MarginMatch: Improving Semi-Supervised Learning with Pseudo-Margins

Tiberiu Sosea Cornelia Caragea
University of Illinois Chicago
tsosea2@uic.edu cornelia@uic.edu

Abstract

We introduce MarginMatch, a new SSL approach combining consistency regularization and pseudo-labeling, with its main novelty arising from the use of unlabeled data training dynamics to measure pseudo-label quality. Instead of using only the model's confidence on an unlabeled example at an arbitrary iteration to decide if the example should be masked or not, MarginMatch also analyzes the behavior *of the model on the pseudo-labeled examples as the training progresses, to ensure low quality predictions are masked out. MarginMatch brings substantial improvements on four vision benchmarks in low data regimes and on two large-scale datasets, emphasizing the importance of enforcing high-quality pseudo-labels. Notably, we obtain an improvement in error rate over the state-of-the-art of* 3.25% *on CIFAR-100 with only* 25 *labels per class and of* 3.78% *on STL-10 using as few as* 4 *labels per class. We make our code available at* https://github.com/tsosea2/MarginMatch.

1. Introduction

Deep learning models have seen tremendous success in many vision tasks [14, 22, 27, 42, 43]. This success can be attributed to their scalability, being able to produce better results when they are trained on large datasets in a supervised fashion [15, 27, 34, 35, 43, 47]. Unfortunately, large labeled datasets annotated for various tasks and domains are difficult to acquire and demand considerable annotation effort or domain expertise. Semi-supervised learning (SSL) is a powerful approach that mitigates the requirement for large labeled datasets by effectively making use of information from unlabeled data, and thus, has been studied extensively in vision [4, 5, 23, 25, 30, 36, 38, 39, 44–46].

Recent SSL approaches integrate two important components: consistency regularization [46, 49] and pseudo-labeling [25]. Consistency regularization works on the assumption that a model should output similar predictions when fed perturbed versions of the same image, whereas pseudo-labeling uses the model's predictions of unlabeled examples as labels to train against. For example, Sohn et al. [41] introduced FixMatch that combines consistency regularization on weak and strong augmentations with pseudo-labeling. FixMatch relies heavily on a high-confidence threshold to compute the unsupervised loss, disregarding any pseudo-labels whose confidence falls below this threshold. While training using only high-confidence pseudo-labels has shown to consistently reduce the confirmation bias [1], this rigid threshold allows access only to a small amount of unlabeled data for training, and thus, ignores a considerable amount of unlabeled examples for which the model's predictions do not exceed the confidence threshold. More recently, Zhang et al. [49] introduced FlexMatch that relaxes the rigid confidence threshold in FixMatch to account for the model's learning status of each class in that it adaptively scales down the threshold for a class to encourage the model to learn from more examples from that class. The flexible thresholds in FlexMatch allow the model to have access to a much larger and diverse set of unlabeled data to learn from, but lowering the thresholds can lead to the introduction of wrong pseudo-labels, which are extremely harmful for generalization. Interestingly, even when the high-confidence threshold is used in FixMatch can result in wrong pseudo-labels. See Figure 1 for incorrect pseudo-labels detected in the training set after we apply FixMatch and FlexMatch on ImageNet. We posit that a drawback of FixMatch and FlexMatch and in general of any pseudo-labeling approach is that they use the confidence of the model only at the current iteration to enforce quality of pseudo-labels and completely ignore model's predictions at prior iterations.

