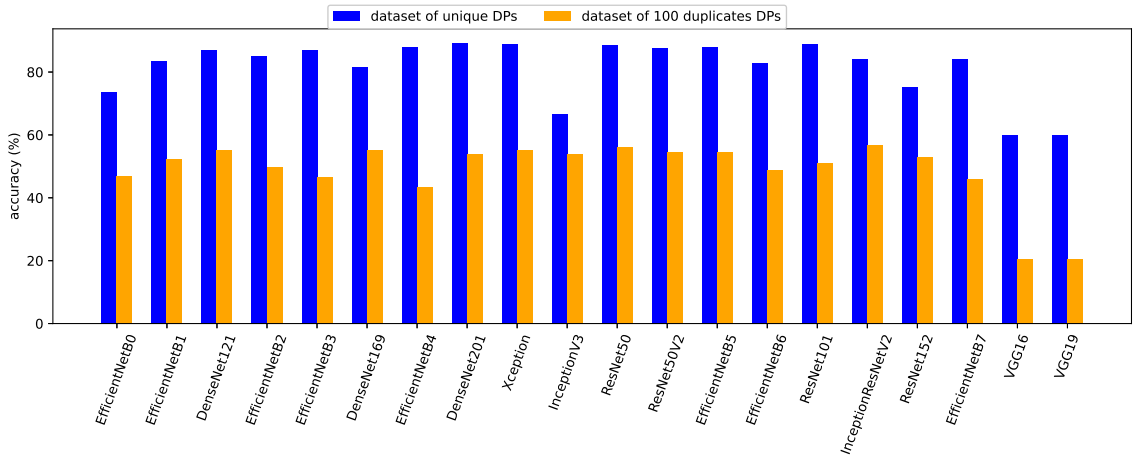
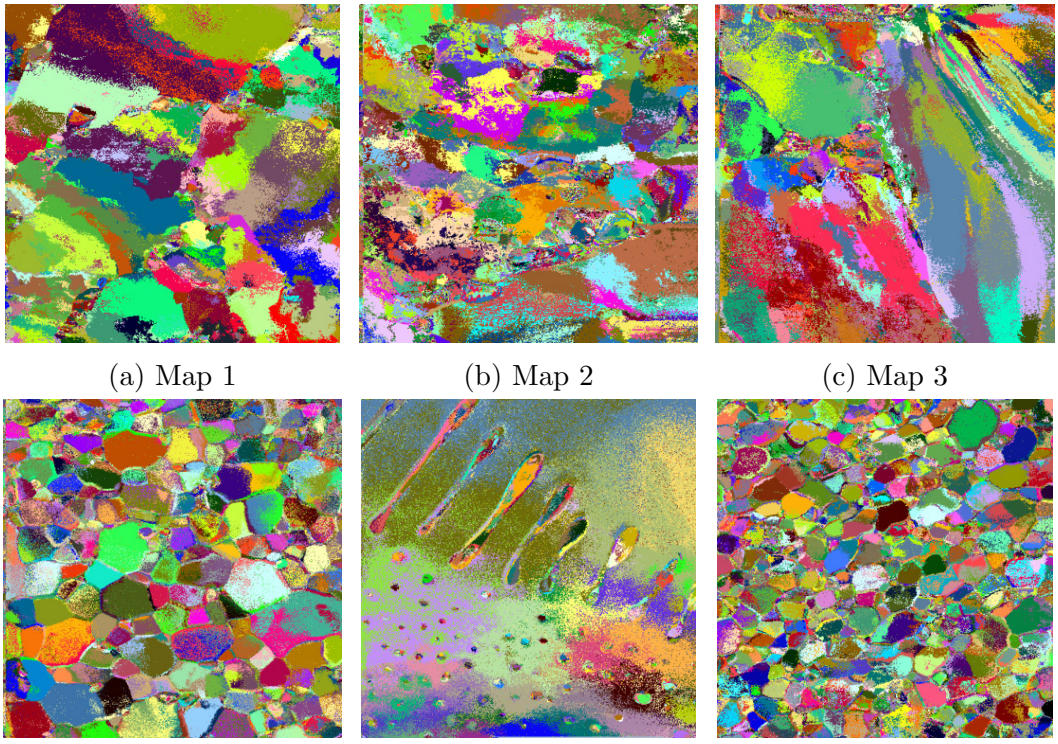


Figure 4.10: Classification results of SOTA DL models, accuracy scores are on experiments with two datasets, the unique DPs dataset and the 100 DPs duplicates dataset.

observe a significant drop of performance when the models are trained and evaluated on the dataset with 100 DP duplicates at most. The class label distribution does not significantly shift between these two dataset as shown in Table 4.5, our interpretation of this phenomenon relies on the fact that the SOTA DL models fail to learn meaningful representations of TEM data when trained with limited diffraction TEM data containing noisy duplicated DPs, thus, making them inadequate to be used for TEM data analysis. To remediate to this inefficiency we introduce in the following a new DL model, designed in the spirit of analyzing TEM diffraction data in experimental conditions. But first, we introduce the segmentation map of a micrograph as the result of a deterministic function that maps unique DPs with a unique label and Euler angles, to a unique identification. Figure 4.11 presents the segmentation maps of all considered micrographs. We implement the encoding function that maps the angles and phase label to a unique identifier using label encoder, these identifier are then used are indices for the XKCD color list[1] to retrieve the segmentation map.

By definition of the mapping function, a segmentation map aggregates the information of both phase determination and Euler's orientation maps. Therefore, it is essential to facilitate the interaction between DL models and existing software for TEM data analysis, correctly predicted segmentation maps will significantly reduce the cost of post-processing analysis of the data with deterministic approaches such as template matching.

[1] https://blog.xkcd.com/2010/05/03/color-survey-results/



(a) Map 1 (b) Map 2 (c) Map 3

(d) Map 10T (e) Map as-cast (f) Map 100T

Figure 4.11: Encoded segmentation maps for all considered micrographs.

