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

Harsh Rangwani[*1], **Shrinivas Ramasubramanian**[*1], **Sho Takemori**[*2], **Takashi Kato**[2], **Yuhei Umeda**[2], **and R. Venkatesh Babu**[1]

[1]Video Analytics Lab, Indian Institute of Science, Bengaluru, India
[2]Fujitsu Limited, Kanagawa, Japan
harshr@iisc.ac.in, shrinivas.ramasubramanian@gmail.com, takemori.sho@fujitsu.com, kato.takashi_01@fujitsu.com, umeda.yuhei@fujitsu.com, venky@iisc.ac.in

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 accuracy whereas practical machine learning systems can have complex goals (e.g. maximizing the minimum of recall across classes etc.) that are non-decomposable in nature. 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 can better optimize the desired non-decomposable metric utilizing unlabeled data, 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 [4, 13, 35, 43]. 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 [35]. Hence, the development of these algorithms has resulted in a vast reduction in the requirement for 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 [22], augmentations [42, 35], 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 [47, 29] are among the state-of-the-art (SOTA) methods for semi-supervised learning.

[*]Equal Contribution. Code: https://github.com/val-iisc/CostSensitiveSelfTraining

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

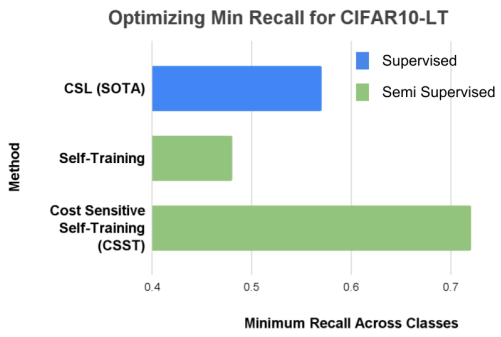


Figure 1: We show a comparison of the SOTA CSL [24] method with the Self-training-based Semi-Supervised methods, for optimizing the minimum recall objective on the CIFAR10-LT dataset. Our proposed CSST framework produces significant gains in the desired metric leveraging additional unlabeled data through our proposed weighted novel consistency regularizer and thresholding mechanism.

Despite the popularity of self-training methods, most of the works [42, 1, 35] 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 [23] used for federated learning, classifier coverage for minority classes for ensuring fairness [8], 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 [24, 34]. However, as supervision can be expensive, in this work we aim to answer the question: *Can we optimize non-decomposable objectives using self-training-based methods developed for semi-supervised learning?*

We first demonstrate that vanilla self-training methods (e.g., FixMatch [35], UDA [42], etc.) can produce unsatisfactory results for 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 [40]. We demonstrate the practical application by optimizing various non-decomposable metrics by plugging existing methods (*e.g.* FixMatch [35] 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).

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 [35], UDA [42] 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 the 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 5 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 metrics relevant to 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 [6] (Tab. 1).

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

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

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,

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

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.

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

Many of these metrics are **non-decomposable**, i.e., one cannot compute these metrics by simply calculating the average of scores on individual examples. Optimizing 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 [24]. Some more complex non-decomposable objectives for classification can be reduced to CSL [26, 36, 24]. 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:

$$\max_F \min_{\boldsymbol{\lambda}\in\Delta_{K-1}} \sum_{i\in[K]} \lambda_i \frac{C_{ii}[F]}{\pi_i}. \tag{3}$$

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 [24, Sec. 2]:

$$\max_F \min_{\boldsymbol{\lambda}\in\mathbb{R}^K_{\geq 0}} \frac{1}{K}\sum_{i\in[K]} C_{ii}[F]/\pi_i + \sum_{j\in[K]} \lambda_j \left(\sum_{i\in[K]} C_{ij}[F] - 0.95/K\right). \tag{4}$$

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 [24].

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 [24]. Following [24], we introduce calibrated loss functions for given gain matrix $\mathbf{G}$. We let $p_m : \mathcal{X} \to \Delta_{K-1} \subset \mathbb{R}^K$ be a prediction function of a model, where Δ_{K-1} is the $K-1$-dimensional probability simplex. For a gain matrix $\mathbf{G}$, the corresponding loss function is given as a combination of logit adjustment [21] and loss re-weighting [28]. We decompose the gain matrix $\mathbf{G}$ as $\mathbf{G} = \mathbf{MD}$, 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 model prediction $p_m(x)$, the hybrid loss is defined as follows:

$$\ell^{\mathrm{hyb}}(y, p_m(x)) = -\sum_{i\in[K]} M_{yi} \log\left(\frac{(p_m(x))_i / D_{ii}}{\sum_{j\in[K]} (p_m(x))_j / D_{jj}}\right). \tag{5}$$

To make the dependence of $\mathbf{G}$ explicit, we also denote $\ell^{\mathrm{hyb}}(y, p_m(x))$ as $\ell^{\mathrm{hyb}}(y, p_m(x); \mathbf{G})$. The average loss on training sample $S \subset \mathcal{X}$ is defined as $\mathcal{L}^{\mathrm{hyb}}(\mathcal{X}) = \frac{1}{|S|}\sum_{x\in S} \ell^{\mathrm{hyb}}(u, p_m(x))$. Narasimhan and Menon [24] proved that the hybrid loss is calibrated, that is learning with ℓ^{hyb} gives the Bayes optimal classifier for $\mathbf{G}$ (c.f., [24, 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 $\ell^{\mathrm{hyb}}(y, p_m(x))$ is denoted by $\ell^{\mathrm{LA}}(y, p_m(x))$.

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 [40, 22, 42, 35]. 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 [40, 35]. 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 [35] and its effectiveness has been proved theoretically [40]. This section provides theoretical analysis of a self-training based SSL algorithm for non-decomposable objectives by generalizing [40]. 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 in Appendix for theoretical results in the paper.

In Sec. 2, we considered non-decomposable metrics and their reduction to cost-sensitive learning objectives defined by Eq. (2) using a gain matrix. In this section, we consider an equivalent objective using the notion of weighted error. For weight matrix $w = (w_{ij})_{1 \leq i,j \leq K}$ 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 distribution $\mathbf{P}(x \mid y = i)$. If $w = \mathrm{diag}(1/K, \ldots, 1/K)$, then this coincides with the usual balanced error [21]. Since $C_{ij}[F] = \mathbf{E}_{(x,y)\sim D}\left[\mathbb{1}(y = i, F(x) = j)\right] = \mathbf{P}(y = i) - \mathbf{P}(y = i)\mathbf{E}_{x \sim P_i}\left[\mathbb{1}(F(x) \neq j)\right]$, we can write:

$$G_{ij} C_{ij}[F] = G_{ij}(\mathbf{P}(y = i) - \mathbf{P}(y = i)\mathbf{E}_{x \sim P_i}\left[\mathbb{1}(F(x) \neq j)\right]) = G_{ij}(\pi_i - \pi_i \mathbf{E}_{x \sim P_i}\left[\mathbb{1}(F(x) \neq j)\right])$$

Here π_i is the class prior $\mathbf{P}(y = i)$ for $1 \leq i \leq K$ as before. Hence CSL objective (2) i.e. $\max_F \sum_{i,j} G_{ij} C_{ij}[F]$ is equivalent to minimizing the negative term above i.e. $G_{ij}\pi_i \mathbf{E}_{x \sim P_i}\left[\mathbb{1}(F(x) \neq j)\right]$ which is same as $\mathrm{Err}_w(F)$ with $w_{ij} = G_{ij}\pi_i$ for $1 \leq i, j \leq K$. Hence, the *notion of weighted error is equivalent to CSL*, which we will also use later for deriving loss functions. We further note that if we add a matrix with the same columns ($c\mathbb{1} \geq 0$) to the gain matrix $\mathbf{G}$, still the maximizers of CSL (2) are the same as the original problem. Hence, without loss of generality, we assume $w_{ij} \geq 0$. We assume $w \neq \mathbf{0}$, i.e., $|w|_1 > 0$ for avoiding degenerate solutions.

In the previous work [40], it is assumed that there exists a ground truth classifier $F^\star : \mathcal{X} \to [K]$ and the supports of distributions $\{P_i\}_{1 \leq i \leq K}$ are disjoint. However, if supports of distributions $\{P_i\}_{1 \leq i \leq K}$ are disjoint, a solution of the minimization problem $\min_F \mathrm{Err}_w(F)$ is independent of w in some cases. More precisely, if $w = \mathrm{diag}(w_1, \ldots, w_K)$ i.e. 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 [24, 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 [40]. See Fig. 2 for an intuitive explanation.

3.2 Weighted Consistency Regularizer

For improving accuracy, the consistency in prediction is equally important across the distributions P_i for $1 \leq i \leq K$ [35, 40]. However, for our case of weighted error, if the entries of the i_0^{th} row of the weight matrix w are larger than the other entries for some i_0, then the consistency of model predictions on examples drawn from the distribution P_{i_0} are more important than those on the other

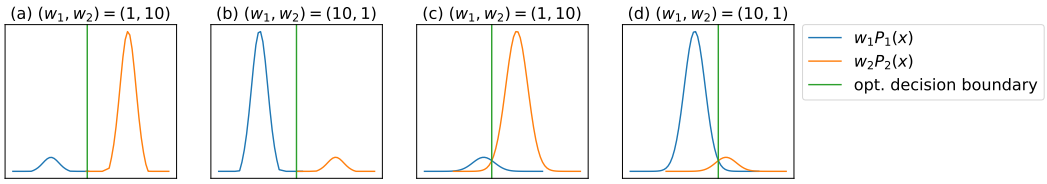


Figure 2: Using a simple example, we explain a difference in theoretical assumptions compared to [40] that assumes $\{P_i\}_{1 \le i \le K}$ have disjoint supports. Here, we consider the case when $K = 2$, $w = \mathrm{diag}(w_1, w_2)$, and P_1 and P_2 are distributions on $\mathcal{X} \subset \mathbb{R}$. (a), (b): In a perfect setting where two distributions P_1 and P_2 have disjoint supports, the Bayes optimal classifier for the CSL is identical to the ground truth classifier ($x \mapsto \mathrm{argmax}_i \, P_i(x)$) for any choices of weights (w_1, w_2). (c), (d): In a more generalized setting, the Bayes optimal classifier $x \mapsto \mathrm{argmax}_i \, w_i P_i(x)$ depends on the choice of weights (i.e., gain matrix). The optimal decision boundary (the green line) for the CSL moves to the right as w_1/w_2 increases.

examples. In this case, we require more restrictive consistency regularizer for distribution for P_{i_0}. Thus, we require a weighted (or cost-sensitive) consistency regularizer, which we define 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:

$$\min_F \mathrm{Err}_w(F) \quad \text{subject to } R_{\mathcal{B},w}(F) \le \beta. \tag{6}$$

A solution of the problem (6) is denoted by F^*. We let $F_{\mathrm{pl}} : \mathcal{X} \to [K]$ a pseudo labeler (a classifier) 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 [35] using consistency regularizer on data augmentation obtains classifier F , that achieves high accuracy (i.e., low balanced errors) for balanced datasets. Also in Appendix 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 [40], 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 $Q(\mathcal{N}(S)) \ge c(Q(S))Q(S)$ for any measurable $S \subseteq \mathcal{X}$.

The c-expansion property implies that if $Q(S)$ decreases, then 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. The c-expansion property (on each P_i) 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 [40], where $a \in (0,1)$ and $\widetilde{c} > 1$ (see Sec. N.2 in Appendix). In addition, we also show that it is also satisfied for mixtures of Gaussians and mixtures of manifolds (see Example 8 in Appendix for more details). Thus, the c-expansion property is a general property satisfied for a wide class of distributions.

ASSUMPTION 3. For 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 for a non-increasing function $c : (0,1] \to [1,\infty)$.

3.4 Cost-Sensitive Self-Training with Weighted Consistency Regularizer

In this section we first introduce the assumptions on the pseudo labeler F_{pl} and then introduce the theoretical Cost-Sensitive Self-Training (CSST) objective. F_{pl} can be any classifier satisfying the following assumption, however, typically it is a classifier trained on a labeled dataset.

ASSUMPTION 4. We assume that $\text{Err}_w(F_{\text{pl}}) + \text{Err}_w(F^*) \leq |w|_1$. Let $\gamma = c(p_w)$, where $p_w = \frac{\text{Err}_w(F_{\text{pl}})+\text{Err}_w(F^*)}{|w|_1}$. We also assume $\gamma > 3$.

