Self-Training: A Survey

Massih-Reza Amini[†], Vasilii Feofanov[†], Loïc Pauletto[†], Liès Hadjadj[†],
Émilie Devijver[†], Yury Maximov[⋆]
[†]University Grenoble Alpes, CNRS [⋆]Los Alamos National Laboratory
Grenoble, France Los Alamos, USA

Abstract

Self-training methods have gained significant attention in recent years due to their effectiveness in leveraging small labeled datasets and large unlabeled observations for prediction tasks. These models identify decision boundaries in low-density regions without additional assumptions about data distribution, using the confidence scores of a learned classifier. The core principle of self-training involves iteratively assigning pseudo-labels to unlabeled samples with confidence scores above a certain threshold, enriching the labeled dataset and retraining the classifier. This paper presents self-training methods for binary and multi-class classification, along with variants and related approaches such as consistency-based methods and transductive learning. We also briefly describe self-supervised learning and reinforced self-training. Furthermore, we highlight popular applications of self-training and discuss the importance of dynamic thresholding and reducing pseudo-label noise for performance improvement.

To the best of our knowledge, this is the first thorough and complete survey on self-training.

1 Introduction

Self-training is a branch of semi-supervised learning and has emerged as a prominent approach within the machine learning domain, addressing the core challenge of leveraging both labeled and unlabeled data for improved inference. This approach is particularly valuable in scenarios where labeled examples are scarce but there is an abundance of unlabeled data available for training. Self-training has proven highly relevant in a range of applications, such as computer vision, natural language processing, and speech recognition, where the acquisition of labeled data can be costly and time-consuming [93, 19, 29, 64]. By iteratively assigning pseudo-labels to unlabeled samples and retraining the classifier, self-training methods enhance the model's performance and generalization capabilities.

1.1 Central hypothesis

In general, it remains unclear how unlabeled data can be used in training and what value it can bring. The basic assumption in semi-supervised learning, called smoothness, stipulates that two examples in a high density region should have identical class labels [14, 3]. This means that if two points are part of the same group or cluster, their class labels will most likely be the same. If they are separated by a low density zone, on the other hand, their desired labels should be different. Hence, if the examples of the same class form a partition, unlabeled training data might aid in determining the partition boundary more efficiently than if just labeled training examples were utilized.

1.2 Three main semi-supervised learning families

There are three main families of semi-supervised methods, each with its own adaptation of the smoothness hypothesis. These adaptations are usually referred to as assumptions, albeit loosely, since they rather represent different paradigms for implementing semi-supervised learning.

Data clustering uses a mixture model and assigns class labels to groups using the labeled data they include; and it constitutes the working principle of generative approaches [40]. The cluster assumption, which underpins these approaches, asserts that if two examples are in the same group, they are likely to be in the same class (Figure 1 (a)). This hypothesis may be explained as follows: if a group is formed by a large number of instances, it is rare that they belong to different classes. This does not imply that a class is constituted by a single group of examples, but rather that two examples from distinct classes are unlikely to be found in the same cluster.

If we consider the partitions of instances to be high density areas, a form of the cluster assumption known as low density separation entails determining the decision boundary over low density regions (Figure 1 (b)), and it constitutes the basis of discriminant techniques. The main difference between generative and discriminant techniques is that discriminant approaches find directly the prediction function without making any assumption on how the data are generated [1].

Density estimation is often based on a notion of distance, which may become meaningless for high dimensional spaces. A third hypothesis, known as the manifold assumption, stipulates that in high-dimensional spaces, instances reside on low-dimensional topological spaces that are locally Euclidean (Figure 1 (c)), which is supported by a variety of semi-supervised models called graphical approaches [6].

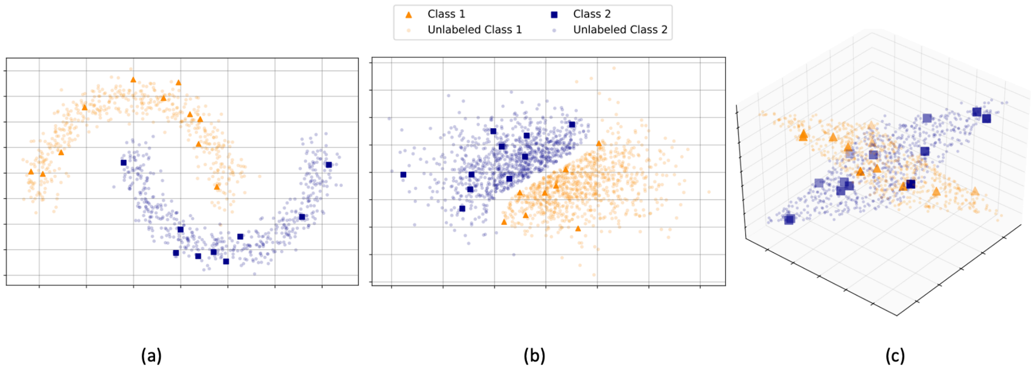

Figure 1: Illustration of three main hypotheses made in semi-supervised learning: (a) cluster assumption, (b) low-density separation and (c) manifold assumption.

1.3 Compatibility

Although semi-supervised algorithms have been successfully applied in many situations, there have been cases where unlabeled data have been shown to have no effect on the performance of a learning task [75]. Several attempts have been made in recent years to investigate the value of unlabeled data in the training process [12], and the capacity of semi-supervised learning approaches to generalize [52]. The bulk of these studies are founded on the notion of compatibility defined by [4], and they strive to exhibit the connection between the marginal data distribution and the target function to be learned. According to these findings, unlabeled data will be beneficial for training only if such a relationship exists.

In generative approaches, the marginal distribution is viewed as a mixture of class conditional distributions, and when compared to the supervised case, semi-supervised learning has been shown to achieve lower finite-sample error bounds in some general cases, or a faster rate of error convergence in others [12, 52, 75]. In this line, [7] showed that accessing the marginal distribution on unlabeled training data would not provide sample size guarantees superior to those obtained by supervised learning unless very strong assumptions about conditional distribution on class labels are made.

For graph-based approaches, [57] provided a context in which such algorithms may be studied and perhaps justified; the key finding of the study is that unlabeled data can help learning in some situations by explicitly defining the structure of the data through a manifold.

Finally, discriminant approaches mostly embed a margin maximization method that searches the decision boundary in low-density regions by pushing it from the unlabeled data [36]. In this survey we focus on self-training algorithms that follow this principle by assigning pseudo-labels to high-confidence unlabeled training examples and include these pseudo-labeled samples in the learning process. While various surveys have explored semi-supervised learning in recent years [83, 92], none have specifically emphasized self-training, which has emerged as the predominant approach in the field, widely applied across various applications.

1.4 Paper structure

The reminder of this paper is organized as follows.

In Section 2, we go over the self-training method in detail. First, we present the framework and notations used throughout the paper in Section 2.1, then we describe the general self-training algorithm in Section 2.2, also introduced in Algorithm 1. Then, we describe pseudo-labeling methods and its variants in Section 2.3, and we discuss the self-training with two classifiers in Section 2.4. Those methods are summed up in Table 1. Finally, we provide some insights into current theoretical studies in Section 2.6.

Other related approaches are described in Section 3. First, we detail the transductive learning context in Section 3.1, and the consistency-based approaches in Section 3.2. Going beyond traditional semi-supervised learning, we investigate the extension of self-training in domain adaptation in Section 2.5, delve into self-supervised learning in Section 3.3, and explore reinforced self-training in Section 3.4.

Section 4 reviews application of self-training methods in different domains, such as natural language processing in Section 4.1, computer vision in Section 4.2 and more generally in knowledge-driven applications in Section 4.3, with speech recognition, anomaly detection and genomics and proteomics.

The views and future prospects are discussed in Section 5.

2 Self-Training

Within this section, we present the fundamental aspects of the self-training approach. Initially, we introduce the framework and notation, followed by a comprehensive exploration of the core concept behind the self-training algorithm, which is further delineated in Algorithm 1. In Section 2.3 and Section 2.4, we present significant contributions directly linked to the standard algorithm. We organize these contributions effectively in Table 1. To conclude, we delve into the theoretical aspects in Section 2.6.

2.1 Semi-supervised framework and notations

We consider classification problems where the input and the output spaces are respectively $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{-1, +1\}$ or $\mathcal{Y} = \{1, \ldots, K\}$. We further suppose available a set of labeled training examples $S = (\mathbf{x}_i, y_i)_{1 \leqslant i \leqslant m} \in (\mathcal{X} \times \mathcal{Y})^m$ generated from a joint probability distribution $\mathbb{P}(\mathbf{x}, y)$ (denoted as $\mathcal{D}$) and a set of unlabeled training examples $X_{\mathcal{U}} = (\mathbf{x}_i)_{m+1 \leqslant i \leqslant m+u} \in \mathcal{X}^u$ supposed to be drawn from the marginal distribution $\mathbb{P}(\mathbf{x})$.

The classic case corresponds to when $m \ll u$, and the issue is thrown into the unsupervised learning framework if S is empty. The opposite extreme scenario is when $X_{\mathcal{U}}$ is empty and the problem is reduced to supervised learning. Given a hypothesis set of functions $\mathcal{H}$ mapping $\mathcal{X}$ to $\mathcal{Y}$, the learner receives a labeled set S and an unlabeled set $X_{\mathcal{U}}$ and outputs a hypothesis $h \in \mathcal{H}$ which is assumed to have a generalization error $R(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathbb{1}_{h(\mathbf{x})\neq y}]$ smaller than if just S was used to find the prediction function, where by $\mathbb{1}_\pi$ we denote the indicator function equal to 1 if the predicate π is true and 0 otherwise.

In practice, classifiers are defined based on a scoring function f from a class of functions $\mathcal{F} = \{f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}\}$, and for an example $\mathbf{x}$ the corresponding classification function h outputs the class for which the score of f is the highest:

$$h(\mathbf{x}) = \mathrm{argmax}_{y \in \mathcal{Y}} f(\mathbf{x}, y).$$

We define the margin $\rho_f(\mathbf{x}, y)$ of a function f for an example $\mathbf{x} \in \mathcal{X}$ and a class $y \in \mathcal{Y}$ as

$$\rho_f(\mathbf{x}, y) = f(\mathbf{x}, y) - \max_{y' \neq y} f(\mathbf{x}, y').$$

In the binary case, $\mathcal{Y} = \{-1, +1\}$, we define the unsigned margin of a classification function $f \in \mathcal{F}$ over an example $\mathbf{x} \in \mathcal{X}$ [2] as

$$m_f(\mathbf{x}) = |\rho_f(\mathbf{x}, +1)|.$$

In the multi-class classification case, $\mathcal{Y} = \{1, \ldots, K\}$, the unsigned margin [24] is defined as

$$m_f(\mathbf{x}) = \sum_{y \in \mathcal{Y}} f(\mathbf{x}, y) \rho_f(\mathbf{x}, y).$$

The maximization of the unsigned margin tends to find a decision boundary that passes through low density regions and hence follows the low density separation assumption.