The accurate prediction of a segmentation map requires a DL model to learn to distinguish between real DPs. Intuitively, differentiating DP images seems more accessible than the orientation prediction and has a better chance to generalize to unseen DP orientations. Consequently, we will focus on a specifically designed DL model for solving the segmentation map prediction problem for the rest of this study. This model is denoted as Multi-task Pairwise Siamese Network (MPSN). Siamese networks [22] are a particular class of DL models that internally have a twin or more identical subnetworks. Siamese architecture is used in many applications, from anomaly detection and classification to similarity and representation learning, etc.[37, 35]. The subnetworks in the siamese architecture are trained by mirroring the gradients during the backpropagation, they can take pairs or triplets as inputs, a rich literature exists on different training algorithms and losses, such as the contrastive loss or the triplet loss (for more details regarding siamese networks, please refer to [35]). Furthermore, siamese networks are known to be more robust to class imbalance by implementing a particular sampling strategy for the inputs during training, and they were also successfully applied in representation learning algorithms; they focus on learning a latent representation with a semantic similarity, encoded in the similarity measure

used during training.

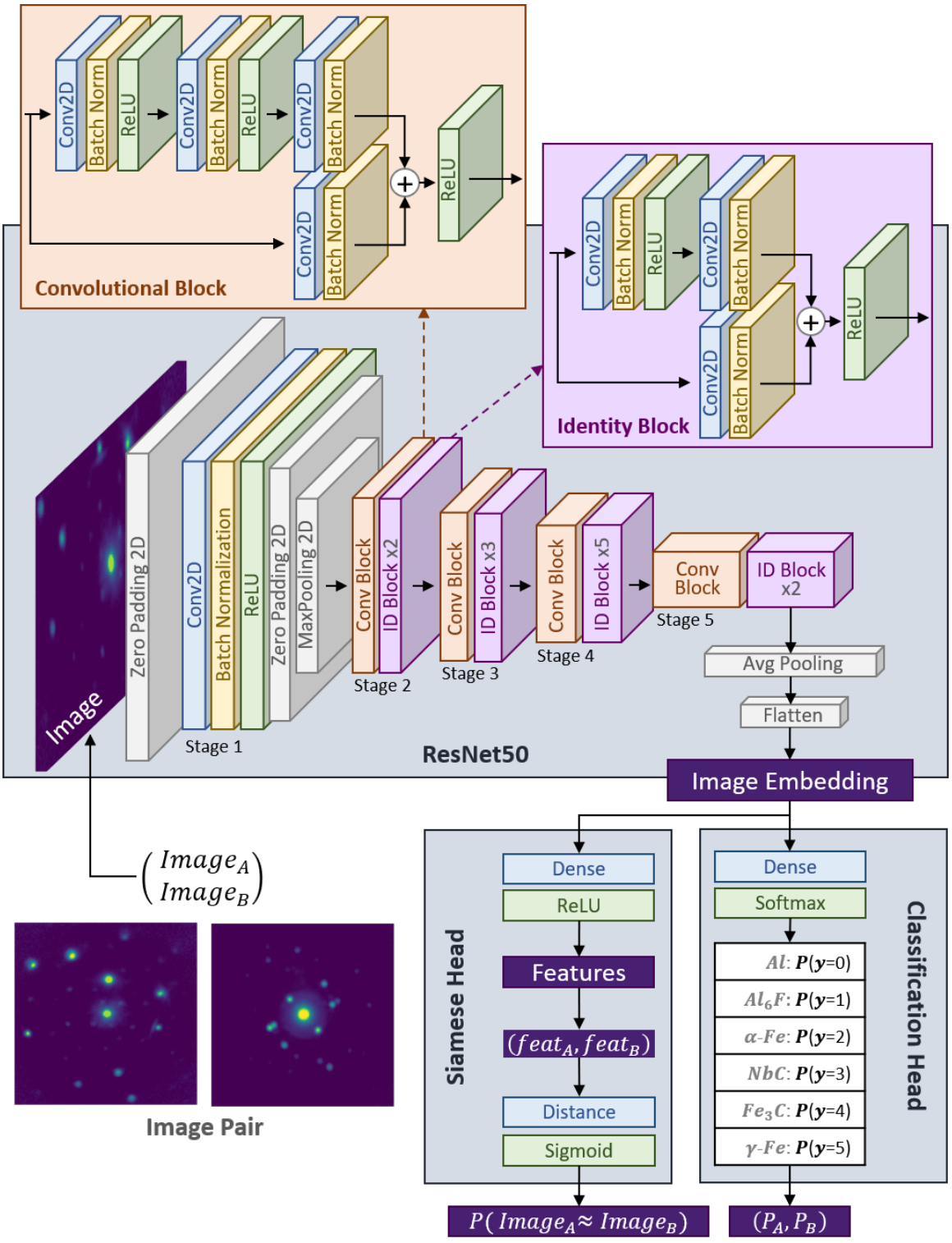


Figure 4.12: Multi Pairwise Siamese Network (MPSN) architecture.

For the MPSN architecture, we make use of the siamese idea in order to train our

DL model to learn to distinguish between DP image pairs of the same phase class. Figure 4.12 illustrates the architecture of the Multi-task Pairwise Siamese Network. The MPSN model takes as input DP image pairs $(\mathbf{x}_0, \mathbf{x}_1)$ from the same phase and a binary label vector y of size 7, the first coefficient of y corresponds to the label pair, which takes one for positive pairs and zero for negative pairs. A positive pair, in our case, is a pair of PD images of the same class with the same Euler orientation, meaning DP duplicates. Negative pairs, on the other hand, are PD image pairs of the same phase but at different Euler orientations. the latter 6 coefficients correspond to the one hot encoding of phase labels. The image pairs are sequentially forwarded through an image encoder. In our case, we chose the ResNet50 DL model [59] as the encoder by removing the dense top layer of the classifier and flattening the latent representation of the model as the image embedding space. This architecture is depicted in Figure 4.12.

This embedding is then connected to two separate heads. The siamese head is composed of a dense layer with relu activation for feature extraction. The feature vectors are used to estimate the distance metric. Then the result is mapped to the 0-1 space with the sigmoid function $\mathcal{S}(x) = 1/(1+e^{-x})$. The second head is a standard classification architecture, composed of a dense layer with the same number of units as the number of classes and a softmax activation function defined as $\sigma(x)_i = e^{x_i}/(\sum_j x_j)$ for each coordinate i of x, to map the logits to a probability distribution vector. The image pairs are also forwarded sequentially to obtain the probability vectors for phase prediction of each DP image.