Since c is non-increasing, γ (as a function of $\text{Err}_w(F_{\text{pl}})$) is a non-increasing function of $\text{Err}_w(F_{\text{pl}})$ (and $\text{Err}_w(F^*)$). Therefore, the assumption $\gamma > 3$ roughly requires that $\text{Err}_w(F_{\text{pl}})$ is "small". We provide concrete conditions for $\text{Err}_w(F_{\text{pl}})$ that satisfy $\gamma > 3$ in the case of mixture of isotropic d-dimenional Gaussians for a region $\mathcal{B}(x)$ defined by r in Appendix (Example 9). In the example, we show that the condition $\gamma > 3$ is satisfied if $\text{Err}_w(F_{\text{pl}}) < 0.17$ in the case when $r = 1/(2\sqrt{d})$ and satisfied if $\text{Err}_w(F_{\text{pl}}) < 0.33$ in the case when $r = 3/(2\sqrt{d})$, where $\mathcal{X} \subseteq \mathbb{R}^d$. Since we assume $\text{Err}_w(F^*)$ is negligible compared to $\text{Err}_w(F_{\text{pl}})$ (Assumption 1), the former condition in Assumption 4 is approximately equivalent to $\text{Err}_w(F_{\text{pl}}) \leq |w|_1$ which is satisfied by the definition of Err_w.

We define $L^{(i)}_{\text{0-1}}(F, F') = \mathbf{E}_{x\sim P_i}\left[\mathbb{1}(F(x) \neq F'(x))\right]$. Then, we consider the following objective:

$$\min_F \mathcal{L}_w(F), \quad \text{where } \mathcal{L}_w(F) = \frac{\gamma+1}{\gamma-1} L_w(F, F_{\text{pl}}) + \frac{2\gamma}{\gamma-1} R_{\mathcal{B},w}(F). \tag{7}$$

Here $L_w(F, F_{\text{pl}})$ is defined as $\sum_{i,j\in[K]} w_{ij} L^{(i)}_{\text{0-1}}(F, F_{\text{pl}})$. The above objective corresponds to cost-sensitive self-training (with F_{pl}) objective with weighted consistency regularization. We provide following theorem which relates the weighted error of classifier $\hat{F}$ learnt using the above objective to the weighted error of the pseudo labeler (F_{pl}).

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

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

REMARK. Since both $\text{Err}_w(F^*)$ and $R_{\mathcal{B},w}(F^*) \leq \beta$ are negligible compared to $\text{Err}_w(F_{\text{pl}})$ and $\gamma > 3$, Theorem 5 asserts that $\hat{F}$ learnt by minimizing semi-supervised loss $L_w(F, F_{\text{pl}})$ 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. The above theorem is a generalization of [40, Theorem 4.3], which provided a similar result for balanced 0-1-error in the case of distributions with disjoint supports. In Appendix Sec. D, following [40, 39], we also provide a generalization bound for $\text{Err}_w(F)$ using all-layer margin [39] in the case when classifiers are neural networks.

4 CSST in Practice

In the previous section, we proved that by using self-training (CSST), we can achieve a superior classifier $\hat{F}$ in comparison to pseudo labeler F_{pl} through weighted consistency regularization. As we have established the equivalence of the weighted error Err_w to the CSL objective expressed in terms of $\mathbf{G}$ (Sec. 3.1) , we can theoretically optimize a given non-decomposable metric expressed by $\mathbf{G}$ better using CSST, utilizing the additional unlabeled data via self-training and weighted consistency regularization. We now show how CSST can be used in practice for optimizing non-decomposable metrics in the case of neural networks.

The practical self-training methods utilizing consistency regularization (e.g., FixMatch [35], etc.) for semi-supervised learning have supervised loss $\mathcal{L}_s$ for labeled and consistency regularization 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 $\mathcal{L}_s$ can be modified conveniently based on the desired non-decomposable metric by using suitable $\mathbf{G}$ (Sec. 2.1). We will now introduce the novel weighted consistency loss and its corresponding thresholding mechanism for unlabeled data in CSST, used 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 [22, 35, 42] 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 [24] 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:

$$\ell_u^{\mathrm{wt}}(\hat{p}_m(x), p_m(\mathcal{A}(x)), \mathbf{G}) = -\sum_{i=1}^{K}(\mathbf{G}^{\mathbf{T}}\hat{p}_m(x))_i \log(p_m(\mathcal{A}(x))_i) \tag{8}$$

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

As the above loss is similar in nature to the cost sensitive losses introduced by Narasimhan and Menon [24] (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 ($\mathcal{L}_u^{wt}$) for training overparameterized deep networks. We further show in Appendix Sec. B that the above loss ℓ_u^{wt} approximately minimizes the theoretical weighted consistency regularization term $R_{\mathcal{B},w}(F)$ defined in Sec. 3.4.

Threshold Mechanism for CSST. In the usual semi-supervised learning formulation [35] we use the confidence threshold ($\max_i(p_m(x)_i) > \tau$) 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 $\mathbf{G}$ as only a few samples cross the threshold (Fig. 4). 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 ([24] 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. This proposition demonstrates that for a particular sample the high confidence $p_m(x)$ may not be optimal based on $\mathbf{G}$. 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:

$$\mathcal{L}_u^{wt}(B_u) = \frac{1}{|B_u|}\sum_{x\in B_u}\mathbb{1}_{(\mathcal{D}_{KL}(\mathrm{norm}(\mathbf{G}^T\hat{p}_m(x))\ ||\ p_m(x))\leq\tau)}\ell_u^{\mathrm{wt}}(\hat{p}_m(x), p_m(\mathcal{A}(x)), \mathbf{G}) \tag{9}$$

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. 4 and Tab. 3. We will now empirically incorporate CSST by introducing consistency based losses and thresholding mechanism for unlabeled data, into the popular semi-supervised methods of FixMatch [35] and Unsupervised Data Augmentation for Consistency Training (UDA) [42]. The exact expression for the weighted consistency losses utilized for UDA and FixMatch have been provided in the Appendix Sec. G.2 and G.3 .

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 ($\rho = 10$)	
	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

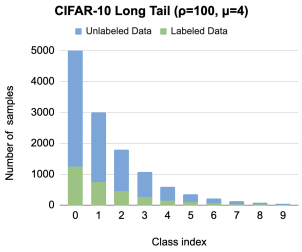


Figure 3: CIFAR-10 Long tail distribution $\rho = 100, \mu = 4$.

5 Experiments

We demonstrate that the proposed CSST framework shows significant gains in performance on both vision and NLP tasks on imbalanced 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 data distribution and the unlabeled samples are much more abundant (μ times) the labeled. The frequency of samples follows an exponentially decaying long-tailed distributed as seen in Fig. 3, which closely imitates the distribution of real-world data [37, 14]. For CIFAR-10 [15], IMDb [20] and DBpedia-14 [18], we use $\rho = 100$ and $\rho = 10$ for CIFAR-100 [15] and ImageNet-100 [31] [2] datasets. We compare our method against supervised methods of ERM, Logit Adjustment (LA) [21] and Cost Sensitive Learning (CSL) [25] 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) [45], specifically WRN-28-2 and WRN-28-8 for CIFAR-10 and CIFAR-100 respectively. For ImageNet, we use a ResNet-50 [9] network for our experiments and finetuned DistilBERT(base uncased) [33] for IMDb and DBpedia-14 datasets. We divide the balanced held-out set for each dataset equally into validation and test sets. A detailed list of hyper-parameters and additional experiments can be found in the Appendix Tab. 6 and Sec. O respectively.

Maximizing Worst-Case Recall. For CIFAR-10, IMDb, and DBpedia-14 datasets, we maximize the minimum recall among all classes (Eq. (3)). Given the low number of samples per class for datasets with larger number of classes like CIFAR-100 and ImageNet-100, we pick objective (10). 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 recall objective can be mathematically formulated as:

$$\max_{F} \min_{(\lambda_{\mathcal{H}}, \lambda_{\mathcal{T}}) \in \Delta_1} \frac{\lambda_{\mathcal{H}}}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \frac{C_{ii}[F]}{\pi_i} + \frac{\lambda_{\mathcal{T}}}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \frac{C_{ii}[F]}{\pi_i}. \tag{10}$$

The corresponding gain matrix $\mathbf{G}$ is 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. (9) with the corresponding ℓ_u^{wt} being substituted by ℓ_u^{LA} as define in Sec. 2.1. Also for labeled samples we use $\mathcal{L}_s^{\text{LA}}$ 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 $\boldsymbol{\lambda}$ 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 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. As the $\mathbf{G}$ is a diagonal matrix for this objective, the proposed KL-Based Thresholding here is equivalent to the confidence based threshold of FixMatch in this case. Despite the equivalence of thresholding mechanism, we see significant gains in min-recall (Tab. 2) just using the regularization term. We discuss their equivalance in Appendix Sec. H.

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

[2] 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

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 $\text{cov}_i[F] = \frac{1}{K}, \forall\, i \in [K]$. Along similar lines to objective (10) we modify the objective (4) to maximize the average recall subject to the both the average head and tail class coverage (HT Coverage) being above a given threshold of $\frac{0.95}{K}$. The $\mathbf{G}$ is non-diagonal here and $G_{ij} = (\mathbb{1}_{j\in\mathcal{H}}\frac{\lambda_{\mathcal{H}}}{|\mathcal{H}|} + \mathbb{1}_{j\in\mathcal{T}}\frac{\lambda_{\mathcal{T}}}{|\mathcal{T}|}) + \frac{\delta_{ij}}{K\pi_i}$.

$$\max_{F} \min_{(\lambda_{\mathcal{H}},\lambda_{\mathcal{T}})\in\mathbb{R}^2_{\geq 0}} \sum_{i\in[K]} \frac{C_{ii}[F]}{K\pi_i} + \lambda_{\mathcal{H}}\left(\sum_{i\in[K],j\in\mathcal{H}} \frac{C_{ij}[F]}{|\mathcal{H}|} - \frac{0.95}{K}\right) + \lambda_{\mathcal{T}}\left(\sum_{i\in[K],j\in\mathcal{T}} \frac{C_{ij}[F]}{|\mathcal{T}|} - \frac{0.95}{K}\right). \tag{11}$$

As these objectives corresponds to a non-diagonal $\mathbf{G}$ as shown in Eq. (4) 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. (9)). 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 further note 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. 5a) 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. 5b 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 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.

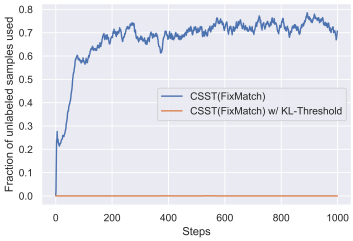
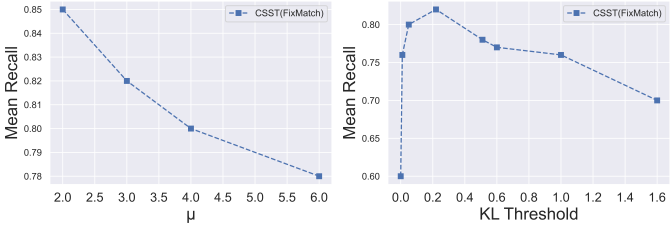


Figure 4: Fraction of unlabelled data used for maximizing average recall under coverage constraints for CIFAR-100 ($\rho = 10, \mu = 4$) (Sec. 5).



(a) (b)

Figure 5: 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)

6 Related Work

Self-Training. Self-training algorithms have been popularly used for the tasks of semi-supervised learning [1, 43, 35, 17] and unsupervised domain adaptation [32, 48]. In recent years several regularizers which enforce consistency in the neighborhood (either an adversarial perturbation [22] or augmentation [42]) 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, unlike the general non-decomposable metrics we consider.

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 [19, 44, 7] and adaptive margin [3, 44] based techniques. For overparameterized models Narasimhan et al. [26] 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 [25, 30, 27]. However most work has focused on linear models leaving scope for works in context of deep neural networks. Sanyal et al. [34] train DNN using reweighting strategies for optimizing metrics, Huang et al. [10] use a reinforcement learning strategy to optimize complex metrics, and Kumar et al. [16] 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 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) that involves minimizing a cost-sensitive error on pseudo labels and consistency regularization. We show theoretically that we can obtain classifiers that can better optimize the desired non-decomposable metric than the original model used for obtaining pseudo labels, under similar data distribution assumptions as used for theoretical analysis of Self-training. We then apply CSST to practical and effective self-training method of FixMatch and UDA, incorporating a novel regularizer and thresholding mechanism based on a 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.

Acknowledgements: This work was supported in part by Fujitsu Research grant. Harsh Rangwani is supported by Prime Minister's Research Fellowship (PMRF).