2.2 Self-training: the idea

Self-training, also known as decision-directed or self-taught learning machine, is one of the earliest approach in semi-supervised learning [70] that has risen in popularity in recent years.

To determine the decision boundary on low density regions, the idea behind self-training algorithms is to consider a pseudo-labeling strategy for assigning pseudo-labels to the examples of $X_{\mathcal{U}}$. This strategy can be characterized by a function, called pseudo-labeler [87, 25]:

$$\Phi_\ell : \mathcal{X} \times \mathcal{F} \to \mathcal{X} \times \mathcal{Y}.$$

We denote $\tilde{y}$ the pseudo-label of an unlabeled $\mathbf{x} \in X_{\mathcal{U}}$ for a score function $f \in \mathcal{F}$ assigned by Φ_ℓ and $X_{\bar{\mathcal{U}}}$ the set of pseudo-labeled examples.

The self-learning strategy is an iterative wrapper algorithm that starts by learning a supervised classifier on the labeled training set S. Then, at each iteration, the current classifier selects a part of the unlabeled data, $X_{\bar{\mathcal{U}}}$, and assigns pseudo-labels to them using the classifier's predictions [95].

These pseudo-labeled unlabeled examples are removed from $X_{\mathcal{U}}$ and a new supervised classifier is trained over $S \cup X_{\bar{\mathcal{U}}}$, by considering these pseudo-labeled unlabeled data as additional labeled examples. To do so, the classifier

Algorithm 1. Self-Training

```
Input : $S = (\mathbf{x}_i, y_i)_{1 \leqslant i \leqslant m}$, $X_{\mathcal{U}} = (\mathbf{x}_i)_{m+1 \leqslant i \leqslant m+u}$.
$k \leftarrow 0$, $X_{\bar{\mathcal{U}}} \leftarrow \emptyset$.
repeat
    Train $f^{(k)}$ on $S \cup X_{\bar{\mathcal{U}}}$
    $\Pi_k \leftarrow \{\Phi_\ell(\mathbf{x}, f^{(k)}), \mathbf{x} \in X_{\mathcal{U}}\}$   ▷ Pseudo-labeling
    $X_{\bar{\mathcal{U}}} \leftarrow X_{\bar{\mathcal{U}}} \cup \Pi_k$
    $X_{\mathcal{U}} \leftarrow X_{\mathcal{U}} \setminus \{\mathbf{x} \mid (\mathbf{x}, \tilde{y}) \in \Pi_k\}$
    $k \leftarrow k + 1$
until $X_{\mathcal{U}} = \emptyset \vee \Pi_k = \emptyset$
Output : $f^{(k)}, X_{\mathcal{U}}$, $X_{\bar{\mathcal{U}}}$
```

$h \in \mathcal{H}$ that is learned at the current iteration is the one that minimizes a regularized empirical loss over S and $X_{\bar{\mathcal{U}}}$:

$$\frac{1}{m} \sum_{(\mathbf{x},y)\in S} \ell(h(\mathbf{x}), y) + \frac{\gamma}{|X_{\bar{\mathcal{U}}}|} \sum_{(\mathbf{x},\tilde{y})\in X_{\bar{\mathcal{U}}}} \ell(h(\mathbf{x}), \tilde{y}) + \lambda \|h\|^2$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ is an instantaneous loss most often chosen to be the cross-entropy loss, γ is a hyperparameter for controlling the impact of pseudo-labeled data in learning, and λ is the regularization hyperparameter. This process of pseudo-labeling and learning a new classifier continues until the unlabeled set $X_{\mathcal{U}}$ is empty or there is no more unlabeled data to pseudo-label. The pseudo-code of the self-training algorithm is shown in Algorithm 1.

2.3 Pseudo-labeling strategies

Pseudo-labeling is a crucial component of self-training methods, where a portion of unlabeled data is selected for pseudo-labeling to avoid overfitting to the initial classifier. In the following we will review different techniques that have been proposed to determine the subset of examples to pseudo-label, each with its strengths and weaknesses.

2.3.1 Threshold-Based Methods

A classical assumption, that stems from the low density separation hypothesis, is to suppose that the classifier learned at each step makes the majority of its mistakes on observations close to the decision boundary.

In the case of binary classification, preliminary research suggested to assign pseudo-labels only to unlabeled observations for which the current classifier is the most confident [81]. Hence, considering thresholds θ^- and θ^+ defined for respectively the negative and the positive classes, the pseudo-labeler assigns a pseudo-label $\tilde{y}$ to an instance $\mathbf{x} \in X_{\mathcal{U}}$ such that:

$$\tilde{y} = \begin{cases} +1, & \text{if } f(\mathbf{x}, +1) \geqslant \theta^+, \\ -1, & \text{if } f(\mathbf{x}, -1) \leqslant \theta^-, \end{cases} \tag{1}$$

and $\Phi_\ell(\mathbf{x}, f) = (\mathbf{x}, \tilde{y})$. An unlabeled example $\mathbf{x}$ that does not satisfy the conditions equation 1 is not pseudo-labeled; i.e. $\Phi_\ell(\mathbf{x}, f) = \emptyset$.

Intuitively, thresholds should be set to high absolute values as pseudo-labeling examples with low confidence would increase chances of assigning wrong labels. However, thresholds of very high value imply excessive trust in the confidence measure underlying the model, which, in reality, can be biased due to the small labeled sample size. Using several iterations makes also the situation more intricate as at every iteration the optimal threshold might be different.

One way to select the thresholds is to set them equal to the average of respectively positive and negative predictions [81]. In this line, and in the context of multi-class classification [44] used Neural Networks as the supervised classifier and chose the most confident class to infer pseudo-labels for unlabeled data using the current model' outputs. The pseudo-labeled examples were then added to the labeled training set and treated similarly as labeled examples.

These methods are simple and intuitive and they are effective in leveraging high-confidence predictions. However, they present a risk of assigning wrong labels if thresholds are not set appropriately and excessive trust in the confidence measure can be biased due to small labeled sample size.

2.3.2 Proportion-Based Methods

[100] adapted the idea of [81] for multi-class classification by not choosing thresholds but rather fixing a proportion p of the most confident unlabeled data to be pseudo-labeled and then increasing this proportion at each iteration of the algorithm until $p = 0.5$ was reached. Following this idea, [11] revisited the concept of pseudo-labeling by discussing the iterative process of

assigning pseudo-labels to unlabeled data and emphasized the resilience of pseudo-labeling to out-of-distribution samples.

These methods are adaptive to the confidence level of the model and reduce the risk of overfitting to low-confidence predictions. However, they may include incorrect labels if the initial proportion is too high and generally require careful tuning of the proportion parameter.

2.3.3 Curriculum Learning-Based Methods

These methods use curriculum learning to pseudo-label easy-to-learn observations before moving on to more complex ones. [95] proposed an adaptation of curriculum learning to pseudo-labeling, which entails in learning a model using easy-to-learn observations before moving on to more complex ones. The principle is that at the step k of the algorithm, the pseudo-labeler selects unlabeled examples having predictions that are in the $(1-\alpha_k)^{th}$ percentile of the distribution of the maximum probability predictions assumed to follow a Pareto distribution, and where $\alpha_k \in [0,1]$ is an hyperparameter that varies from 0 to 1 in increments of 0.2. Later on, [21] proposed a method that combines CLIP's text-image alignment and SAM's mask generation capabilities, along with a multi-source curriculum learning strategy to address noise and excessive focus issues, gradually improving semantic alignment and segmentation precision.

These approaches gradually increase the complexity of pseudo-labeled data and reduce the risk of incorrect labels in early iterations. However they require careful tuning of the hyperparameter α_k and may not be effective if the distribution of predictions is not well-behaved.

2.3.4 Majority vote classifiers

Considering the distribution of predictions over unlabeled data, and the majority vote classifiers, such as Random Forest or Adaboost [68], it is possible to automatically choose a threshold for pseudo-labeling.

Formally, the learning of a majority vote classifier with partially labeled data can be defined as follows.

After observing the training set $S \cup X_{\bar{\mathcal{U}}}$, the task of the learner is to choose a posterior distribution Q over a set of hypothesis $\mathcal{H}$ such that the Q-weighted majority vote classifier B_Q defined by:

$$\forall \mathbf{x} \in \mathcal{X}, B_Q(\mathbf{x}) = \text{argmax}_{y \in \mathcal{Y}} \mathbb{E}_{h \sim Q} \left[\mathbb{1}_{h(\mathbf{x})=y} \right],$$

will have the smallest possible risk on examples of $X_{\mathcal{U}}$. The associated Gibbs classifier, G_Q, is defined as the random choice of a classifier h according to the distribution Q, and its error over an unlabeled set $X_{\mathcal{U}}$ is given by:

$$\hat{R}_u(G_Q) = \frac{1}{u} \sum_{\mathbf{x}' \in X_{\mathcal{U}}} \mathbb{E}_{h \sim Q}[\mathbb{1}_{h(\mathbf{x}') \neq y'}],$$

where, for every unlabeled example $\mathbf{x}' \in X_{\mathcal{U}}$ we refer to y' as its true unknown class label. For binary and multi-class classification respectively, [2] and [24] showed that a tight upper-bound on the Gibbs classifier's risk that holds with high probability over the random choice of S and $X_{\mathcal{U}}$, guarantees a tight bound on the error of the Bayes classifier over the unlabeled training set:

$$\hat{R}_u(B_Q) = \frac{1}{u} \sum_{\mathbf{x}' \in X_{\mathcal{U}}} \mathbb{1}_{B_Q(\mathbf{x}') \neq y'}.$$

This bound is mainly based on the distribution of predictions over unlabeled data and the derivations can be extended to bound the risk of voted majority classifiers having margins greater than a threshold θ, $\hat{R}_{u \wedge \theta}(B_Q)$, defined as:

$$\hat{R}_{u \wedge \theta}(B_Q) = \frac{1}{u} \sum_{\mathbf{x}' \in X_{\mathcal{U}}} \mathbb{1}_{B_Q(\mathbf{x}') \neq y' \wedge m_{B_Q}(\mathbf{x}') > \theta},$$

with a slight abuse of notation for m_{B_Q}. One of the practical aspects that arises from this result is the possibility to specify a threshold θ which minimizes an upper-bound of the conditional risk of a voted majority classifier B_Q over the unlabeled training set, $X_{\mathcal{U}}$, defined as:

$$\hat{R}_{u|\theta}(B_Q) = \frac{\hat{R}_{u \wedge \theta}(B_Q)}{\frac{1}{u} \sum_{\mathbf{x} \in X_{\mathcal{U}}} \mathbb{1}_{m_{B_Q}(\mathbf{x}) \geqslant \theta}},$$

where the denominator is the proportion of the unlabeled examples with the confidence higher than the threshold θ, and the numerator is the joint Bayes risk on $X_{\mathcal{U}}$. Thus, the criterion can be interpreted as a trade-off between the number of examples going to be pseudo-labeled and the error they induce. Furthermore, these bounds are shown to be tight in the case where the majority vote classifier makes its error mostly on low margin regions [25]. [24] demonstrated that this technique outperforms conventional fixed-threshold pseudo-labeling strategies on different multi-class classification problems.