The MPSN model is trained in an MTL fashion with a composite loss function balanced by a hyperparameter λ, constituted of a binary cross-entropy for the siamese head and another cross-entropy loss for the classification head:

$$\begin{aligned}\ell_\lambda(\mathbf{x}_0, \mathbf{x}_1, y) = &- \lambda[(y_0 \log(f^s(\mathbf{x}_0, \mathbf{x}_1; \theta^e, \theta^s)) + (1 - y_0) \log(1 - f^s(\mathbf{x}_0, \mathbf{x}_1; \theta^e, \theta^s)))] \\ &- (1 - \lambda) \sum_{i=0}^{1} \sum_{k=1}^{6} y_k \log(f^c(\mathbf{x}_i; \theta^e, \theta^c)).\end{aligned} \tag{4.4}$$

Where $f^s(.; \theta^e, \theta^s)$ is the siamese parametric function, with the shared encoder parameters θ^e, and the task-specific parameters θ^s, and $f^c(.; \theta^e, \theta^c)$ the classification parametric function with the shared encoder parameters θ^e, and the task-specific parameters θ^c. MPSN is trained to optimize the following minimization formulation:

$$\underset{\substack{\theta^e \\ \theta^s, \theta^c}}{\text{minimize}} \quad \sum_t \ell_\lambda(\mathbf{x}_0^{(t)}, \mathbf{x}_1^{(t)}, y^{(t)}) \tag{4.5}$$

Where t refers to the t th pair in the created sample set of pairs. The strategy for sampling image pairs is crucial for standard siamese models as same as for MPSN model. Algorithm 4 details the sampling strategy designed to train this model with TEM diffraction data, it takes as input the TEM diffraction data $\mathcal{S}$ of size n composed of DP images $\mathbf{x}_i$, their corresponding phase label p_i, and segment id c_i output of the segmentation mapping function. It returns a pair generator $\mathcal{G}(\mathcal{S})$ which will be used as source of training data for training the MPSN model. With the generator at hand, the model is trained like any other standard DL model by specifying the number of gradient steps per epoch. As mentioned previously, we utilize all the micrographs for training the model except for map 1 and 100T, which are kept for the final evaluation. The same optimization routine was carried out for all SOTA models and the Multi-task Pairwise Siamese Network.

Algorithm 4 Pair sampling generator for MPSN training

Require: $\mathcal{S} = \{(\mathbf{x}_i, p_i, c_i)\}_{i=1}^n$ a set of labeled TEM DP images, with $p_i \in [1, \ldots, 6]$ and $c_i \in C$, where C is the set of all code ids in $\mathcal{S}$.
Partition the set $\mathcal{S}$ in a hierarchical partition $\mathcal{P}(p, c)$.
Set a boolean variable *pos* to True.
Create a generator $\mathcal{G}$ on $\mathcal{S}$ with the following internal routine:
    When next sample is asked:
    Randomly sample p from $[1, \ldots, 6]$.
    if *pos* is True **then**
        Randomly sample c from C_p the subset of C of all id codes of phase p.
        Randomly sample an image pair $(\mathbf{x}, \mathbf{x}')$ from the partition $\mathcal{P}(p, c)$.
        Return the labeled positive pair $(\mathbf{x}, \mathbf{x}', pos, p)$.
        Set *pos* to False.
    else
        Randomly sample c and c' from C_p, such that $c \neq c'$.
        Randomly sample $\mathbf{x}$ from $\mathcal{P}(p, c)$ and $\mathbf{x}'$ from $\mathcal{P}(p, c')$.
        Return the labeled negative pair $(\mathbf{x}, \mathbf{x}', pos, p)$.
        Set *pos* to True.
Output : pair sampling generator $\mathcal{G}(\mathcal{S})$

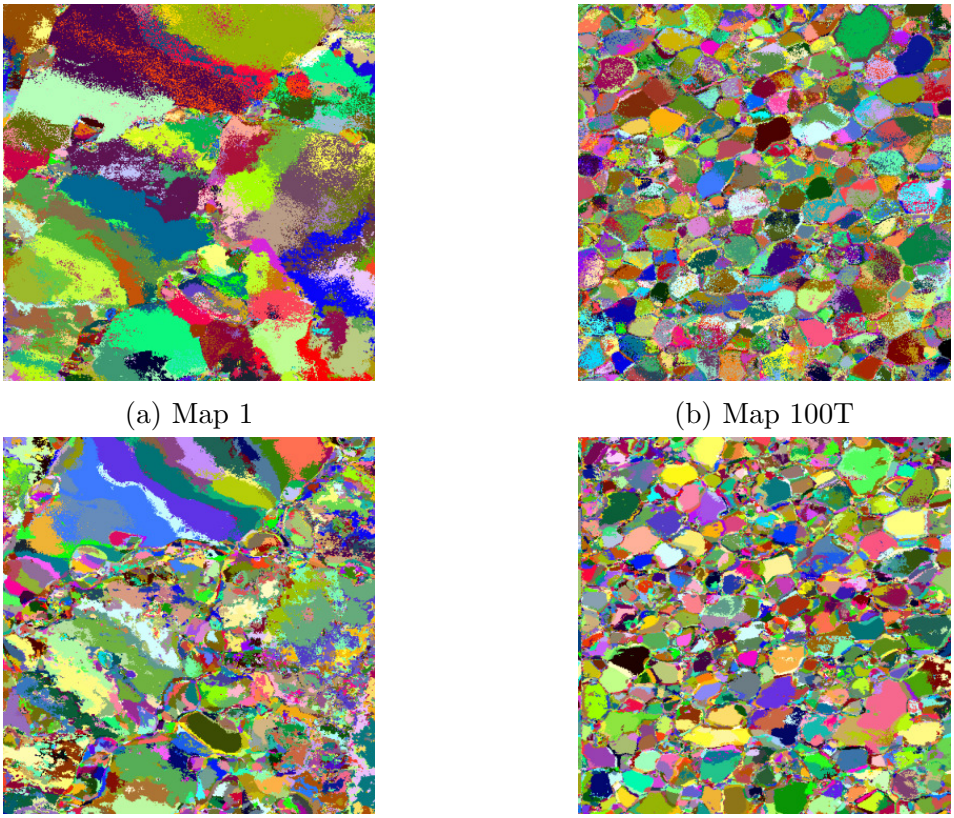
After training the model, we can predict the class label for the phase determination map and the segmentation map from preprocessed DPs using the algorithm below, which takes as input the TEM diffraction data of a micrograph and return, phase labels and segmentation ids to construct the maps.