In this paper, we propose MarginMatch, a new SSL approach that monitors the *behavior* of the model on the unlabeled examples as the training progresses, from the beginning of training until the current iteration, instead of using only the model's current *belief* about an unlabeled example (i.e., its confidence at the current iteration) to decide if the example should be masked or not. We estimate a pseudo-label's contribution to learning and generalization by introducing pseudo-margins of unlabeled examples averaged across training iterations. Pseudo-margins of unlabeled examples extend the margins from machine learning [3, 11, 18, 33] which provide a measure of confidence of the outputs of the model and capture the difference between the output for the correct

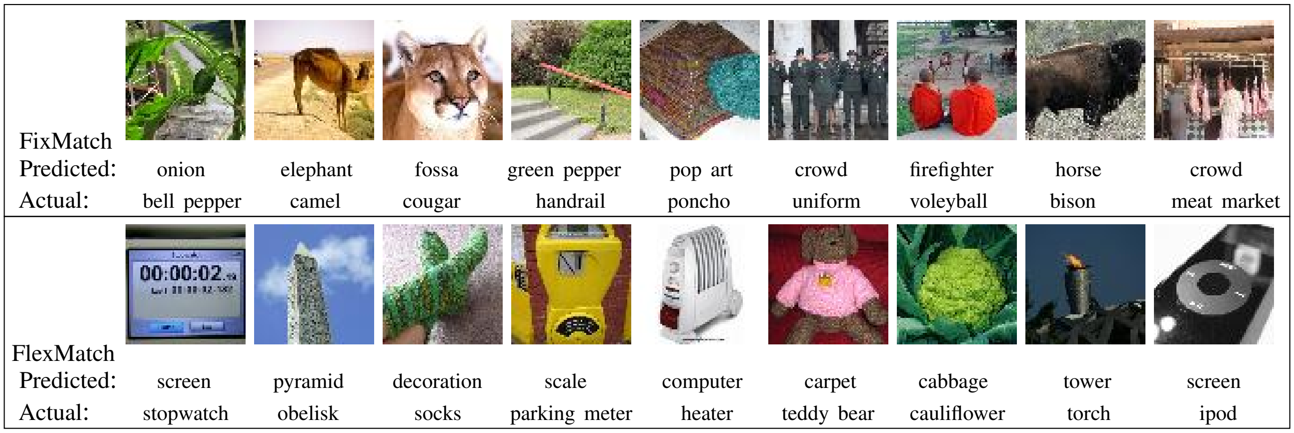

Figure 1. Incorrect pseudo-labels propagated until the end of the training process for FixMatch and FlexMatch on ImageNet.

(gold) label and the other labels. In our case, the pseudo-margins capture how much larger the assigned logit (the logit corresponding to the argmax of the model's prediction) is compared with all other logits at iteration t. Similar to FlexMatch, in MarginMatch we take advantage of the flexible confidence thresholds to allow the model to learn from larger and more diverse sets of unlabeled examples, but unlike FlexMatch, we train the model itself to identify the characteristics of mislabeled pseudo-labels simply by monitoring the model's training dynamics on unlabeled data over the iterations.

We carry out comprehensive experiments using established SSL experimental setups on CIFAR-10, CIFAR-100 [21], SVHN [31], STL-10 [8], ImageNet [10], and WebVision [26]. Despite its simplicity, our findings indicate that MarginMatch produces improvements in performance over strong baselines and prior works on all datasets at no additional computational cost. Notably, compared to current state-of-the-art, on CIFAR-100 we see 3.02% improvement in error rate using only 4 labels per class and 3.78% improvement on STL-10 using the same extremely label-scarce setting of 4 labels per class. In addition, on ImageNet [10] and WebVision [26] we find that MarginMatch pushes the state-of-the-art error rates by 0.97% on ImageNet and by 0.79% on WebVision.

Our contributions are as follows:

1. We introduce a new SSL approach which we call MarginMatch that enforces high pseudo-label quality during training. Our approach allows access to a large set of unlabeled data to learn from (thus, incorporating more information from unlabeled data) and, at the same time, monitors the training dynamics of unlabeled data as training progresses to detect and filter out potentially incorrect pseudo-labels.
2. We show that MarginMatch outperforms existing works on six well-established computer vision benchmarks showing larger improvements in error rates especially on challenging datasets, while achieving similar convergence performance (or better) than prior works.
3. We perform a comprehensive analysis of our approach and indicate potential insights into why our MarginMatch substantially outperforms other SSL techniques.