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. 2, 10

[2] Sergey G Bobkov. An isoperimetric inequality on the discrete cube, and an elementary proof of the isoperimetric inequality in gauss space. *The Annals of Probability*, 25(1):206–214, 1997. 17

[3] 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. 10

[4] 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

[5] Marco Chiani, Davide Dardari, and Marvin K Simon. New exponential bounds and approximations for the computation of error probability in fading channels. *IEEE Transactions on Wireless Communications*, 2(4):840–845, 2003. 20

[6] 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, 16

[7] 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. 10

[8] 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. 2

[9] 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. 8

[10] 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. 10

[11] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-4020. URL https://aclanthology.org/P18-4020. 25

[12] Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546. 29

[13] 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

[14] 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. 8

[15] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. Technical report, University of Toronto. 8, 26

[16] 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. 10

[17] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 10

[18] 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. 8, 26

[19] Yi Lin, Yoonkyung Lee, and Grace Wahba. Support vector machines for classification in nonstandard situations. *Machine learning*, 46(1):191–202, 2002. 10

[20] 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. 8, 26

[21] 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, 8

[22] 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, 4, 7, 10

[23] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019. 2

[24] Harikrishna Narasimhan and Aditya K Menon. Training over-parameterized models with non-decomposable objectives. *Advances in Neural Information Processing Systems*, 34, 2021. 2, 3, 4, 7, 16, 24, 29

[25] 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. 8, 10

[26] 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, 10

[27] 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. 10

[28] 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

[29] 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

[30] Shameem Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. Optimizing f-measures by cost-sensitive classification. *Advances in neural information processing systems*, 27, 2014. 10

[31] 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. 8, 26

[32] 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. 10

[33] 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. 8, 28

[34] 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. 2, 10

[35] 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, 4, 5, 6, 7, 10, 24, 25, 28, 29

[36] 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

[37] 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. 8

[38] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10857–10866, June 2021. 29

[39] 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. 6, 21

[40] 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, 4, 5, 6, 15, 17, 21, 22, 29

[41] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020. 28

[42] 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, 4, 7, 10

[43] 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, 10

[44] 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. 10

[45] 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. 8

[46] Bowen Zhang, Yidong Wang, Wenxin Hou, HAO WU, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18408–18419. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/995693c15f439e3d189b06e89d145dd5-Paper.pdf. 29

[47] 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

[48] 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. 10

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, Assumption 3, 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 will release our code here: https://github.com/val-iisc/CostSensitiveSelfTraining
   (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 Section 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 Tab. 9
   (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 L 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 L in Appendix
   (b) Did you mention the license of the assets? [Yes] See Section L 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]

Appendix

Table of Contents

A Limitations and Negative Societal Impacts **16**

A.1 Limitations of our Work 16

A.2 Negative Societal Impact 16

B Connection between Minimization of Weighted Consistency Regularizer Loss (Eq. (8)) and Theoretical Weighted Consistency ($R_{\mathcal{B},w}(F)$ in Sec. 3.4) **16**

C Additional Examples and Proof of Theorem 5 **16**

C.1 Examples for Theoretical Assumptions 17

C.2 Proof of Theorem 5 Assuming a Lemma 17

C.3 Proof of Lemma 11 19

C.4 Miscellaneous Proofs for Examples 20

D All-Layer Margin Generalization Bounds **21**

E Proof of Proposition 6 **22**

F Notation **24**

G Objective **24**

G.1 Logit Adjusted Weighted Consistency Regularizer 24

G.2 `CSST`(FixMatch) 24

G.3 `CSST`(UDA) 25

H Threshold mechanism for diagonal Gain Matrix **25**

I Dataset **26**

J Algorithms **26**

K Details of Experiments and Hyper-parameters **27**

L Code, License, Assets and Computation Requirements **28**

L.1 Code and Licenses of Assets 28

L.2 Computational Requirements 28

M Statistical Analysis **28**

N Additional Details **29**

N.1 Formal Statement Omitted in Sec. 2.2 29

N.2 Comparison with the $(a, \widetilde{c})$-expansion Property in [40] 29

O Additional Experiments **29**

A Limitations and Negative Societal Impacts

A.1 Limitations of our Work

At this point we only consider optimizing objectives through CSST which can be written as a linear combination of entries of confusion matrix. Although there are important metrics like Recall, Coverage etc. which can be expressed as a linear form of confusion matrix. However, there do exist important metrics like Intersection over Union (IoU), Q-Mean etc. which we don't consider in the current work. We leave this as an open direction for further work.

Also in this work we considered datasets where unlabeled data distribution doesn't significantly differ from the labeled data distribution, developing robust methods which can also take into account the distribution shift between unlabeled and labeled is an interesting direction for future work.

A.2 Negative Societal Impact

Our work has application in fairness domain [6], where it can be used to improve performance of minority sub-groups present in data. These fairness objectives can be practically enforced on neural networks through the proposed CSST framework. However these same algorithms can be tweaked to artifically induce bias in decision making of trained neural networks, for example by ignoring performance of models on certain subgroups. Hence, we suggest deployment of these models after through testing on all sub groups of data.

B Connection between Minimization of Weighted Consistency Regularizer Loss (Eq. (8)) and Theoretical Weighted Consistency ($R_{\mathcal{B},w}(F)$ in Sec. 3.4)

The theoretical weighted consistency regularization can be approximated under strong augmentation $\mathcal{A}$ as $R_{\mathcal{B},w}(F) \approx \sum_{i,j\in[K]} w_{ij}\mathbf{E}_{x\sim P_i}\left[\mathbf{1}(F(\mathcal{A}(x)) \neq F(x))\right]$. Noting that $\mathbf{1}(F(\mathcal{A}(x)) \neq F(x)) \leq \mathbf{1}(F(\mathcal{A}(x)) \neq j) + \mathbf{1}(F(x) \neq j)$ for any class label j, this value of $R_{\mathcal{B},w}(F)$ is bounded as follows:

$$
\begin{aligned}
R_{\mathcal{B},w}(F) &\approx \sum_{i,j\in[K]} w_{ij}\mathbf{E}_{x\sim P_i}\left[\mathbf{1}(F(\mathcal{A}(x)) \neq F(x))\right] \\
&\leq \sum_{i,j\in[K]} w_{ij}\mathbf{E}_{x\sim P_i}\left[\mathbf{1}(F(\mathcal{A}(x)) \neq j))\right] + \sum_{i,j\in[K]} w_{ij}\mathbf{E}_{x\sim P_i}\left[\mathbf{1}(F(x) \neq j)\right].
\end{aligned}
$$

Assuming $F(x)$ to be composed of a classifier made of neural network model output $p_m(x)$. For consistency regularization, we choose samples which have a low second term by thresholding on confidence for which we obtain the pseudo-label $F(x) = \hat{p}_m(x)$. In that case the minimization of (the upper bound of) $R_{\mathcal{B},w}(F)$ is approximately equivalent to cost-sensitive learning (CSL):

$$
\min_F \sum_{i,j\in[K]} w_{ij}\mathbf{E}_x\left[\mathbf{1}(F(\mathcal{A}(x)) \neq j))\right].
$$

The above objective can be equivalently expressed in form of gain matrix $\mathbf{G}$ where $G_{i,j} = w_{i,j}/(\pi_i)$. Hence, the above minimization can be effectively done for different gain matrices based on corresponding non-decomposable metric 2.1. Following Narasimhan and Menon [24], we can effectively minimize the $R_{\mathcal{B},w}(F)$ using the calibrated CSL loss function by plugging in the pseudo label $\hat{p}_m(x)$ in place of ground truth label ($\mathbf{y}$). The final weighted regularization function can be seen as:

$$
\ell^{hyb}(\hat{p}_m(x), p_m(\mathcal{A}(x)); \mathbf{G}) \tag{12}
$$

The above regularization loss is same as the weighted consistency regularization loss in Eq. (8), which leads to minimization of $\mathcal{D}_{KL}(\text{norm}(\mathbf{G}^{\mathbf{T}}\hat{p}_m(x))||p_m(\mathcal{A}(x)))\ \forall x \in \mathcal{X}$ as seen in Prop. 6. Hence, the minimization of theoretical consistency regularization term is achieved by minimizing the weighted consistency loss.

C Additional Examples and Proof of Theorem 5

In this section, we provide some examples for assumptions introduced in Sec. 3 and a proof of Theorem 5. We provide proof of these examples in Sec. C.4.

C.1 Examples for Theoretical Assumptions

The following example (Example 8) shows that the c-expansion (Definition 2) property is satisfied for mixtures of Gaussians and mixtures of manifolds.

EXAMPLE 8. By [40, Examples 3.4, 3.5], the c-expansion property is satisfied for mixtures of isotropic Gaussian distributions and mixtures of manifolds. More precisely, in the case of mixtures of isotropic Gaussian distributions, i.e., if Q is given as mixtures of $\mathcal{N}(\tau_i, \frac{1}{d}I_{d\times d})$ for $i = 1, \dots, n$ with some $n \in \mathbb{Z}_{\geq 1}$ and $\tau_i \in \mathbb{R}^d$, and $\mathcal{B}(x)$ is an ℓ_2-ball with radius r then by [2, (13)] and [40, Section B.2], Q satisfies the c-expansion property with $c(p) = R_h(p)/p$ for $p > 0$ and $h = 2r\sqrt{d}$ (c.f., [40, section B.2]). Here $R_h(p) = \Phi(\Phi^{-1}(p) + h)$ and Φ is the cumulative distribution function of the standard normal distribution on $\mathbb{R}$.

In Sec. 3.4, we required the assumption that $\gamma > 3$ (Assumption 4) and remarked that it roughly requires $\mathrm{Err}_w(F_{\mathrm{pl}})$ is "small". The following example provides explicit conditions for $\mathrm{Err}_w(F_{\mathrm{pl}})$ that satisfy the assumption using a toy example.

EXAMPLE 9. Using a toy example provided in Example 8, we provide conditions that satisfy the assumption $\gamma > 3$ approximately. To explain the assumption $\gamma > 3$, we assume that $\mathcal{P}_w$ is given as a mixture of isotropic Gaussians and $\mathcal{B}(x)$ is ℓ_2-ball with radius r as in Example 8. Furthermore, we assume that $|w|_1 = 1$ and $\mathrm{Err}_w(F^*)$ is sufficiently small compared to $\mathrm{Err}_w(F_{\mathrm{pl}})$. Then, $p_w = \mathrm{Err}_w(F_{\mathrm{pl}}) + \mathrm{Err}_w(F^*) \approx \mathrm{Err}_w(F_{\mathrm{pl}})$. Using this approximation, since $\mathcal{P}_w$ satisfies the c-expansion property with $c(p) = R_{2r\sqrt{d}}(p)/p$, if $r = \frac{1}{2\sqrt{d}}$ then, the condition $\gamma > 3$ is satisfied when $\mathrm{Err}_w(F_{\mathrm{pl}}) < 0.17$. If $r = \frac{3}{2\sqrt{d}}$ then, the condition $\gamma > 3$ is satisfied when $\mathrm{Err}_w(F_{\mathrm{pl}}) < 0.33$.

In Assumption 1, we assumed that both of $\mathrm{Err}_w(F^*)$ and $R_{\mathcal{B},w}(F^*)$ are small. The following example suggests the validity of this assumption.

EXAMPLE 10. In this example, we assume w is a diagonal matrix $\mathrm{diag}(w_1, \dots, w_K)$. For simplicity, we normalize w so that $\sum_{i\in[K]} w_i = 1$. As in [40, Example 3.4], we assume that P_i is given as isotropic Gaussian distribution $\mathcal{N}(\tau_i, \frac{1}{d}I_{d\times d})$ with $\tau_i \in \mathbb{R}^d$ for $i = 1, \dots, K$ and $\mathcal{B}(x)$ is an ℓ^2-ball with radius $\frac{1}{2\sqrt{d}}$. Furthermore, we assume $\inf_{1\leq i<j\leq K} \|\tau_i - \tau_j\|_2 \gtrsim \frac{\sqrt{\log d}}{\sqrt{d}}$ and $\sup_{i,j\in[K]} \frac{w_i}{w_j} = o(d)$, where the latter assumption is valid for high dimensional datasets (e.g., image datasets). Then it can be proved that there exists a classifier F such that $R_{\mathcal{B},w}(F) = O(\frac{1}{d^c})$ and $\mathrm{Err}_w(F) = O(\frac{1}{d^c})$, where $c > 0$ is a constant (we can take F as the Bayes-optimal classifier for Err_w). Thus, this suggests that Assumption 1 is valid for datasets with high dimensional instances.

The statement of Example 9 follows from numerical computation of $R_{2r\sqrt{d}}(p)/p$. We provide proofs of Examples 8 and 10 in Sec. C.4.

C.2 Proof of Theorem 5 Assuming a Lemma

Theorem 5 can be deduced from the following lemma (by taking $H = F^*$ and $\mathcal{L}_{Q,H}(\widehat{F}) \leq \mathcal{L}_{Q,H}(F^*)$), which provides a similar result to [40, Lemma A.8].

