ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240648

FlexSSL : A Generic and Efficient Framework for Semi-Supervised Learning

Huiling Qin[a], Xianyuan Zhan[b,*], Yuanxun Li[c] and Yu Zheng[d]

[a]Beijing Normal University
[c]Institute for AI Industry Research (AIR), Tsinghua University
[b]KingSoft
[d]JD iCity, JD Technology, JD Intelligent Cities Research

Abstract. Semi-supervised learning holds great promise for many real-world applications, due to its ability to leverage both unlabeled and expensive labeled data. However, most semi-supervised learning algorithms still heavily rely on the limited labeled data to infer and utilize the hidden information from unlabeled data. We note that any semi-supervised learning task under the self-training paradigm also hides an auxiliary task of discriminating label observability. Jointly solving these two tasks allows full utilization of information from both labeled and unlabeled data, thus alleviating the problem of over-reliance on labeled data. This naturally leads to a new generic and efficient learning framework without the reliance on any domain-specific information, which we call FlexSSL. The key idea of FlexSSL is to construct a semi-cooperative "game", which forges cooperation between a main self-interested semi-supervised learning task and a companion task that infers label observability to facilitate main task training. We show with theoretical derivation of its connection to loss re-weighting on noisy labels. Through evaluations on a diverse range of tasks, we demonstrate that FlexSSL can consistently enhance the performance of semi-supervised learning algorithms.

1 Introduction

Despite the huge success of supervised learning with deep neural networks, they require large amounts of labeled data for model training to achieve high performance [20, 23]. Collecting labeled data can be very difficult and costly in practice [21]. The challenge can be further exacerbated as in many cases, human or machine labeled data may contain errors or noises [22, 28]. Semi-Supervised Learning (SSL) [7] that leverages both labeled and unlabeled data has become an increasingly promising yet still challenging area, and is widely applicable to many real-world problems.

To utilize the hidden information in the unlabeled data, a general and popular semi-supervised learning paradigm is through self-training [39, 17]. Self-training uses a previously learned model to predict labels for the unlabeled data (pseudo-labeling) which are then used for subsequent model training [39, 17, 16, 34, 13]. Despite some empirical successes, self-training methods still suffer from two core challenges, which are *over-reliance on labeled data* and *error accumulation*. Most semi-supervised learning methods [39, 17, 16, 31, 38, 15] assume the labeled and unlabeled data follow the same or similar data distribution, thus the data-label mapping mechanisms learned from the labeled data are somewhat transferable to unlabeled data. Essentially, the algorithm is still reinforcing the information in the labeled data rather than mining additional information from the unlabeled data. Second, the pseudo-labels of unlabeled data can be incorrectly predicted. Using such biased information for training in subsequent epochs could increase confidence in erroneous predictions, and eventually leading to a vicious circle of error accumulation [5, 2]. The situation can be even worse when the labeled data contain errors or noises, as the model learned from labeled data will make more mistakes, resulting in more severe error accumulation.

Alleviating the issue of over-reliance on labeled data requires extracting additional sources of information from the unlabeled data. Some recent approaches introduce domain-specific data augmentation and consistency regularization [27] to enforce the stability of the predictions with respect to the transformations of the unlabeled data (e.g. data augmentation on images such as rotation, flip and shift, etc.). Although such data perturbation or augmentation are well-defined for images, they are not directly transferable to general settings due to strong reliance on the domain-specific knowledge, hence the success of these methods are mostly restricted to image classification tasks [27, 18, 38, 31]. We provide a new insight by noting that, under the self-training paradigm, every semi-supervised learning task hides another auxiliary task of discriminating whether the data label is real or a pseudo-label predicted by a machine oracle. Jointly solving the main task together with the auxiliary task allows sufficient utilization of the hidden information in unlabeled data without the need of any domain-specific information.

To break the vicious cycle of error accumulation, one needs to not always trust the machine-labeled data. A common approach in semi-supervised learning literature is to use stricter and complex "rules" for selecting high-quality pseudo-labels, but at the cost of extra high computation burden during the label selection process [26, 41]. An alternative idea is to incorporate the reliability or confidence of data labels and treat them differently during training. A connected approach in supervised learning is to use the notion of "soft labels" with confidence probabilities or weights, which have been validated in a number of noisy label learning and pseudo-labeling problems [22, 28, 30, 1]. This treatment can also help semi-supervised learning algorithms to gain stable learning from noisy labels.

Combining previous insights, we develop a new generic and efficient semi-supervised learning framework, called FlexSSL. FlexSSL

* Corresponding Author. Email: zhanxianyuan@air.tsinghua.edu.cn

contains three ingredients. First, it simultaneously solves a companion task by learning a discriminator as a critic to predict label observability (whether it is a true label or a pseudo-label), which is used to mine additional information in the unlabeled data while facilitating main task learning. Second, the output of the discriminator is in fact a measure of label reliability, thus can be interpreted as the confidence probabilities of labels. This converts the original problem into a soft label learning problem. Last but not least, FlexSSL introduces a cooperative-yet-competitive learning scheme to boost the performance of both the main task and the companion task. Specifically, FlexSSL forges cooperation between both tasks by providing extra information to each other (the main task provides prediction loss, companion task provides label confidence measures), leading to improved performance of both tasks. Moreover, we consider the main task model to be self-interested which tries to challenge the discriminator by providing as little information as possible. This design forms a semi-cooperative "game" with a partially adversarial main task model, which can be shown equivalent to a loss reweighting mechanism on noisy soft labels.

The proposed FlexSSL can be easily incorporated into a wide range of SSL models (e.g. image classification, label propagation and data imputation) with limited changes or extra computation cost on the original method. We show with empirical experiments that FlexSSL consistently achieves superior performance and effectiveness compared with the original and self-training version of different SSL algorithms. Moreover, FlexSSL naturally provides the data confidence measures from the discriminator as a byproduct of the learning process, which offers additional interpretability of the semi-supervised learning task. This can be particularly useful for many practical applications, while also providing extra benefits for scenarios with noisy data labels.