2. MarginMatch

Notation Let $L = \{(x_1, y_1), ..., (x_B, y_B)\}$ be a batch of size B of **labeled** examples and $U = \{\hat{x}_1, ..., \hat{x}_{\nu B}\}$ be a batch of size νB of **unlabeled** examples, where ν is the batch-wise ratio of unlabeled to labeled examples. Let $p_\theta(y|x)$ denote the class distribution produced by model θ on input image x and $\hat{p}_\theta(y|x)$ denote the argmax of this distribution as a one-hot label. Let also $H(p, q)$ denote the cross-entropy between two probability distributions p and q.

2.1. Background

Consistency reg [39] is an important component in recent semi-supervised learning approaches and relies on the continuity assumption [2, 23] that the model should output similar predictions on multiple perturbed versions of the same input x. As mentioned above, examples of two such approaches are FixMatch [41] and FlexMatch [49] that use consistency regularization at their core combined with psedo-labeling. In psedo-labeling [25], a model itself is used to assign artificial labels for unlabeled data and only artificial labels whose largest class probability is above a predefined confidence threshold are used during training.

Specifically, FixMatch [41] predicts artificial labels for unlabeled examples using a weakly-augmented version of each unlabeled example and then employs the artificial labels as pseudo-labels to train against but this time using a strongly-augmented version of each unlabeled example. That is, FixMatch minimizes the following batch-wise consistency loss on unlabeled data:

$$\mathcal{L}_u = \sum_{i=1}^{\nu B} \mathbb{1}(\max(p_\theta(y|\pi(\hat{x}_i))) > \tau) \quad \times$$
$$H(\hat{p}_\theta(y|\pi(\hat{x}_i)), p_\theta(y|\Pi(\hat{x}_i))) \quad (1)$$

where τ is a confidence threshold, π and Π are weak and strong augmentations, respectively, and $\mathbb{1}$ is the indicator function. Therefore, the low-confidence examples (lower than τ) are completely ignored despite containing potentially useful information for model training.

FlexMatch [49] argues that using a *fixed* threshold τ to filter the unlabeled data ignores the learning difficulties of different classes, and thus, introduces class-dependent thresholds, which are obtained by adaptively scaling τ depending on the learning status of each class. FlexMatch assumes that a class with fewer examples above the fixed threshold τ has a greater learning difficulty, and hence, it adaptively lowers the threshold τ to encourage more training examples from this class to be learned. The learning status α_c for a class c is simply computed as the number of unlabeled examples that are predicted in class c and pass the fixed threshold τ:

$$\alpha_c = \sum_{i=1}^{n} \mathbb{1}(\max(p_\theta(y|\pi(\hat{x}_i))) > \tau)\mathbb{1}(\hat{p}_\theta(y|\pi(\hat{x}_i)) = c) \quad (2)$$

where n is the total number of unlabeled examples. This learning effect is then normalized and used to obtain the class-dependent threshold for each class c:

$$\mathcal{T}_c = \frac{\alpha_c}{\max\limits_c(\alpha_c)} \times \tau \quad (3)$$

In practice, FlexMatch iteratively computes new thresholds after each complete pass through unlabeled data, hence we can parameterize $\mathcal{T}_c$ as $\mathcal{T}_c^t$, denoting the threshold obtained at iteration t. The unlabeled loss is then obtained by plugging in the adaptive threshold $\mathcal{T}_c^t$ in Eq. 1:

$$\mathcal{L}_u = \sum_{i=1}^{\nu B} \mathbb{1}(\max(p_\theta(y|\pi(\hat{x}_i))) > \mathcal{T}^t_{\hat{p}_\theta(y|\pi(\hat{x}_i))}) \quad \times$$
$$H(\hat{p}_\theta(y|\pi(\hat{x}_i)), p_\theta(y|\Pi(\hat{x}_i))) \quad (4)$$

The aforementioned works use the confidence of the model *solely at the current iteration* to enforce quality of pseudo-labels. We believe this is not sufficient as it provides only a myopic view of the model's behavior (i.e., its confidence) on unlabeled data (at a single iteration) and may result in wrong pseudo-labels even when the confidence threshold is high enough (e.g., if the model is miscalibrated or overly-confident [13]). Figure 1 shows examples of images that are added to the training set with a wrong pseudo-label for both FixMatch and FlexMatch. These types of unlabeled examples, which are incorrectly pseudo-labeled and used during training are particularly harmful for deep neural networks, which can attain zero training error on any dataset, even on randomly assigned labels [50], resulting in poor generalization capabilities.