2.3.5 Adaptive Thresholding Methods

[15] highlighted two major issues with self-training: the snowball effects of cascading pseudo-labeling mistakes and random sampling of tiny samples (called data bias). The authors suggest two-phase solutions to address these problems for image classification using deep learning. First, they proposed a classification head to separate the creation and use of pseudo labels in order to reduce training errors. An additional head is utilized to receive the pseudo-labels and carry out training on unlabeled data while the default head is used for classification and pseudo-labeling.

To address the limitations of confidence thresholding in self-training, [87] proposed an approach by analyzing the relationship between the ideal threshold and the model's learning status. This approach adapts the confidence threshold self-adaptively based on the model's progress. Additionally, it includes a self-adaptive class fairness regularization penalty to encourage diverse predictions during early training stages. In this line, [16] proposed a method to address the quantity-quality trade-off in pseudo-labeling. This approach aims to maintain both the quantity and quality of pseudo-labels during training, thereby effectively utilizing unlabeled data. This approach employs a truncated Gaussian function to weight samples based on their confidence, serving as a softer alternative to the confidence threshold. Furthermore, the authors introduced a uniform alignment approach to improve the utilization of weakly-learned classes.

Dynamic adjustment of thresholds reduces the risk of incorrect labels, and encourages diverse predictions during early training stages. However, it requires additional computational overhead to adapt thresholds and may introduce complexity in the training process.

2.4 Self-training with two classifiers

In the wake of works utilizing only a single classifier in self-training algorithms, new studies have been proposed with the use of two classifiers, where each model learns on the output of the other [90, 17, 39]. Most of these techniques are based on the idea of consensus in predictions between two classifiers and were inspired by the seminal work of [9] who proposed the co-training algorithm.

In co-training, examples are defined by two modalities that are comparable but not entirely correlated. Each view of an example is expected to contain complementary information about the data and if there are enough labeled training data, each of them is supposed to be sufficient for learning. The main principle is to learn a classifier on each view, taking initially the available labeled examples as the training set. Then, one of the classifiers assigns pseudo-labels to unlabeled data, and the other one uses them to retrain the model by including them into its training set. At each iteration, the classifiers alternately switch their roles, thereby co-training each other. As for self-training algorithms with a single classifier, this procedure continues until there are no more unlabeled instances to be pseudo-labeled. In practice, several studies artificially generated the two modalities for classification problems where examples are mono-viewed and described by a vector representation. These approaches create the two modalities out of one by selecting at random the set of features that should correspond to each modality from the initial set of features; and their efficiency was empirically proved on various applications [79].

Co-training and self-training share several fundamental principles. Both methods involve iteratively assigning pseudo-labels to unlabeled data and using these pseudo-labeled examples to improve the model. In self-training, a single model assigns pseudo-labels based on its confidence. In co-training, two or more models collaborate to assign pseudo-labels, with each model providing pseudo-labels for the data on which it is most confident. Furthermore, both approaches leverage unlabeled data to enhance the performance of the model(s) involved in training. In self-training, the model uses its own predictions to generate pseudo-labels for unlabeled data. In co-training, multiple models use their complementary strengths to generate pseudo-labels for unlabeled data, thereby improving the overall model performance by relying on confidence measures to select which unlabeled examples to pseudo-label. In self-training, the model's confidence in its predictions determines which examples are pseudo-labeled. In co-training, each model's confidence in its predictions determines which examples it will pseudo-label for the other model(s). Finally, both approaches involve iterative improvement of the model(s) through the incorporation of pseudo-labeled data. In self-training, the model is retrained with the augmented dataset that includes pseudo-labeled examples. In co-training, the models are retrained with the augmented dataset that includes pseudo-labeled examples provided by the other model(s).

Without splitting the input feature set, [17] proposed Cross Pseudo Supervision for semantic segmentation in images. This method employs two

	Base classifier		Classification		Threshold		Noise
	Single	Double	Binary	Multi-class	Fixed	Optimized	Account
Scudder [70]	✓	–	✓	–	✓	–	–
Joachims. [36]	✓	–	✓	–	✓	–	–
Amini et al. [2]	✓	–	✓	–	–	✓	–
Hadjadj et al. [32]	✓	–	✓	–	–	✓	✓
Tur et al. [81]	✓	–	–	✓	✓	–	–
Cascante et al. [11]	✓	–	–	✓	✓	–	–
Chen et al. [15]	✓	–	–	✓	✓	–	✓
Feofanov et al. [24]	✓	–	–	✓	–	✓	–
Zhang et al. [95]	✓	–	–	✓	–	✓	–
Chen et al. [16]	✓	–	–	✓	–	✓	–
Wang et al. [87]	✓	–	–	✓	–	✓	–
Blum et al. [9]	–	✓	✓	–	✓	–	✓
Tarvainen and Valpola [79]	–	✓	–	✓	✓	–	–
Xie et al. [90]	–	✓	–	✓	✓	–	–
Karamanolakis et al. [39]	–	✓	–	✓	✓	–	–
Chen et al. [17]	–	✓	–	✓	✓	–	–
Ghiasi et al. [30]	–	✓	–	✓	✓	–	–
Du et al. [22]	–	✓	–	✓	✓	–	–

Table 1: A summary of principal self-training algorithms, based on pseudo-labeling with one or two classifiers, introduced in Section 2.3 and 2.4.

neural networks as supervised classifiers having the same images as inputs. Each neural-network is learned at every mini-batch by considering the pseudo-labels generated by the other network for unlabeled instances as ground-truths. In multi-task learning, [30] proposed to independently train specialized teachers using labeled datasets. These teachers then label an unlabeled dataset, creating a multitask pseudo-labeled dataset. Subsequently, a student model is trained using the pseudo-labeled dataset, employing multi-task learning to learn from various datasets and tasks simultaneously. Finally, the feature representations of the student model are evaluated across six vision tasks, including image recognition, to assess its perWe ance and generalization capabilities.

The learnability of co-training was studied under the PAC framework [82], which also accounts for noise in the class labels of unlabeled examples caused by pseudo-labeling. The injection of noisy labels in the pseudo-labeling step is in fact inherent to any self-training algorithm. Taking into account noisy labels in training a model was first considered in supervised learning, following the paradigm of learning with an imperfect supervisor in which training data contains an unknown portion of imperfect labels [33]. Most of these studies tackle this problem from an algorithmic point of view, employing regularization or estimating mislabeling errors by modeling the transition probability between noisy and true labels [41, 60].

Table 1 summarizes the main self-training approaches presented so far by emphasizing their key aspects.

2.5 Self-training under Domain Shift

Recently, self-training has expanded its scope beyond semi-supervised learning and has found extensive application to the learning problems where available data is subject to a distribution shift. In unsupervised domain adaptation, where the objective is to transfer knowledge from a labeled source domain to an unlabeled target one, self-training become a popular alternative to discrepancy minimization methods [28]. In this case, self-training aims to progressively correct the domain shift by including more and more pseudo-labeled target examples to the source training set. This is particularly relevant for gradual domain adaptation, where unlabeled instances from intermediate domains are available [72].

When intermediate domains are not given, it is important to ensure that pseudo-labeled target examples are reliable and are not biased towards the source data. While [100] approached this issue by carefully choosing a pseudo-labeling policy, [67] learn a representation via a tri-training scheme, in which the student is trained on target data pseudo-labeled by agreement of two teachers. [49] alternate between two gradient steps: (1) to train a source classification head that generates pseudo-labels, (2) to train a target classification head using pseudo-labeled data under the constraint that it predicts well on source data.

As the discrepancy between the source and the target can be large, the prediction confidence may exhibit a strong bias failing to distinguish between correct and wrong pseudo-labels. Therefore, several works focus specifically on model calibration and uncertainty estimation including the Monte-Carlo dropout [55], and prediction agreement of diversified linear heads [58].

2.6 Theoretical studies

Several studies have recently looked into the theoretical properties of self-training algorithms.

In this line, [88] suggest a new concept of expansion defined as the quantity of data dispersion in an example's neighbor, where the term neighbor refers to adding adversarial perturbations [53] or augmentations [77] to the example. The study establishes distributional guarantees of self-training when the label distribution meets such expansion properties and classes are suitably separated according to neighbors. The study generates finite sample bounds for Deep Neural Networks (DNNs) by combining generalization bounds with DNN generalization bounds. Experiments with a Generative Adversarial Network (GAN) are also used to verify the expansion assumption.

[26] examine a self-training algorithm with linear models for the binary classification using gradient-based optimization of the cross-entropy loss after supervised learning with a small number of samples. The classifier is a mixture model with concentration and anti-concentration properties. The authors show that utilizing $O(d/\epsilon^2)$ unlabeled observations in the self learning algorithm, with d the number of input variables, suffices to achieve the classification error of the Bayes-optimal classifier up to an ϵ error if the initial pseudo-labeling strategy has a classification error smaller than an absolute constant C_{err}. Furthermore, the authors demonstrate that a constant number of labeled examples is sufficient for optimal performance in a self-training algorithm by demonstrating that using only $O(d)$ labeled examples, the standard gradient descent algorithm can learn a pseudo-labeling strategy with a classification error no more than C_{err}.