2 Preliminaries

2.1 Problem Statement

We formulate the semi-supervised learning problem as learning a model $f(\cdot)$ with input data $x \in X$ to predict the label $y \in Y$. In SSL, only a partial set of data labels Y_L with ground truth are given in the training set, the rest are unlabeled Y_U ($Y = Y_L \bigcup Y_U$). For convenience of later discussion, we denote L as the set of indices of data samples with labels and U as the set of indices for unlabeled data. In many real-world SSL problems, the size of labeled data $|L|$ is often limited. In extreme cases, Y_L may potentially contain errors or noises. We further define a mask vector M to denote the observability of labels over each data sample. We set $M_i \in M$ equals to 1 if $i \in L$ and $M_j \in M$ equals to 0 if $j \in U$. To develop a generic SSL framework, we consider a diverse set of task settings, include:

- **Inductive classification:** a typical SSL task setting is to construct a classifier to predict labels $y \in Y$ for any object in the input space $x \in X$. Common examples include semi-supervised image classification tasks [27, 31, 38], where only a subset of samples are given the known labels Y_L and the rest labels Y_U are unknown.
- **Label propagation:** another class of SSL tasks under transductive setup is to use all inputs X and observed labels Y_L to train a classifier to predict on unseen labels Y_U. Examples include classifying nodes in a graph given only small subset of node labels Y_L.
- **Data imputation:** a special SSL task under regression setting, in which the missing state of input and output data are strongly correlated [24, 25]. In data imputation, the inputs X are partially filled and the labels Y are equivalent to a reconstructed version of X obtained through regression.

2.2 Auxiliary Task in SSL Problems

We begin our discussion by first noting that, if we feed the unknown labels Y_U with the model f predicted labels $\tilde{Y}_U = \hat{Y}_U^{(k)}$ ($\hat{Y}_U^{(k)} = f(X)$ represents the labels from the k-th round of pseudo-labeling) or other machine-generated labels, there actually hides an auxiliary task of discriminating whether the data label is real or a pseudo-label. Denote all the data labels under pseudo-labeling as $\tilde{Y} = Y_L \bigcup \tilde{Y}_U$. We can train a discriminator $d(\cdot)$ to tell the confidence measure $p \in P$ of whether $x \to \tilde{y}$ ($x \in X, \tilde{y} \in \tilde{Y}$) is a valid mapping. This is always learnable since the ground truth of P is exactly the observability mask M. With a slight abuse of notation, we formulate the main SSL task A and the auxiliary companion task B as follows:

$$\begin{aligned} A &: f(X) = \hat{Y}, \qquad \mathcal{L}_A = loss_A(Y_L, \hat{Y}_L) \\ B &: d(X, \tilde{Y}) = P, \qquad \mathcal{L}_B = loss_B(P, M) \end{aligned} \tag{1}$$

where $loss_A$ is the original loss function of the main task and $loss_B$ can be any binary classification loss function between P and M, such as the binary cross entropy (BCE) loss. Jointly solving the above two tasks allows exploiting the underlying relationship between input data X and data label Y from another angle, which can potentially provide more information to facilitate main task training.

3 The FlexSSL Framework

Note that the two tasks specified in Problem (1) are independent with each other, it is not possible to gain much learning benefit unless we introduce some form of coupling between the main model f and the discriminator d. In this section, we introduce FlexSSL to jointly learn f and d in a principled way. It forges cooperation between a self-interested main task and an unselfish companion task for boosted performance. We show by theoretical derivation that the introduction of auxiliary tasks under the pseudo-labeling paradigm will lead to a soft-label learning formulation of the original SSL task.

3.1 Alternative Formulation Under FlexSSL

We consider an alternative formulation of Problem (1) by modifying the loss of main task $\mathcal{L}_A$ and inputs of the companion task to forge cooperation and information sharing:

$$\begin{aligned} &A : f(X) = \hat{Y},\ \mathcal{L}_A = loss_A(\tilde{Y}, \hat{Y}) + \alpha \cdot loss_W(\tilde{Y}, \hat{Y}, P) \\ &B : d(X, \hat{Y}, g(\tilde{Y}, \hat{Y})) = P, \quad \mathcal{L}_B = loss_B(P, M) \end{aligned} \tag{2}$$

where $g(\tilde{y}, \hat{y})$ is some form of information provided by the main task. We can simply model the information $g(\cdot)$ as the element-wise loss term of the main task, i.e. $g(\tilde{y}, \hat{y}) = loss_A(\tilde{y}, \hat{y})$, such as the cross entropy loss for classification tasks and mean square error (MSE) loss for regression tasks. Moreover, $loss_W(\tilde{Y}, \hat{Y}, P)$ is a corrective additional loss term impacted by the label confidence measure information P provided by the discriminator d, and α is its weight parameter. Due to the existence of corrective information from $loss_W$, we now train the main task with all data labels $\tilde{Y}$ rather than observed labels Y_L, thus allows sufficient utilization of information from the unlabeled data. Figure 1 illustrates the learning strategy of FlexSSL.

The design of FlexSSL has several interesting properties. First, since additional information are cooperatively provided by both tasks to each other, they can be both learned better. Second, a less well-learned model f will produce large element-wise error $loss_A(\tilde{y}, \hat{y})$

