Cost-Sensitive Self-Training for Optimizing Non-Decomposable Metrics

Anonymous Author(s)
Affiliation

Abstract

Self-training-based semi-supervised learning algorithms have enabled the learning of highly accurate deep neural networks, using only a fraction of labeled data. However, the majority of work on self-training has focused on the objective of improving the accuracy whereas practical machine learning systems can have complex goals i.e. maximizing the minimum of recall across classes that are non-decomposable. In this work, we introduce the Cost-Sensitive Self-Training (`CSST`) framework which generalizes the self-training-based methods for optimizing non-decomposable metrics. We prove that our framework is able to better optimize the desired non-decomposable metric, under similar data distribution assumptions made for the analysis of self-training. Using the proposed `CSST` framework we obtain practical self-training methods (for both vision and NLP tasks) for optimizing different non-decomposable metrics using deep neural networks. Our results demonstrate that `CSST` achieves an improvement over the state-of-the-art in majority of the cases across datasets and objectives.

1 Introduction

In recent years, semi-supervised learning algorithms are increasingly being used for training deep neural networks [3, 9, 31, 37]. These algorithms lead to accurate models by leveraging the unlabeled data in addition to the limited labeled data present. For example, it's possible to obtain a model with minimal accuracy degradation ($\leq 1\%$) using 5% of labeled data with semi-supervised algorithms compared to supervised models trained using 100% labeled data [31]. Hence, the development of these algorithms has resulted in a vast reduction in the requirement of expensive labeled data.

Self-training is one of the major paradigms for semi-supervised learning. It involves obtaining targets (*e.g.* pseudo-labels) from a network from the unlabeled data, and using them to train the network further. The modern self-training methods also utilize additional regularizers that enforce prediction consistency across input transformations (e.g., adversarial perturbations [18], augmentations [36, 31], etc.), enabling them to achieve high performance using only a tiny fraction of labeled data. Currently, the enhanced variants of self-training with consistency regularization [40, 25] are among the state-of-the-art (SOTA) methods for semi-supervised learning.

Despite the popularity of self-training methods, most of the works [36, 1, 31] have focused on the objective of improving prediction accuracy. However, there are nuanced objectives in real-world based on the application requirements. Examples include minimizing the worst-case recall [19] used for federated learning, classifier coverage for minority classes for ensuring fairness [6], etc. These objectives are complex and cannot be expressed just by using a loss function on the prediction of a single input (i.e., non-decomposable). There has been a considerable effort in optimizing non-decomposable objectives for different supervised machine learning models [20, 30]. However, as supervision can be expensive in this work we aim to answer the following novel question: *Can we optimize non-decomposable objectives using self-training-based methods developed for semi-supervised learning?*

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

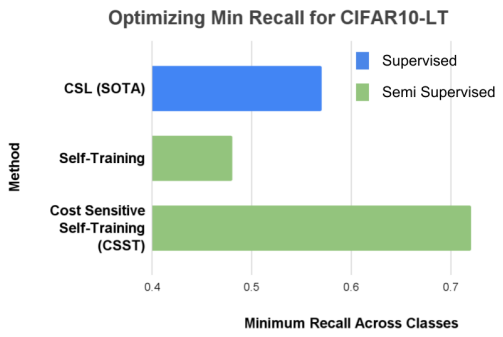


Figure 1: We show a comparison of the SOTA CSL [20] method with the Self-training-based Semi-Supervised methods, for optimizing the minimum recall objective on the CIFAR10-LT dataset. Our proposed CSST framework produces significant gains in the desired metric leveraging additional unlabeled data through our proposed weighted novel consistency regularizer and thresholding mechanism.

We first demonstrate that vanilla self-training methods (e.g., FixMatch [31], UDA [36], etc.) can produce unsatisfactory results for other non-decomposable metrics (Fig. 1). We then generalize the Cost-Sensitive Loss for Self-Training by introducing a novel weighted consistency regularizer, for a particular non-decomposable metric. Further, for training neural networks we introduce appropriate loss functions and pseudo label selection (thresholding) mechanisms considering the non-decomposable metric we aim to optimize. We also prove that we can achieve better performance on desired non-decomposable metric through our framework utilizing self-training, under similar assumptions on data distributions as made for theoretical analysis of self-training [35].We demonstrate the practical application by optimizing various non-decomposable metrics by plugging existing methods (*e.g.* FixMatch [31] etc.) into our framework. Our framework leads to a significant average improvement in desired metric of minimizing worst-case recall while maintaining similar accuracy (Fig. 1). We will release our code and framework publicly to promote reproducible research.

In summary: **a)** we introduce a Cost-Sensitive Self-Training (CSST) framework for optimizing non-decomposable metrics that utilizes unlabeled data in addition to labeled data. (Sec. 4) **b)** we provably demonstrate that our CSST framework can leverage unlabeled data to achieve better performance over baseline on desired non-decomposable metric (Sec. 3) **c)** we show that by combining CSST with self-training frameworks (*e.g.* FixMatch [31], UDA [36] etc.) leads to effective optimization of non-decomposable metrics, resulting in significant improvement over vanilla baselines. (Sec. 5)

2 Preliminaries

2.1 Non-Decomposable Objectives and Reduction to Cost-Sensitive Learning

We consider K-class classification problem with an instance space $\mathcal{X}$ and the set of labels $\mathcal{Y} = [K]$. The data distribution on $\mathcal{X} \times [K]$ is denoted by D. For $i \in [K]$, we denote by π_i the class prior $\mathbf{P}(y = i)$. Notations commonly used across paper are in Table 4 present in Appendix.

For a classifier $F : \mathcal{X} \to [K]$, we define confusion matrix $\mathbf{C}[F] \in \mathbb{R}^{K \times K}$ by $C_{ij}[F] = \mathbf{E}_{(x,y)\sim D}\left[\mathbb{1}(y = i, F(x) = j)\right]$. Many objectives for classification can be defined as functions of entries of confusion matrices such as class-coverage, recall and accuracy to name a few. We introduce more complex metrics, which are of practical importance in the case of imbalanced distributions [4]. (Tab. 1) A classifier often tends to suffer low recalls on tail (minority) classes in such cases. Therefore, one may want to maximize the worst case recall.

$$\max_{F} \min_{i \in [K]} \mathrm{rec}_i[F].$$

Table 1: Metrics defined using entries of a confusion matrix.