[96] study the generalization ability of self-training in the case where the base classifier is a two-layer neural network with the second layer weights all fixed to one, and assuming that the ground truth is realizable, the labels are observed without noise, and the labeled and unlabeled instances are drawn from two isotropic Gaussian distributions. The authors show that, given some plausible assumptions about the initial point and the amount of unlabeled training examples, the algorithm converges to the ground truth with fewer observations than needed when no unlabeled data is provided. The reader can refer to [98] for a broader context. [96] extend their main result to a more general setting, where it is shown that the model still converges towards a given convex combination of the ground truth and the initial point, and is guaranteed to outperform the initial supervised model, without fixing any requirement on the number of labeled training examples.

[32] propose a first bound over the misclassification error of a self-training method which utilizes half-spaces as the base classifier in the case where class labels of examples are supposed to be corrupted by a Massart noise model. Under this assumption, it is shown that the use of unlabeled data in the

proposed self-training algorithm does not degrade the performance of the first half-space trained over the labeled training data.

[78] study the identifiability of self-training approaches. In addressing the bias in the conventional risk estimator, the proposed method, named Inverse Propensity Weighting, involves assigning weights to examples based on the inverse of their propensity scores-representing the probability of a class label being observed. The study introduces two estimators for the missing data mechanism, one of which is derived through the maximization of the observed likelihood. Furthermore, a likelihood ratio test is suggested to evaluate the informativeness of the labels, determining whether they exhibit non-random missing patterns.

Some other works studied self-training from a theoretical perspective when a distribution shift takes place. [18] proves that self-training can help to avoid spurious features, while [42] derived an upper-bound on the error of self-training in the case of gradual shifts.

3 Related and unrelated approaches

In semi-supervised learning, there are two main other areas of research that are related to self-training. The first, known as transductive learning, is based on the low density separation assumption and tends to give class labels for only the set of unlabeled training samples. The second method, referred to as consistency learning, uses classifier predictions over unlabeled data as a confidence indicator and constrains model outputs to be comparable for similar unlabeled examples without assigning pseudo-labels.

In this section, we also go a bit further, and introduce different context where self-training has been used and extended. First, we present self-supervised learning, which, despite its similar name with self-training, is an entirely separate technique that employs unlabeled data to train or pre-train a model. Finally, we introduce reinforced self-training, that merges elements of reinforcement learning with self-training principles by integrating a scoring function based on a learned reward model and employing offline reinforcement learning objectives for model fine-tuning.

3.1 Transductive learning

The goal of transductive learning, as previously stated, is to assign pseudolabels to samples from an unlabeled training set, $X_{\mathcal{U}}$. As this set is finite, the considered function class $\mathcal{F}$, for finding the transductive prediction function, is also finite. $\mathcal{F}$ can be defined using a nested structure according to the structural risk minimization principle, $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \ldots \subseteq \mathcal{F}$ [84]. Transductive techniques often employ the distribution of unsigned margins of unlabeled examples to guide the search for a prediction function, limiting it to following the low density separation assumption in order to find the best function class among the current ones.

Transductive approaches also assume that the function class's structure should reflect prior knowledge of the learning problem at hand, and that it should be built in such a way that the correct prediction of class labels of labeled and unlabeled training examples is contained in a function class $\mathcal{F}_j$ of small size with a high probability. In particular, the Transductive Support Vector Machines (TSVM) [36] developed for the binary case is based on this paradigm. The approach looks for the optimal hyperplane in a feature space that separates the best labeled examples while avoiding high density areas. TSVM does this by building a structure on a function class $\mathcal{F}$ and sorting the outputs of unlabeled samples by their margins. The solutions to the associated optimization problem are the pseudo-labels of unlabeled examples for which the resulting hyperplane separates the examples of both labeled and unlabeled training sets with the largest margin.

[73] extended this idea to the multi-class classification case with Neural Networks. Similar to TSVM, class labels of unlabeled examples are treated as variables, and the algorithm tries to determine their optimal values, along with the optimal NNs parameter set get by minimizing a cross-entropy loss estimated over both labeled and unlabeled training sets through an iterative training process. The authors employ the MinMax Feature regularization to constrain the neural network to learn features of same-class images to be close, and features of different classes to be separated by a preset margin, in order to overcome incorrect label estimations on outliers and noisy samples.

Transductive learning is particularly useful in the context of proprietary and closed APIs in natural language processing, including few-shot classification. [20] proposed a scenario where pre-trained model embeddings are served through a gated API with compute-cost and data-privacy constraints, and introduces transductive inference as a solution. The authors present a new parameter-free transductive regularizer based on the Fisher-Rao loss, which fully utilizes unlabeled data without sharing labels with the API provider, and demonstrate its superiority through extensive experiments. More recently, [54] studied the problem of adversarially robust learning in the transductive setting. For classes with bounded VC dimension, the authors propose a simple transductive learner that correctly labels adversarially perturbed test examples with a robust error rate linear in the VC dimension, providing an exponential improvement over the best-known inductive setting bounds, albeit with a more restrictive notion of optimal robust error.

3.2 Consistency-based approaches

Early studies in this line, see for example [99] for binary classification, were proposed to learn a single classifier defined from a scoring function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ penalized for quick changes in its predictions. The similarity matrix $\mathbf{W} = [W_{ij}]_{\substack{1\leqslant i\leqslant u\\ 1\leqslant j\leqslant u}}$, constructed over the unlabeled training data, is used to measure the similarity between instances. The penalization is mostly expressed as a regularization term in the learning objective. As an example, adapting the work of [99] to multi-class classification, the penalization term can be written as:

$$\Omega_{\mathbf{W}}(X_{\mathcal{U}}) = \sum_{i,j=1}^{u} W_{ij} \|f(\mathbf{x}_{m+i}, .) - f(\mathbf{x}_{m+j}, .)\|^2$$

where for a given example $\mathbf{x}$, $f(\mathbf{x}, .) = (f(\mathbf{x}, k))_{k\in\mathcal{Y}}$ is the vector class predictions of f. In terms of learning, $\Omega_{\mathbf{W}}$ can be seen as a regularization term, constraining the model to have the same predictions on similar unlabeled instances.

Other types of penalization have been studied in the literature. [52] suggested an approach that partitions partially labeled data and then uses labeled training samples to identify dense clusters having predominant classes with a fraction of non-predominant classes below a given threshold extending earlier results on supervised classification [37]. In this situation, the proposed penalization term measures the learner's inability to predict the predominant classes of the identified clusters which in turn constrains the supervised classifier to be consistent with the structure of the dense clusters.

In this line, [65] consider non-decomposable metrics with consistency regularization by giving a cost-sensitive framework that consists of minimizing a cost-sensitive error on pseudo labels and consistency regularization. They demonstrate theoretically that they can build classifiers that can maximize the required non-decomposable measure more effectively than the original model used to produce pseudo-labels under comparable data distribution assumptions.

Without explicitly stating a penalization term, consistency learning was extended to cases with two classifiers. The Mean-Teacher approach [79] is perhaps one of the earliest popular techniques that have been proposed in this context. This method employs Neural Networks (NNs) as supervised classifiers, and it is based on the assumption that two close models with the same input should make the same prediction. One of the models is called the teacher, while the other is referred to as the student. These two NN models are structurally identical, and their weights are related in that the teacher's weights are an exponential moving average [43] of the student' weights. In this scenario, the student model is the only one that is trained over the labeled training set, and the consistency loss is computed between the teacher's probability distribution prediction and the student's one using the mean square error or the Kullback-Leibler divergence.

More recently, [22] provide a two-stage method to reduce label propagation errors; where in the first phase, the gradients of the student loss are computed and utilized to update the teacher. In the second stage, the teacher assigns pseudo-labels which are then utilized to train the current student.

Also, [27] proposed a new contrastive semi-supervised learning approach for data-efficient language modeling. The method uses hard prompts for sentence representation and integrates prompt-based pseudo labeling with a mask language model through a contrast loss, which pulls together similar samples and pushes apart dissimilar ones. Additionally, mask consistency training is employed to align word predictions from weak and strong augmentations. This approach improves model generalization and leverages unlabeled data for few-shot text classification,

3.3 Self-supervised Learning

Although similar in names, self-training is a completely different approach than self-supervised learning which has demonstrated encouraging results and has become an active area of research [59].

In self-supervised learning, a model acquires the ability to make predictions regarding various facets of its input data, all without the necessity of explicit labeled training data. Rather than depending on labeled data, self-supervised learning harnesses the inherent structure present in the input data and autonomously generates guidance to train the model. This procedure involves the formulation of a pretext task, also referred to as a proxy task, wherein the model is trained to make predictions concerning a specific relationship inherent in the data. For instance, in the domain of computer vision, a pretext task might involve rotating images within a predefined range of angles, followed by training a supervised model to predict these angles.

Once the model has undergone training on the pretext task, the knowledge it has gained in this process can be applied to downstream tasks that do require labeled data. Consequently, by learning from extensive amounts of unlabeled data, self-supervised learning empowers the acquisition of robust data representations, capitalizing on the abundant, freely available unlabeled data resources.

Common approaches in self-supervised learning include predicting the order of shuffled image patches or their orientation [91], reconstructing corrupted images [23], filling in missing words in a sentence [8], or predicting future frames in a video sequence [69]. These pretext tasks encourage the model to capture meaningful representations of the input data, which can then be used for various downstream tasks, such as image classification, object detection, or natural language processing.

3.4 Reinforced self-training

A recent innovative approach, called Reinforced self-training (ReST) has emerged, particularly notable for its application in conditional language modeling [31, 74]. This approach operates through two distinct loops: the inner loop, called “Improve”, which concentrates on refining the policy using a fixed dataset, and the outer loop, called “Grow”, which involves expanding the dataset by sampling from the most recent policy.

In the domain of conditional language modeling, ReST follows a systematic sequence of steps. Initially, during the Grow phase, the language model policy, originally a supervised policy, generates multiple output predictions for each context, thereby enriching the training dataset. Subsequently, in the Improve stage, the expanded dataset undergoes ranking and filtering using a scoring function. The language model then undergoes fine-tuning on the refined dataset using an offline reinforcement learning objective, with the potential for repeating this process with an increasing filtering threshold. The resultant policy from this iterative process is subsequently employed in the following Grow phase.

ReST may find niche suitability in specific applications or scenarios where reinforcement learning principles enhance model performance through learned reward signals. In contrast, classical self-training techniques possess a broader applicability and have been employed across a wide spectrum of semi-supervised learning tasks without necessitating reinforcement learning frameworks.

4 Applications

In this section, we will concentrate on the most popular applications where self-training was employed, although this technique may be extended and used to a variety of additional machine learning tasks. The goal of our presentation here is not to be thorough, but rather to focus on the main features of self-training that were used in the literature among the selected applications.