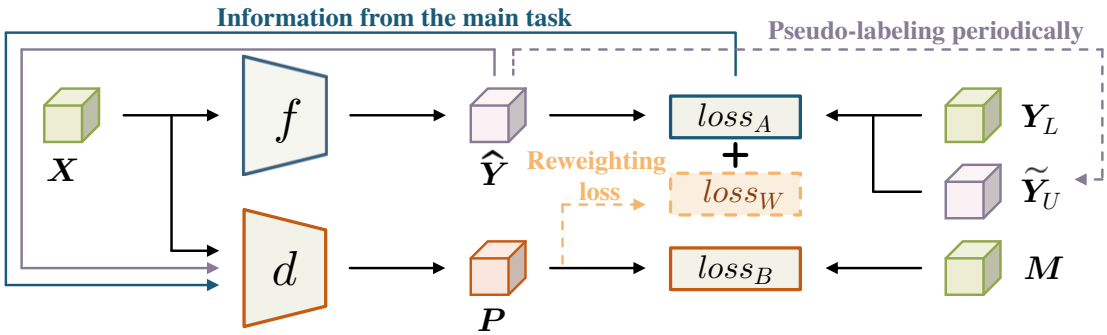


Figure 1: Illustration of the learning strategy in FlexSSL

for a predicted label $\hat{y}$, which in turn provides more information in $g(\cdot)$ for the discriminator d to distinguish whether a label is real or a pseudo-label. As f is learned better during the cooperative learning process, when g no longer provides useful information for discriminating the label observability, then we may have reason to believe f has achieved satisfactory inference performance. Hence minimizing the information in $g(\cdot)$ coincides with improving the learning of f. This suggests that if we alter the learning direction of f using $loss_W$ to encourage it providing as little information in $g(\cdot)$ as possible for the discriminator d, then we might obtain a better learned f. This in fact forms a cooperative-yet-competitive "game" between both tasks, with the main task being self-interested and in some sense partially adversarial to the discriminator d. Under this design, the main task and companion task in FlexSSL are neither fully cooperative as in multi-objective optimization [8] nor fully adversarial to each other as in adversarial learning [10, 19], but balance the benefit of cooperation and competition. Finally, it also allows the discriminator to detect errors or noises in observed data labels. As the erroneous (x, y) data pairs may possess different $x \to y$ mapping patterns against normal data, thus are more likely to be identified as pseudo-labels. Such negative impact can be further corrected using $loss_W$ with label confidence measure P.

3.2 *Derivation of the Additional Loss Term* $loss_W$

Until this point, the only mystery about FlexSSL is the exact form of $loss_W$ that embodies the self-interested behavior for the main task model f. We resort to calculus of variations to provide theoretical derivation of $loss_W$.

Under the alternative formulation (2), both the discriminator d and its loss function $loss_B$ are impacted by the information $g(\cdot)$ provided by f, thus they are now functionals of f (i.e. function of a function). For simplicity, we can express d and $loss_B$ as $d(x, g(f))$ and $loss_B(x, d, g(f))$. Note that under empirical risk minimization, we can write $\mathcal{L}_B$ as where $p_{data}(x)$ is the distribution of training dataset $\mathcal{D}$ and $\Omega_{\mathcal{D}}$ is its domain. Observe that $\mathcal{L}_B$ has a natural integral form of the functional $F(x, d, f) = p_{data}(x) \cdot loss_B(x, d, g(f))$, which is commonly studied in functional analysis. We can thus define $\mathcal{L}_B$ as a new functional $J(f, d)$.

Although the discriminator d aims to minimize its loss $\mathcal{L}_B$ (or $J(d, f)$), as mentioned previously, we want the main task model f to challenge the discriminator d by providing as little information in g as possible. This essentially requires changing the behavior of f to be adversarial to d such that $J(d, f)$ attains its maxima. This results in solving following minimax optimization problem for $J(d, f)$:

$$\min_{d} \max_{f} J(d, f) \tag{3}$$

We wish to examine how the variation of f impacts $J(d, f)$. To simplify the analysis, we first focus on solving the inner maximization problem with respect to f, and consider d as an unknown external functional decided by the outer minimization problem.

According to calculus of variation, the extrema (maxima or minima) of a functional can be obtained by solving the associated Euler-Lagrange equation: $F_f - \frac{\mathrm{d}}{\mathrm{d}x} F_{f'} = 0$, where $F_f = \partial F / \partial f$ and $F_{f'} = \partial F / \partial f'$. In our case $\frac{\mathrm{d}}{\mathrm{d}x} F_{f'} = 0$ as f' does not appear in functional $J(d, f)$. As F is a function of d, and d is a function of $g(f)$, therefore, it is necessary that $F_f = \frac{\partial F}{\partial d} \cdot \frac{\partial d}{\partial g} \cdot \frac{\partial g}{\partial f} = 0$ holds. Let θ_f be the model parameters of f, it also suggests that

$$\begin{aligned} \frac{\partial F}{\partial d} \cdot \frac{\partial d}{\partial g} \cdot \frac{\partial g}{\partial f} \cdot \frac{\partial f}{\partial \theta_f} &\overset{(i)}{=} \frac{\partial F}{\partial d} \cdot \frac{\partial d}{\partial g} \cdot \nabla_{\theta_f} loss_A \\ &\overset{(ii)}{=} \frac{\partial d}{\partial g} \cdot \frac{\partial F}{\partial d} \cdot \nabla_{\theta_f} loss_A = 0 \end{aligned} \tag{4}$$

The first equation (i) is due to our design of letting $g(\cdot)$ be the element-wise loss function of f ($g(\tilde{y}, \hat{y}) = loss_A(\tilde{y}, \hat{y})$). For the second equation (ii), as d, g and F are real-valued functionals of x and f, which is the same for their partial derivatives $\partial F / \partial d$ and $\partial d / \partial g$. Assume the continuity of both $\partial F / \partial d$ and $\partial d / \partial g$ are satisfied, as the set of real-valued continuous functions is a commutative ring [12], hence we can swap their order for convenience.