Algorithm 5 MPSN prediction of a segmentation map

Require: $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^{n}$ a set of unlabeled TEM DP images, *mpsn* a trained MPSN model, and $\kappa = 0.8$ a confidence threshold.
Predict the phase label of all images in $\mathcal{S}$ with the classification head of *mpsn*.
Group the images in $\mathcal{S}$ by their phase prediction in $S[p]$.
Set $L = []$ and $c = 0$.
for each group g in $S[p]$ **do**
  while there are images in g **do**
    Randomly sample $\mathbf{x}$ from g.
    Predict pair labels on pairs $(g, duplicate(\mathbf{x}))$ with the classification head of *mpsn* using the confidence level κ.
    Add set of the positive pairs from the prediction to L with labels (p, c).
    Remove this set from g.
    Set $c = c + 1$
  end while
end for
Output : L the set of labeled TEM DP images, given unlabeled as input in $\mathcal{S}$.

The algorithm above takes advantage of both predictions in the MPSN model to retrieve the segmentation map of a given micrograph, the parameter κ determines the precision of the retrieved map, higher values will result in a segmentation map with a high number of segments, whereas values close to 0.5 will produce segmentations with fewer and larger segments. The figure below shows the predicted segmentation maps of the micrographs 1 and 100T by the MPSN model.

Figure 4.13 shows the segmentation maps of map 1 and 100T, predicted by the trained MPSN model using the algorithms described above. For the purpose of visualization clarity we fixed κ to 0.7 rather than the default value of 0.8. We observe that the predicted are visually similar to the ground truth maps, note that for the segmentation maps, we are interested in the segment shapes and not there color. The color id is just a code to differentiate between different segment. In our case, we suppose that each segment contains a unique DP with a given phase label and orientation. Especially for the micrograph of map 100T, the MPSN model is able to retrieved the mosaic patterns in the Euler's angle map, this confirms our intuition that using DL to differentiate between DP pairs may leads to better results than by addressing the regression task of angles, in particular when generalizing to unseen orientations.



(a) Map 1 (b) Map 100T

(c) Predicted map 1 (d) Predicted map 100T

Figure 4.13: First row corresponds to the segmentation maps of the micrographs 1 and 100T, in the second row are the ones predicted by the MPSN model with $\kappa = 0.7$.

Given the predicted segmentation maps by the MPSN model, we can recover Euler's orientation map and the phase map by querying a deterministic approach, such as template matching to label a single DP from each segment and propagate the labels of this DP through all the segments. Table 4.6 and Table 4.7 present the precision of these retrieved maps with respect to the ground truth maps provided by ASTAR. In addition to the metrics used in the previous section, we introduce the reduction metric as the proportion of queries saved when relying on label propagation with the predicted segmentation maps rather than querying all DPs of the micrograph. Note that with smaller κ, we obtain segmentation maps with fewer segments, which significantly reduces the number of queries needed to retrieve the Euler's orientation and phase's maps, as shown in the tables. On the other hand, high κ values ensure better precision in predicting the phases and angles.

Figure 4.14 and Figure 4.15 show the predicted orientation and phase maps by label propagation, given the segmentation map predicted by the MPSN model with a confidence level $\kappa = 0.8$, for the micrographs of map 1 and map 100T respectively.

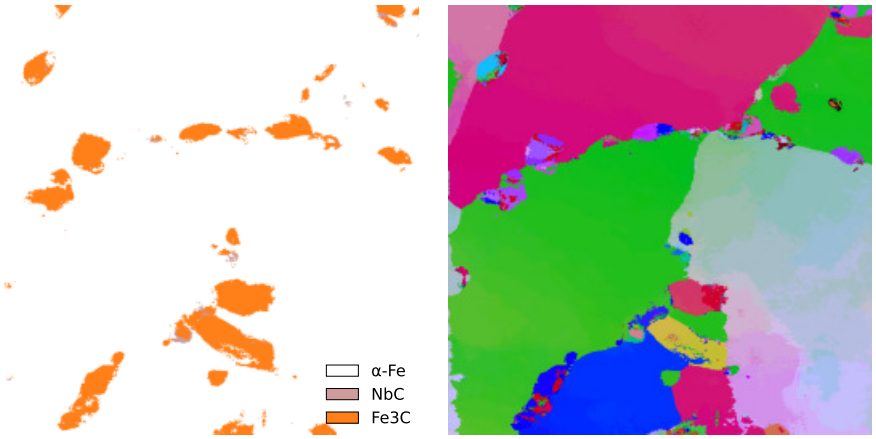
κ	reduction	accuracy	balanced accuracy	maae (degrees)		
				ϕ_1	Φ	ϕ_2
0.6	**99.58**%	95.7%	55.9%	10.0	5.3	7.2
0.7	98.59%	97.5%	63.5%	5.7	3.2	4.7
0.8	94.70%	98.4%	75.7%	3.6	2.1	3.2
0.9	66.54%	**99.4**%	**88.8**%	**1.3**	**0.8**	**1.2**

Table 4.6: Accuracy, balanced accuracy and mean absolute angle error of the MPSN model predictions on map 1, the reduction is in terms of number of queries.

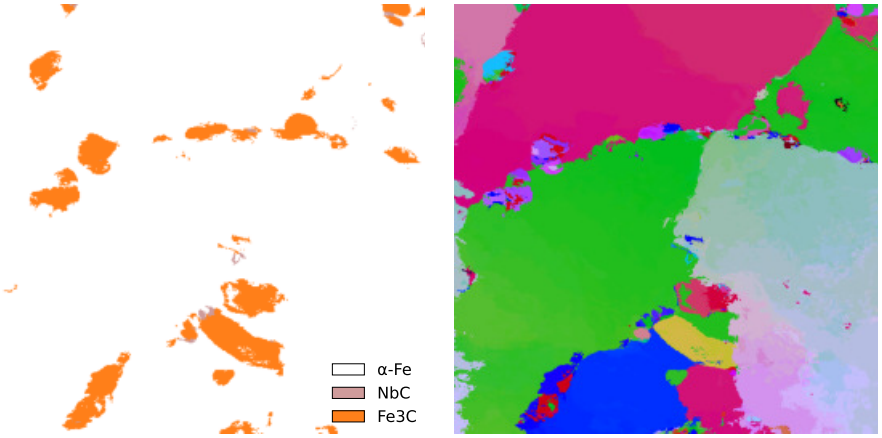
κ	reduction	accuracy	balanced accuracy	maae (degrees)		
				ϕ_1	Φ	ϕ_2
0.6	**99.50**%	99.2%	66.6%	50.9	39.3	51.0
0.7	97.84%	99.3%	73.3%	22.4	18.1	24.3
0.8	92.37%	99.6%	87.9%	11.3	9.0	12.8
0.9	65.67%	**99.9**%	**95.2**%	**4.8**	**3.9**	**5.8**

Table 4.7: Accuracy, balanced accuracy and mean absolute angle error of the MPSN model predictions on map 100T, the reduction is in terms of number of queries.