4.1 Natural Language Processing

Co-training is perhaps one of the preliminary self-training techniques which was applied to web pages classification [9]. In the paper, the content of a web page has been divided into two sets of words: those that appear on the page and those that appear in hyperlinks pointing to the page. The main hypothesis here is that each of the set of words contain sufficient information for the classification task and that there are enough labeled data to efficiently learn two supervised classifiers. Both theoretical and empirical studies of co-training show that if examples have two redundant but not entirely correlated views, then unlabeled data may be used to augment the original labeled training data to find more robust classifiers. However, the drawback of this strategy is that in general, text data is mono-view. For bag-of-word representation of texts, a solution was to split the set of words in two random sets, considered as two distinct views of a text [56], as mentioned in Section 2.4. However, this idea cannot be generalized to sequential models that could be used as base classifiers in co-training.

Other current self-training techniques in NLP are mostly built on the concept of co-training and employ two base classifiers that are learned over each other's predictions. In this line, [89] proposed a Named Entity Recognition (NER) strategy that consists in automatically detecting and classifying named entities, with a first NER model trained on labeled training data serving as a teacher to predict the probability distribution of entity labels for each token in the unlabeled set. The pseudo-labeled data with such soft labels are then used to train a student NER model for the unlabeled set and the process of pseudo-labeling and training is repeated until convergence as in co-training. For the task of Relation Extraction (RE) which consists in obtaining a predefined semantic relation between two entities in a given sentence, [93] proposed an approach which classifies the pseudo-labeled instances generated from a teacher into confident, ambiguous and hard sets. In the training of the student model, the confident and ambiguous instances are subsequently interpreted as positive and set-negatives observations, respectively. Based on these ideas, [50] proposed several self-training methods for NLP, demonstrating that neural language models can be enhanced through synthetic data generation strategies such as pseudo-labeling, prototypical label embeddings, and synthetic text generation. These methods were shown to significantly improve performance in tasks like dialog response retrieval, dialog policy learning, and question answering, achieving state-of-the-art results and reducing dependence on human annotation.

Lately [97] proposed a self-training approach that reduces the annotation cost for structured label spaces using active learning. The method leverages partial annotation by selecting only the most informative substructures for annotation and incorporates the model's automatic predictions as pseudo-labels for unannotated sub-structures. By employing an error estimator to adaptively decide the partial selection ratio, the approach effectively combines partial annotation with self-training, reducing annotation cost across four structured prediction tasks compared to strong full annotation baselines.

4.2 Computer Vision

As in NLP, the two variants of self-training with one or two classifiers, mainly referred to as student and teacher in the literature, are mainly considered for image classification. Most recent approaches use neural networks as base classifiers and rely on these models' ability to learn efficient representations of images, proposing various strategies to either improve the representation or reduce the effect of noise injection during the pseudo-labeling phase of self-training.

The most common strategy with student and teacher base classifiers is arguably the one proposed by [90], in which an EfficientNet model trained on labeled ImageNet images is used as a teacher to create pseudo labels on unlabeled ones. A larger EfficientNet is subsequently employed as a student model, being trained on a mix of labeled and pseudo-labeled images. This training involves altering the input images using various techniques like dropout, stochastic depth, and data augmentation. The objective is for the model to learn a representation of images that remains consistent despite these alterations. This procedure is repeated by reversing the roles of the student and the teacher. The input of the teacher model is not altered throughout the training process. The main motivation advanced is to ensure that the pseudo labels be as accurate as possible. Empirical evidence from various image collections demonstrates the effectiveness of this strategy.

[77] proposed a self-training approach called FixMatch that combines consistency regularization with a confidence-based mechanism to select high-confidence pseudo-labeled examples for training. The algorithm applies to the same image two different data augmentations procedures, called weak (flip-and-shift) and strong (more heavy distortions) augmentations.

As in the previous case, these perturbations helps to increase diversity and improve the model's robustness on the unlabeled images. The authors introduce a consistency loss term that encourages the consistency between the model's hard output of the weakly-augmented version and the model's soft output of the strongly-augmented version of the same unlabeled image. They demonstrate that the model learns to provide more trustworthy and accurate results by minimizing the discrepancy between these predictions. In order to decrease the influence of possibly inaccurate pseudo-labels on the learning process, the loss is evaluated only on those unlabeled data from the batch that have the confidence higher than a fixed threshold.

This idea has then been adapted to various correlated tasks, including object detection, image segmentation [19], remote sensing [35] and video anomaly detection [51], among others. [15] proposed an improvement of FixMatch by introducing two novel features. First of all, they introduce a separate classification head that is used to assign pseudo-labels and trained using labeled data only in order to avoid possible label noise from wrong pseudo-labels. Secondly, they improve the feature learning by introducing an adversarial classification head whose goal is to approximate the worst possible error on unlabeled data. All these approaches employ a constant predefined threshold across all classes to choose unlabeled data for training, disregarding varying learning conditions and complexities among different classes.

To tackle this concern, [95] introduced a curriculum learning technique to utilize unlabeled data based on the model's learning progress. The essence of this strategy involves dynamically adapting thresholds for distinct classes during each time step, enabling the inclusion of insightful unlabeled data and their corresponding pseudo-labels. This approach has been successfully applied to many domains, including object detection [46], medical image classification [62], human action recognition [85] and facial expression identification [71].

4.3 Knowledge-driven applications

Through the incorporation of domain expertise, recent studies have developed more sophisticated self-training systems that reduce label noise in the pseudo-labeling phase across diverse applications. In the subsequent sections, we will consider advances made in this context in the domains of speech recognition, anomaly detection, genomics and proteomics.

4.3.1 Speech Recognition

Newly developed methods have introduced filtering mechanisms that are congruent with domain knowledge for end-to-end speech recognition. These mechanisms establish rules that assess pseudo-labels using criteria specific to the domain. For example by using filters to verify if certain phonetic patterns that are common in the domain, are present in the pseudo-labels [29]. Similar techniques incorporate phonetic information relevant to the domain to validate pseudo-labels. In these approaches, incorrectly labeled examples that violate phonetic constraints are discarded from training the model [48].

Other approaches integrate domain-specific language models in the the pseudo-label generation process in order to ensure that the generated labels adhere to the linguistic nuances and terminologies of the domain. In this line, [38] introduced a self-training approach, with one base classifier combined with a language model for pseudo-labeling. Their approach involves implementing tailored filtering methods designed to address common errors arising from sequence-to-sequence models, alongside an inventive ensemble technique for enhancing the breadth of pseudo-label variations. Furthermore [76] presents a self-training approach for automatic speech recognition in low-resource settings, specifically focusing on the Punjabi language. The proposed method generates highly accurate pseudo-labels for unlabeled Punjabi speech, resulting in a significant improvement in word error rate compared to state-of-the-art approaches.

As in image classification, alternative methods for speech recognition apply data-augmentation techniques, tailored to the unique aspects of the domain, to enhance the robustness of the model's predictions and consequently the quality of pseudo-labels. In this sense, [5] employed a text-to-speech system to generate audio training data from text-only sources.

4.3.2 Anomaly Detection

Leveraging domain knowledge to mitigate label noise in pseudo-labels within self-training approaches has also been considered in anomaly detection. In this case, the understanding of the anomaly patterns and characteristics specific to the domain are incorporated in the model. In this regard, [64] identified common anomaly types, their features and potential sources of noise and performed time domain analysis. Also, [80] proposed a method for video anomaly detection that employs an unsupervised approach using isolation trees and deep features. The method generates initial anomaly and dynamicity scores, which are refined using a cross-branch feed-forward network based on the I3D architecture. It then incorporates a self-training strategy to iteratively improve the model's performance by pseudolabeling and retraining on the most confident samples, achieving competitive accuracy on popular datasets like UCF-Crime, CCTV-Fights, and UBI-Fights. This approach combines appearance and motion information to enhance the quality of anomaly evidence, making it more practical for real-world applications where annotation is burdensome. Alternate strategies focus on simulating anomalies within the unlabeled dataset using domain knowledge. This aids the model in learning from a broad spectrum of anomalies, mitigating the potential of becoming overly specialized in a particular anomaly type [63]. [47] also proposed a weakly supervised Video Anomaly Detection approach using Multi-Sequence Learning (MSL) and a hinge-based MSL ranking loss to reduce selection errors by optimizing sequences of snippets. The method employs a Transformer-based MSL network to learn video-level and snippet-level anomaly scores, refining them through a self-training strategy that gradually reduces sequence length. This approach achieves significant improvements on datasets like ShanghaiTech, UCF-Crime, and XD-Violence.

4.3.3 Genomics and proteomics

Furthermore, datasets in the field of genomics and proteomics encompass a variety of characteristics including gene expression levels, epigenetic markers, and genetic variants. These characteristics have been shown to increase the effectiveness of features used in self-training approaches, together with the selection of important features and their physiologically coherent transformation.

[10] incorporated biological context into feature engineering that integrate unsupervised modeling of datasets relating to human disease with the supervised component that concentrated on training with mouse data. In this context, [66] amalgamated expression data from three distinct humanized mouse models that were subjected to live attenuated yellow fever vaccine challenges in self-training with different base classifiers. The results of this study show that self-training coupled with NRG-HIS/Fluc mice exhibited the most favorable outcomes across the tested human cohorts.

[45] proposed a self-training subspace clustering algorithm based on adaptive confidence for gene expression data. The proposed algorithm enhances the discriminative property of gene expression data using low-rank representation with distance penalty and introduces a semi-supervised clustering objective function with label confidence. To mitigate the negative impact of mislabeled data, an adaptive adjustment strategy based on the gravitational search algorithm is employed.

[34] applied domain-specific quality control steps to clean and pre-process the data. This included filtering out low-quality samples, normalizing data to account for technical biases, and addressing batch effects that can introduce noise. By doing so, they ensured that the unlabeled data that is feed into the self-training pipeline is as accurate as possible. [13] utilized reference databases and annotation resources related to genomics. These resources provide information about genes, functional elements, pathways, and biological processes. Incorporating this information into the pseudo-labeling process has been shown to lead to more accurate predictions by aligning them with known biological knowledge. [94] applied network analysis techniques to identify interactions between genes and proteins. The authors demonstrated that Pathway enrichment analysis can help identify genes that are functionally related and likely to be co-regulated. This information has been shown to guide the self-training process to produce more coherent and biologically plausible pseudo-labels.