LEMMA 11. *Let H be a classifier and Q a probability measure on $\mathcal{X}$ satisfying c-expansion property. We put $\gamma_H = c(Q(\{x \in \mathcal{X} : F_{\mathrm{pl}}(x) \neq H(x)\}))$. For a classifier F, we define $\mathcal{S}_{\mathcal{B}}(F)$ by $\mathcal{S}_{\mathcal{B}}(F) = \{x \in \mathcal{X} : F(x) = F(x') \quad \forall x' \in \mathcal{B}(x)\}$. For a classifier F, we define $\mathcal{L}_{Q,H}(F)$ by*

$$\frac{\gamma_H + 1}{\gamma_H - 1} Q(\{x \in \mathcal{X} : F(x) \neq F_{\mathrm{pl}}(x)\}) + \frac{2\gamma_H}{\gamma_H - 1} Q(\mathcal{S}^c_{\mathcal{B}}(F)) + \frac{2\gamma_H}{\gamma_H - 1} Q(\mathcal{S}^c_{\mathcal{B}}(H)) - Q(\{x \in \mathcal{X} : F_{\mathrm{pl}}(x) \neq H(x)\}),$$

where $\mathcal{S}^c_{\mathcal{B}}(F)$ denotes the complement of $\mathcal{S}_{\mathcal{B}}(F)$. Then, we have $Q(\{x \in \mathcal{X} : F(x) \neq H(x)\}) \leq \mathcal{L}_{Q,H}(F)$ for any classifier F.

In this subsection, we provide a proof of Theorem 5 assuming Lemma 11. We provide a proof of the lemma in the next subsection. For a classifier F, we define $\mathcal{M}(F)$ as $\{x \in \mathcal{X} : F(x) \neq F^*(x)\}$ and $\mathcal{M}_{\mathrm{pl}}(F)$ as $\{x \in \mathcal{X} : F(x) \neq F_{\mathrm{pl}}(x)\}$. We define $\widetilde{\mathcal{L}}_w(F)$ by

$$\widetilde{\mathcal{L}}_w(F) = \mathcal{L}_w(F) + \frac{2\gamma}{\gamma - 1} R_{\mathcal{B},w}(F^*) - \mathcal{P}_w(\{x \in \mathcal{X} : F_{\mathrm{pl}}(x) \neq F^*(x)\}).$$

We note that $\widetilde{\mathcal{L}}_w(F) - \mathcal{L}_w(F)$ does not depend on F.

Proof of Theorem 5. We let $Q = \mathcal{P}_w$ and $H = F^*$ in Lemma 11 and denote γ_H in the lemma by γ'. Since $w_{ij} \geq 0$ and $\mathbf{E}_{x\sim P_i}[\mathbb{1}(F_{\mathrm{pl}}(x) \neq F^*(x))] \leq \mathbf{E}_{x\sim P_i}[\mathbb{1}(F_{\mathrm{pl}}(x) \neq j)] + \mathbf{E}_{x\sim P_i}[\mathbb{1}(F^*(x) \neq j)]$ for any i, j, we have the following:

$$\begin{aligned}
|w|_1 \mathcal{P}_w(\mathcal{M}(F_{\mathrm{pl}})) &= \sum_{i,j\in[K]} w_{ij} \mathbf{E}_{x\sim P_i}[\mathbb{1}(F_{\mathrm{pl}}(x) \neq F^*(x))] \\
&\leq \sum_{i,j\in[K]} w_{ij} \left\{\mathbf{E}_{x\sim P_i}[\mathbb{1}(F_{\mathrm{pl}}(x) \neq j)] + \mathbf{E}_{x\sim P_i}[\mathbb{1}(F^*(x) \neq j)]\right\} \\
&= \mathrm{Err}_w(F_{\mathrm{pl}}) + \mathrm{Err}_w(F^*).
\end{aligned}$$

Here, the first equation follows from the definition of $\mathcal{P}_w$. Thus, we obtain $\mathcal{P}_w(\mathcal{M}(F_{\mathrm{pl}})) \leq p_w$. By definition of γ and γ' ($\gamma = c(p_w)$ and $\gamma' = c(\mathcal{P}_w(\mathcal{M}(F_{\mathrm{pl}})))$) and the assumption that c is non-increasing, we have the following:

$$\gamma \leq \gamma'. \tag{13}$$

We note that

$$\begin{aligned}
\mathrm{Err}_w(F) &= \sum_{i,j\in[K]} w_{ij} \mathbf{E}_{x\sim P_i}[\mathbb{1}(F(x) \neq j)] \\
&\leq \sum_{i,j\in[K]} w_{ij} \mathbf{E}_{x\sim P_i}[\mathbb{1}(F(x) \neq F^*(x))] + \sum_{i,j\in[K]} w_{ij} \mathbf{E}_{x\sim P_i}[\mathbb{1}(F^*(x) \neq j)] \\
&= |w|_1 \mathcal{P}_w(\mathcal{M}(F)) + \mathrm{Err}_w(F^*).
\end{aligned} \tag{14}$$

Here the first inequality holds since $\mathbb{1}(F(x) \neq j) \leq \mathbb{1}(F(x) \neq F^*(x)) + \mathbb{1}(F^*(x) \neq j)$ for any x and j. By (14) and Lemma 11, the error is upper bounded as follows:

$$\begin{aligned}
\mathrm{Err}_w(F) \leq \mathrm{Err}_w(F^*) &+ \frac{\gamma'+1}{\gamma'-1}|w|_1 \mathcal{P}_w(\mathcal{M}_{\mathrm{pl}}(F)) \\
&+ \frac{2\gamma'}{\gamma'-1}|w|_1 \mathcal{P}_w(\mathcal{S}^c_{\mathcal{B}}(F)) + \frac{2\gamma'}{\gamma'-1}|w|_1 \mathcal{P}_w(\mathcal{S}^c_{\mathcal{B}}(F^*)) - |w|_1 \mathcal{P}_w(\mathcal{M}(F_{\mathrm{pl}})).
\end{aligned}$$

By (13), we obtain

$$\begin{aligned}
\mathrm{Err}_w(F) \leq \mathrm{Err}_w(F^*) &+ \frac{\gamma+1}{\gamma-1}|w|_1 \mathcal{P}_w(\mathcal{M}_{\mathrm{pl}}(F)) \\
&+ \frac{2\gamma}{\gamma-1}|w|_1 \mathcal{P}_w(\mathcal{S}^c_{\mathcal{B}}(F)) + \frac{2\gamma}{\gamma-1}|w|_1 \mathcal{P}_w(\mathcal{S}^c_{\mathcal{B}}(F^*)) - |w|_1 \mathcal{P}_w(\mathcal{M}(F_{\mathrm{pl}})).
\end{aligned} \tag{15}$$

By definition of $\mathcal{L}_w$ and letting $F = \widehat{F}$, we have the following:

$$\begin{aligned}
&\mathrm{Err}_w(\widehat{F}) \leq \mathrm{Err}_w(F^*) + \mathcal{L}_w(\widehat{F}) + \frac{2\gamma}{\gamma-1} R_{\mathcal{B},w}(F^*) - |w|_1 \mathcal{P}_w(\{x \in \mathcal{X} : F_{\mathrm{pl}}(x) \neq F^*(x)\}) \\
&\leq \mathrm{Err}_w(F^*) + \mathcal{L}_w(F^*) + \frac{2\gamma}{\gamma-1} R_{\mathcal{B},w}(F^*) - |w|_1 \mathcal{P}_w(\{x \in \mathcal{X} : F_{\mathrm{pl}}(x) \neq F^*(x)\}) \\
&= \mathrm{Err}_w(F^*) + \frac{2}{\gamma-1}|w|_1 \mathcal{P}_w(\mathcal{M}(F_{\mathrm{pl}})) + \frac{4\gamma}{\gamma-1} R_{\mathcal{B},w}(F^*) \\
&\leq \mathrm{Err}_w(F^*) + \frac{2}{\gamma-1}(\mathrm{Err}_w(F_{\mathrm{pl}}) + \mathrm{Err}_w(F^*)) + \frac{4\gamma}{\gamma-1} R_{\mathcal{B},w}(F^*) \\
&= \frac{2}{\gamma-1}\mathrm{Err}_w(F_{\mathrm{pl}}) + \frac{\gamma+1}{\gamma-1}\mathrm{Err}_w(F^*) + \frac{4\gamma}{\gamma-1} R_{\mathcal{B},w}(F^*).
\end{aligned}$$

Here, the second inequality holds since $\widehat{F}$ is a minimizer of $\mathcal{L}_w$, the third inequality follows from $\mathbb{1}(F^*(x) \neq F_{\mathrm{pl}}(x)) \leq \mathbb{1}(F^*(x) \neq j) + \mathbb{1}(F_{\mathrm{pl}}(x) \neq j)$ for any j. Thus, we have the assertion of the theorem. □

C.3 Proof of Lemma 11

We decompose $\mathcal{M}(F) \cap \mathcal{S}_{\mathcal{B}}(F) \cap \mathcal{S}_{\mathcal{B}}(H)$ into the following three sets:

$$\begin{aligned}
\mathcal{N}_1 &= \{x \in \mathcal{S}_{\mathcal{B}}(F) \cap \mathcal{S}_{\mathcal{B}}(H) : F(x) = F_{\mathrm{pl}}(x), \text{ and } F_{\mathrm{pl}}(x) \neq H(x)\}, \\
\mathcal{N}_2 &= \{x \in \mathcal{S}_{\mathcal{B}}(F) \cap \mathcal{S}_{\mathcal{B}}(H) : F(x) \neq F_{\mathrm{pl}}(x), F_{\mathrm{pl}}(x) \neq H(x), \text{ and } F(x) \neq H(x)\}, \\
\mathcal{N}_3 &= \{x \in \mathcal{S}_{\mathcal{B}}(F) \cap \mathcal{S}_{\mathcal{B}}(H) : F(x) \neq F_{\mathrm{pl}}(x) \text{ and } F_{\mathrm{pl}}(x) = H(x)\}.
\end{aligned}$$

LEMMA 12. *Let $S = \mathcal{S}_{\mathcal{B}}(F) \cap \mathcal{S}_{\mathcal{B}}(H)$ and $V = \mathcal{M}(F) \cap \mathcal{M}(F_{\mathrm{pl}}) \cap S$. Then, we have $\mathcal{N}(V) \cap \mathcal{M}^c(F) \cap S = \emptyset$ and $\mathcal{N}(V) \cap \mathcal{M}^c(F_{\mathrm{pl}}) \cap S \subseteq \mathcal{M}_{\mathrm{pl}}(F)$. Here $\mathcal{M}_{\mathrm{pl}}(F)$ is defined as $\{x \in \mathcal{X} : F(x) \neq F_{\mathrm{pl}}(x)\}$.*

Proof. We take any element x in $\mathcal{N}(V) \cap S$. Since $x \in \mathcal{N}(V)$ and definition of neighborhoods, there exists $x' \in \mathcal{M}(F_{\mathrm{pl}}) \cap \mathcal{M}(F) \cap S$ such that $\mathcal{B}(x) \cap \mathcal{B}(x') \neq \emptyset$. Since $x, x' \in \mathcal{S}_{\mathcal{B}}(F)$, F takes the same values on $\mathcal{B}(x)$ and $\mathcal{B}(x')$. By $\mathcal{B}(x) \cap \mathcal{B}(x') \neq \emptyset$, F takes the same value on $\mathcal{B}(x) \cup \mathcal{B}(x')$. It follows that $F(x) = F(x')$. Since we have $x, x' \in \mathcal{S}_{\mathcal{B}}(H)$, similarly, we see that $H(x) = H(x')$. By $x' \in \mathcal{M}(F)$, we have $F(x) = F(x') \neq H(x') = H(x)$. Therefore, we have $x \in \mathcal{M}(F)$. Thus, we have proved that $x \in \mathcal{N}(V) \cap S$ implies $x \in \mathcal{M}(F)$, i.e., $\mathcal{N}(V) \cap S \subseteq \mathcal{M}(F)$. Therefore, we have $\mathcal{N}(V) \cap \mathcal{M}^c(F) \cap S = \emptyset$. This completes the proof of the first statement. Next, we assume $x \in \mathcal{N}(V) \cap \mathcal{M}^c(F_{\mathrm{pl}}) \cap S$. Then, we have $F(x) \neq H(x)$ and $F_{\mathrm{pl}}(x) = H(x)$. Therefore, we obtain $F(x) \neq F_{\mathrm{pl}}(x)$. This completes the proof. □

LEMMA 13. *Suppose that assumptions of Lemma 11 hold. We define q as follows:*

$$q = \frac{Q(\mathcal{M}_{\mathrm{pl}}(F) \cup \mathcal{S}_{\mathcal{B}}^c(F) \cup \mathcal{S}_{\mathcal{B}}^c(H))}{\gamma_H - 1}. \tag{16}$$

Then, we have $Q(\mathcal{S}_{\mathcal{B}}(F) \cap \mathcal{S}_{\mathcal{B}}(H) \cap \mathcal{M}(F_{\mathrm{pl}}) \cap \mathcal{M}(F)) \leq q$. In particular, noting that $\mathcal{N}_1 \cup \mathcal{N}_2 \subseteq \mathcal{S}_{\mathcal{B}}(F) \cap \mathcal{S}_{\mathcal{B}}(H) \cap \mathcal{M}(F_{\mathrm{pl}}) \cap \mathcal{M}(F)$, we have $Q(\mathcal{N}_1 \cup \mathcal{N}_2) \leq q$.