Metric	Definition
Recall ($\mathrm{rec}_i[F]$)	$\frac{C_{i,i}[F]}{\sum_j C_{i,j}[F]}$
Coverage ($\mathrm{cov}_i[F]$)	$\sum_j C_{j,i}[F]$
Precision ($\mathrm{prec}_i[F]$)	$\frac{C_{i,i}[F]}{\sum_k C_{k,i}[F]}$
Worst Case Recall	$\min_i \frac{C_{i,i}[F]}{\sum_j C_{i,j}[F]}$
Mean Accuracy	$\frac{\sum_i C_{i,i}[F]}{\sum_{i,j} C_{i,j}[F]}$

Similarly, on long-tailed datasets, the tail classes suffer from low coverage, lower than their respective priors. An interesting objective in such circumstances is to maximise the mean recall, subject to the coverage

being within a given margin.

$$\max_F \frac{1}{K} \sum_{i \in [K]} \mathrm{rec}_i[F] \quad \text{s.t. } \mathrm{cov}_j[F] \geq \frac{0.95}{K}, \forall j \in [K]. \tag{1}$$

These metrics except for accuracy are **non-decomposable**, i.e., one cannot compute these metrics by simply calculating the average of scores on individual examples. Optimising for these metrics can be regarded as instances of cost-sensitive learning (CSL). More specifically, *optimization problems of the form which can be written as a linear combination of $G_{i,j}$ and $C_{ij}[F]$ will be our focus in this work* where $\mathbf{G}$ is a $K \times K$ matrix.

$$\max_F \sum_{i,j \in [K]} G_{ij} C_{ij}[F], \tag{2}$$

The entry G_{ij} represents the reward associated with predicting class j when the true class is i. The matrix $\mathbf{G}$ is called a gain matrix [20]. Some more complex non-decomposable objectives for classification can be reduced to CSL [22, 32, 20]. For instance, the aforementioned two complex objectives can be reduced to CSL using continuous relaxation or a Lagrange multiplier as bellow. Let $\Delta_{K-1} \subset \mathbb{R}^K$ be the $K-1$-dimensional probability simplex. Then, maximizing the minimum recall is equivalent to the saddle-point optimization problem: $\max_F \min_{\boldsymbol{\lambda} \in \Delta_{K-1}} \sum_{i \in [K]} \lambda_i \frac{C_{ii}[F]}{\pi_i}$. Thus, for a fixed $\boldsymbol{\lambda}$, the corresponding gain matrix is given as a diagonal matrix $\mathrm{diag}(G_1, \ldots, G_K)$ with $G_i = \lambda_i / \pi_i$ for $1 \leq i \leq K$. Similarly, using Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^K_{\geq 0}$, Eq. (1) is rewritten as a max-min optimization problem [20, Sec. 2]: $\max_F \min_{\boldsymbol{\lambda} \in \mathbb{R}^K_{\geq 0}} \frac{1}{K} \sum_{i \in [K]} C_{ii}[F]/\pi_i + \sum_{j \in [K]} \lambda_j \left(\sum_{i \in [K]} C_{ij}[F] - 0.95/K \right)$. In this case, the corresponding gain matrix $\mathbf{G}$ is given as $G_{ij} = \frac{\delta_{ij}}{K \pi_i} + \lambda_j$, where δ_{ij} is the Kronecker's delta. One can solve these max-min problems by alternatingly updating $\boldsymbol{\lambda}$ (using exponented gradient or projected gradient descent) and optimizing the cost-sensitive objectives [20]. The next subsection introduces calibrated loss functions for CSL.

2.2 Loss Functions for Non-Decomposable Objectives

The cross entropy loss function is appropriate for optimizing accuracy for deep neural networks, however, learning with CE can suffer low performance for cost-sensitive objectives [20]. Following [20], we introduce calibrated loss functions for given gain matrix $\mathbf{G}$. For $x \in \mathcal{X}$, we let $\boldsymbol{s}(x) : \mathcal{X} \to \mathbb{R}^K$ be a score of a model on instance x so that the prediction $\boldsymbol{p}_m(x)$ is given as softmax i.e. $\left(\frac{\exp(s_y(x))}{\sum_{i \in [K]} \exp(s_i(x))} \right)_{y \in [K]}$. For a gain matrix $\mathbf{G}$, the corresponding loss function is given as a combination of logit adjustment [17] and loss re-weighting [24]. We decompose the gain matrix $\mathbf{G}$ as $\mathbf{G} = \mathbf{M}\mathbf{D}$, where $\mathbf{D} = \mathrm{diag}(G_{11}, \ldots, G_{KK})$ be a diagonal matrix, with $D_{ii} > 0, \forall i \in [K]$ and $\mathbf{M} \in \mathbb{R}^{K \times K}$. For $y \in [K]$ and a score function $\mathbf{s}(x)$, the hybrid loss is defined as follows:

$$\ell^{\mathrm{hyb}}(y, \mathbf{s}(x)) = -\sum_{i \in [K]} M_{yi} \log \left(\frac{\exp(s_i(x) - \log(D_{ii}))}{\sum_{j \in [K]} \exp(s_j(x) - \log(D_{jj}))} \right). \tag{3}$$

The average loss on training sample $S \subset \mathcal{X}$ is defined as $\mathcal{L}^{\mathrm{hyb}}(\mathcal{X}) = \frac{1}{|S|} \sum_{x \in S} \ell^{\mathrm{hyb}}(u, \mathbf{s}(x))$. Narasimhan and Menon [20] proved that the hybrid loss is calibrated, that is learning with ℓ^{hyb} gives the Bayes optimal classifier for $\mathbf{G}$ (c.f., [20, Proposition 4], of which we provide a formal statement in Sec. N.1). If $\mathbf{G}$ is a diagonal matrix (i.e., $\mathbf{M} = \mathbf{1}_K$), the hybrid loss is called the logit adjusted (LA) loss and $\ell^{\mathrm{hyb}}(y, \mathbf{s}(x))$ is denoted by $\ell^{\mathrm{LA}}(y, \mathbf{s}(x))$.