General observations The key observations made in these applications reveal that, in pseudo-labeling, employing fixed thresholds often yields suboptimal outcomes, underscoring the importance of dynamic thresholding for optimal results. Furthermore, enhancing pseudo-label noise improves both generalization and class differentiation. In Appendix A, we will show the impact of dynamic thresholding on pseudo-labeling across general benchmarks and examine the noise considerations in two image classification collections studied in [15].

5 Conclusion, Limitations and Perspectives

In this survey, we provided an overview of self-training approaches for semi-supervised learning that have received increasing attention in recent years.

First, we discussed the various strategies for selecting unlabeled samples for pseudo-labeling that have been proposed. We emphasized the significance of considering margin distributions across unlabeled data as a pivotal factor in the development of these strategies. Next, we provided an overview of the diverse variants of self-training explored in the literature, along with relevant approaches. Furthermore, we examined recent theoretical advancements in this research domain and outlined the principal characteristics of self-training employed in several widely recognized applications. Lastly, we explored the impact of fundamental aspects of self-training on a range of benchmark datasets.

Limitations Despite the promising results and widespread adoption of self-training, several limitations remain. One significant challenge is the sensitivity of self-training methods to the quality of pseudo-labels. Incorrect or noisy pseudo-labels can propagate errors through the training process, leading to suboptimal performance. Additionally, the effectiveness of self-training can be highly dependent on the initial labeled dataset. If the initial dataset is not representative or is too small, the self-training process may not yield significant improvements.

Another limitation is the computational cost associated with iterative pseudo-labeling and retraining. Self-training methods often require multiple iterations, which can be time-consuming and resource-intensive, especially for large-scale datasets. Furthermore, the selection of appropriate thresholds for pseudo-labeling remains a critical and often challenging task, as it requires a delicate balance between the quantity and quality of pseudo-labels.

Future Work To address these limitations, future research should focus on developing more robust and efficient self-training algorithms. This could involve exploring adaptive thresholding techniques that dynamically adjust the confidence thresholds based on the model's learning status. Additionally, incorporating uncertainty quantification methods to estimate the reliability of pseudo-labels could help mitigate the impact of noisy labels.

Another promising direction is the integration of self-training with other semi-supervised learning techniques, such as consistency regularization and multi-view training, to leverage their complementary strengths. Furthermore, extending self-training to new domains, such as medical imaging and industrial time-series data, could open up new avenues for practical applications.

While the self-training approach is currently in widespread use, there are extensive opportunities for future research. Presently, the majority of studies have concentrated on perturbation-based deep learning, particularly in the domains of visual, text, and audio applications. However, there exist numerous other domains, such as industrial time-series or medical data, where the application of self-training could prove highly beneficial.

Also, recent research emphasizes the potential of exploring self-training methods from a theoretical standpoint, particularly in addressing the challenge of training a final classifier on data with noisy labels [32]. It has also been demonstrated that accurately estimating the confidence of pseudo-labels is crucial for effective self-training [58]. Therefore, theoretically establishing the correlation between performance and the level of uncertainty in pseudo-labeling could be a valuable direction for future research, especially in analyzing self-training within the context of learning problems affected by domain shifts.

References

[1] M.-R. Amini and P. Gallinari. Semi-supervised logistic regression. In European Conference in Artificial Intelligence - ECAI, pages 390–394, 2002.

[2] M.-R. Amini, F. Laviolette, and N. Usunier. A transductive bound for the voted classifier with an application to semi-supervised learning. In Advances in Neural Information Processing Systems 21 - NeurIPS, pages 65–72, 2008.

[3] M.-R. Amini and N. Usunier. Learning with Partially Labeled and Interdependent Data. Springer, 2015.

[4] M.-F. Balcan and A. Blum. An augmented PAC model for semi-supervised learning. In Semi-Supervised Learning, pages 396–419. MIT, 2006.

[5] M. Bartelds, N. San, B. McDonnell, D. Jurafsky, and M. Wieling. Making more of little data: Improving low-resource automatic speech recognition using data augmentation. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, pages 715–729, 2023.

[6] M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. Machine Learning, 56(1-3):209–239, 2004.

[7] S. Ben-David, T. Lu, and D. Pál. Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning. In Conference on Learning Theory - COLT, pages 33–44, 2008.

[8] D. Biswas, S. Linzbach, D. Dimitrov, H. Jabeen, and S. Dietze. Broadening BERT vocabulary for knowledge graph construction using wikipedia2vec. In Joint proceedings of the 1st workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM) and the 2nd challenge on Language Models for Knowledge Base Construction (LM-KBC), 2023.

[9] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In Conference on Learning Theory - COLT, pages 92–100, 1998.

[10] D. K. Brubaker, E. A Proctor, K. M. Haigis, and D.A. Lauffenburger. Computational translation of genomic responses from experimental model systems to humans. PLoS computational biology, 15(1):e1006286, 2019.

[11] P. Cascante-Bonilla, F. Tan, Y. Qi, and V. Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In AAAI Conference on Artificial Intelligence, pages 6912–6920, 2021.

[12] V. Castelli and T.M. Cover. On the exponential value of labeled samples. Pattern Recognition Letters, 16(1):105–111, 1995.

[13] K.-L. Chan, R. Rosli, T. V. Tatarinova, M. Hogan, M. Firdaus-Raih, and L. Eng-Ti Low. Seqping: gene prediction pipeline for plant genomes using self-training gene models and transcriptomic data. BMC Bioinformatics, 17, 2017.

[14] O. Chapelle, B. Schölkopf, and A. Zien. Semi-Supervised Learning. The MIT Press, 2010.

[15] B. Chen, J. Jiang, X. Wang, P. Wan, J. Wang, and M. Long. Debiased self-training for semi-supervised learning. In Advances in Neural Information Processing Systems - NeurIPS, pages 32424–32437, 2022.

[16] H. Chen, R. Tao, Y. Fan, Y. Wang, J. Wang, B. Schiele, X. Xie, B. Raj, and M. Savvides. Softmatch: Addressing the quantity-quality tradeoff in semi-supervised learning. In The Eleventh International Conference on Learning Representations ICLR, 2023.

[17] X. Chen, Y. Yuan, G. Zeng, and J. Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In Conference on Computer Vision and Pattern Recognition - CVPR, pages 2613–2622, 2021.

[18] Y. Chen, C. Wei, A. Kumar, and T. Ma. Self-training avoids using spurious features under domain shift. In Advances in Neural Information Processing Systems - NeurIPS, pages 21061–21071, 2020.

[19] T. Cheng, X. Wang, S. Chen, Q. Zhang, and W. Liu. Boxteacher: Exploring high-quality pseudo labels for weakly supervised instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 3145–3154, June 2023.

[20] P. Colombo, V. Pellegrain, M. Boudiaf, M. Tami, V. Storchan, I. Ayed, and P. Piantanida. Transductive learning for textual few-shot classification in API-based embedding models. In Proceedings of the Confer-

ence on Empirical Methods in Natural Language Processing EMNLP, pages 4214–4231, 2023.

[21] Qiyuan Dai and Sibei Yang. Curriculum point prompting for weakly-supervised referring image segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13711–13722, June 2024.

[22] Y. Du, Y. Shen, H. Wang, J. Fei, W. Li, L. Wu, R. Zhao, Z. Fu, and Q. Liu. Learning from future: A novel self-training framework for semantic segmentation. In Advances in Neural Information Processing Systems - NeurIPS, pages 4749–4761, 2022.

[23] Y. Fang, L. Dong, H. Bao, X. Wang, and F. Wei. Corrupted image modeling for self-supervised visual pre-training. In The 11^{th} International Conference on Learning Representations - ICLR, 2023.

[24] V. Feofanov, E. Devijver, and M.-R. Amini. Transductive bounds for the multi-class majority vote classifier. In AAAI Conference on Artificial Intelligence, pages 3566–3573, 2019.

[25] V. Feofanov, E. Devijver, and M.-R. Amini. Multi-class probabilistic bounds for majority vote classifiers with partially labeled data. Journal of Machine Learning Research, 25(104):1–47, 2024.

[26] S. Frei, D. Zou, Z. Chen, and Q. Gu. Self-training converts weak learners to strong learners in mixture models. In International Conference on Artificial Intelligence and Statistics - AISTATS, pages 8003–8021, 2022.

[27] R. Fu, C. Chen, S. Yan, X. Wang, and H. Chen. Consistency-based semi-supervised learning for oriented object detection. Knowledge-Based Systems, 304:112534, 2024.

[28] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky. Domain-adversarial training of neural networks. Journal of Machine Learning Research, 17(59):1–35, 2016.

[29] M. Gheini, T. Likhomanenko, M. Sperber, and H. Setiawan. Joint speech transcription and translation: Pseudo-labeling with out-of-distribution data. In Findings of the Association for Computational Linguistics: ACL 2023, pages 7637–7650, 2023.

[30] G. Ghiasi, B.Zoph, E. D. Cubuk, Q. V. Le, and T.-Y. Lin. Multi-task self-training for learning general representations. In International Conference on Computer Vision - ICCV, pages 8836–8845, 2021.

[31] C. Gulcehre, T. Le Paine, S. Srinivasan, K. Konyushkova, L. Weerts, A. Sharma, A. Siddhant, A. Ahern, M. Wang, C. Gu, W. Macherey, A. Doucet, O. Firat, and N. de Freitas. Reinforced self-training (rest) for language modeling, 2023.

[32] L. Hadjadj, M.-R. Amini, and S. Louhichi. Self-training of halfspaces with generalization guarantees under massart mislabeling noise model. In International Joint Conference on Artificial Intelligence - IJCAI, pages 3777–3785, 2023.

[33] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In Advances in Neural Information Processing Systems - NeurIPS, pages 3124–3132, 2018.

[34] K. Huang, C. Xiao, L. M. Glass, C. W. Critchlow, G. Gibson, and J. Sun. Machine learning applications for therapeutic tasks with genomics data. Patterns, 2(10), 2021.

[35] W. Huang, Y. Shi, Z. Xiong, Q. Wang, and X. X. Zhu. Semi-supervised bidirectional alignment for remote sensing cross-domain scene classification. ISPRS Journal of Photogrammetry and Remote Sensing, 195:192–203, 2023.

[36] T. Joachims. Transductive inference for text classification using support vector machines. In International Conference on Machine Learning - ICML, pages 200–209, 1999.

[37] B. Joshi, M.-R. Amini, I. Partalas, F. Iutzeler, and Y. Maximov. Aggressive sampling for multi-class to binary reduction with applications to text classification. In Advances in Neural Information Processing Systems - NeurIPS, pages 4235–4243, 2017.