Proof. We let $S = \mathcal{S}_{\mathcal{B}}(F) \cap \mathcal{S}_{\mathcal{B}}(H)$ and $V = \mathcal{M}(F) \cap \mathcal{M}(F_{\mathrm{pl}}) \cap S$ as before. Then by Lemma 12, we have

$$\begin{aligned}
\mathcal{N}(V) \cap V^c \cap S &= (\mathcal{N}(V) \cap \mathcal{M}^c(F) \cap S) \cup (\mathcal{N}(V) \cap \mathcal{M}^c(F_{\mathrm{pl}}) \cap S) \\
&\subseteq \emptyset \cup \mathcal{M}_{\mathrm{pl}}(F) = \mathcal{M}_{\mathrm{pl}}(F).
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\mathcal{N}(V) \cap V^c &= \mathcal{N}(V) \cap V^c \cap (S \cup S^c) \\
&= (\mathcal{N}(V) \cap V^c \cap S) \cup (\mathcal{N}(V) \cap V^c \cap S^c) \\
&\subseteq \mathcal{M}_{\mathrm{pl}}(F) \cup S^c.
\end{aligned}$$

Thus, by the c-expansion property, we have

$$\begin{aligned}
Q(\mathcal{M}_{\mathrm{pl}}(F) \cup S^c) &\geq Q(\mathcal{N}(V) \cap V^c) \\
&\geq Q(\mathcal{N}(V)) - Q(V) \\
&\geq (c(Q(V)) - 1)\, Q(V).
\end{aligned}$$

Since $V \subseteq \mathcal{M}(F_{\mathrm{pl}})$, c is non-increasing, and $\gamma_H > 1$, we have $Q(V) \leq Q(\mathcal{M}_{\mathrm{pl}}(F) \cup S^c)/(\gamma_H - 1) \leq q$. This completes the proof. □

The following lemma provides an upper bound of $Q(\mathcal{N}_3)$.

LEMMA 14. *Suppose that the assumptions of Lemma 11 hold. We have*

$$Q(\mathcal{N}_3) \leq q + Q(\mathcal{S}_{\mathcal{B}}^c(F) \cup \mathcal{S}_{\mathcal{B}}^c(H)) + Q(\mathcal{M}_{\mathrm{pl}}(F)) - Q(\mathcal{M}(F_{\mathrm{pl}})),$$

where q is defined by (16).

Proof. We let $S = \mathcal{S}_{\mathcal{B}}(F) \cap \mathcal{S}_{\mathcal{B}}(H)$. First, we prove

$$\mathcal{N}_3 \sqcup \left(\mathcal{M}_{\mathrm{pl}}^c(F) \cap S\right) = \mathcal{N}_1 \sqcup \left(\mathcal{M}^c(F_{\mathrm{pl}}) \cap S\right). \tag{17}$$

Here, for sets A, B, we denote union $A \cup B$ by $A \sqcup B$ if the union is disjoint. By definition, we have $\mathcal{N}_1 = S \cap \mathcal{M}_{\mathrm{pl}}^c(F) \cap \mathcal{M}(F_{\mathrm{pl}})$ and $\mathcal{N}_3 = S \cap \mathcal{M}_{\mathrm{pl}}(F) \cap \mathcal{M}^c(F_{\mathrm{pl}})$. Thus, we have

$$\begin{aligned}
&\mathcal{N}_3 \cup \left(\mathcal{M}_{\mathrm{pl}}^c(F) \cap S\right) \\
&= (S \cap \mathcal{M}_{\mathrm{pl}}(F) \cap \mathcal{M}^c(F_{\mathrm{pl}})) \cup \left(\mathcal{M}_{\mathrm{pl}}^c(F) \cap S\right) \\
&= S \cap \left\{(\mathcal{M}_{\mathrm{pl}}(F) \cap \mathcal{M}^c(F_{\mathrm{pl}})) \cup \mathcal{M}_{\mathrm{pl}}^c(F)\right\} \\
&= S \cap \left(\mathcal{M}^c(F_{\mathrm{pl}}) \cup \mathcal{M}_{\mathrm{pl}}^c(F)\right).
\end{aligned}$$

Similarly, we have the following:

$$\begin{aligned}
&\mathcal{N}_1 \cup (\mathcal{M}^c(F_{\mathrm{pl}}) \cap S) \\
&= \left(S \cap \mathcal{M}_{\mathrm{pl}}^c(F) \cap \mathcal{M}(F_{\mathrm{pl}})\right) \cup (\mathcal{M}^c(F_{\mathrm{pl}}) \cap S) \\
&= S \cap \left(\mathcal{M}^c(F_{\mathrm{pl}}) \cup \mathcal{M}_{\mathrm{pl}}^c(F)\right).
\end{aligned}$$

Since disjointness is obvious by definition, we obtain (17). Next, we note that the following holds:

$$\begin{aligned}
Q(\mathcal{M}_{\mathrm{pl}}^c(F) \cap S) &= Q(\mathcal{M}_{\mathrm{pl}}^c(F)) - Q(\mathcal{M}_{\mathrm{pl}}^c(F) \cap S^c) \\
&\geq Q(\mathcal{M}_{\mathrm{pl}}^c(F)) - Q(S^c). \qquad (18)
\end{aligned}$$

By (17), we obtain the following:

$$\begin{aligned}
Q(\mathcal{N}_3) &= Q(\mathcal{N}_1) + Q(\mathcal{M}^c(F_{\mathrm{pl}}) \cap S) - Q(\mathcal{M}_{\mathrm{pl}}^c(F) \cap S) \\
&\leq Q(\mathcal{N}_1) + Q(\mathcal{M}^c(F_{\mathrm{pl}})) - Q(\mathcal{M}^c(F_{\mathrm{pl}}) \cap S) \\
&\leq Q(\mathcal{N}_1) + Q(\mathcal{M}^c(F_{\mathrm{pl}})) - Q(\mathcal{M}_{\mathrm{pl}}^c(F)) + Q(S^c) \\
&\leq q + Q(\mathcal{M}^c(F_{\mathrm{pl}})) - Q(\mathcal{M}_{\mathrm{pl}}^c(F)) + Q(S^c) \\
&= q - Q(\mathcal{M}(F_{\mathrm{pl}})) + Q(\mathcal{M}_{\mathrm{pl}}(F)) + Q(S^c).
\end{aligned}$$

Here the second inequality follows from (18) and the third inequality follows from Lemma 13. This completes the proof. □

Now, we can prove Lemma 11 as follows.

Proof of Lemma 11.

$$\begin{aligned}
Q(\mathcal{M}(F)) &= Q(\mathcal{M}(F) \cap \mathcal{S}_{\mathcal{B}}(F) \cap \mathcal{S}_{\mathcal{B}}(H)) + Q\left(\mathcal{M}(F) \cap (\mathcal{S}_{\mathcal{B}}^c(F) \cup \mathcal{S}_{\mathcal{B}}^c(H))\right) \\
&\leq Q(\mathcal{N}_1 \cup \mathcal{N}_2) + Q(\mathcal{N}_3) + Q(\mathcal{S}_{\mathcal{B}}^c(F) \cup \mathcal{S}_{\mathcal{B}}^c(H)) \\
&\leq 2q + 2Q(\mathcal{S}_{\mathcal{B}}^c(F) \cup \mathcal{S}_{\mathcal{B}}^c(H)) + Q(\mathcal{M}_{\mathrm{pl}}(F)) - Q(\mathcal{M}(F_{\mathrm{pl}})).
\end{aligned}$$

Here, the last inequality follows from Lemmas 13 and 14. Since q satisfies $q \leq \frac{Q(\mathcal{M}_{\mathrm{pl}}(F)) + Q(\mathcal{S}_{\mathcal{B}}^c(F) + Q(\mathcal{S}_{\mathcal{B}}^c(H)))}{\gamma_H - 1}$ by (16), we have our assertion. □

C.4 Miscellaneous Proofs for Examples

Proof of Example 8. In Example 8, we stated that $p \mapsto R_h(p)/p$ is non-increasing. This follows from the concavity of R_h and $\lim_{p \to +0} R_h(p) = 0$. In fact, we can prove the concavity of R_h by $\frac{d^2 R_h}{dp^2}(p) = -h \exp\left(\frac{\xi^2}{2} - h\xi - \frac{1}{2}h^2\right) \leq 0$, where $\xi = \Phi^{-1}(p)$. □

Proof of Example 10. For each $i, j \in [K]$, $w_i P_i(x) \geq w_j P_j(x)$ is equivalent to $(x - \tau_i) \cdot v_{ji} \leq \frac{\|\tau_i - \tau_j\|}{2} + \frac{2(\log w_i - \log w_j)}{d\|\tau_i - \tau_j\|}$, where $v_{ji} = \frac{\tau_j - \tau_i}{\|\tau_i - \tau_j\|}$. Thus, for each $i \in [K]$, we have $\bigcap_{j \in [K] \setminus \{i\}} X_{ij} \subseteq \mathcal{S}_{\mathcal{B}}(F_{\mathrm{opt}})$. Here X_{ij} is defined as $\{x \in \mathcal{X} : (x - \tau_i) \cdot v_{ji} \leq \frac{\|\tau_i - \tau_j\|}{2} + \frac{2(\log w_i - \log w_j)}{d\|\tau_i - \tau_j\|} - \frac{r}{2}\}$. For any $w \in \mathbb{R}^d$ with $\|w\|_2 = \sqrt{d}$ and $a > 0$, we have $P_i(\{x \in \mathcal{X} : (x - \tau_i) \cdot w > a\}) = 1 - \Phi(a) \leq \frac{1}{2}\exp(-a^2/2)$ (c.f., [5]). Thus, $P_i(X_{ij}^c) \leq \frac{1}{2}\exp(-da_{ij}^2/2)$, where $a_{ij} = \frac{\|\tau_i - \tau_j\|}{2} + \frac{2(\log w_i - \log w_j)}{d\|\tau_i - \tau_j\|} - \frac{r}{2}$. By assumptions, we have $a_{ij}\sqrt{d} \gtrsim \sqrt{\log d}$. Therefore, $P_i(X_{ij}^c) = O(\frac{1}{poly(d)})$. It follows that $P_i(\mathcal{S}_{\mathcal{B}}^c(F_{\mathrm{opt}})) \leq \sum_{j \in [K] \setminus \{i\}} P_i(X_{ij}^c) = O(\frac{1}{poly(d)})$. Thus, we have $R_{\mathcal{B},w}(F_{\mathrm{opt}}) = O(\frac{1}{poly(d)})$. By the same way, we can prove that $\mathrm{Err}_w(F_{\mathrm{opt}}) = O(\frac{1}{poly(d)})$. □

D All-Layer Margin Generalization Bounds

Following [40, 39], we introduce all layer margin of neural networks and provide generalization bounds of CSST. In this section, we assume that classifier $F(x)$ is given as $F(x) = \mathrm{argmax}_{1\le i\le K}\,\Phi_i(x)$, where Φ is a neural network of the form

$$\Phi(x) = (f_p \circ f_{p-1} \circ \cdots \circ f_1)(x).$$

Here $f_i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$ with $d_0 = d$ and $d_p = K$. We assume that each f_i belongs to a function class $\mathcal{F}_i \subset \mathrm{Map}(\mathbb{R}^{d_{i-1}}, \mathbb{R}^{d_i})$. We define a function class $\mathcal{F}$ to which Φ belongs by

$$\mathcal{F} = \{\Phi : \mathbb{R}^d \to \mathbb{R}^K : \Phi(x) = (f_p \circ \cdots \circ f_1)(x), \quad f_i \in \mathcal{F}_i, \forall i\}.$$

For example, for $b > 0$, $\mathcal{F}_i$ is given as $\{h \mapsto W\phi(h) : W \in \mathbb{R}^{d_{i-1}\times d_i}, \|W\|_{\mathrm{fro}} \le b\}$ if $i > 1$ and $\{h \mapsto Wh : W \in \mathbb{R}^{d_0\times d_1}, \|W\|_{\mathrm{fro}} \le b\}$ if $i = 1$, where ϕ is a link function (applied on $\mathbb{R}^{d_i}$ entry-wise) with bounded operator norm (i.e., $\|\phi\|_{\mathrm{op}} := \sup_{x\in\mathbb{R}^{d_i}\setminus\{0\}} \|\phi(x)\|_2/\|x\|_2 < \infty$) and $\|W\|_{\mathrm{fro}}$ denotes the Frobenius norm of the matrix. However, we do not assume the function class $\mathcal{F}_i$ does not have this specific form. We assume that each function class $\mathcal{F}_i$ is a normed vector space with norm $\|\cdot\|$. In the example above, we consider the operator norm, i.e., if $f(h) = \phi(Wh)$, $\|f\|$ is defined as $\|f\|_{\mathrm{op}}$. Let $x_1, \ldots, x_n$ be a finite i.i.d. sequence of samples drawn from $\mathcal{P}_w$. We denote the corresponding empirical distribution by $\widehat{P}_w$, i.e., for a measurable function f on $\mathcal{X}$, $\mathbf{E}_{x\sim\widehat{P}_w}[f] = \sum_{i=1}^n f(x_i)$.