2.2. Proposed Approach: MarginMatch

We now introduce MarginMatch, our new SSL approach that uses the model's training dynamics on unlabeled data to improve pseudo-label data quality. Our approach leverages consistency regularization with weak and strong augmentations and pseudo-labeling, but instead of using only the model's current *belief* (i.e., its confidence at the current iteration) to decide if an unlabeled example should be used for training or not, our MarginMatch monitors the training dynamics of unlabeled data over the iterations by investigating the *margins* (a measure of confidence) of the outputs of the model [3]. The margin of a training example is a well established metric in machine learning [3, 11, 18, 33] that quantifies the difference between the logit corresponding to the assigned ground truth label and the largest other logit.

In our SSL formulation, we redefine the concept of margins to *pseudo-margins* of unlabeled examples since no ground truth labels are available for the unlabeled data. Let c be the pseudo-label (or the argmax of the prediction, i.e., $\hat{p}_\theta(y|\pi(\hat{x}))$) at iteration t on unlabeled example $\hat{x}$ after applying weak augmentations. We define the *pseudo-margin* (PM) of $\hat{x}$ with respect to pseudo-label c at iteration t as follows:

$$\text{PM}_c^t(\hat{x}) = z_c - max_{c!=i}(z_i) \quad (5)$$

where z_c is the logit corresponding to the assigned pseudo-label c and $\max_{c!=i}(z_i)$ is the largest *other* logit corresponding to a label i different from c. To monitor the model's predictions on $\hat{x}$ with respect to pseudo-label c from the beginning of training to iteration t, we average all the margins with respect to c from the first iteration until t and obtain the average pseudo-margin (APM) as follows:

$$\text{APM}_c^t(\hat{x}) = \frac{1}{t}\sum_{j=1}^{t} \text{PM}_c^j(\hat{x}) \quad (6)$$

Here c acts as the "ground truth" label for the APM calculation. Note that if at a prior iteration t', the assigned pseudo-label is different from c (say c'), then the APM calculation at iteration t' is done with respect to c' (by averaging all margins with respect to c' from 1 to t'). In practice, we maintain a vector of pseudo-margins for all classes accumulated over the training iterations and dynamically retrieve the accumulated pseudo-margin value of the argmax class c to obtain the APM_c^t at iteration t.

Intuitively, if c is the pseudo-label of $\hat{x}$ at iteration t, then PM_c^t with respect to class c at iteration t will be positive. In contrast, if the argmax of the model prediction on $\hat{x}$ at a previous iteration $t' < t$ is different from c, then $PM_c^{t'}$ at t'