2.3 Consistency Regularizer for Semi-Supervised Learning

Modern self-training methods not only leverage pseudo labels, but also forces consistent predictions of a classifier on augmented examples or neighbor examples [35, 18, 36, 31]. More formally, a classifier F is trained so that the consistent regularizer $R(F)$ is small while a supervised loss or a loss between pseudo labeler are minimized [35, 31]. Here the consistency regularizer $R(F)$ is defined as

$$\mathbf{E}_x \left[\mathbb{1}(F(x) \neq F(x'), \exists x' \text{ s.t. } x' \text{ is a neighbor of an augmentation of } x) \right].$$

In existing works, consistency regularizers are considered for optimization of accuracy. In the subsequent sections, we consider consistency regularizers for cost-sensitive objectives.

3 Cost-Sensitive Self-Training for Non-Decomposable Metrics

3.1 CSL and Weighted Error

In the case of accuracy or 0-1-error, a self-training based SSL algorithm using a consistency regularizer achieves the state-of-the-art performance across a variety of datasets [31] and its effectiveness has been proved theoretically [35]. This section provides theoretical analysis of a self-training based SSL algorithm for non-decomposable objectives by generalizing [35]. More precisely, the main result of this section (Theorem 5) states that an SSL method using consistency regularizer improves a given pseudo labeler for non-decomposable objectives. We provide all the omitted proofs for theoretical results in the paper, in Appendix.

In Sec. 2, we considered non-decomposable metrics and their reduction to cost-sensitive learning objectives defined by (2) using a gain matrix. In this section, we consider an equivalent objective using a weighted error. For weight matrix $w = (w_{ij})_{1 \le i,j \le K}$ and a classifier $F : \mathcal{X} \to [K]$, a weighted error is defined as follows:

$$\mathrm{Err}_w(F) = \sum_{i,j \in [K]} w_{ij} \mathbf{E}_{x \sim P_i} \left[\mathbb{1}(F(x) \neq j) \right],$$

where, $P_i(x)$ denotes the class conditional probability $\mathbf{P}(x \mid y = i)$. If $w = \mathrm{diag}(1/K, \ldots, 1/K)$, then this coincides with the balanced error [17]. Since $\mathbf{E}_{(x,y) \sim D} \left[\mathbb{1}(y = i, F(x) = j) \right] = \mathbf{P}(y = i) - \mathbf{P}(y = i) \mathbf{E}_{x \sim P_i} \left[\mathbb{1}(F(x) \neq j) \right]$, minimizing $\mathrm{Err}_w(F)$ with respect to F is equivalent to CSL (2) with gain matrix $\mathbf{G}$ where $G_{ij} = (w_{ij}/\pi_i)_{1 \le i,j \le K}$. We note that if we add a gain matrix $\mathbf{G}$ to a matrix with the same columns, then the maximizers of CSL (2) are the same as the original problem. Hence, without loss of generality, we assume $w_{ij} \ge 0$. We also assume that $w \neq \mathbf{0}$, i.e., $|w|_1 > 0$.

In the previous work [35], it is assumed that there exists a ground truth classifier $F^\star : \mathcal{X} \to [K]$ and the classes are disjoint. However, if supports of distributions $\{P_i\}_{1 \le i \le K}$ are disjoint, a solution of the minimization problem $\min_F \mathrm{Err}_w(F)$ may be independent of w in some cases. More precisely, if $w = \mathrm{diag}(w_1, \ldots, w_K)$ is a diagonal matrix and $w_i > 0, \forall i$, then the optimal classifier is given as $x \mapsto \mathrm{argmax}_{k \in [K]} w_k P_k(x)$ (this follows from [20, Proposition 1]). If supports are disjoint, then the optimal classifier is the same as $x \mapsto \mathrm{argmax}_{k \in [K]} P_k(x)$, which coincides with the ground truth classifier. Therefore, we do not assume the supports of P_i are disjoint nor a ground truth classifier exists unlike [35]. In Appendix, Sec. N.3, we explain the differences in our theoretical assumptions with respect to Wei et al. [35] more intuitively using a simple example.

3.2 Weighted Consistency Regularizer

In the case of accuracy, the usual consistency regularizer is equally important across the distributions P_i for $1 \le i \le K$ [31, 35]. However, in our case, if the entries of the i_0th row of the weight matrix w are smaller than the other entries for some i_0, then the model predictions on examples drawn from the distribution P_{i_0} are less important than those on the other examples. In this case, we require less restrictive consistency regularizer for distribution P_{i_0} than the other distributions. Thus, we need a weighted (or cost-sensitive) consistency regularizer defined below.

We assume that the instance space $\mathcal{X}$ is a normed vector space with norm $|\cdot|$ and $\mathcal{T}$ is a set of augmentations, i.e., each $T \in \mathcal{T}$ is a map from $\mathcal{X}$ to itself. For a fixed $r > 0$, we define $\mathcal{B}(x)$ by $\{x' \in \mathcal{X} : \exists T \in \mathcal{T} \text{ s.t. } |x' - T(x)| \le r\}$. For each $i \in [K]$, we define conditional consistency regularizer by $R_{\mathcal{B},i}(F) = \mathbf{E}_{x \sim P_i} \left[\mathbb{1} \left(\exists x' \in \mathcal{B}(x) \text{ s.t. } F(x) \neq F(x') \right) \right]$. Then, we define the weighted consistency regularizer by $R_{\mathcal{B},w}(F) = \sum_{i,j \in [K]} w_{ij} R_{\mathcal{B},i}(F)$. If the prediction of classifier F is robust to augmentation $T \in \mathcal{T}$ and small noise, then $R_{\mathcal{B},w}(F)$ is small. For $\beta > 0$, we consider the following optimization objective for finding a classifier F:

$$\min_F \mathrm{Err}_w(F) \quad \text{subject to } R_{\mathcal{B},w}(F) \le \beta. \tag{4}$$

A solution of the problem (4) is denoted by F^*. We let $F_{\mathrm{pl}} : \mathcal{X} \to [K]$ a pseudo labeler (a classifier trained on a labled dataset) with reasonable performance (we elaborate on this in Section 3.4). The following mathematically informal assumption below is required to interpret our main theorem.

ASSUMPTION 1. We assume both β and $\mathrm{Err}_w(F^*)$ are sufficiently small so that they are negligible compared to $\mathrm{Err}_w(F_{\mathrm{pl}})$.