For $\xi = (\xi_1, \ldots, \xi_p) \in \prod_{i=1}^p \mathbb{R}^{d_i}$, we define the perturbed output $\Phi(x, \xi)$ as $\Phi(x, \xi) = h_p(x, \xi)$, where

$$\begin{aligned} h_1(x,\xi) &= f_1(x) + \xi_1\|x\|_2, \\ h_i(x,\xi) &= f_i(h_{i-1}(x,\xi)) + \xi_i\|h_{i-1}(x,\xi)\|_2, \quad \text{for } 2 \le i \le p. \end{aligned}$$

Let $x \in \mathcal{X}$ and $y \in [K]$. We define $\Xi(\Phi, x, y)$ by $\{\xi \in \prod_{i=1}^p \mathbb{R}^{d_i} : \mathrm{argmax}_i\,\Phi_i(x,\xi) \ne y\}$. Then, the all-layer margin $m(\Phi, x, y)$ is defined as

$$m(\Phi, x, y) = \min_{\xi\in\Xi(\Phi,x,y)} \|\xi\|_2,$$

where $\|\xi\|_2$ is given by $\sqrt{\sum_{i=1}^p \|\xi_i\|_2^2}$. Following [40], we define a variant of the all-layer margin that measures robustness of Φ with respect to input transformations defined by $\mathcal{B}(x)$ as follows:

$$m_{\mathcal{B}}(\Phi, x) := \min_{x'\in\mathcal{B}(x)} m(F, x', \mathrm{argmax}_i\,\Phi_i(x)).$$

ASSUMPTION 15 (c.f. [39], Condition A.1)**.** Let $\mathcal{G}$ be a normed space with norm $\|\cdot\|$ and $\epsilon > 0$. We say $\mathcal{G}$ satisfies the ϵ^{-2} covering condition with complexity $\mathcal{C}_{\|\cdot\|}(\mathcal{G})$ if for all $\epsilon > 0$, we have

$$\log \mathcal{N}_{\|\cdot\|}(\epsilon, \mathcal{G}) \le \frac{\mathcal{C}_{\|\cdot\|}(\mathcal{G})}{\epsilon^2}.$$

Here $\mathcal{N}_{\|\cdot\|}(\epsilon, \mathcal{G})$ the ϵ-covering number of $\mathcal{G}$. We assume function class $\mathcal{F}_i$ satisfies the ϵ^{-2} covering condition with complexity $\mathcal{C}_{\|\cdot\|}(\mathcal{F}_i)$ for each $1 \le i \le p$.

Throughout this section, we suppose that Assumption 15 holds. Essentially, the following two propositions follows were proved by Wei et al. [40]:

PROPOSITION 16 (c.f., [40], Lemma D.6)**.** *With probability at least $1-\delta$ over the draw of the training data, for all $t \in (0, \infty)$, any $\Phi \in \mathcal{F}$ satisfies the following:*

$$R_{\mathcal{B},w}(F) = \mathbf{E}_{\widehat{P}_w}[\mathbf{1}(m_{\mathcal{B}}(\Phi, x) \le t)] + \widetilde{O}\left(\frac{\sum_{i=1}^p \mathcal{C}_{\|\cdot\|_{\mathrm{op}}}(\mathcal{F}_i)}{t\sqrt{n}}\right) + \zeta,$$

where $\zeta = O\left(\sqrt{\frac{\log(1/\delta)+\log n}{n}}\right)$ is a lower order term and $F(x) = \mathrm{argmax}_{i\in[K]}\,\Phi_i(x)$.

PROPOSITION 17 (c.f., [40], Theorem D.3)**.** *With probability at least $1-\delta$ over the draw of the training data, for all $t \in (0, \infty)$, any $\Phi \in \mathcal{F}$ satisfies the following:*

$$L_w(F, F_{\mathrm{pl}}) = \mathbf{E}_{\widehat{P}_w}[\mathbf{1}(m(\Phi, x, F_{\mathrm{pl}}(x)) \le t)] + \widetilde{O}\left(\frac{\sum_{i=1}^p \mathcal{C}_{\|\cdot\|_{\mathrm{op}}}(\mathcal{F}_i)}{t\sqrt{n}}\right) + \zeta,$$

where $\zeta = O\left(\sqrt{\frac{\log(1/\delta)+\log n}{n}}\right)$ is a lower order term and $F(x) = \mathrm{argmax}_{i\in[K]}\,\Phi_i(x)$.

REMARK. Although we have proved Theorem 5 following [40], we had to provide our own proof due to some differences in theoretical assumptions (e.g., in our case there does not necessarily exist the ground-truth classifier, although they assumed the expansion property for each P_i, we assume the expansion property for the weighted probability measure). On the other hand, the proofs of [40, Lemma D.6] and [40, Theorem D.3] work for any distribution P on $\mathcal{X}$ and its empirical distribution $\widehat{P}$. Since $\|w\|_1 \mathcal{P}_w(\mathcal{S}_{\mathcal{B}}^c(F)) = R_{\mathcal{B},w}(F)$ and $\|w\|_1 \mathcal{P}_w(\{x : F(x) \neq F_{\mathrm{pl}}(x)\}) = L_w(F, F_{\mathrm{pl}})$, Proposition 16 and Proposition 17 follow from the corresponding results in [40].

THEOREM 18. *Suppose Assumptions 3, 4, and 15 hold. Then, with probability at least $1-\delta$ over the draw of the training data, for all $t_1, t_2 \in (0, \infty)$, and any neural network Φ in $\mathcal{F}$, we have the following:*

$$\begin{aligned}\mathrm{Err}_w(F) = \frac{\gamma+1}{\gamma-1}\mathbf{E}_{\widehat{P}_w}\left[\mathbf{1}(m(\Phi, x, F_{\mathrm{pl}}(x)) \leq t_1)\right] + \frac{2\gamma}{\gamma-1}\mathbf{E}_{\widehat{P}_w}\left[\mathbf{1}(m_{\mathcal{B}}(\Phi, x) \leq t_2)\right] \\ - \mathrm{Err}_w(F_{\mathrm{pl}}) + 2\mathrm{Err}_w(F^*) + \frac{2\gamma}{\gamma-1}R_{\mathcal{B},w}(F^*) \\ + \widetilde{O}\left(\frac{\sum_{i=1}^p \mathcal{C}_{\|\cdot\|_{\mathrm{op}}}(\mathcal{F}_i)}{t_1\sqrt{n}}\right) + \widetilde{O}\left(\frac{\sum_{i=1}^p \mathcal{C}_{\|\cdot\|_{\mathrm{op}}}(\mathcal{F}_i)}{t_2\sqrt{n}}\right) + \zeta,\end{aligned}$$

where $\zeta = O\left(\sqrt{\frac{\log(1/\delta)+\log n}{n}}\right)$ is a lower order term and $F(x) = \mathrm{argmax}_{i\in[K]}\, \Phi_i(x)$.

Proof. By (15) with $H = F^*$ and $-\mathbf{1}(F_{\mathrm{pl}}(x) \neq F^*(x)) \leq -\mathbf{1}(F_{\mathrm{pl}}(x) \neq j) + \mathbf{1}(F^*(x) \neq j)$ for any $x \in \mathcal{X}$ and $j \in [K]$, we obtain the following inequality:

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

Then, the statement of the theorem follows from Proposition 16 and Proposition 17. □

E Proof of Proposition 6

Proof. Let the average weighted consistency loss be $\mathcal{L}_u^{wt} = \frac{1}{|B|}\sum_{x\in B} \ell_u^{wt}(\hat{p}_m(x), p_m(\mathcal{A}(x)), \mathbf{G})$ this will be minimized if for each of $x \in B$ the $\ell_u^{wt}(\hat{p}_m(x), p_m(\mathcal{A}(x)), \mathbf{G})$ is minimized. This expression can be expanded as:

$$\begin{aligned}\ell_u^{wt}(\hat{p}_m(x), p_m(\mathcal{A}(x)), \mathbf{G}) &= -\sum_{i=1}^K (\mathbf{G}^T\hat{p}_m(x))_i \log(p_m(\mathcal{A}(x))_i) \\ &= -C\sum_{i=1}^K \frac{(\mathbf{G}^T\hat{p}_m(x))_i}{\sum_{j=1}^m \mathbf{G}^T\hat{p}_m(x))_j} \log(p_m(\mathcal{A}(x))_i) \\ &= C \times \mathrm{H}(\mathrm{norm}(\mathbf{G}^T\hat{p}_m(x)) \mid\mid p_m(\mathcal{A}(x))).\end{aligned}$$

Here we use H to denote the cross entropy between two distributions. As we don't backpropogate gradients from the $\hat{p}_m(x)$ (pseudo-label) branch of prediction network we can consider $C = \sum_{j=1}^m \mathbf{G}^T\hat{p}_m(x))_j$ as a constant in our analysis. Also adding a constant term of entropy $\mathrm{H}(\mathrm{norm}(\mathbf{G}^T\hat{p}_m(x)))$ to cross entropy term and dropping constant C doesn't change the outcome of minimization. Hence we have the following:

$$\begin{aligned}\min_{p_m} \mathrm{H}(\mathrm{norm}(\mathbf{G}^T\hat{p}_m(x)) \mid\mid p_m(\mathcal{A}(x))) &= \min_{p_m} \mathrm{H}(\mathrm{norm}(\mathbf{G}^T\hat{p}_m(x)) \mid\mid p_m(\mathcal{A}(x))) \\ &\quad + \mathrm{H}(\mathrm{norm}(\mathbf{G}^T\hat{p}_m(x))) \\ &= \min_{p_m} \mathcal{D}_{KL}(\mathrm{norm}(\mathbf{G}^T\hat{p}_m(x)) || p_m(\mathcal{A}(x))).\end{aligned}$$

This final term is the $\mathcal{D}_{KL}(\mathbf{G}^T\hat{p}_m(x)||p_m(\mathcal{A}(x))$ which is obtained by using the identity $\mathcal{D}_{KL}(p, q) = \mathrm{H}(p, q) + \mathrm{H}(p)$ where p, q are the two distributions. □

Table 5: Table of Notations used in Paper

$\mathcal{Y}$	:	Label space
$\mathcal{X}$	:	Instance space
K	:	Number of classes
π_i	:	prior for class i
F	:	a classifier model
D	:	data distribution
$\boldsymbol{\lambda}$	:	Lagrange multiplier
λ_u	:	coefficient of unlabeled loss
$\text{rec}_i[F]$	:	recall of ith class for a classifier F
$\text{acc}[F]$	:	accuracy for a classifier F
$\text{prec}_i[F]$	:	precision of ith class for a classifier F
$\text{cov}_i[F]$	:	coverage for i^{th} class for a classifier F
$\mathbf{G}$	:	a $K \times K$ matrix
$\mathbf{D}$	:	a $K \times K$ diagonal matrix
$\mathbf{M}$	:	a $K \times K$ matrix
μ	:	ratio of labelled to unlabelled samples
B	:	batch size for FixMatch
ℓ_u^{wt}	:	loss for unlabelled data using pseudo label
ℓ_s^{hyb}	:	loss for labelled data
$\mathcal{L}_u^{\text{hyb}}$	:	average loss for unlabelled data using pseudo label on a batch of samples
$\mathcal{L}_s^{\text{hyb}}$	:	average loss for labelled data on a batch of samples
H	:	cross entropy function
$\mathcal{A}$	:	a $\mathcal{X} \to \mathcal{X}$ function that is stochastic in nature and applies a strong augmentation to it
α	:	a $\mathcal{X} \to \mathcal{X}$ function that is stochastic in nature and applies a weak augmentation to it
ρ	:	imbalance factor
B	:	batch size of samples
B_s	:	batch of labelled samples
B_u	:	batch of unlabelled samples
x	:	an input sample, $x \in \mathcal{X}$
$\hat{p}_m$	:	a pseudo label generating function
p_m	:	distribution of confidence for a model's prediction on a given sample
w	:	a $K \times K$ weight matrix that corresponds to a gain matrix $\mathbf{G}$
$\text{Err}_w(F)$	:	weighted error of F that corresponds to the objective of CSL
$\mathcal{P}_w$	:	weighted distribution on $\mathcal{X}$
P_i	:	class conditional distribution of samples for class i
$R_{\mathcal{B},w}(F)$	:	theoretical weighted (cost sensitive) consistency regularizer
F_{pl}	:	a pseudo labeler
$L_w(F, F_{\text{pl}})$	:	weighted error between F and F_{pl}
$\mathcal{L}_w(F)$	:	theoretical CSST loss
c	:	a non-increasing function used in the definition of the c-expansion property (Definition 2)
γ	:	a value of c defined in Assumption 4
β	:	an upper bound of $R_{\mathcal{B},w}(F)$ in the optimization problem (6)
S^c	:	the complement of a set S

F Notation

We provide the list of notations commonly used in the paper in Table 5.