Algorithm 1 MarginMatch

```
Require: Labeled data $L$; unlabeled data $U$; erroneous examples $E$; maximum number of iterations $T$; number of classes $C+1$ ($C$ original classes plus one virtual class of erroneous examples); $\theta$ model; $\pi$ weak augmentations; $\Pi$ strong augmentations.
Initialize the Average Pseudo-Margin ($APM$) threshold $\gamma^1$ at the first iteration to a small value (e.g., $\gamma^1 = -\infty$).
for t = 1 to $T$ do
    Estimate learning status $\alpha_c$ (using Eq. 2) and calculate the class-wise flexible thresholds $\mathcal{T}_c^t$ (using Eq. 3) for each class $c$.
    while $U$ not exhausted do
        Labeled batch $L_b = \{(x_1, y_1), ..., (x_B, y_B)\}$, unlabeled batch $U_b = \{\hat{x}_1, ..., \hat{x}_{\nu B}\}$, erroneous (or mislabeled) batch $E_b = \{(\tilde{x}_1, C+1), ..., (\tilde{x}_B, C+1)\}$
        for $x \in U_b \cup E_b$ do
            Compute logits $z_c$ for each class $c$ after applying weak augmentations when $x \in U_b$ and strong augmentations when $x \in E_b$.
            Calculate pseudo-margin $PM_c^t$ (using Eq. 5) and update Average $PM_c^t$ (using Eq. 6) for each $c = 1$ to $C+1$.
        end for
        Minimize $\mathcal{L} = \mathcal{L}_s + \lambda(\mathcal{L}_u + \mathcal{L}_e)$
            $\mathcal{L}_s = \frac{1}{B}\sum_{i=1}^{B} H(y_i, p_\theta(y|\pi(x_i)))$
            $\mathcal{L}_u = \sum_{i=1}^{\nu B} \mathbb{1}(\text{AM}^t_{\hat{p}_\theta(y|\pi(\hat{x}_i))}(\hat{x}_i) > \gamma^t) \times \mathbb{1}(\max(p_\theta(y|\pi(\hat{x}_i))) > \mathcal{T}^t_{\hat{p}_\theta(y|\pi(\hat{x}_i))}) \times H(\hat{p}_\theta(y|\pi(\hat{x}_i)), p_\theta(y|\Pi(\hat{x}_i)))$
            $\mathcal{L}_e = \sum_{i=1}^{B} H(C+1, p_\theta(y|\Pi(\tilde{x}_i)))$
    end while
    Update $\gamma^{t+1}$ as the $95^{th}$ percentile erroneous sample $APM_{C+1}^t$.
end for
```

with respect to c will be negative. Therefore, if over the iterations, the model predictions do not agree frequently with the pseudo-label c from iteration t and the model fluctuates significantly between iterations on the predicted label, the APM for class c will have a low, likely negative value. Similarly, if the model is highly uncertain of the class of $\hat{x}$ (reflected in a high entropy of the class probability distribution), the APM for class c will have a low value. These capture the characteristics of mislabeled examples or of those harmful for training. Motivated by these observations, MarginMatch leverages the APM of the assigned pseudo-label c and compares it with an APM threshold to mask out pseudo-labeled examples with low APMs. Formally, the unlabeled loss in MarginMatch is:

$$\mathcal{L}_u = \sum_{i=1}^{\nu B} \mathbb{1}(\text{AM}^t_{\hat{p}_\theta(y|\pi(\hat{x}_i))}(\hat{x}_i) > \gamma^t) \quad \times \\ \mathbb{1}(\max(p_\theta(y|\pi(\hat{x}_i))) > \mathcal{T}^t_{\hat{p}_\theta(y|\pi(\hat{x}_i))}) \quad \times \\ H(\hat{p}_\theta(y|\pi(\hat{x}_i)), p_\theta(y|\Pi(\hat{x}_i))) \quad (7)$$

where γ^t is the APM threshold at iteration t, estimated as explained below, and $\mathcal{T}^t_{\hat{p}_\theta(y|\pi(\hat{x}_i))}$ is the flexible threshold estimated as in FlexMatch [49]. To train our model, we adopt the best practices [41, 49] and optimize the weighted combination of the supervised and unsupervised losses:

$$\mathcal{L} = \mathcal{L}_s + \lambda\mathcal{L}_u \quad (8)$$

where the supervised loss is given by:

$$\mathcal{L}_s = \frac{1}{B}\sum_{i=1}^{B} H(y_i, p_\theta(y|\pi(x_i))) \quad (9)$$