As d is determined by the outer minimization problem of (3), thus d is unkown and it is not possible to obtain the exact $\partial d / \partial g$ by only inspecting the inner maximization problem. Instead, we consider another solution of Eq.(4) by letting $\frac{\partial F}{\partial d} \cdot \nabla_{\theta_f} loss_A = 0$. Directly solving this functional equation for $\forall x \in \mathcal{D}$ is still intractable, as $\frac{\partial F}{\partial d}$ and $\nabla_{\theta_f} loss_A$ can be complicated functions and the $p_{data}(x)$ in F is typically unknown. However, we can consider a relaxed and tractable condition by computing the its integration over $\Omega_{\mathcal{D}}$. As $\Omega_{\mathcal{D}}$ is bounded when $\mathcal{D}$ is finite, thus the final integral is still 0,

$$\begin{aligned} 0 &= \int_{\Omega_{\mathcal{D}}} \frac{\partial F}{\partial d} \cdot \nabla_{\theta_f} loss_A \mathrm{d}x \\ &= \int_{\Omega_{\mathcal{D}}} p_{data}(x) \cdot \frac{\partial loss_B(x, d, f)}{\partial d} \cdot \nabla_{\theta_f} loss_A \mathrm{d}x \\ &\approx \sum_{x \in \mathcal{D}} \frac{\partial loss_B(x, d, f)}{\partial d} \nabla_{\theta_f} loss_A \end{aligned} \tag{5}$$

As an example, suppose we use BCE loss for $loss_B$, and write p_i as the output value of $d(X, \hat{Y}, g(\tilde{Y}, \hat{Y}))$ with index i, the condition (5) becomes:

$$-\sum_{i \in L} \frac{1}{p_i} \nabla_{\theta_f} loss_A(\tilde{y}_i, \hat{y}_i) \quad + \sum_{j \in U} \frac{1}{1 - p_j} \nabla_{\theta_f} loss_A(\tilde{y}_j, \hat{y}_j) = 0 \tag{6}$$

Note that the LHS of above equation can be perceived as the gradient of a new term $L_f(\tilde{Y}, \hat{Y}, P)$:

$$\begin{aligned} -L_f(\tilde{Y}, \hat{Y}, P) = &\sum_{i \in L} \frac{1}{p_i} \cdot loss_A(\tilde{y}_i, \hat{y}_i) \\ &- \sum_{j \in U} \frac{1}{1-p_j} \cdot loss_A(\tilde{y}_j, \hat{y}_j) \end{aligned} \tag{7}$$

Consequently, minimizing $-L_f$ with respect to θ_f ($\frac{\partial L}{\partial \theta_f} = 0$) satisfies the condition (5), which is derived from the necessary condition of solving the inner maximization problem of $J(d, f)$ specified in (3). The negative sign before $-L_f$ is to ensure gradient ascent update in the direction of F_f to maximize $J(d, f)$. We now uncover the exact form of $loss_W$, which can be set as $loss_W(\tilde{Y}, \hat{Y}, P) = -\mathcal{L}_f(\tilde{Y}, \hat{Y}, P)$ to realize the self-interested behavior of f. The resulting $loss_W$ can be considered as an additional reweighting loss that complements the original loss term $loss_A$ of the main task model f based on the estimated confidence P from the discriminator d.

3.3 Interpretation of FlexSSL

Adding $loss_W$ to the original loss term $loss_A$, we can obtain the complete loss for the main task as:

$$\begin{aligned} \mathcal{L}_A = &\sum_{i \in L} \left(1 + \frac{\alpha}{p_i}\right) \cdot loss_A(y_i, \hat{y_i}) \\ &+ \sum_{j \in U} \left(1 - \frac{\alpha}{1-p_j}\right) \cdot loss_A(\tilde{y}_j, \hat{y_j}) \end{aligned} \tag{8}$$

This essentially transforms the original main task into a cost-sensitive learning problem [40] by imposing following soft-labeling weights on the errors of each data label as:

$$\text{Soft-labeling weights} = \begin{cases} 1 + \alpha/p_i, & i \in L \\ 1 - \alpha/(1-p_j), & j \in U \end{cases} \tag{9}$$

Above soft-labeling weights induce very different behaviors on the loss $loss_A(\tilde{y}, \hat{y})$ of labeled and pseudo-labeled samples. As p_i represents the judgment of the discriminator d on whether y_i is labeled or not. For labeled samples, the loss are boosted by additional weight of α/p_i, which means if the discriminator d judges an labeled samples $y_i \in Y_L$ to be unlabeled or unreliable (confidence $p_i \to 0$), the loss of this sample will get significantly boosted. This forces the main task model f to pay more attention to infer the problematic labeled samples. As unlabeled samples $y_j \in Y_U$ are periodically filled with pseudo-labels generated by f (initially filled with random values), FlexSSL does not fully trust these labels and may even choose to enlarge the gap with the pseudo-label when $p > 1 - \alpha$ ($\alpha \leq 1$). Specifically, the more likely an label y_j is considered reliable by the discriminator ($p_j \to 1$), the stronger it forces the learning process to ignore further improvement on inferring y_j.

Note that FlexSSL is a very general framework, one can use different choices of loss function $loss_B$ for the discriminator, which may likely give rise to different forms of soft-labeling weights. In Table 1, we present the soft-labeling weights (with p_i represents $d(x_i, g(f(x_i)))$ for simplicity) of three commonly used classification loss functions for the discriminator loss ($loss_B$) derived under the FlexSSL framework. Since the continuity of $\partial loss_B / \partial d$ needs to be satisfied in the derivation of Eq. (4). The derivative of BCE loss can potentially result in $1/p_i$ and $1/(1-p_i)$ with infinite values, violating the continuity assumption. We thus clip their values as $\min\{1/p_i, H\}$ and $\min\{1/(1-p_i), H\}$ in our practical algorithm, where H is set to 10 in our implementation. The same clip operation is also applicable under other loss terms.

4 Experiments

FlexSSL can be easily incorporated in a variety of SSL tasks. In this section, we evaluate the universality, effectiveness, and interpretability of the FlexSSL. We choose three representative SSL tasks and apply FlexSSL on their official implementations to evaluate the performance of FlexSSL, including: image classification [11] for inductive classification, label propagation [15] for transductive classification and data imputation [9] for regression. We also conduct additional experiments that compare FlexSSL with some advanced SSL methods on the image classification tasks to evaluate the efficiency and effectiveness of FlexSSL.