These results show that the MPSN model architecture can drastically reduce the cost of analyzing TEM data with deterministic approaches by over 90%, while providing a high accuracy on the resulted MAPS after the analysis. Furthermore, we show from our experimental protocol that the model can generalize to new experimental TEM data without the need of training on simulated data or fine-tuning. These arguments support the realistic deployment of such a DL approach in an interactive setting with an existing deterministic method to significantly speed up The TEM data analysis for a real-time use case.

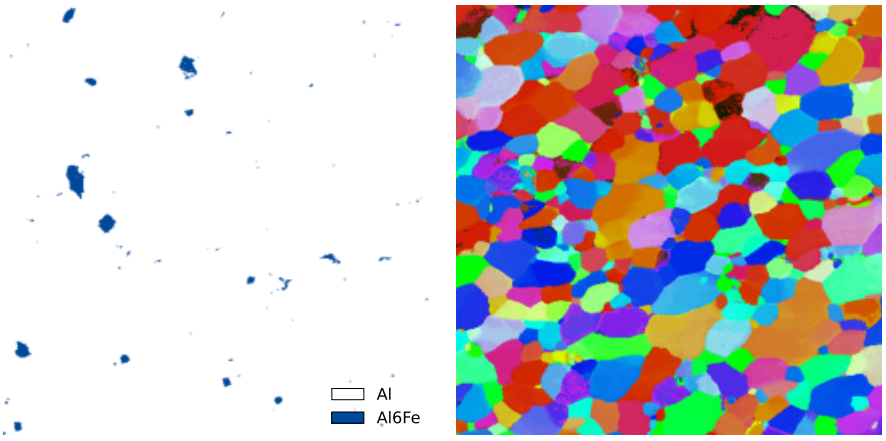


(a) Phases and Euler's orientation maps of micrograph of map 1.

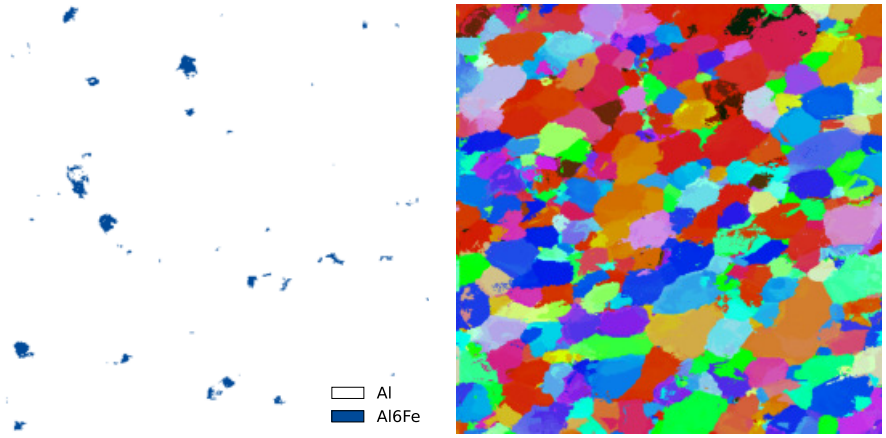


(b) Predicted phases and Euler's orientation maps using the MPSN model with $\kappa = 0.8$.

Figure 4.14: Phases and Euler's orientation maps found using the segmentation map predicted by the MPSN model on map 1.



(a) Phases and Euler's orientation maps of micrograph of map 100T.



(b) Predicted phases and Euler's orientation maps using the MPSN model with $\kappa = 0.8$.

Figure 4.15: Phases and Euler's orientation maps found using the segmentation map predicted by the MPSN model on map 100T.

4.4 Conclusion

In this work, we investigated the potential of deep learning for the rapid automation of TEM data analysis. We proposed a preprocessing of simulated and experimental DPs with descriptors. We demonstrated the value of multi-task learning for the regression task of orientation prediction and the classification task of phase determination and transfer learning by successfully learning to analyze simulated TEM diffraction data and applying it to experimental diffraction data. In parallel, we performed an extensive comparative study of SOTA DL models to analyze diffraction data with a realistic experimental protocol. Furthermore, we introduced the segmentation map a new visualization support to allow DL-based solutions to interact in an interactive way with existing deterministic approaches for TEM data analysis. In this context, we address the problem of TEM data analysis for DL as the task of differentiating between experimental DPs, in order to predict the corresponding segmentation map. We proposed a new DL model, namely the Multi-task Pairwise Siamese Network with dedicated training and inference procedures. The model is able to drastically reduce labeling cost in terms of the number of queries of traditional deterministic algorithms for TEM analysis while showing favorable results in the prediction of maps using label propagation on the predicted segmentation map with promising performance in generalizing to unseen micrographs.

In the current state of the presented methods, it appears that accurate phase determination can be reached both based on training on simulated images and directly on experimental data. In the latter case, the main ussie is obviously the amount of data available and the diversity of data, since experimental maps contain a large proportion of duplicates. As for the simulated data strategy, in order to become more user-friendly, future work should also address how a pre-trained model could be easily transferred to different experimental configurations (camera length, precession conditions, etc.). Concerning the prediction of orientation angle, promising precision has been obtained on some Euler angles but not all, and further research is need to improve this precision to a degree where the noise on experimental maps can be reduced to observe relevant features in most microstructures. The possibility to implement a real time approximation of the phase and orientation map during TEM acquisition is not out of reach given the results presented here, and would be a valuable asset for more useful data collection by the experimentalists.

Chapter 5

Conclusions and Future Perspectives

The work in this thesis has concentrated on demonstrating the relevance of implicit information in the design of machine learning algorithms. This study direction has been extensively investigated by the community, with its different subdomains in semi-supervised learning, active learning, and transfer learning. We demonstrated that implicit information may take numerous forms and is especially useful in learning circumstances where labeled information is sparse and expensive. This implicit information can also come from a variety of sources, such as data distribution, a learning algorithm's knowledge model, or simulations. We studied these many sources of implicit information by developing tailored learning algorithms to recognize and exploit this knowledge to improve learning efficiency.