[38] J. Kahn, A. Lee, and A. Hannun. Self-training for end-to-end speech recognition. In International Conference on Acoustics, Speech and Signal Processing - ICASSP, pages 7084–7088, 2020.

[39] G. Karamanolakis, S. Mukherjee, G. Zheng, and A. Awadallah. Self-training with weak supervision. In North American Conference on Chinese Linguistics - NAACL, pages 845–863, 2021.

[40] D.P. Kingma, S. Mohamed, D. Jimenez, and M. Welling. Semi-supervised learning with deep generative models. In Advances in Neural Information Processing Systems - NeurIPS, 2014.

[41] A. Krithara, M.-R. Amini, J.-M. Renders, and C. Goutte. Semi-supervised document classification with a mislabeling error model. In Advances in Information Retrieval , 30th European Conference on IR Research, ECIR, pages 370–381, 2008.

[42] A. Kumar, T. Ma, and P. Liang. Understanding self-training for gradual domain adaptation. In International Conference on Machine Learning, pages 5468–5479. PMLR, 2020.

[43] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In International Conference on Learning Representations - ICLR, 2017.

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

[45] Dan Li, Hongnan Liang, Pan Qin, and Jia Wang. A self-training subspace clustering algorithm based on adaptive confidence for gene expression data. Frontiers in Genetics, 14, 2023.

[46] G. Li, X. Li, Y. Wang, W. Yichao, D. Liang, and S. Zhang. Dtg-ssod: Dense teacher guidance for semi-supervised object detection. In Advances in Neural Information Processing Systems - NeurIPS, pages 8840–8852, 2022.

[47] S. Li, Fang Liu, and Licheng Jiao. Self-training multi-sequence learning with transformer for weakly supervised video anomaly detection. In AAAI Conference on Artificial Intelligence, 2022.

[48] S. Ling, C. Shen, M. Cai, and Z. Ma. Improving pseudo-label training for end-to-end speech recognition using gradient mask. In International Conference on Acoustics, Speech and Signal Processing - ICASSP, pages 8397–8401, 2022.

[49] H. Liu, J. Wang, and M. Long. Cycle self-training for domain adaptation. Advances in Neural Information Processing Systems - NeurIPS, 34:22968–22981, 2021.

[50] H. Luo. Self-Training for Natural Language Processing. PhD thesis, Massachusetts Institute of Technology, USA, 2022.

[51] H. Lv, Z. Yue, Q. Sun, B. Luo, Z. Cui, and H. Zhang. Unbiased multiple instance learning for weakly supervised video anomaly detection. In Conference on Computer Vision and Pattern Recognition - CVPR, pages 8022–8031, 2023.

[52] Y. Maximov, M.-R. Amini, and Z. Harchaoui. Rademacher complexity bounds for a penalized multi-class semi-supervised algorithm. Journal of Artificial Intelligence Research, 61:761–786, 2018.

[53] T. Miyato, S. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. Pattern Analysis and Machine Intelligence, 41(8):1979–1993, 2019.

[54] O. Montasser, S. Hanneke, and N. Srebro. Transductive robust learning guarantees. In Proceedings of The 25th International Conference on Artificial Intelligence and Statistics - AISTATS, pages 11461–11471, 2022.

[55] S. Mukherjee and A. Awadallah. Uncertainty-aware self-training for few-shot text classification. Advances in Neural Information Processing Systems - NeurIPS, 33:21199–21212, 2020.

[56] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In Proceedings of the International Conference on Information and Knowledge Management - CIKM, pages 86–93, 2000.

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

[58] A. Odonnat, V. Feofanov, and I. Redko. Leveraging ensemble diversity for robust self-training in the presence of sample selection bias. International Conference on Artificial Intelligence and Statistics - AISTATS, 2024.

[59] U. Ozbulak, H. J. Lee, B. Boga, E. Timothy Anzaku, H. Park, A. Van Messem, W. De Neve, and J. Vankerschaver. Know your self-supervised learning: A survey on image-based generative and discriminative training. Transactions on Machine Learning Research, 2023.

[60] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In Conference on Computer Vision and Pattern Recognition - CVPR, pages 2233–2241, 2017.

[61] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg. Scikit-learn: Machine learning in python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[62] Z. Peng, D. Zhang, S. Tian, W. Wu, L. Yu, S. Zhou, and S. Huang. Faxmatch: Multi-curriculum pseudo-labeling for semi-supervised medical image classification. Medical Physics, 50(5):3210–3222, 2023.

[63] C. Qiu. Self-Supervised Anomaly Detection with Neural Transformations. doctoral thesis, Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau, 2023.

[64] C. Qu, H. Zhang, R. Zhang, S. Zou, L. Huang, and H. Li. Multiclass anomaly detection of bridge monitoring data with data migration between different bridges for balancing data. Applied Sciences, 13(13), 2023.

[65] H. Rangwani, S. Ramasubramanian, S. Takemori, K. Takashi, Y. Umeda, and V.B. Radhakrishnan. Cost-sensitive self-training for optimizing non-decomposable metrics. In Advances in Neural Information Processing Systems - NeurIPS, pages 26994–27007, 2022.

[66] D. Ravinder. Using machine learning to increase the predictive value of humanized mouse models for the human immune response to YFV-17D. Masters of engineering in biomedical engineering, Massachusetts Institute of Technology, 2021.

[67] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. In International Conference on Machine Learning, pages 2988–2997. PMLR, 2017.

[68] R.E. Schapire, Y. Freund, P. Barlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In International Conference on Machine Learning - ICML, pages 322–330, 1997.

[69] M. C. Schiappa, Y. S. Rawat, and M. Shah. Self-supervised learning for videos: A survey. ACM Computing Surveys, 2022.

[70] H. Scudder. Adaptive communication receivers. IEEE Transactions on Information Theory, 11(2):167–174, 1965.

[71] N. Shabbir and R. Kumar Rout. Fgbcnn: A unified bilinear architecture for learning a fine-grained feature representation in facial expression recognition. Image and Vision Computing, 137, 2023.

[72] L. Shi and W. Liu. Adversarial self-training improves robustness and generalization for gradual domain adaptation. Advances in Neural Information Processing Systems - NeurIPS, 36, 2024.

[73] W. Shi, Y. Gong, C. Ding, Z. Ma, X. Tao, and N. Zheng. Transductive semi-supervised deep learning using min-max features. In European Conference on Computer Vision - ECCV, pages 311–327, 2018.

[74] A. Singh, J. D Co-Reyes, R. Agarwal, A. Anand, P. Patil, X. Garcia, P. J. Liu, J. Harrison, J. Lee, K. Xu, A. T. Parisi, A. Kumar, A; A Alemi, A. Rizkowsky, A. Nova, B. Adlam, B. Bohnet, G.F. Elsayed, H. Sedghi, I. Mordatch, I.Simpson, I. Gur, J. Snoek, J. Pennington, J. Hron, K. Kenealy, K. Swersky, K. Mahajan, L. A Culp, L. Xiao, M. Bileschi, N. Constant, R. Novak, R. Liu, T. Warkentin, Y. Bansal, E. Dyer, B. Neyshabur, J. Sohl-Dickstein, and N. Fiedel. Beyond human data: Scaling self-training for problem-solving with language models. Transactions on Machine Learning Research, 2024.

[75] A. Singh, R. Nowak, and J. Zhu. Unlabeled data: Now it helps, now it doesn't. In Advances in Neural Information Processing Systems - NeurIPS, pages 513–521, 2008.

[76] S. Singh, F. Hou, and R. Wang. A novel self-training approach for low-resource speech recognition. In 24th Annual Conference of the International Speech Communication Association, Interspeech, pages 1588–1592, 2023.

[77] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. Raffel, E. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In Advances in Neural Information Processing Systems - NeurIPS, pages 596–608, 2020.

[78] A. Sportisse, H. Schmutz, O. Humbert, C. Bouveyron, and P.-A. Mattei. Are labels informative in semi-supervised learning? Estimating and leveraging the missing-data mechanism. In Proceedings of the 40th International Conference on Machine Learning - ICML, pages 32521–32539, 2023.

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

[80] K. Vijay Thakare, Y. Raghuwanshi, D. P. Dogra, H. Choi, and I.-J. Kim. Dyannet: A scene dynamicity guided self-trained video anomaly detection network. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision - WACV, pages 5541–5550, January 2023.

[81] G. Tür, D. Tür, and R.-E. Schapire. Combining active and semi-supervised learning for spoken language understanding. Speech Communication, 45(2):171–186, 2005.

[82] L.G. Valiant. A theory of the learnable. Communications of the ACM, 27(11):1134–1142, 1984.

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

[84] V. Vapnik. Statistical Learning Theory. Wiley-Interscience, 1998.

[85] Chenxi Wang, Jingzhou Luo, Xing Luo, Haoran Qi, and Zhi Jin. V-dixmatch: A semi-supervised learning method for human action recognition in night video sensing. IEEE Sensors Journal, 24(14):21857–21865, 2024.

[86] Y. Wang, H. Chen, Y. Fan, W. Sun, R. Tao, W. Hou, R. Wang, L. Yang, Z. Zhou, L.-Z. Guo, H. Qi, Z. Wu, Y.-F. Li, S. Nakamura, W. Ye, M. Savvides, B. Raj, T. Shinozaki, B. Schiele, J. Wang, X. Xie, and Y. Zhang. USB: A unified semi-supervised learning benchmark for classification. In Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, pages 3938–3961, 2022.

[87] Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj, B. Schiele, and X. Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. In The Eleventh International Conference on Learning Representations - ICLR, 2023.

[88] C. Wei, K. Shen, Y. Chen, and T. Ma. Theoretical analysis of self-training with deep networks on unlabeled data. In International Conference on Learning Representations - ICLR, 2021.

[89] Q. Wu, Z. Lin, B. F. Karlsson, J.-G. Lou, and B. Huang. Single-/multi-source cross-lingual ner via teacher-student learning on unlabeled data in target language. In Annual Conference of the Association for Computational Linguistics - ACL, pages 6505–6514, 2020.

[90] Q. Xie, M.-T. Luong, E. H. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. In Conference on Computer Vision and Pattern Recognition - CVPR, pages 10684–10695, 2020.

[91] M. Xu, S. Yoon, A. Fuentes, and D. S. Park. A comprehensive survey of image augmentation techniques for deep learning. Pattern Recognition, 137:109347, 2023.

[92] X. Yang, Z. Song, I. King, and Z. Xu. A survey on deep semi-supervised learning. IEEE Transactions on Knowledge and Data Engineering, 35(9):8934–8954, 2023.