- **Image classification:** We use RestNet18 for image classification on the fashion-mnist dataset [36], with 6,000/10,000 training/testing set partition. And we further use CNN-13 as the backbone architecture of some advanced SSL methods on CIFAR-10 and CIFAR-100 datasets with 4000 labeled samples, including MT [33], UDA [37], Mixmatch [4], FlexMatch [41], and UPS [26]. The training set only contains 10% of the original dataset which aims to increase the task difficulty.
- **Label propagation:** We trained a two-layer GCN [15] integrated with FlexSSL and evaluate the prediction accuracy on Cora dataset [29]. The test set consists of 1000 labeled examples extracted from a knowledge graph with 2,708 nodes and 5,429 edges.
- **Data imputation:** We use a deep denoising autoencoder (DAE) [9] for data imputation under regression setting. We use the UCI online news popularity dataset [3] for evaluation. The data is split the training/testing set with ratio of 0.8/0.2.

For all tasks, we remove certain proportion (*missing rate*) of data labels in the training set and treat them as unlabeled data in the semi-supervised setting. At initial step of FlexSSL, as the model is not yet trained to provide pseudo-labels, we fill the initial pseudo-labels for unlabeled data with random labels. For the regression task (data imputation), we compute the mean μ of labeled data and fill the missing entries with μ adding random noises; for the classification tasks, we fill the missing label Y_U by randomly selecting a data label.

4.1 Performance Evaluation Across Diverse Tasks

Performance enhancement. We compare in Table 2 our method with the original model in three representative SSL tasks to demonstrate the flexibility and accuracy of FlexSSL under diverse problem settings. FlexSSL consistently achieves superior performance over all tasks, which shows the universality of FlexSSL under diverse SSL problem setting. We also use 10 different levels of missing rates to verify the robustness of FlexSSL. By randomly dropping some labels, we increase the missing rate of the training set from 0% (fully supervised) to 90%. As label propagation and data imputation tasks are not well-defined under fully supervised setting, only the image classification task under 0% missing rates is presented in the table. It is observed in Table 2 that most of the original models experience substantial performance deterioration with the increase of missing rate. By contrast, FlexSSL remains high level of robustness with limited performance drop in most tasks.

Surprisingly, for inductive image classification task, we observe that in the case of 0% missing rate, FlexSSL can achieve even

Discriminator loss	Loss function	Weights for $i \in L$	Weights for $j \in U$
BCE loss	$-\sum_{i\in L}\log p_i - \sum_{j\in U}\log(1-p_j)$	$1+\alpha/p_i$	$1-\frac{\alpha}{1-p_j}$
Exponential loss	$\sum_{i\in L} e^{-p_i} + \sum_{j\in U} e^{p_j}$	$1+\alpha e^{-p_i}$	$1-\alpha e^{p_j}$
Logistic loss	$\sum_{i\in L}\log(1+e^{-p_i}) + \sum_{j\in U}\log(1+e^{p_j})$	$1+\frac{\alpha e^{-p_i}}{1+e^{-p_i}}$	$1-\frac{\alpha e^{p_j}}{1+e^{p_j}}$

Table 1: Soft-labeling weights under different forms of discriminator loss functions

Missing rate (%)	Image classification Accuracy (%)		Label propagation Accuracy (%)		Data imputation MSE	
	ResNet18	ResNet18+FlexSSL	GCN	GCN+FlexSSL	DAE	DAE+FlexSSL
00	83.85 ± 0.12	86.23 ± 0.13	-	-	-	-
10	83.57 ± 0.12	84.69 ± 0.08	82.18 ± 1.59	84.33 ± 0.56	0.8055	0.7899
20	83.06 ± 0.16	85.15 ± 0.13	82.18 ± 1.69	84.00 ± 1.81	0.9029	0.8716
30	82.89 ± 0.21	85.38 ± 0.16	81.80 ± 1.75	83.33 ± 0.68	0.9335	0.8867
40	82.43 ± 0.13	84.98 ± 0.21	82.28 ± 2.69	83.00 ± 0.95	0.9761	0.9229
50	81.94 ± 0.16	84.61 ± 0.21	81.93 ± 2.56	81.33 ± 2.28	0.9726	0.9412
60	81.23 ± 0.16	83.87 ± 0.20	80.03 ± 6.37	82.33 ± 2.10	0.9787	0.9727
70	79.70 ± 0.13	82.90 ± 0.21	77.84 ± 7.52	83.67 ± 3.73	1.1593	1.1582
80	78.84 ± 0.35	81.31 ± 0.42	74.33 ± 7.00	80.33 ± 1.00	1.1429	1.1425
90	76.02 ± 0.41	77.96 ± 0.47	69.66 ± 7.66	74.26 ± 0.53	-	

Table 2: Comparison of the original models and the FlexSSL-integrated version for different tasks (with mean $\pm$ std). The std results in the imputation task are omitted because they are less than 1e-04. Results averaged over 10 random seeds.

higher accuracy compared with the fully supervised base model ($86.2\pm0.13\%$ vs $83.85\pm0.13\%$). Moreover, FlexSSL achieves similar level of accuracy as the fully supervised base model even with 60% missing rate. These results suggests that FlexSSL can also be used in supervised learning problems to boost model performance.

Learning efficiency. We further compare the performance of FlexSSL with the widely used self-training-based pseudo-labeling method [16, 13] in different tasks. Figure 2 shows the learning curves of image classification, label propagation, and data imputation under missing rates of 40%, 20%, and 30%, respectively. In Figure 2, we observe that the learning speed of the main task model with the corrective reweighting loss term $loss_W$ in FlexSSL is significantly faster than the original model. This is primarily due to the use of carefully reweighted soft labels to correct unreliable label information in FlexSSL, as compared to the use of hard pseudo-labels as true label for model learning in self-training.