Further research into new approaches to develop learning algorithms to recognize and apply implicit information is critical. To solve our daily activities, we create effective ways of combining contextual knowledge and prior experiences. However, without enough labeled data, these activities remain beyond the capabilities of machine intelligence.

Future goals should be to achieve human intelligence not by completing a specific set of problems, as was previously regarded to be a milestone for artificial intelligence, but rather by developing learning models and algorithms that better blend memory, experience, and perception. With developing study paths such as few-shot learning or self-learning, the research community has already set its course to move in this direction. In my opinion, the research done in this area will be important in advancing AI to the next level.

Bibliography

[5] Naoki Abe and Hiroshi Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 1–9, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[6] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017.

[7] J. A. Aguiar, M. L. Gong, R. R. Unocic, T. Tasdizen, and B. D. Miller. Decoding crystallography from high-resolution electron imaging and diffraction datasets with deep learning. *Science Advances*, 5(10):1949, 2019.

[8] Alnur Ali, R. Caruana, and Ashish Kapoor. Active learning with model selection. In *AAAI*, 2014.

[9] Massih R. Amini, Nicolas Usunier, and François Laviolette. A transductive bound for the voted classifier with an application to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 65–72, 2009.

[10] Massih-Reza Amini and Nicolas Usunier. *Learning with Partially Labeled and Interdependent Data*. Springer, 2015.

[11] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, May 2002.

[12] Maria-Florina Balcan and Avrim Blum. An augmented PAC model for semi-supervised learning. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, pages 396–419. The MIT Press, 2006.

[13] Yoram Baram, Ran El-Yaniv, and Kobi Luz. Online choice of active learning algorithms. *J. Mach. Learn. Res.*, 5:255–291, 2004.

[14] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(85):2399–2434, 2006.

[15] Shai Ben-David and Tyler Lu and. Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning. In *21st Annual Conference on Learning Theory - COLT*, pages 33–44, 2008.

[16] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[17] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[18] K. Bhatia, K. Dahiya, H. Jain, A. Mittal, Y. Prabhu, and M. Varma. The extreme classification repository: Multi-label datasets and code, 2016.

[19] Thomas Bonis and Steve Oudot. A fuzzy clustering algorithm for the mode-seeking framework. *Pattern Recognition Letters*, 102:37 – 43, 2018.

[20] Arturo Bonnin, Ricard Borràs, and Jordi Vitrià. A cluster-based strategy for active learning of rgb-d object detectors. In *ICCV Workshops*, pages 1215–1220. IEEE, 2011.

[21] G. Boole. *An Investigation of the Laws of Thought: On Which Are Founded the Mathematical Theories of Logic and Probabilities*. Creative Media Partners, LLC, 2015.

[22] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, page 737–744, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[23] Roberto Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. 03 2009.

[24] Guénaël Cabanes, Younès Bennani, Renaud Destenay, and André Hardy. A new topological clustering algorithm for interval data. *Pattern Recognition*, 46(11):3030 – 3039, 2013.

[25] G. Carlsson. *The Shape of Data*, page 16–44. London Mathematical Society Lecture Note Series. Cambridge University Press, 2012.

[26] Gunnar Carlsson and Rickard Brüel Gabrielsson. Topological approaches to deep learning. In Nils A. Baas, Gunnar E. Carlsson, Gereon Quick, Markus Szymik, and Marius Thaule, editors, *Topological Data Analysis*, pages 119–146, Cham, 2020. Springer International Publishing.

[27] C.B. Carter and D.B. Williams. *Transmission Electron Microscopy: Diffraction, Imaging, and Spectrometry*. Springer International Publishing, 2016.

[28] Vittorio Castelli and Thomas M. Cover. On the exponential value of labeled samples. *Pattern Recognit. Lett.*, 16(1):105–111, 1995.

[29] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT Press, 2006.

[30] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. The MIT Press, 2006.

[31] Nitesh Chawla and Grigoris Karakoulas. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research - JAIR*, 23, 09 2011.

[32] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, SCG '09, page 237–246, New York, NY, USA, 2009. Association for Computing Machinery.

[33] Frédéric Chazal, Leonidas J. Guibas, Steve Y. Oudot, and Primoz Skraba. Scalar field analysis over point cloud data. *Discrete & Computational Geometry*, 46(4):743, 2011.

[34] Liangyu Chen, Yutong Bai, Siyu Huang, Yongyi Lu, Bihan Wen, Alan Loddon Yuille, and Zongwei Zhou. Making your first choice: To address cold start problem in vision active learning. *ArXiv*, abs/2210.02442, 2022.

[35] Davide Chicco. *Siamese Neural Networks: An Overview*, pages 73–94. Springer US, New York, NY, 2021.

[36] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017.

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

[38] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, May 1994.

[39] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[40] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 208–215, New York, NY, USA, 2008. Association for Computing Machinery.

[41] Ilias Diakonikolas, Themis Gouleakis, and Christos Tzamos. Distribution-independent pac learning of halfspaces with massart noise. In *Advances in Neural Information Processing Systems*, volume 32, pages 4749–4760. Curran Associates, Inc., 2019.

[42] Ilias Diakonikolas and Daniel M. Kane. Hardness of learning halfspaces with massart noise, 2020.

[43] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[44] Amandine Duchaussoy, Xavier Sauvage, Kaveh Edalati, Zenji Horita, Gilles Renou, Alexis Deschamps, and Frédéric De Geuser. Structure and mechanical behavior of ultrafine-grained aluminum-iron alloy stabilized by nanoscaled intermetallic particles. *Acta Materialia*, 167:89–102, 2019.

[45] John C. Duchi. Introductory lectures on stochastic convex optimization. *Park City Mathematics Series, Graduate Summer School Lectures*, 2016.

[46] A. Dvoretzky, J. Kiefer, and J. Wolfowitz. Asymptotic Minimax Character of the Sample Distribution Function and of the Classical Multinomial Estimator. *The Annals of Mathematical Statistics*, 27(3):642 – 669, 1956.

[47] Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28(4):511–533, November 2002.

[48] Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction.* American Mathematical Society, 2010.

[49] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June 2008.