Assumption 1 assumes existence of an optimal classifier F^* that minimizes the error $\mathrm{Err}_w(F)$ (i.e. Bayes Optimal) among the class of classifiers which are robust to data augmentation (i.e. low weighted consistency $R_{\mathcal{B},w}(F)$). As we operate in case of overparameterized neural networks such a classifier F^* is bound to exist, but is unknown in our problem setup. In the case of the balanced error, the validity of this assumption is justified by the fact that the existing work [31] using consistency regularizer on data augmentation obtains classifier F , that achieves high accuracy (i.e., low balanced errors) for balanced datasets. Also in Appendix Sec. C.1, we provide an example that supports the validity of the assumption in the case of Gaussian mixtures and diagonal weight matrices.

3.3 Expansion Property

For $x \in \mathcal{X}$, we define the neighborhood $\mathcal{N}(x)$ of x by $\{x' \in \mathcal{X} : \mathcal{B}(x) \cap \mathcal{B}(x') \neq \emptyset\}$. For a subset $S \subseteq \mathcal{X}$, neighborhood of S is defined as $\mathcal{N}(S) = \cup_{x \in S} \mathcal{N}(x)$. Similarly to [35], we consider the following property on distributions.

DEFINITION 2. Let $c : (0, 1] \to [1, \infty)$ be a non-increasing function. For a distribution Q on $\mathcal{X}$ we say Q has c-expansion property if $Q(\mathcal{N}(S)) \geq c(Q(S))Q(S)$ for any measurable $S \subseteq \mathcal{X}$.

The c-expansion property implies that if $Q(S)$ decreases, then the "the expansion factor" $Q(\mathcal{N}(S))/Q(S)$ increases. This is a natural condition, because it roughly requires that if $Q(S)$ is small, then $Q(\mathcal{N}(S))$ is large compared to $Q(S)$. For intuition let us consider a ball of radius l depicting $S \subset \mathbb{R}^d$ with volume $Q(S)$ and its neighborhood $\mathcal{N}(S)$ expands to a ball with radius $l + 1$. The expansion factor here would be $((l + 1)/l)^d$, hence as l (i.e. $Q(S)$) increases $(1 + 1/l)^d$ (i.e. $Q(\mathcal{N}(S))/Q(S)$) decreases. Hence it's natural to expect c to be a non-decreasing function. In addition, we also show that it is also satisfied for mixtures of Gaussians and mixtures of manifolds (see Example 3 in Appendix for more details). The c-expansion property considered here is equivalent to the $(a, \widetilde{c})$-expansion property, which is shown to be realistic for vision and used for theoretical analysis of self-training in [35], where $a \in (0, 1)$ and $\widetilde{c} > 1$ (see Sec. N.2 in Appendix).

3.4 Cost-Sensitive Self-Training with Weighted Consistency Regularizer

Let $F_{\mathrm{pl}} : \mathcal{X} \to [K]$ be a pseudo labeler. F_{pl} can be any classifier satisfying the following assumption, however, typically it is a classifier trained on a labeled dataset.

ASSUMPTION 4. Define weighted probability measure $\mathcal{P}_w$ on $\mathcal{X}$ by $\mathcal{P}_w(U) = \frac{\sum_{i,j \in [K]} w_{ij} P_i(U)}{\sum_{i,j \in [K]} w_{ij}}$ for $U \subseteq \mathcal{X}$. We assume $\mathcal{P}_w$ satisfies c-expansion property for a non-increasing function $c : (0, 1] \to [1, \infty)$. We assume that $\mathrm{Err}_w(F_{\mathrm{pl}}) + \mathrm{Err}_w(F^*) \leq |w|_1$. Let $\gamma = c(p_w)$, where $p_w = \frac{\mathrm{Err}_w(F_{\mathrm{pl}}) + \mathrm{Err}_w(F^*)}{|w|_1}$. We also assume $\gamma > 3$.

Since c is non-increasing, γ is a non-increasing function of $\mathrm{Err}_w(F_{\mathrm{pl}})$ (and $\mathrm{Err}_w(F^*)$). Therefore, the assumption $\gamma > 3$ roughly requires that $\mathrm{Err}_w(F_{\mathrm{pl}})$ is "small". We provide concrete conditions for $\mathrm{Err}_w(F_{\mathrm{pl}})$ that satisfy $\gamma > 3$ in the case of mixture of isotropic Gaussians in Appendix (Example 8). We define $L_{\text{0-1}}^{(i)}(F, F') = \mathbf{E}_{x \sim P_i} [\mathbb{1}(F(x) \neq F'(x))]$. Then, we consider the following objective:

$$\min_F \mathcal{L}_w(F), \quad \text{where } \mathcal{L}_w(F) = \frac{\gamma + 1}{\gamma - 1} L_w(F, F_{\mathrm{pl}}) + \frac{2\gamma}{\gamma - 1} R_{\mathcal{B},w}(F). \tag{5}$$

Here $L_w(F, F_{\mathrm{pl}})$ is defined as $\sum_{i,j \in [K]} w_{ij} L_{\text{0-1}}^{(i)}(F, F_{\mathrm{pl}})$. The following theorem is a generalization of [35, Theorem 4.3], which provided a similar result for balanced 0-1-error in the case of distributions with disjoint supports.

THEOREM 5. *Any learnt classifier $\widehat{F}$ using the loss function $\mathcal{L}_w$ (i.e., $\mathrm{argmin}_F \mathcal{L}_w(F)$) satisfies:*

$$\mathrm{Err}_w(\widehat{F}) \leq \frac{2}{\gamma - 1} \mathrm{Err}_w(F_{\mathrm{pl}}) + \frac{\gamma + 1}{\gamma - 1} \mathrm{Err}_w(F^*) + \frac{4\gamma}{\gamma - 1} R_{\mathcal{B},w}(F^*).$$

REMARK. Since both $\mathrm{Err}_w(F^*)$ and $R_{\mathcal{B},w}(F^*) \leq \beta$ are negligible compared to $\mathrm{Err}_w(F_{\mathrm{pl}})$ and $\gamma > 3$, Theorem 5 asserts that $\hat{F}$ learnt by minimizing semi-supervised loss $L_w(F, F_{\mathrm{pl}})$ with the consistency regularizer $R_{\mathcal{B},w}(F)$ can achieve superior performance than the pseudo labeler F_{pl} in terms of the weighted error Err_w. In Appendix Sec. D, following [35, 34], we also provide a generalization bound for $\mathrm{Err}_w(F)$ using all-layer margin [34] in the case when classifiers are neural networks.