[93] J. Yu, X. Wang, J. Zhao, C. Yang, and W. Chen. STAD: self-training with ambiguous data for low-resource relation extraction. In Proceedings of the 29th International Conference on Computational Linguistics - COLING, pages 2044–2054, 2022.

[94] Z. Yu, Y. Su, Y. Lu, Y. Yang, F. Wang, S. Zhang, Y. Chang, K.-C. Wong, and X. Li. Topological identification and interpretation for single-cell gene regulation elucidation across multiple platforms using scMGCA. Nature Communications, 14(400), 2023.

[95] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In Advances in Neural Information Processing Systems - NeurIPS, pages 18408–18419, 2021.

[96] S. Zhang, M. Wang, S. Liu, P.-Y. Chen, and J. Xiong. How unlabeled data improve generalization in self-training? a one-hidden-layer theoretical analysis. In International Conference on Learning Representations - ICLR, 2022.

[97] Z. Zhang, E. Strubell, and E. H. Hovy. Data-efficient active learning for structured prediction with partial annotation and self-training. In Findings of the Association for Computational Linguistics: EMNLP, pages 12991–13008. Association for Computational Linguistics, 2023.

[98] K. Zhong, Z. Song, P. Jain, P. L. Bartlett, and I. S. Dhillon. Recovery guarantees for one-hidden-layer neural networks. In International Conference on Machine Learning - ICML, pages 4140–4149, 2017.

[99] X. Zhu, Z. Ghahramani, and J.D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In International Conference on Machine Learning - ICML, pages 912–919, 2003.

[100] Y. Zou, Z. Yu, B. Kumar, and J. Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In European conference on computer vision - ECCV, pages 289–305, 2018.

A Empirical Study

In this section, we will evaluate the effectiveness and performance of the self-training algorithm. This assessment will focus on two key features discussed in the preceding sections: the handling of noise in the pseudo-labeling phase and the impact of threshold selection. Our primary focus will be on scenarios with sufficient labeled training data, allowing for the development of an initial supervised complex model, and scenarios with severely limited labeled training data, where using complex baseline classifiers like deep learning models is not feasible.

A.1 Noise Account

We first consider the case where the initial labeled training set allows to train deep neural networks and examine the effects of taking into account noise in the pseudo-labeling process along with the dynamic selection of the threshold on CIFAR-10 and CIFAR-100 [86]. Both datasets contain 32x32 pixel RGB images belonging to respectively 10 and 100 classes; 50000 examples are used for training and 10000 samples for test.

We consider the debiased self-training approach (DST) [15] to address the presence of noise in pseudo-labeling, in conjunction with the FlexMatch method [95] for the dynamic threshold determination in pseudo-labeling. As outlined in Section 2.3, DST involves training a dedicated head on pseudo-labeled examples, allowing the model to implicitly capture and account for noise inherent in the pseudo-labels.

For FlexMatch, we followed the same experimental protocol than [95]. In this case, Wide ResNet (WRN) was used as the base classifier in self-training. Parameter learning was accomplished using stochastic gradient descent (SGD) with a momentum coefficient of 0.9. The initial rate was set to $\eta_0 = 0.03$ with a cosine learning rate decay schedule as $\eta = \eta_0 \cos(7\pi t/16T)$, where t denotes the current training step and T is the total training step set at 2^{20}. Additionally, exponential moving averaging with a momentum of 0.999 was implemented and the batch size for labeled data was fixed to 64. For DST, we used the code made available by the authors[1].

We compared FlexMatch with and without the DST approach denoted respectively by FM and FM+DST. We also compared self-training with WRN

[1] https://github.com/thuml/Debiased-Self-Training

trained in fully supervised manner. Each experiment was repeated 5 times by changing the seed at each time. Figure 2 presents the average accuracy of different models on the test set for the same number of initial labeled training samples per class within the set $\{4, 10, 20, 50\}$ for both datasets. In both datasets, considering label noise within pseudo-labels leads to improved performance, with the improvement being more pronounced in the case of CIFAR-100.

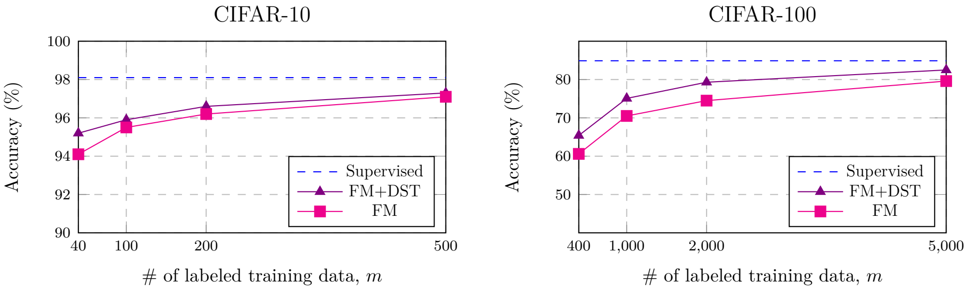

Figure 2: Comparisons in terms of Accuracy on CIFAR-10 and CIFAR-100 for a varying number of labeled training data. "Supervised" refers to the fully supervised learning ($m = 50000$, $u = 0$).

In CIFAR-100, classes are structured into 20 superclasses, each comprising 5 related classes, addressing noise in this more complex task aids in class differentiation and enhances the model's ability to generalize. It is worth noting that with a greater number of initial labeled training examples, the gap between the FM and FM+DST approaches narrows, as the model becomes more proficient with the increased labeled data and makes fewer errors in pseudo-labeling.

A.2 The impact of threshold selection

We now study the effect of selecting automatically the threshold for pseudo-labeling on 9 publicly available data sets proposed for semi-supervised learning[2]. The characteristics of these datasets are presented in Table 2. It is worth noting that certain datasets contain only a limited number of labeled training examples, comprising just a few hundred instances and accounting

[2] https://archive.ics.uci.edu/ml/index.php

for less than 1% of the total training examples. This condition underscores the suitability of employing complex base classifiers.

Data set	# of labeled examples	# of unlabeled examples	Dimension	# of classes
	m	u	d	K
Vowel	99	891	10	11
Protein	129	951	77	8
PageBlocks	1094	4379	10	5
Isolet	389	7408	617	26
HAR	102	10197	561	6
Pendigits	109	10883	16	10
Letter	400	19600	16	26
Fashion	175	69825	784	10
MNIST	175	69825	784	10

Table 2: Characteristics of data sets used in our experiments, d and K correspond to respectively the dimension of the input space and the number of classes.

In the experimentation, Random Forest was employed instead using the scikit-learn implementation [61] with 200 trees of maximum depth while leaving other parameters at their default values. The primary objective was to assess and contrast the classifier's performance in two scenarios: the supervised scenario (denoted by RF) and the self-training scenario where pseudo-labeling is automatically conducted following the approach introduced by [24][3] (denoted by PL*). Additionally, we investigated the impact of setting the pseudo-labeling threshold at predefined values from the set $\theta \in \{0.5, 0.7, 0.9\}$ (denoted by PL$_\theta$).

The automatic pseudo-labeling strategy selects the threshold which minimizes the bound of the error of the Random Forest classifier over the unlabeled training samples.

Results are resumed in Table 3. Experiments are repeated 20 times by choosing randomly the labeled training examples, and $^\downarrow$ indicates that performance is statistically worse than the best result, shown in bold, according to the Wilcoxon rank-sum test.

These results suggest that the effectiveness of self-training heavily relies on the method used to determine the pseudo-labeling threshold. When the threshold is automatically determined, self-training (i.e. PL*) can perform

[3] https://github.com/vfeofanov/trans-bounds-maj-vote

Data set	RF	PL$_{\theta=0.5}$	PL$_{\theta=0.7}$	PL$_{\theta=0.9}$	PL*
Vowel	$.586 \pm .028$	$.489^{\downarrow}\pm .016$	$.531^{\downarrow}\pm .034$	$.576^{\downarrow}\pm .028$	$.586 \pm .026$
Protein	$.764^{\downarrow}\pm .032$	$.653^{\downarrow}\pm .024$	$.687^{\downarrow} \pm .036$	$.724^{\downarrow} \pm .018$	$.781 \pm .034$
PageBlocks	$.965 \pm .003$	$.931^{\downarrow}\pm .003$	$.964 \pm .004$	$.965 \pm .002$	$.966 \pm .002$
Isolet	$.854^{\downarrow} \pm .016$	$.648^{\downarrow} \pm .018$	$.7^{\downarrow} \pm .04$	$.861^{\downarrow} \pm .08$	$.875 \pm .014$
HAR	$.851 \pm .024$	$.76^{\downarrow}\pm .04$	$.81^{\downarrow}\pm .041$	$.823^{\downarrow}\pm .035$	$.854 \pm .026$
Pendigits	$.863^{\downarrow}\pm .022$	$.825^{\downarrow}\pm .022$	$.839^{\downarrow}\pm .036$	$.845^{\downarrow}\pm .024$	$.884 \pm .022$
Letter	$.711 \pm .011$	$.062^{\downarrow}\pm .011$	$.651^{\downarrow}\pm .015$	$.673\ ^{\downarrow}\pm .015$	$.717 \pm .013$
Fashion	$.718 \pm .022$	$.625^{\downarrow}\pm .014$	$.64^{\downarrow}\pm .04$	$.68^{\downarrow}\pm .014$	$.723 \pm .023$
MNIST	$.798^{\downarrow}\pm .015$	$.665^{\downarrow}\pm .012$	$.705^{\downarrow}\pm .055$	$.823^{\downarrow}\pm .045$	$.857 \pm .013$

Table 3: Classification performance using the accuracy score on 9 publicly available data set. Best results are shown in bold and the sign $^{\downarrow}$ shows if the performance is statistically worse than the best result on the level 0.01 of significance.

competitively, indicating that this approach has the potential to improve results compared to the supervised RF.

However, when a fixed threshold is applied, self-training tends to yield inferior results compared to the supervised learning approach. This suggests that an arbitrarily chosen threshold might not effectively capture the underlying patterns in the data for the pseudo-labeling process, leading to suboptimal performance.

Moreover, when the threshold is too low as for $\theta \in \{0.5, 0.7\}$, pseudo-labeling is likely to produce label noise and degrade the performance of self-training with respect to the supervised RF classifier in all cases. When the threshold it is too high (i.e. $\theta = 0.9$), self-training becomes competitive compared to RF on Isolet and MNIST, but the quantity of pseudo-labeled unlabeled examples seems not to be sufficient to learn efficiently.

In summary, the findings emphasize the importance of a dynamic and adaptive threshold selection mechanism when implementing self-training.