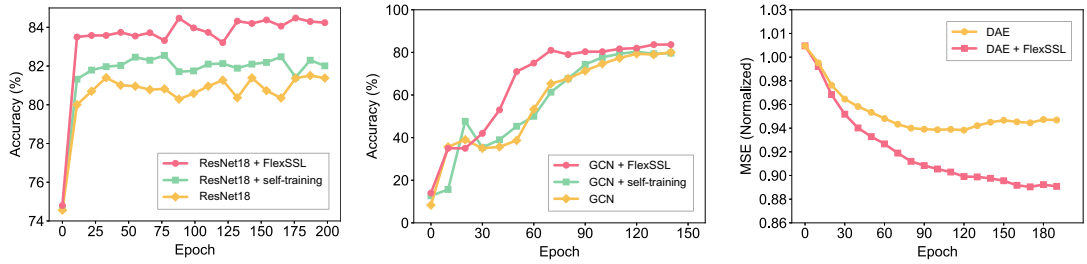
Robustness to missing data. Figure 3 shows the accuracy of the original, self-trained and FlexSSL-integrated versions of models for tasks with different missing rates. We observe that FlexSSL maintains high accuracy under various missing rates and consistently outperforms the original and self-training versions. This is achieved through the cost-sensitive learning in FlexSSL for different data samples to maximally utilize the hidden information from both labeled and unlabeled data, which lead to more robust and accurate inference. In the label propagation task, although self-training increases the training samples by periodically pseudo-labeling high confidence unlabeled data, the accuracy of the model could decreases due to the involvement of wrong labeling information, which results in subsequent error accumulation and performance drop. FlexSSL, however, maintains high accuracy in spite of high missing rate by continuously soft pseudo-labeling the data to ensure reliable pseudo-labels are properly used to facilitate main task training.

Robustness to error accumulation. To further investigate the impact of error accumulation in pseudo-labels during model training, we record the accuracy of pseudo-labels after each round of pseudo-labeling in self-training. The test accuracy of these samples in the FlexSSL version of the model are also recorded for comparison. To ensure the stability of self-training, we add pseudo-labels every 10 training epochs. Different from self-training that only label high confidence unlabeled samples, FlexSSL introduces pseudo-labels for all unlabeled data Y_U but with soft-labeling weights. Figure 4 shows the error accumulation trends for self-training and FlexSSL in the label propagation task. When the missing rate is low (Figure 4(a)), self-training maintains a high accuracy at the beginning, but model performance will fluctuate and decrease with the involvement of more potentially problematic pseudo-labels. On the other hand, when the missing rate is high and available information is scarce (Figure 4(b)), gradually involving pseudo-labeled data is beneficial for self-training, resulting in slowly increasing test accuracy, although the model still struggles to achieve high accuracy. Despite the occurrence of error accumulation in self-training, we barely observe such phenomenon in FlexSSL. The test accuracy of the pseudo-labels remains high and stable across different rounds of pseudo-labeling, which clearly suggests the ability of FlexSSL in avoiding error accumulation in pseudo-labels.

4.2 Performance and Efficiency Comparison with SSL SOTA

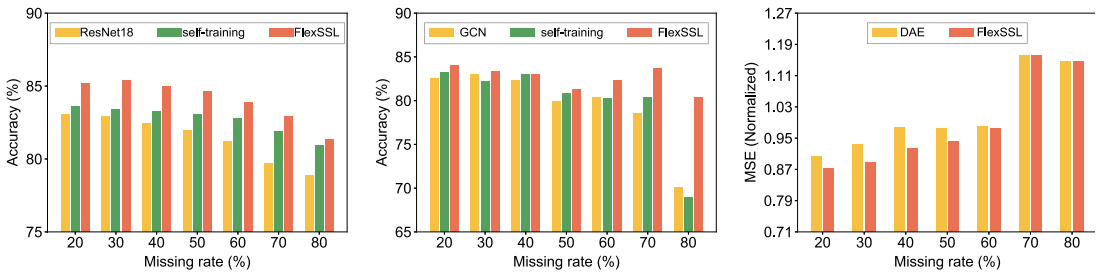
To further validate the performance and efficiency of FlexSSL, we also conduct experiments on the semi-supervised image classification task against advanced SSL methods on CIFAR-10 and CIFAR-100 datasets. Table 3 presents the mean and standard deviation results, as well as the training time until convergence of each method. FlexSSL achieves superior performance and efficiency over almost all advanced SSL methods. It also achieves comparable results to the SOTA method UPS [26], but with only 1/10th of the training time. Under the same hardware setting, it takes more than 120 hours to train UPS on both CIFAR-10 and CIFAR-100 to achieve 93% and 58% accuracy respectively; whereas FlexSSL takes only 10 hours and 2.5 hours to reach final convergence, reducing the training time by at least 92% to reach the similar level of accuracy.

Meanwhile, we plot the convergence curves of the top-2 methods with the highest accuracy in Table 3 to compare the efficiency of the training process in Figure 5. UPS employs costly label confidence measures combined with the negative sample learning to select



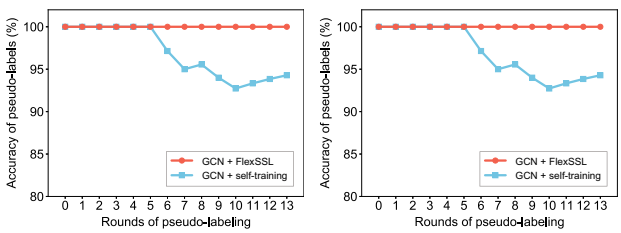
(a) Image classification (b) Label propagation (c) Data imputation

Figure 2: The convergence of test accuracy under different methods and tasks



(a) Image classification (b) Label propagation (c) Data imputation

Figure 3: Performance under different missing rate



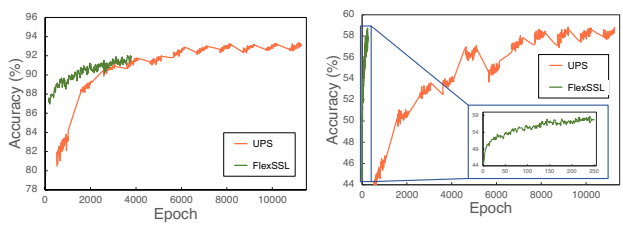
(a) 20% missing rate (b) 70% missing rate

Figure 4: Impact of Error accumulation on accuracy of pseudo-labels in the label propagation task