Naively, to optimize the objective (5), the distribution of unlabeled dataset should coincide with $\mathcal{P}_w$, or the optimizer has to have access to a large set of samples drawn from $\mathcal{P}_w$. However, this is a restrictive assumption in practice. In Sec. 4, we address this issue by focusing on unlabeled instances whose pseudo labels are of high confidence and utilizing the LA loss or hybrid loss introduced in Sec. 2 to minimize the weighted objective $\mathcal{L}_w(F)$. As previously explained, existing works [31] also filtered out unlabeled instances. However, we shall empirically and theoretically show that a naive application of an existing threshold method does not work in our case.

4 CSST in Practice

The self-training methods utilizing consistency regularization in addition to to average supervised loss $\mathcal{L}_s$ also have a consistency loss for unlabeled samples (i.e. $\mathcal{L}_u$) with a thresholding mechanism to select unlabeled samples. The final loss for training the network is $\mathcal{L}_s + \lambda_u \mathcal{L}_u$ where λ_u is the hyperparameter. The supervised loss ℓ_s can be modified easily based on $\mathbf{G}$, for optimizing desired non-decomposable metric (Sec. 2.1). We will now introduce how can we modify the consistency loss and thresholding mechanism to introduce CSST for optimizing desired non-decomposable metric.

Weighted Consistency Regularization. As the idea of consistency regularization is to enforce consistency between model prediction on different augmentations of input, this is usually achieved by minimizing some kind of divergence $\mathcal{D}$. A lot of recent works [18, 31, 36] in semi-supervised learning use $\mathcal{D}_{\mathrm{KL}}$ to enforce consistency between the model's prediction on unlabeled data and its augmentations, $p_m(x)$ and $p_m(\mathcal{A}(x))$. Here $\mathcal{A}$ usually denotes a form of strong augmentation. Across these works, the distribution of confidence of the model's prediction is either sharpened or used to get a hard pseudo label to obtain $\hat{p}_m(x)$. As we aim to optimized the cost-sensitive learning objective, we aim to match the distribution of normalized distribution (i.e. $\mathrm{norm}(\mathbf{G}^{\mathbf{T}}\hat{p}_m(x)) = \mathbf{G}^{\mathbf{T}}\hat{p}_m(x)/\sum_i (\mathbf{G}^{\mathbf{T}}\hat{p}_m(x))_i$) (Proposition 2[20] also in Prop. 7) with the $p_m(\mathcal{A}(x))$ by minimizing the KL-Divergence between these. We now propose to use the following weighted consistency regularizer loss function for optimizing the same:

$$\ell_u^{\mathrm{wt}}(\hat{p}_m(x), p_m(\mathcal{A}(x), \mathbf{G}) = -\sum_{i=1}^{K} (\mathbf{G}^{\mathbf{T}}\hat{p}_m(x))_i \log(p_m(\mathcal{A}(x))_i) \tag{6}$$

PROPOSITION 6. *The minimizer of* $\mathcal{L}_u^{wt} = \frac{1}{|D|}\sum_{x\in D} \ell_u^{wt}(\hat{p}_m(x), p_m(\mathcal{A}(x), \mathbf{G})$ *leads to minimization of KL Divergence i.e.* $\mathcal{D}_{KL}(\mathrm{norm}(\mathbf{G}^{\mathbf{T}}\hat{p}_m(x))||p_m(\mathcal{A}(x)))\ \forall x \in D$.

As the above loss is similar in nature to the cost sensitive losses introduced by Narasimhan and Menon [20] (Sec. 2.1) we can use the logit-adjusted variants (i.e. ℓ^{LA} and ℓ^{hyb} based on type of $\boldsymbol{G}$) of these in our final loss formulations for training overparameterized deep networks. We formalize the connection of practical weighted consistency regularizer to theory (Sec. 3.4) in Appendix Sec. B.

Threshold Mechanism for CSST. In the usually semi-supervised learning formulation we use the $\max_i(p_m(x)_i) > \tau$ as the function to select samples for which consistency regularization term is non-zero. We find that this leads to sub-optimal results in particularly the case of non-diagonal $\mathbf{G}$ due to few samples being able to cross the threshold (Fig. 3). As in the case of cost-sensitive loss formulation the samples may not achieve the high confidence to cross the threshold of consistency regularization. This is also theoretically justified by the following proposition:

PROPOSITION 7 ([20] Proposition 2)**.** *Let* $p_m^{opt}(x)$ *be the optimal softmax model function obtained by optimizing the cost-sensitive objective in Eq. 2 by averaging weighted loss function* $\ell^{wt}(y, p_m(x)) = -\sum_{i=1}^{K} G_{y,i} \log \frac{(p_m(x)_i)}{\sum_j p_m(x)_j}$. *Then optimal* $p_m^{opt}(x)$ *is:* $p_m^{opt}(x) = \frac{G_{y,i}}{\sum_j G_{y,j}} = \mathrm{norm}(\mathbf{G}^T\mathbf{y})\forall(x, y)$

Here $\mathbf{y}$ is the one-hot representation vector for a label y. We now propose our novel way of thresholding samples for which consistency regularization is applied in CSST. Our thresholding method takes into account the objective of optimizing the non-decomposable metric by taking $\mathbf{G}$ into account. We propose to use the threshold on KL-Divergence of the softmax of the sample $p_m(x)$ with the optimal softmax (i.e. $\mathrm{norm}(\mathbf{G}^T\hat{p}_m(x))$) for a given $\mathbf{G}$ corresponding to the pseudo label (or sharpened) $\hat{p}_m(x)$, using which we modify the consistency regularization loss term ℓ_u given below:

$$\mathcal{L}_u^{wt}(B_u) = \frac{1}{|B_u|}\sum_{x\in B_u} \mathbb{1}_{(\mathcal{D}_{KL}(\mathrm{norm}(\mathbf{G}^T\hat{p}_m(x))\ ||\ p_m(x))\leq\tau)} \ell_u^{\mathrm{wt}}(\hat{p}_m(x), p_m(\mathcal{A}(x), \mathbf{G})) \tag{7}$$

Table 2: Results of maximizing the worst-case recall over *all classes* (col 2–3) and over just the head and tail classes (col 4–7).