[50] Mark Fanty and Ronald Cole. Spoken letter recognition. In R. P. Lippmann, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 220–226. Morgan-Kaufmann, 1991.

[51] Vasilii Feofanov, Emilie Devijver, and Massih-Reza Amini. Transductive bounds for the multi-class majority vote classifier. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*, pages 3566–3573, 2019.

[52] Chazal Frederic, Guibas Leonidas J., Oudot Steve Y., and Skraba Primoz. Persistence-based clustering in riemannian manifolds. *J. ACM*, 60(6), 2013.

[53] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2022.

[54] H. Goldstein. *Classical Mechanics*. Addison-Wesley series in physics. Addison-Wesley Publishing Company, 1980.

[55] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17, pages 529–536. MIT Press, 2005.

[56] Isabelle Guyon, Gavin C. Cawley, Gideon Dror, and Vincent Lemaire. Results of the active learning challenge. In Isabelle Guyon, Gavin Cawley, Gideon Dror, Vincent Lemaire, and Alexander Statnikov, editors, *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, volume 16

of *Proceedings of Machine Learning Research*, pages 19–45, Sardinia, Italy, 16 May 2011. PMLR.

[57] Allen Hatcher. *Algebraic topology*. Cambridge Univ. Press, Cambridge, 2000.

[58] Jean-Claude Hausmann. *On the Vietoris-Rips complexes and a cohomology theory for metric spaces*, pages 175–188. Prospects in topology : proceedings of a conference in honor of William Browder. Princeton University Press, Princeton, N.J., 1995. ID: unige:12821.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[60] Clara Higuera, Katheleen J. Gardiner, and Krzysztof J. Cios. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PLOS ONE*, 10(6):1–28, 2015.

[61] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[62] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[63] Rong Hu, Brian Mac Namee, and Sarah Jane Delany. Off to a good start: Using clustering to select the initial training set in active learning. In *FLAIRS*, 2010.

[64] Gao Huang, Zhuang Liu, Geoff Pleiss, Laurens Van Der Maaten, and Kilian Weinberger. Convolutional networks with dense connectivity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[65] Itseez. Open source computer vision library. https://github.com/itseez/opencv, 2015.

[66] Deng Jia, Dong Wei, Socher Richard, Li Li-Jia, Li Kai, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. pages 248–255, 2009.

[67] Yi Jiang, Dong Chen, Xin Chen, Tangyi Li, Guo-Wei Wei, and Feng Pan. Topological representations of crystalline compounds for the machine-learning prediction of materials properties. *npj Computational Materials*, 7(1):28, Feb 2021.

[68] Haifeng Jin, Qingquan Song, and Xia Hu. Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 1946–1956, New York, NY, USA, 2019. Association for Computing Machinery.

[69] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[70] Jaeho Kang, Kwang Ryel Ryu, and Hyuk chul Kwon. Using cluster-based sampling to select initial training set for active learning in text classification. In *PAKDD*, 2004.

[71] Leonard Kaufman and Peter Rousseeuw. *Finding Groups in Data: An Introduction To Cluster Analysis*. 01 1990.

[72] Daniel Khashabi. Learning halfspaces; literature review and some recent results. 2016.

[73] Jun-ichi Kikuchi and Kazuma Yasuhara. *Transmission Electron Microscopy (TEM)*. John Wiley & Sons, Ltd, 2012.

[74] Georg Krempl, Tuan Cuong Ha, and Myra Spiliopoulou. Clustering-based optimised probabilistic active learning (copal). In Nathalie Japkowicz and Stan Matwin, editors, *Discovery Science*, pages 101–115, Cham, 2015. Springer International Publishing.

[75] Georg Krempl, Daniel Kottke, and Vincent Lemaire. Optimised probabilistic active learning (opal). *Machine Learning*, 100(2):449–476, Sep 2015.

[76] Jesse H. Krijthe. Rssl: Semi-supervised learning in r. In Bertrand Kerautret, Miguel Colom, and Pascal Monasse, editors, *Reproducible Research in Pattern Recognition*, pages 104–115, Cham, 2017. Springer International Publishing.

[77] Aditi S. Krishnapriyan, Joseph Montoya, Maciej Haranczyk, Jens Hummelshøj, and Dmitriy Morozov. Machine learning with persistent homology and chemical word embeddings improves prediction accuracy and interpretability in metal-organic frameworks. *Scientific Reports*, 11(1):8888, Apr 2021.

[78] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30*, pages 6402–6413, 2017.

[79] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the Institute of Radio Engineers*, 86(11):2278–2323, 1998.

[80] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In William W. Cohen and Haym Hirsh, editors, *Machine Learning Proceedings 1994*, pages 148–156. Morgan Kaufmann, San Francisco (CA), 1994.

[81] Yu-Feng Li and Zhi-Hua Zhou. Towards Making Unlabeled Data Never Hurt. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1081–1088, 2011.

[82] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[83] Edwin Lughofer. Single-pass active learning with conflict and ignorance. *Evolving Systems*, 3(4):251–271, 2012.

[84] PY Lum, G Singh, A Lehman, T Ishkanov, M Vejdemo-Johansson, M Alagappan, J Carlsson, and G Carlsson. Extracting insights from the shape of complex data using topology. *Scientific reports*, 3:1236, 2013.

[85] Ben H. Martineau, Duncan N. Johnstone, Antonius T. J. van Helvoort, Paul A. Midgley, and Alexander S. Eggeman. Unsupervised machine learning applied to scanning precession electron diffraction data. *Advanced Structural and Chemical Imaging*, 5(1):3, Mar 2019.

[86] Pascal Massart and Elodie Nédélec. Risk bounds for statistical learning. *Ann. Statist.*, 34(5):2326–2366, 10 2006.

[87] Yury Maximov, Massih-Reza Amini, and Zaïd Harchaoui. Rademacher complexity bounds for a penalized multi-class semi-supervised algorithm. *Journal of Artificial Intelligence Research*, 61:761–786, 2018.

[88] Jun Meng, Zheng Chang, Peng Zhang, Wenhao Shi, and Yushi Luan. lncrna-lstm: Prediction of plant long non-coding rnas using long short-term memory based on p-nts encoding. In De-Shuang Huang, Zhi-Kai Huang, and Abir Hussain, editors, *Intelligent Computing Methodologies*, pages 347–357, Cham, 2019. Springer International Publishing.