Methods	CIFAR-10 (time)	CIFAR-100 (time)
MT	11.41 ± 0.25 (10h)	45.36 ± 0.20 (6h)
UDA	8.21 (19h)	45.07 (18h)
Mixmatch	10.88 (5h)	45.66 (13h)
Flexmatch	8.08 (17.5h)	42.26 (5h)
UPS	**6.39** ± **0.03** (120h)	**40.77** ± **0.10** (120h)
FlexSSL	**7.85** ± **0.03** (10h)	**41.37** ± **0.12** (2.5h)

Table 3: Error rate (%) and training time on the CIFAR-10 and CIFAR-100 test set with 4000 labels. We bold the top two highest scores.

high-quality pseudo-labels to improve model performance, but these also make the algorithm computationally expensive. By contrast, FlexSSL does not contain any pseudo-label selection process, but simply pseudo-label all unlabeled samples and use the soft-labeling weights to unevenly update the loss of different samples. The great computational efficiency of FlexSSL combined with its generality and applicability to a wide range of SSL tasks, demonstrate its great potential for solving various real-world large-scale problems.



(a) CIFAR-10 (b) CIFAR-100

Figure 5: Learning curves of CCS and UPS on the CIFAR-10 and CIFAR-100 dataset with 4000 labels

4.3 Additional Ablation Study

In this section we present the ablation study of the weight parameter α under the image classification task to analyze the impact of α in FlexSSL. We selected three different levels of missing rates in our experiments - low (30%), medium (50%) and high (80%). Figure 7 shows the classification accuracy obtained by FlexSSL using different α values under different missing rates. We observe that a larger α will make the main task favors more on minimizing the reweighting loss rather than simply minimizing the difference between model predicted labels and the given labels $\tilde{Y}$. In our exeriment results in Figure 6, it actually suggests that with the α taking value from the recommended range $[0.1, 0.9]$, FlexSSL is not very sensitive to α. The maximum difference of accuracy between the worst and best choice of α is only about 2% under different missing rates. The insensitivity to α is another advantage of FlexSSL, as we do not need to perform complex parameter tuning in order to learn a robust and high performance model, which is very implementation friendly.

4.4 Interpretability of the Discriminator Outputs

To evaluate the judgement of the discriminator d, we analyze its output value distribution on the labeled and intentionally introduced

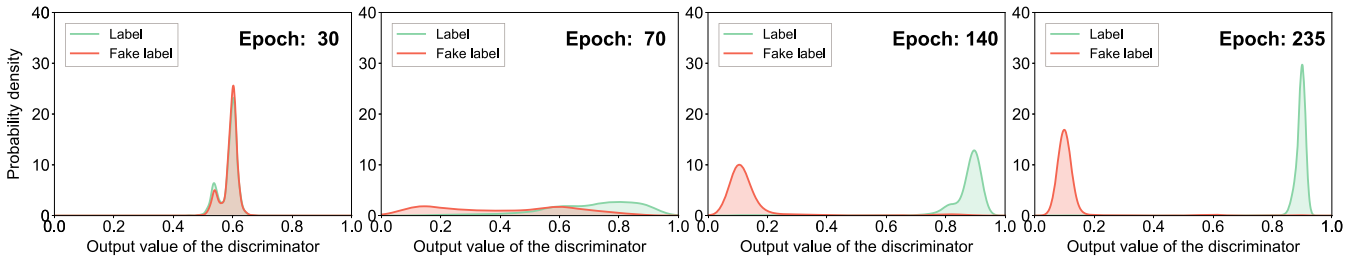
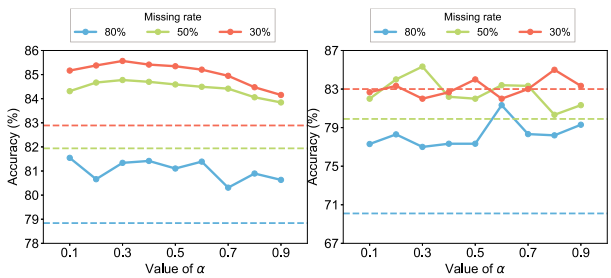


Figure 6: The probability distribution given by the discriminator during the training process



(a) Image classification (b) Label propagation

Figure 7: Test accuracy under different weight parameter α

fake/noisy labeled data in different training stages of image classification task. The four plots in Figure 6 clearly show the transition of an improved discriminator, which gradually gains better capability to discriminate true labels ($p \to 1$) and fake/noisy labels ($p \to 0$). The interpretability of the discriminator is a byproduct of FlexSSL, which can be particularly useful for many real-world FlexSSL tasks or scenarios involving error or noise in labeled data.

5 Related Work

5.1 Self-training via pseudo-labeling.

Pseudo-labeling [35, 39, 17] is one of the most widely used self-training method in semi-supervised learning, which uses a supervised model that is iteratively trained on both labeled and pseudo-labeled data in previous iterations of the algorithm. The basic assumption in most algorithms [39, 17, 16, 34, 13] is that an unlabeled instance can be labeled for training if the underlying model has high confidence in its prediction. When the model makes inaccurate predictions, it is likely to cause error accumulation that negatively impacts training [5, 2]. To alleviate this problem, many studies [42, 43, 14] guide the model learning based on various label confidence measures and avoid the model from being overly confident on incorrect pseudo-labels. The above works only focus on leveraging both labeled and unlabeled data, but not fully applicable to scenarios with noisy labeled data. FlexSSL considers the contribution of both labeled and unlabeled data through soft labeling, which better addresses aforementioned drawbacks.

5.2 Learning from noisy labels.

The lack of high-quality labels is common in many real-world scenarios [22, 28]. Erroneous or noisy labels can severely degrade the generalization performance. Learning from noisy labels [32] is becoming an increasingly important task in modern deep learning applications. Early studies [6] require unbiased gradient estimates of the loss to provide learning guarantees. Some recent studies [22, 28, 30, 1] provide unbiased estimators for evaluating and training under noise data. However, these methods still require that the observed data have a relatively well-behaved (noise-free) underlying distribution, thus the noisy data can be weighted according to the same distribution. Satisfying such requirements is difficult in practice and the observed distribution may be biased across the data. In contrast, FlexSSL bypasses the dependence on the original data distribution and learn label confidence measures adaptive using an additional discriminator to facilitate model training.