G Objective

G.1 Logit Adjusted Weighted Consistency Regularizer

As we have introduced weighted consistency regularizer in Eq. (8) for utilizing unlabeled data, we now provide logit adjusted variant of it for training deep networks in this section. We provide logit adjusted term for $\ell_u^{\mathrm{wt}}(\hat{p}_m(x), p_m(\mathcal{A}(x), \mathbf{G})$ below:

$$\begin{aligned}
\ell_u^{\mathrm{wt}}(\hat{p}_m(x), p_m(\mathcal{A}(x), \mathbf{G}) &= -\sum_{i=1}^{K} (\mathbf{G}^{\mathbf{T}} \hat{p}_m(x))_i \log(p_m(\mathcal{A}(x))_i) \\
&= -\sum_{i=1}^{K} (\mathbf{G}^{\mathbf{T}} \hat{p}_m(x))_i \log\left(\frac{\exp(p_m(\mathcal{A}(x))_i)}{\sum_{j=1}^{K} \exp(p_m(\mathcal{A}(x))_j)}\right) \\
&= -\sum_{i=1}^{K} (\mathrm{D}^{\mathbf{T}} \mathrm{M}^{\mathbf{T}} \hat{p}_m(x))_i \log\left(\frac{\exp(p_m(\mathcal{A}(x))_i)}{\sum_{j=1}^{K} \exp(p_m(\mathcal{A}(x))_j)}\right)
\end{aligned}$$

The above expression comes from the decomposition $\mathbf{G} = \mathrm{MD}$. The above loss function can be converted into it's logit adjusted equivalent variant by following transformation as suggested by Narasimhan and Menon [24] which is equivalent in terms of optimization of deep neural networks:

$$\ell_u^{\mathrm{wt}}(\hat{p}_m(x), p_m(\mathcal{A}(x), \mathbf{G}) \equiv -\sum_{i=1}^{K} (\mathrm{M}^{\mathbf{T}} \hat{p}_m(x))_i \log\left(\frac{\exp(p_m(\mathcal{A}(x))_i - \log(\boldsymbol{D}_{ii}))}{\sum_{j=1}^{K} \exp(p_m(\mathcal{A}(x))_j - \log(\boldsymbol{D}_{jj}))}\right) \tag{19}$$

The above loss is the consistency loss ℓ_u^{wt} that we practically implement for `CSST`. Further in case $\hat{p}_m(x)$ is a hard pseudo label y as in FixMatch, the above weighted consistency loss reduces to $\ell^{\mathrm{hyb}}(y, p_m(\mathcal{A}(x)))$. Further in case the gain matrix G is diagonal the above loss will converge to $\ell^{\mathrm{LA}}(y, p_m(\mathcal{A}(x)))$. Thus the weighted consistency regularizer can be converted to logit adjusted variants ℓ^{LA} and ℓ^{hyb} based on $\mathbf{G}$ matrix.

G.2 `CSST`(FixMatch)

In FixMatch, we use the prediction made by the model on a sample x after applying a weak augmentation α and is used to get a hard pseudo label for the models prediction on a strongly augmented sample i.e. $\mathcal{A}(x)$. The set of weak augmentations include horizontal flip, We shall refer to this pseudo label as $\hat{p}_m(x)$. The list of strong augmentations are given in Table 12 of Sohn et al. [35]. Weak augmentations include padding, random horizontal flip and cropping to the desired dimensions (32X32 for CIFAR and 224X224 for ImageNet). Given a batch of labeled and unlabeled samples B_s and B_u, `CSST` modifies the supervised and un-supervised component of the loss function depending upon the non-decomposable objective and its corresponding gain matrix $\mathbf{G}$ at a given time during training. We assume that in the dataset, a sample x, be it labeled or unlabeled is already weakly augmented. `vanilla` FixMatch's supervised componenet of the loss function is a simple cross entropy loss whereas in our `CSST`(FixMatch) it is replaced by ℓ_s^{hyb} .

$$\mathcal{L}_s^{\mathrm{hyb}} = \frac{1}{|B_s|} \sum_{x,y \in B_s} \ell^{\mathrm{hyb}}(y, s(x)). \tag{20}$$

$$\mathcal{L}_u^{\mathrm{wt}} = \frac{1}{|B_u|} \sum_{x \in B_u} \mathbb{1}_{(\mathcal{D}_{KL}(\mathrm{norm}(\mathbf{G}^T \hat{p}_m(x)) \,||\, p_m(x)) \leq \tau)} \ell_u^{\mathrm{wt}}(\hat{p}_m(x), p_m(\mathcal{A}(x)), \mathbf{G})). \tag{21}$$

The component of the loss function for unlabeled data (i.e. consistency regularization) is where one of our contributions w.r.t the novel thresholding mechanism comes into light. `vanilla` FixMatch selects unlabeled samples for which consistency loss is non-zero, such that the model's confidence on

the most likely predicted class is above a certain threshold. We rather go for a threshold mechanism that select based on the basis of degree of distribution match to a target distribution based on $\mathbf{G}$. The final loss function $\mathcal{L} = \mathcal{L}_s^{hyb} + \lambda_u \mathcal{L}_u^{\text{wt}}$, i.e. a linear combination of $\mathcal{L}_s^{\text{hyb}}$ and $\mathcal{L}_u^{wt}$. Since for FixMatch we are dealing with Wide-ResNets and ResNets which are deep networks, as mentioned in Section G.1, we shall use the alternate logit adjusted formulation as mentioned in Eq. (19) as substitute for ℓ_u^{wt} in Eq. (21).

G.3 CSST(UDA)

The loss function of UDA is a linear combination of supervised loss and consistency loss on unlabeled samples. The former is the cross entropy (CE) loss, while the latter for the unlabeled samples minimizes the KL-divergence between the model's predicted label distribution on an input sample and its augmented sample. Often the predicted label distribution on the unaugmented sample is sharpened. The augmentation we used was a English-French-English backtranslation based on the MarianMT [11] fast neural machine translation model. In UDA supervised component of the loss is annealed using a method described as Training Signal Annealing (TSA), where the CE loss is considered only for those labeled samples whose $\max_i p_m(x)_i < \tau_t$, where t is a training time step. We observed that using TSA in a long tailed setting leads to overfitting on the head classes and hence chose to not include the same in our final implementation.

CSST modifies the supervised and unsupervised component of the loss function in UDA depending upon a given objective and its corresponding gain matrix $\mathbf{G}$ at a given time during training. The supervised component of the loss function for a given constrained optimization problem and a gain matrix $\mathbf{G}$, is the hybrid loss ℓ_s^{hyb}. For the consistency regularizer part of the loss function, we minimize the KL-divergence between a target distribution and the model's prediction label distribution on its augmented version. The target distribution is $\text{norm}(\mathbf{G}^T \hat{p}_m(x))$, where $\hat{p}_m(x)$ is the sharpened prediction of the label distribution by the model. Given a batch of labeled and unlabeled samples B_s and B_u, the final loss function in CSST(UDA) is a linear combination of $\mathcal{L}_s^{hyb}$ and $\mathcal{L}_u^{wt}$, i.e $\mathcal{L} = \mathcal{L}_s^{hyb} + \lambda_u \mathcal{L}_u^{wt}$.

$$\mathcal{L}_s^{\text{hyb}} = \frac{1}{|B_s|} \sum_{x,y \in B_s} \ell^{\text{hyb}}(p_m(x), y). \tag{22}$$

$$\mathcal{L}_u^{wt} = \frac{1}{|B_u|} \sum_{x \in B_u} \mathbb{1}_{(\mathcal{D}_{KL}(\text{norm}(\mathbf{G}^T \hat{p}_m(x)) \,||\, p_m(x)) \leq \tau)} \ell_u^{\text{wt}}(\hat{p}_m(x), p_m(\mathcal{A}(x), \mathbf{G})). \tag{23}$$

Since for UDA, we are dealing with DistilBERT, as mentioned in Section G.1, we shall use the alternate formulation as mentioned in Eq. (19) as substitute for ℓ_u^{wt} in Eq. above.

H Threshold mechanism for diagonal Gain Matrix

Consider the case when the gain matrix is a diagonal matrix. The loss function $\mathcal{L}_u^{wt}(B_u)$ as defined in (9) makes uses of a threshold function that selects samples based on the KL divergence based threshold between the target distribution as defined by the gain matrix $\mathbf{G}$ and the models predicted distribution of confidence over the classes.

$$\text{Threshold function } := \mathbb{1}_{(\mathcal{D}_{KL}(\text{norm}(\mathbf{G}^T \hat{p}_m(x)) \,||\, p_m(x)) \leq \tau)} \tag{24}$$

Since $\mathbf{G}$ is a diagonal matrix and the pseudo-label $\hat{p}_m(x)$ is one hot, the $\text{norm}(\mathbf{G}^T \hat{p}_m(x))$ is a one-hot vector. The threshold function's KL divergence based criterion can be expanded as follows where $\hat{y}$ is the pseudo-label's maximum class's index:

$$\mathcal{D}_{KL}(\text{norm}(\mathbf{G}^T \hat{p}_m(x)) || p_m(x)) = -\log p_m(x)_{\hat{y}} < \tau \tag{25}$$

The above equations represents a threshold on the negative log-confidence of the model's prediction for a given unlabeled sample, for the pseudo-label class ($\hat{y}$). This can be further simplified to $p_m(x)_{\hat{y}} \geq \exp(-\tau)$ which is simply a threshold based on the model's confidence. Since pseudo-label is generated from the model's prediction, this threshold is nothing but a selection criterion to select only those samples whose maximum confidence for a predicted hard pseudo-label is above a fixed threshold. This is identical to the threshold function which is used in Fixmatch [35] i.e. $\max(p_m(x)) \geq \exp(-\tau)$. In FixMatch this $\exp(-\tau)$ is set to 0.95.

I Dataset

CIFAR-10 and CIFAR-100 [15]. are image classification datasets of images of size 32 X 32. Both the datasets have a size of 50k samples and by default, they have a uniform sample distribution among its classes. CIFAR-10 has 10 classes while CIFAR-100 has 100 classes. The test set is a balanced set of 10k images.

ImageNet-100 [31]. is an image classification dataset carved out of ImageNet-1k by selecting the first 100 classes. The distribution of samples is uniform with 1.3k samples per class. The test set contains 50 images per class. All have a resolution of 224X224, the same as the original ImageNet-1k dataset.

IMDb[20]. dataset is a binary text sentiment classification dataset. The data distribution is uniform by default and has a total 25k samples in both trainset and testset. In this work, we converted the dataset into a longtailed version of $\rho = 10, 100$ and selected 1k labeled samples while truncating the labels of the rest and using them as unlabeled samples.

DBpedia-14[18]. is a topic classification dataset with a uniform distribution of labeled samples. The dataset has 14 classes and has a total of 560k samples in the trainset and 70k samples in the test set. Each sample, apart from the content, also has title of the article that could be used for the task of topic classification. In our experiments, we only make use of the content.

J Algorithms

We provide a detailed description of algorithms used for optimizing non decomposable objectives through `CSST`(FixMatch) ans `CSST`(UDA). Algorithm 1 is used for experiments in Section 5 for maximizing worst-case recall (i.e. min recall using `CSST`(FixMatch) and `CSST`(UDA)). Algorithm 2 is used for experiments in Section 5 for maximizing recall under coverage constraints (i.e. min coverage experiments on CIFAR10-LT, CIFAR100-LT and ImageNet100-LT).