Method	CIFAR10-LT ($\rho = 100$)		CIFAR100-LT ($\rho = 10$)		Imagenet100-LT ($\rho = 100$)	
	Avg. Rec	Min Rec	Avg. Rec	Min HT Rec	Avg. Rec	Min HT Rec
ERM	0.52	0.26	0.36	0.14	0.40	0.30
LA	0.51	0.38	0.36	0.35	0.48	0.47
CSL	0.64	0.57	0.43	0.43	0.52	0.52
Vanilla (FixMatch)	0.78	0.48	0.63	0.36	0.58	0.49
CSST (FixMatch)	0.76	0.72	0.63	0.61	0.64	0.63

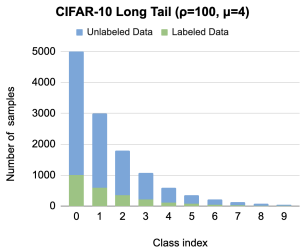


Figure 2: CIFAR-10 Long tail distribution $\rho = 100, \mu = 4$.

We name this proposed combination of KL-Thresholding and weighted consistency regularization as CSST in our experimental results. We find that for non-diagonal gain matrix $\mathbf{G}$ the proposed thresholding plays a major role in improving performance over supervised learning. This is demonstrated by comparison of CSST and CSST w/o KL-Thresholding (without proposed thresholding mechanism) in Fig. 3 and Tab. 3. We will now introduce CSST by introducing consistency based losses and threshold mechanism for unlabeled data (i.e. replacing l_u part of the data) into the popular semi-supervised methods of FixMatch [31] and Unsupervised Data Augmentation for Consistency Training (UDA) [36]. The exact expression for the weighted consistency losses utilized for UDA and FixMatch have been provided in the Appendix Sec. H.2 and H.3 .

5 Experiments

We demonstrate that the proposed CSST framework shows significant gains in performance on both vision and natural language processing datasets, with an imbalance ratio defined on the training set as $\rho = \frac{\max_i P(y=i)}{\min_i P(y=i)}$. We assume the labeled and unlabeled samples come from a similar distribution and only differ by the ratio of total number of samples (μ). The frequency of samples follows an exponentially decaying long-tailed distributed as seen in Fig. 2, which closely imitates the distribution of real-world long-tailed datasets [33, 10]. For CIFAR-10 [11], IMDb [16] and DBpedia-14 [14], we use $\rho = 100$ and $\rho = 10$ for CIFAR-100 [11] and ImageNet-100 [27] [1] datasets. For CIFAR-100, at $\rho = 100,\ \mu = 4$ we only have 1 labeled sample for the tail classes, hence we chose a lower imbalance factor. In case of IMDb and DBpedia-14, we set the total number of labeled samples to 1,000. In all our experiments on CIFAR-10, CIFAR-100 and ImageNet-100 datasets, we set $\mu = 4$. We compare our method against supervised methods of ERM, Logit Adjustment (LA) [17] and Cost Sensitive Learning (CSL) [21] trained on the same number of labeled samples as used by semi-supervised learning methods, along with vanilla semi-supervised methods of FixMatch (for vision) and UDA (for NLP tasks). We use WideResNets(WRN) [39], specifically WRN-28-2 and WRN-28-8 for CIFAR-10 and CIFAR-100 respectively. For ImageNet we use a ResNet-50 [7] network for our experiments and finetuned DistilBERT(base uncased) [29] for IMDb and DBpedia-14 datasets. The validation and test sets have a uniform distribution and have 5,000 samples each. We use Stochastic Gradient Descent(SGD) with cosine learning rate schedule. A detailed list of hyper-parameters and additional experimental details can be found in the Appendix Table 4.

Maximizing Worst-Case Recall. For CIFAR-10, IMDb, and DBpedia-14 datasets, we maximize the minimum recall among all classes. Given the nature of distribution of number of samples per class for datasets with larger number of classes like CIFAR-100 and ImageNet-100, we pick an easier objective of maximizing the minimum of average recall of head classes or tail classes. We define the head classes ($\mathcal{H}$) and tail classes ($\mathcal{T}$) as the first 90 classes and last 10 classes respectively. The min HT objective can be mathematically formulated as: $\max_F \min\left(\sum_{i\in\mathcal{H}} \mathrm{rec}_i[F]/|\mathcal{H}|,\ \sum_{i\in\mathcal{T}} \mathrm{rec}_i[F]/\,|\mathcal{T}|\right)$. The corresponding gain matrix $\mathbf{G}$ is $\mathrm{diag}(\frac{\lambda_{\mathcal{H}}}{\pi_1|\mathcal{H}|}, \frac{\lambda_{\mathcal{H}}}{\pi_2|\mathcal{H}|}, \ldots, \frac{\lambda_{\mathcal{T}}}{\pi_{K-1}|\mathcal{T}|}, \frac{\lambda_{\mathcal{T}}}{\pi_K|\mathcal{T}|})$. Since $\mathbf{G}$ is diagonal here, we use CSST(FixMatch) loss function Eq. 7 with the corresponding ℓ_u^{wt} being substituted by ℓ_u^{LA} as define in Sec. 2.1. Also for labeled samples we use $\mathcal{L}_s^{LA}$ as $\mathbf{G}$ is diagonal, we then combine the loss and train network using SGD. Each few steps of SGD, were followed by an update on the λ and $\mathbf{G}$ based on the uniform validation set (See Alg. 1 in Appendix). We find that CSST(FixMatch) significantly outperforms the other baselines in terms of the min recall and min Head-Tail recall for

[1] https://www.kaggle.com/datasets/ambityga/imagenet100

Table 3: Results of maximizing the mean recall subject to coverage constraint *all classes* (col 2–3) and over the head and tail classes (col 4–7). Proposed CSST(FixMatch) approach compares favorably to ERM,LA,CSL vanilla(FixMatch) and CSST(FixMatch) w/o KL-Thresh.. It is the best at both maximizing mean recall and coming close to satisfying the coverage constraint.