6 Discussion

In the semi-supervised learning problem, the learning ability of a model can be improved from two aspects, one is to increase the amount of information of the labeled data from the positive direction, such as data augmentation based methods, it obtains a larger amount of labeled data by performing certain transformations on the labeled data, and employ consistency regularization in the training process to improve the robustness and accuracy of the model. And the other is to try to mine the hidden information from the unlabeled data, such as pseudo-labeling based methods. FlexSSL focuses on utilizing the large amount of unlabeled data in semi-supervised scenarios, and by soft-labeling the sample data during the training process, the model is able to mine more information hidden in the unlabeled data.

As two different semi-supervised learning strategies, FlexSSL does not conflict with data augmentations and consistency regularization. As the main task model in FLexSSL is trained on all data samples, with additional reweighting loss posed on samples. It is possible to use both types of SSL methods together to further improve the learning ability of the model as long as the consistency regularization loss can be explictly formulated and considered as part of the main task loss $loss_A$. But our goal is to develop a general SSL framework, applicable to a wide range of SSL problems, hence techniques with domain-restrictions such as consistency regularization are not evaluated in this paper.

7 Conclusion

We introduce a novel and efficient semi-supervised learning framework called FlexSSL, which can be easily incorporated into a wide range of SSL models for enhanced performance without the need of any domain-specific knowledge. FlexSSL improves semi-supervised model learning by constructing a cooperative-yet-competitive "game" that establishes cooperation between a main self-interested semi-supervised learning task and a companion task of inferring the observability of labels. Meanwhile, it also offers additional confidence measures for the observed and inferred data labels, which can be particularly useful for the practical applications with noisy labels. Through extensive evaluations, we show that FlexSSL can consistently improve the performance and robustness of semi-supervised classification and regression models with great learning efficiency.

References

[1] G. Algan and I. Ulusoy. Meta soft label generation for noisy labels. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7142–7148. IEEE, 2021.

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

[3] A. Asuncion and D. Newman. Uci machine learning repository, 2007.

[4] D. Berthelot, N. Carlini, I. J. Goodfellow, N. Papernot, A. Oliver, and C. Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5050–5060, 2019.

[5] X. Cai, F. Nie, W. Cai, and H. Huang. Heterogeneous image features integration via multi-modal semi-supervised learning model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1737–1744, 2013.

[6] N. Cesa-Bianchi, S. Shalev-Shwartz, and O. Shamir. Online learning of noisy data. *IEEE Trans. Inf. Theory*, 57(12):7907–7931, 2011.

[7] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. The MIT Press, 2006.

[8] K. Deb. Multi-objective optimization. In *Search methodologies*, pages 403–449. Springer, 2014.

[9] L. Gondara and K. Wang. MIDA: multiple imputation using denoising autoencoders. In *Advances in Knowledge Discovery and Data Mining - 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part III*, pages 260–272, 2018.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] E. Hewitt. Rings of real-valued continuous functions. i. *Transactions of the American Mathematical Society*, 64(1):45–99, 1948.

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

[14] J. Kahn, A. Lee, and A. Hannun. Self-training for end-to-end speech recognition. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 7084–7088, 2020.

[15] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[16] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[17] D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[18] L. Liu, Y. Li, and R. T. Tan. Decoupled certainty-driven consistency loss for semi-supervised learning. *arXiv preprint arXiv:1901.05657*, 2019.

[19] A. Matyasko and L. Chau. Improved network robustness with adversary critic. In *NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10601–10610, 2018.

[20] Y. Meng, J. Shen, C. Zhang, and J. Han. Weakly-supervised hierarchical text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6826–6833, 2019.

[21] Y. Meng, Y. Zhang, J. Huang, C. Xiong, H. Ji, C. Zhang, and J. Han. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9006–9017, 2020.

[22] N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari. Cost-sensitive learning with noisy labels. *J. Mach. Learn. Res.*, 18:155:1–155:33, 2017.

[23] Y. Prabhu and M. Varma. Fastxml: a fast, accurate and stable tree-classifier for extreme multi-label learning. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 263–272, 2014.

[24] H. Qin, X. Zhan, Y. Li, X. Yang, and Y. Zheng. Network-wide traffic states imputation using self-interested coalitional learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1370–1378, 2021.

[25] T. W. Richardson, W. Wu, L. Lin, B. Xu, and E. A. Bernal. Mcflow: Monte carlo flow models for data imputation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 14193–14202, 2020.

[26] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[27] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171, 2016.

[28] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1670–1679, 2016.

[29] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3): 93–93, 2008.

[30] Z. Shen, P. Cui, T. Zhang, and K. Kunag. Stable learning via sample reweighting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5692–5699, 2020.

[31] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. Raffel, E. D. Cubuk, A. Kurakin, and C. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[32] H. Song, M. Kim, D. Park, and J. Lee. Learning from noisy labels with deep neural networks: A survey. *CoRR*, abs/2007.08199, 2020.

[33] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*, 2017.

[34] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1195–1204, 2017.

[35] I. Triguero, S. García, and F. Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems*, 42(2):245–284, 2015.

[36] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[37] Q. Xie, Z. Dai, E. H. Hovy, T. Luong, and Q. Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[38] Q. Xie, M. Luong, E. H. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10684–10695, 2020.

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

[40] B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Third IEEE international conference on data mining*, pages 435–442. IEEE, 2003.

[41] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34, 2021.

[42] Y. Zhou, M. Kantarcioglu, and B. M. Thuraisingham. Self-training with selection-by-rejection. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pages 795–803, 2012.

[43] Y. Zou, Z. Yu, X. Liu, B. V. K. V. Kumar, and J. Wang. Confidence regularized self-training. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 5981–5990, 2019.