Algorithm 1 `CSST`-based Algorithm for Maximizing Worst-case Recall

Inputs: Training set S_s(labeled) and S_u(unlabeled) , Validation set S^{val}, Step-size $\omega \in \mathbb{R}_+$, Class priors π
Initialize: Classifier F^0, Multipliers $\boldsymbol{\lambda}^0 \in \Delta_{K-1}$
for $t = 0$ to $T-1$ **do**
  Update $\boldsymbol{\lambda}$:
    $\lambda_i^{t+1} = \lambda_i^t \exp\left(-\omega \cdot \text{recall}_i[F^t]\right), \forall i,$
    $\boldsymbol{\lambda} = \text{norm}(\boldsymbol{\lambda})$
    $\mathbf{G} = \text{diag}(\lambda_1^{t+1}/\pi_1, \ldots, \lambda_K^{t+1}/\pi_K)$
  Compute ℓ_u^{wt}, ℓ_s^{hyb} using $\mathbf{G}$
  Cost-sensitive Learning (CSL) for FixMatch:
    $B_u \sim S_u, B_s \sim S_s$ // Sample batches of data
    $F^{t+1} \in \arg\min_F \sum_{B_u, B_s} \lambda_u \mathcal{L}_u^{\text{wt}} + \mathcal{L}_s^{\text{hyb}}$ // Replaced by few steps of SGD
end for
return F^T

Parameter	CIFAR-10	CIFAR-100	ImageNet-100	IMDb ($\rho = 10$)	IMDb ($\rho = 100$)	DBpedia-14
τ	0.05	0.05	0.05	0.1	0.1	0.1
λ_u	1.0	1.0	1.0	0.1	0.1	0.1
μ	4.0	4.0	4.0	13.8	12.6	133
$\lvert B_s \rvert$	64	64	64	32	32	32
$\lvert B_u \rvert$	256	256	256	128	128	128
lr	3e-3	3e-3	0.1	1e-5	1e-5	1e-5
ω	0.25	0.25	0.1	0.5	0.5	0.5
SGD steps before eval	32	100	500	50	50	100
optimizer	SGD	SGD	SGD	AdamW	AdamW	AdamW
KL-Thresh	0.95	0.95	0.95	0.9	0.9	0.9
Weight Decay	1e-4	1e-3	1e-4	1e-2	1e-2	1e-2
ρ	100	10	10	10	100	100
λ_u	1.0	1.0	1.0	0.1	0.1	0.1
μ	4.0	4.0	4.0	11.0	11.0	11.0
Arch.	WRN-28-2	WRN-28-8	ResNet50	DistilBERT	DistilBERT	DistilBERT

Table 6: This table shows us the detailed hyper parameters used for CSST(FixMatch) for the long tailed datasets CIFAR-10, CIFAR-100, ImageNet-100 and CSST(UDA) on IMDb, DBpedia-14. All the datasets were converted to their respective long tailed versions based on the imbalance factor ρ, and a fraction of the samples were used along with their labels for supervision.

Algorithm 2 CSST-based Algorithm for Maximizing Mean Recall s.t. per class coverage > 0.95/K

Inputs: Training set S_s(labeled) and S_u(unlabeled) , Validation set S^{val}, Step-size $\omega \in \mathbb{R}_+$, Class priors π
Initialize: Classifier F^0, Multipliers $\boldsymbol{\lambda}^0 \in \mathbb{R}_+^K$
for $t = 0$ to $T - 1$ **do**
  Update $\boldsymbol{\lambda}$:
    $\lambda_i^{t+1} = \lambda_i^t - \omega\big(\text{cov}_i[F^t] - \frac{0.95}{K}\big), \forall i$
    $\lambda_i^{t+1} = \max\{0, \lambda_i^{t+1}\}, \forall i \in [K]$ // Projection to $\mathbb{R}_+$
    $\mathbf{G} = \text{diag}(\lambda_1^{t+1}/\pi_1, \ldots, \lambda_K^{t+1}/\pi_K) + \mathbf{1_K}\boldsymbol{\lambda}^\top$
  Compute ℓ_u^{wt}, ℓ_s^{hyb} using $\mathbf{G}$
  Cost-sensitive Learning (CSL) for FixMatch:
    $B_u \sim S_u, B_s \sim S_s$ // Sample batches of data
    $F^{t+1} \in \arg\min_F \sum_{B_u, B_s} \lambda_u \mathcal{L}_u^{\text{wt}} + \mathcal{L}_s^{\text{hyb}}$ // Replaced by few steps of SGD
end for
return F^T

K Details of Experiments and Hyper-parameters

The experiment of $\max_F \min_i \text{recall}_i[f]$ and $\max_F \text{recall}[F]$ s.t. $\text{cov}_i[F] > \frac{0.95}{K}, \forall i \in [K]$ was performed on the long tailed version of CIFAR-10, IMDb($\rho = 10, 100$) and DBpedia-14 datasets. This was because the optimization of the aforementioned 2 objectives is stable for cases with low number of classes. Hence the objective of $\max_F \min(\text{recall}_{\mathcal{H}}[F], \text{recall}_{\mathcal{T}}[F])$ and $\max_F \text{recall}[F]$ s.t. $\min_{\mathcal{H},\mathcal{T}} \text{cov}_{\mathcal{H},\mathcal{T}}[F] > \frac{0.95}{K}$ is a relatively easier objective for datasets with large number of classes, hence were the optimization objectives for CIFAR-100 and ImageNet-100 long tailed datasets. For all experiments for a given dataset, we used the same values for a given common hyperparameter. We ablated the threshold for our novel unlabeled sample selection criterion(τ) and the ratio of labeled and unlabeled samples, given fixed number of unlabeled samples(μ) and are available in Fig. 4b.

L Code, License, Assets and Computation Requirements

L.1 Code and Licenses of Assets

In this work, we use the open source implementation of FixMatch [35] [3] in PyTorch, which is licensed under `MIT` License for educational purpose. Also for NLP experiments we make use of DistillBERT [33] pretrained model available in the HuggingFace [41] library. The code to reproduce the main experiments results can be found at https://github.com/val-iisc/CostSensitiveSelfTraining.

L.2 Computational Requirements

Method	CIFAR-10	CIFAR-100	ImageNet-100
`ERM`	A5000 49m	A5000 6h 47m	RTX3090 15h 8m
`LA`	RTX3090 39m	A5000 6h 9m	A5000 15h 7m
`CSL`	A5000 47m	A5000 6h 40m	A5000 12h
`CSST`(FixMatch) w/o KL-Threshold	4 X A5000 21h 0m	4 X A100 2d 19h 16 m	4 X A5000 2d 13h 19m
`CSST`(FixMatch)	4 X A5000 21h 41m	4 X A5000 2d 11h 52m	4 X A5000 2d 4m

Table 7: Computational requirements and training time (d:days, h:hours, m:minutes) for experiments relevant to vision datasets. As we can see some of the experiments on the larger datasets such as ImageNet requires long compute times.

Method	IMDb($\rho = 10$)	IMDb($\rho = 100$)	DBpedia-14
`ERM`	4 X A5000 25m	4 X A5000 29m	4 X A5000 2h 44m
`UDA`	4 X A5000 44m	4 X A5000 32m	4 X A5000 10h 18m
`CSST`(UDA)	4 X A5000 49m	4 X A5000 35m	4 X A5000 13h 12m

Table 8: Computational requirements and training time(d:days, h:hours, m:minutes) for experiments done on NLP datasets. The DistilBERT model which we are using is pretrained on a language modeling task, hence it requires much less time for training in comparison to vision models which are trained from scratch.

All experiments were done on a variety of GPUs, with primarily Nvidia A5000 (24GB) with occasional use of Nvidia A100 (80GB) and Nvidia RTX3090 (24GB). For finetuning DistilBERT and all experiments with ImageNet-100 dataset we used PyTorch data parallel over 4 A5000s. Training was done till no significant change in metrics was observed. The detailed list of computation used per experiment type and dataset have been tabulated in Table 7 and Table 8.

M Statistical Analysis

We establish the statistical soundness and validity of our results we ran our experiments on 3 different seeds. Due to the computational requirements for some of the experiments ($\approx$2days) we chose to run the experiments on multiple seeds for a subset of tasks i.e. for maximising the minimum recall among all classes for CIFAR-10 LT. We observe that the std. deviation is significantly smaller than the average values for mean recall and min. recall and our performance metrics fall within our std. deviation hence validating the stability and soundness of training.

[3] https://github.com/LeeDoYup/FixMatch-pytorch

Table 9: Avg. and std. deviation of Mean Recall and Min. Recall for CIFAR-10 LT

Method	Mean Recall	Min Recall
ERM	0.52 ± 0.01	0.27 ± 0.02
LA	0.54 ± 0.02	0.37 ± 0.01
CSL	0.63 ± 0.01	0.43 ± 0.04
Vanilla (FixMatch)	0.78 ± 0.01	0.47 ± 0.02
CSST(FixMatch)	0.75 ± 0.01	0.72 ± 0.01

N Additional Details

N.1 Formal Statement Omitted in Sec. 2.2

In Sec. 2.2, we stated that learning with the hybrid loss ℓ^{hyb} gives the Bayes-optimal classifier for the CSL (2). However, due to space constraint, we did not provide a formal statement. In this section, we provide a formal statement of it for clarity.

PROPOSITION ([24] Proposition 4)**.** For any diagonal matrix $\mathbf{D} \in \mathbb{R}^{K\times K}$ with $D_{ii} > 0, \forall i$, $\mathbf{M} \in \mathbb{R}^{K\times K}$, and $\mathbf{G} = \mathbf{MD}$, the hybrid loss ℓ^{hyb} is calibrated for $\mathbf{G}$. That is, for any model-prediction $\widehat{p}_m : \mathcal{X} \to \mathbb{R}^K$ that minimizes $\mathbf{E}_{(x,y)\sim D}\left[\ell^{\text{hyb}}(y, \widehat{p}_m(x))\right]$, the associated classifier $F(x) = \operatorname{argmax}_{y\in[K]} (\widehat{p}_m)_i (x)$ is the Bayes-optimal classifier for CSL (2).

N.2 Comparison with the $(a, \widetilde{c})$-expansion Property in [40]

We compare the c-expansion property with $(a, \widetilde{c})$-expansion property proposed by [40], where $a \in (0, 1)$ and $\widetilde{c} > 1$. Here we say a distribution Q on $\mathcal{X}$ satisfies the $(a, \widetilde{c})$-expansion property if $Q(\mathcal{N}(S)) \geq \widetilde{c}$ for any $S \subset \mathcal{X}$ with $Q(S) \leq a$. If Q satisfies $(a, \widetilde{c})$-expansion property [40] with $\widetilde{c} > 1$, then Q satisfies the c-expansion property, where the function c is defined as follows. $c(p) = \widetilde{c}$ if $p \leq a$ and $c(p) = 1$ otherwise. On the other hand, if Q-satisfies c-expansion property, then for any $a \in (0, 1)$ and $S \subseteq \mathcal{X}$ with $Q(S) \leq a$, we have $Q(\mathcal{N}(S)) \geq c(Q(S))Q(S) \geq c(a)Q(S)$ since c is non-increasing. Therefore, Q satisfies the $(a, c(a))$-expansion property. Thus, we could say these two conditions are equivalent. To simplify our analysis, we use our definition of the expansion property.

In addition, Wei et al. [40] showed that the $(a, \widetilde{c})$-expansion property is realistic for vision. Although they assumed the (a, c)-expansion property for each P_i $(1 \leq i \leq K)$ and we assume the c-expansion property for $\mathcal{P}_w$, it follows that the c-expansion property for $\mathcal{P}_w$ is also realistic for vision, since $\mathcal{P}_w$ is a linear combination of P_i.

O Additional Experiments

In this section, we compare CSST(FixMatch) against contemporary semi-supervised learning techniques namely CReST [38], DARP [12], vanilla(FixMatch) [35] and FlexMatch [46]. We compared these methods on the long-tailed CIFAR-10 ($\rho = 100$) and CIFAR-100 ($\rho = 10$) datasets. The objective for long-tailed CIFAR-10 dataset was to maximise the worst-case recall (3) and average recall, subject to a per-class coverage constraint (4). For CIFAR-100 LT dataset, we compare these methods for the objectives maximizing HT recall (10) and recall under HT coverage constraints(11). For the objectives (3) and (10), DARP achieves the best average recall yet it suffers on the worst-case recall. CSST(FixMatch) outperforms it on CIFAR-100 and the has superior worst-case recall on CIFAR-10. A similar observation can be made for the objectives (4) and (11) where it is only CSST(FixMatch) that has the highest min. coverage and either has superior mean recall or has negligible decrease in the average recall.

Table 10: Comparing CSST(FixMatch) against other Semi-Supervised Learning Methods for long tailed data distribution for the objectives (3) and (10). Although CSST(FixMatch) does not achieve the highest mean recall, it at very little cost to mean recall, achieves the best worst-case recall

Method	**CIFAR10-LT** ($\rho = 100$)		**CIFAR100-LT** ($\rho = 10$)	
	Avg. Rec	**Min Rec**	**Avg. Rec**	**Min HT Rec**
CReST	0.72	0.47	0.52	0.46
DARP	0.81	0.64	0.55	0.54
FlexMatch	0.80	0.48	0.61	0.39
`Vanilla` (FixMatch)	0.78	0.48	0.63	0.36
`CSST`(FixMatch)	0.76	0.72	0.63	0.61

Table 11: Comparing CSST(FixMatch) against other Semi-Supervised Learning Methods for long tailed data distribution for the objectives (4), (11). Only CSST(FixMatch) is the closest to satisfying the constraint yet suffers very little on the avg. recall.

Method	**CIFAR10-LT** ($\rho = 100$, tgt : 0.1)		**CIFAR100-LT** ($\rho = 10$, tgt : 0.01)	
	Avg. Rec	**Min Cov**	**Avg. Rec**	**Min HT Cov**
CReST	0.72	0.052	0.52	0.009
DARP	0.81	0.063	0.55	0.006
FlexMatch	0.80	0.046	0.61	0.006
`Vanilla` (FixMatch)	0.78	0.055	0.63	0.004
`CSST`(FixMatch)	0.80	0.092	0.63	0.010