[89] Omar Montasser, Surbhi Goel, Ilias Diakonikolas, and Nathan Srebro. Efficiently learning adversarially robust halfspaces with noise. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7010–7021. PMLR, 13–18 Jul 2020.

[90] Joydeep Munshi, Alexander M Rakowski, Benjamin Savitzky, Colin Ophus, Matthew L Henderson, Shreyas Cholia, and Maria KY Chan. 4d crystal: Deep learning crystallographic information from electron diffraction images. *Microscopy & Microanalysis*, 27(S1):2774–2776, 2021.

[91] Partha Niyogi. Manifold regularization and semi-supervised learning: Some theoretical analyses. *Journal of Machine Learning Research*, 14(1):1229–1250, 2013.

[92] Gert Nolze. Euler angles and crystal symmetry. *Crystal Research and Technology*, 50(2):188–201, February 2015.

[93] Fábio Perez, Rémi Lebret, and Karl Aberer. Cluster-based active learning. *CoRR*, abs/1812.11780, 2018.

[94] L. Polterovich, D. Rosen, K. Samvelyan, and J. Zhang. *Topological Persistence in Geometry and Analysis*. University Lecture Series. American Mathematical Society, 2020.

[95] Kossar Pourahmadi, Parsa Nooralinejad, and Hamed Pirsiavash. A simple baseline for low-budget active learning. *arXiv preprint arXiv:2110.12033*, 2021.

[96] Edgar Rauch and Laurent Dupuy. Rapid diffraction patterns identification through template matching. *Archives of Metallurgy and Materials*, 50:87–99, 01 2005.

[97] Edgar F. Rauch, Muriel Véron, Stavros Nicolopoulos, and Daniel Bultreys. Orientation and phase mapping in tem microscopy (ebsd-tem like): Applications

to materials science. In *Electron Microscopy XIV*, volume 186 of *Solid State Phenomena*, pages 13–15. Trans Tech Publications Ltd, 4 2012.

[98] Bastian Rieck, Tristan Yates, Christian Bock, Karsten Borgwardt, Guy Wolf, Nicholas Turk-Browne, and Smita Krishnaswamy. Uncovering the topology of time-varying fmri data using cubical persistence. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6900–6912. Curran Associates, Inc., 2020.

[99] Philippe Rigollet. Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research*, 8:1369–1392, 2007.

[100] Joseph D Romano, Trang T Le, William La Cava, John T Gregg, Daniel J Goldberg, Praneel Chakraborty, Natasha L Ray, Daniel Himmelstein, Weixuan Fu, and Jason H Moore. Pmlb v1.0: an open source dataset collection for benchmarking machine learning methods. *arXiv preprint arXiv:2012.00058v2*, 2021.

[101] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In Carla E. Brodley and Andrea Pohoreckyj Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 441–448. Morgan Kaufmann, 2001.

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

[103] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[104] Chuqiao Shi, Michael Cao, David Muller, and Yimo Han. Rapid and semi-automated analysis of 4d-stem data via unsupervised learning. *Microscopy & Microanalysis*, 27(S1):58–59, 2021.

[105] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[106] Oriane Siméoni, Mateusz Budnik, Yannis Avrithis, and Guillaume Gravier. Rethinking deep active learning: Using unlabeled data at model training. *ICPR*, 2019.

[107] Aarti Singh, Robert Nowak, and Jerry Zhu. Unlabeled data: Now it helps, now it doesn't. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2009.

[108] Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. pages 91–100, 01 2007.

[109] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[110] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019.

[111] Ruth Urner, Sharon Wulff, and Shai Ben-David. Plal: Cluster-based active learning. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 376–397, Princeton, NJ, USA, 12–14 Jun 2013. PMLR.

[112] Nicolas Usunier, Massih-Reza Amini, and Cyril Goutte. Multiview semi-supervised learning for ranking multilingual documents. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD*, pages 443–458, 2011.

[113] D Viladot, M. Véron, Mauro Gemmi, F. Peiró, Joaquim Portillo, Sonia Estradé, Joan Mendoza, Nuria Llorca-Isern, and Stavros Nicolopoulos. Orientation and phase mapping in the transmission electron microscope using precession-assisted diffraction spot recognition: State-of-the-art results. *Journal of microscopy*, 252, 07 2013.

[114] Jean-Noël Vittaut, Massih-Reza Amini, and Patrick Gallinari. Learning classification with both labeled and unlabeled data. In *Proceedings of the 13th European Conference on Machine Learning (ECML'02)*, pages 468–476, 2002.

[115] Christoph von der Malsburg. Frank rosenblatt: Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. *Brain Theory*, pages 245–248, 01 1986.

[116] Ellen M. Voorhees. *The Effectiveness & Efficiency of Agglomerative Hierarchic Clustering in Document Retrieval*. PhD thesis, Cornell University, USA, 1985.

[117] Douglas A. Wolfe. *Nonparametrics: Statistical Methods Based on Ranks and Its Impact on the Field of Nonparametric Statistics*, pages 1101–1110. Springer US, Boston, MA, 2012.

[118] Kelin Xia and Guo-Wei Wei. Persistent homology analysis of protein structure, flexibility, and folding. *International Journal for Numerical Methods in Biomedical Engineering*, 30(8):814–844, 2014.

[119] Michael Xu, Abinash Kumar, and James LeBeau. Automating electron microscopy through machine learning and usetem. *Microscopy & Microanalysis*, 27(S1):2988–2989, 2021.

[120] Yan Yan, Romer Rosales, Glenn Fung, and Jennifer G. Dy. Active learning from crowds. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, page 1161–1168, 2011.

[121] Jianwei Yang, Zirun Chen, Wen-Sheng Chen, and Yunjie Chen. Robust affine invariant descriptors. *Mathematical Problems in Engineering*, 2011.

[122] Q. Yang, Y. Zhang, W. Dai, and S.J. Pan. *Transfer Learning*. Cambridge University Press, 2020.

[123] Qiang Yang, Yu Zhang, Wenyuan Dai, and Sinno Jialin Pan. *Transfer Learning*. Cambridge University Press, 2020.

[124] Chengzhu Yu and John H. L. Hansen. Active learning based constrained clustering for speaker diarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(11):2188–2198, 2017.

[125] S. Zaefferer and G. Wu. *Development of a TEM-Based Orientation Microscopy System*, chapter 24, pages 221–228. John Wiley & Sons, Ltd, 2008.

[126] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.