Method	**CIFAR10-LT** Per-class Coverage ($\rho = 100$, tgt : 0.1)		**CIFAR100-LT** Head-Tail Coverage ($\rho = 10$, tgt : 0.01)		**ImageNet100-LT** Head-Tail Coverage ($\rho = 10$, tgt : 0.01)	
	Avg. Rec	**Min Cov**	**Avg. Rec**	**Min HT Cov**	**Avg. Rec**	**Min HT Cov**
ERM	0.52	0.034	0.36	0.004	0.40	0.006
LA	0.51	0.039	0.36	0.009	0.48	0.009
CSL	0.60	0.090	0.45	0.010	0.48	0.010
Vanilla (FixMatch)	0.78	0.055	0.63	0.004	0.58	0.007
CSST(FixMatch) w/o KL-Thresh.	0.67	0.093	0.47	0.010	0.26	0.010
CSST(FixMatch)	0.80	0.092	0.63	0.010	0.58	0.010

Table 4: Results of maximizing the min recall over *all classes* for classification on NLP datasets. Proposed CSST(UDA) approach outperforms ERM and vanilla(UDA) baselines.

Method	**IMDb** ($\rho = 10$)		**IMDb** ($\rho = 100$)		**DBpedia-14** ($\rho = 100$)	
	Avg Rec	**Min Rec**	**Avg Rec**	**Min Rec**	**Avg Rec**	**Min Rec**
ERM	0.79	0.61	0.50	0.00	0.95	0.58
vanilla(UDA)	0.82	0.66	0.50	0.00	0.96	0.65
CSST(UDA)	0.89	0.88	0.77	0.75	0.99	0.97

all datasets, the metrics which we aimed to optimize (Tab. 2), which shows effectiveness of CSST framework. Despite optimizing worst-case recall we find that our framework is still able to maintain reasonable average (Avg.) recall in comparison to baseline vanilla(FixMatch), which demonstrates it's practical applicability.We find that optimizing min recall across NLP tasks of classification on long-tailed data by plugging UDA into CSST(UDA) framework shows similar improvement in performance (Tab. 4). This establishes the generality of our framework to even self-training methods across domain of NLP as well. Since here the gain matrix $\mathbf{G}$ is a diagonal matrix and for the experiments on the vision datasets, FixMatch uses a hard pseudo-label, we discuss the equivalence of our threshold function to that of a confidence based threshold of FixMatch in Appendix Sec. I.

Maximizing Mean Recall Under Coverage Constraints. Maximizing mean recall under coverage constraints objective seeks to result in a model with good average recall, yet at the same time constraints the proportion of predictions for each class to be uniform across classes. The ideal target coverage under a balanced evaluation set(or such circumstances) is given as $\text{cov}_i[F] = \frac{1}{K}, \forall\, i \in [K]$. This objective corresponds to a non-diagonal $\mathbf{G}$ as shown in Sec. 2.1. Hence, for introducing CSST into FixMatch we replace first supervised loss $\mathcal{L}_s$ with $\mathcal{L}_s^{\text{hyb}}$. For the unlabeled data we introduce ℓ^{hyb} in $\mathcal{L}_u^{\text{wt}}$ (Eq. 7). Hence, the final objective $\mathcal{L}$ is defined as, $\mathcal{L} = \mathcal{L}_s^{\text{hyb}} + \lambda_u \mathcal{L}_u^{\text{wt}}$. We update the parameters of the cost-sensitive loss ($\mathbf{G}$ and $\boldsymbol{\lambda}$) periodically after few of SGD on the model parameters (Alg. 2 in Appendix). In this case our proposed thresholding mechanism in CSST(FixMatch) introduced in Sec. 4, leads to effective utilization of unlabled data resulting in improved performance over the naive CSST(FixMatch) without (w/o) KL-Thresholding (Tab. 3). In these experiments, the mean recall of our proposed approach either improves or stays same to the vanilla(FixMatch) implementation but only ours is the one that comes close to satisfying the coverage constraint. We observe that among all the methods, the only methods that come close to satisfying the coverage constraints are the ones with ℓ^{hyb} included in them.

Ablation on amount of unlabeled data. Here, we ablated the total number of labeled sampled while keeping the number of unlabeled samples constant. We observe (in Fig. 4a) that as we increase the number of labeled samples the mean recall improved. This is because more labeled samples helps in better pseudo-labeling on the unlabeled samples and similarly as we decrease the number of labeled sample, the models errors on the pseudo-labels increase causing a reduction in mean recall. Hence, any addtional labeled data can be easily used to improve CSST performance.

Ablation on τ threshold. Fig. 4b shows that when the KL divergence threshold is too high, a large number of samples with very low degree of distribution match are used for generating the sharpened

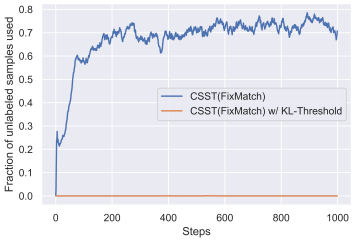
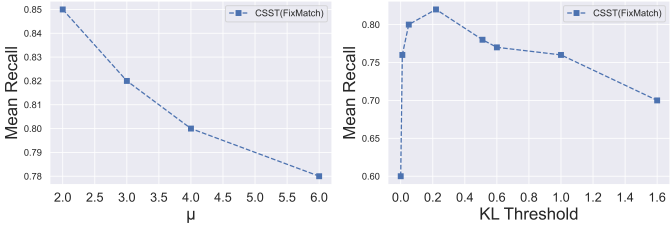


Figure 3: Fraction of unlabelled data used for maximizing average recall under coverage constraints for CIFAR-100 ($\rho = 10, \mu = 4$) (Sec. 5).



(a) (b)

Figure 4: Maximizing average recall under coverage constraints for CIFAR-10 Long tail ($\rho = 100$) (Sec. 5). Fig. shows comparison of (a) increasing the ratio of unlabeled samples to labeled samples given fixed number of unlabeled samples (b) Ablation on KL diveregence based threshold for CSST(FixMatch)

target (or pseudo-labels), this leads to worsening of mean recall as many targets are incorrect. We find that keeping a conservative of $\tau = 0.3$ works well across multiple experiments.

6 Related Work

Self-Training. Self-training algorithms have been popularly used for the tasks of semi-supervised learning [1, 37, 31, 13] and unsupervised domain adaptation [28, 41]. In recent years several regularizers which enforce consistency in the neighborhood (either an adversarial perturbation [18] or augmentation [36]) of a given sample have further enhanced the applicability and performance of self-training methods, when used in conjunction. However, these works have focused mostly on improving the generic metric of accuracy.

Cost-Sensitive Learning. It refers to problem settings where the cost of error differs for a sample based on what class it belongs to. These settings are very important for critical real world applications like disease diagnosis, wherein mistakenly classifying a diseased person as healthy can be disastrous. There have been a pletheora of techniques proposed for these which can be classified into: importance weighting [15, 38, 5] and adaptive margin [2, 38] based techniques. For overparameterized models Narasimhan et al. [22] show that loss weighting based techniques are ineffective and propose a logit-adjustment based cost-sensitive loss which we also use in our framework.

Complex Metrics for Deep Learning. There has been a prolonged effort on optimizing more complex metrics that take into account practical constraints [21, 26, 23]. However most work has focused on linear models leaving scope for works in context of deep neural networks. Sanyal et al. [30] train DNN using reweighting strategies for optimizing metrics, Huang et al. [8] use a reinforcement learning strategy to optimize complex metrics, and Kumar et al. [12] optimize complex AUC (Area Under Curve) metric for a deep neural network. However, all these works have primarily worked in supervised learning setup and are not designed to effectively make use of available unlabeled data.

7 Conclusion

In this work, we aim to optimize the practical non-decomposable metrics readily used in machine learning through Self-training with consistency regularization a class of semi-supervised learning methods. We introduce a cost-sensitive self-training framework (CSST) which involves minimizing a cost-sensitive error on pseudo labels along with consistency regularization. We show theoretically that we can obtain classifiers which can better optimize the desired non-decomposable metric than original model used for obtaining pseudo labels, under similar data distribution assumptions as used for theoretical analysis of Self-training [35]. We then apply CSST to practical and effective self-training method of FixMatch, UDA also incorporating a novel regularizer and thresholding mechanism based on given non-decomposable objective. We find that CSST leads to a significant gain in performance of desired non-decomposable metric, in comparison to vanilla self-training based baseline. Analyzing the CSST framework when the distribution of unlabeled data significantly differs from labeled data, is a good direction to pursue for future work.

References

[1] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/1cd138d0499a68f4bb72bee04bbec2d7-Paper.pdf. 1, 9

[2] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019. 9

[3] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. 1

[4] Andrew Cotter, Heinrich Jiang, Maya R Gupta, Serena Wang, Taman Narayan, Seungil You, and Karthik Sridharan. Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *J. Mach. Learn. Res.*, 20(172):1–59, 2019. 2

[5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 9

[6] Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael P Friedlander. Satisfying real-world goals with dataset constraints. *Advances in Neural Information Processing Systems*, 29, 2016. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[8] Chen Huang, Shuangfei Zhai, Walter Talbott, Miguel Bautista Martin, Shih-Yu Sun, Carlos Guestrin, and Josh Susskind. Addressing the loss-metric mismatch with adaptive loss alignment. In *International conference on machine learning*, pages 2891–2900. PMLR, 2019. 9

[9] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014. 1

[10] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 7

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 7

[12] Abhishek Kumar, Harikrishna Narasimhan, and Andrew Cotter. Implicit rate-constrained optimization of non-decomposable objectives. In *International Conference on Machine Learning*, pages 5861–5871. PMLR, 2021. 9

[13] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 9

[14] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015. 7

[15] Yi Lin, Yoonkyung Lee, and Grace Wahba. Support vector machines for classification in nonstandard situations. *Machine learning*, 46(1):191–202, 2002. 9

[16] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015. 7

[17] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2020. 3, 4, 7

[18] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 1, 3, 6, 9

[19] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019. 1

[20] Harikrishna Narasimhan and Aditya K Menon. Training over-parameterized models with non-decomposable objectives. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2, 3, 4, 6

[21] Harikrishna Narasimhan, Rohit Vaish, and Shivani Agarwal. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. *Advances in neural information processing systems*, 27, 2014. 7, 9

[22] Harikrishna Narasimhan, Harish Ramaswamy, Aadirupa Saha, and Shivani Agarwal. Consistent multiclass algorithms for complex performance measures. In *International Conference on Machine Learning*, pages 2398–2407. PMLR, 2015. 3, 9

[23] Nagarajan Natarajan, Oluwasanmi Koyejo, Pradeep Ravikumar, and Inderjit Dhillon. Optimal classification with multivariate losses. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2016. 9

[24] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952, 2017. 3

[25] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11557–11568, 2021. 1

[26] Shameem Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. Optimizing f-measures by cost-sensitive classification. *Advances in neural information processing systems*, 27, 2014. 9

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 7

[28] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 2988–2997. PMLR, 2017. 9

[29] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 7

[30] Amartya Sanyal, Pawan Kumar, Purushottam Kar, Sanjay Chawla, and Fabrizio Sebastiani. Optimizing non-decomposable measures with deep networks. *Machine Learning*, 107(8): 1597–1620, 2018. 1, 9

[31] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020. 1, 2, 3, 4, 5, 6, 7, 9

[32] Shiv Kumar Tavker, Harish Guruprasad Ramaswamy, and Harikrishna Narasimhan. Consistent plug-in classifiers for complex objectives and constraints. *Advances in Neural Information Processing Systems*, 33:20366–20377, 2020. 3

[33] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 7

[34] Colin Wei and Tengyu Ma. Improved sample complexities for deep neural networks and robust classification via an all-layer margin. In *International Conference on Learning Representations*, 2019. 5

[35] Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations*, 2020. 2, 3, 4, 5, 9

[36] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33: 6256–6268, 2020. 1, 2, 3, 6, 7, 9

[37] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 1, 9

[38] Bianca Zadrozny, John Langford, and Naoki Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Third IEEE international conference on data mining*, pages 435–442. IEEE, 2003. 9

[39] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87. URL https://dx.doi.org/10.5244/C.30.87. 7

[40] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[41] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019. 9

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section A in Appendix
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section A in Appendix
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Assumption 1 and Assumption 4.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section C, D and Section E in Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We promise to release the code and checkpoints after the acceptance
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Refer Section 5, and more detail is available in Sec. K in supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] As our experiments are done on neural networks which are expensive to train we haven't included the error bars in result currently. However, we do plan to provide error bars for a certain subset of experiments in Appendix Table 5..
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section G in Appendix
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Section G in Appendix

(b) Did you mention the license of the assets? [Yes] See Section G in Appendix

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]