Average Pseudo-Margin Threshold Estimation Inspired by Pleiss et al. [33], we propose to estimate the average pseudo-margin threshold γ^t by analyzing the training dynamics of a special category of unlabeled examples, which we force to be *erroneous* or mislabeled examples. That is, to create the sample of *erroneous* examples E, we randomly sample a subset of unlabeled examples from U that we assign to an inexistent (or virtual) class $C+1$ at the beginning of the training process and remove them from U. The purpose of these erroneous examples is to mimic the training dynamics of incorrectly pseudo-labeled (unlabeled) examples and use them as proxy to estimate the cutoff of (potentially) mislabeled pseudo-labels. Since the examples in E *should* belong to one of the C original classes, assigning them to the inexistent class $C+1$ makes them by definition mislabeled (see Appendix A for additional insights into this virtual class). As with all unlabeled examples from U, we compute APM_{C+1}^t for the special category of erroneous examples from E, but unlike the unlabeled examples from U, the erroneous ones from E have a fixed class $C+1$. To mimic the training dynamics of unlabeled examples from U, we use strong augmentations to compute the loss of the erroneous examples from E. That is, given a batch E_b of B erroneous examples, the erroneous sample loss becomes:

$$\mathcal{L}_e = \sum_{i=1}^{B} H(C+1, p_\theta(y|\Pi(\tilde{x}_i))) \quad (10)$$

At iteration t, we use the APMs of the erroneous examples to choose the APM threshold γ^t. We set γ^t as the APM of the 95^{th} percentile erroneous sample. The total loss becomes:

$$\mathcal{L} = \mathcal{L}_s + \lambda(\mathcal{L}_u + \mathcal{L}_e) \quad (11)$$

Our full MarginMatch algorithm is shown in Algorithm 1.

Dataset	CIFAR-10			CIFAR-100			SVHN			STL-10		
#Labels/Class	4	25	400	4	25	100	4	25	100	4	25	100
Pseuso-Labeling	$74.61_{0.26}$	$46.49_{2.20}$	$15.08_{0.19}$	$87.45_{0.85}$	$57.74_{0.28}$	$36.55_{0.24}$	$64.61_{5.60}$	$25.21_{2.03}$	$9.40_{0.32}$	$74.68_{0.99}$	$55.45_{2.43}$	$32.64_{0.71}$
UDA	$10.79_{3.75}$	$5.32_{0.06}$	$4.41_{0.07}$	$48.95_{1.59}$	$29.43_{0.21}$	$23.87_{0.23}$	$5.34_{4.27}$	$4.26_{0.39}$	$1.95_{0.01}$	$37.82_{8.44}$	$9.81_{1.15}$	$6.81_{0.17}$
MixMatch	$45.24_{2.15}$	$12.76_{1.14}$	$7.13_{0.34}$	$62.15_{2.17}$	$41.51_{1.19}$	$28.16_{0.24}$	$46.18_{1.78}$	$3.98_{0.17}$	$3.5_{0.13}$	$34.15_{1.54}$	$8.95_{0.32}$	$10.41_{0.73}$
ReMixMatch	$5.27_{0.19}$	$4.85_{0.13}$	$4.04_{0.12}$	$47.15_{0.76}$	$27.14_{0.23}$	$23.78_{0.12}$	$4.23_{0.31}$	$3.18_{0.04}$	$1.94_{0.06}$	$31.51_{0.75}$	$8.54_{0.48}$	$6.19_{0.24}$
FixMatch	$7.8_{0.28}$	$4.91_{0.05}$	$4.25_{0.08}$	$48.21_{0.82}$	$29.45_{0.16}$	$22.89_{0.12}$	$3.97_{1.18}$	$\mathbf{3.13}_{\mathbf{1.03}}$	$1.97_{0.03}$	$38.43_{4.14}$	$10.45_{1.04}$	$6.43_{0.33}$
FlexMatch	$5.04_{0.06}$	$5.04_{0.09}$	$4.19_{0.01}$	$39.99_{1.62}$	$26.96_{0.08}$	$22.44_{0.15}$	$8.19_{3.20}$	$7.78_{2.55}$	$6.72_{0.30}$	$29.15_{1.32}$	$8.23_{0.13}$	$5.77_{0.12}$
MarginMatch	$\mathbf{4.91}_{\mathbf{0.07}}$	$\mathbf{4.73}_{\mathbf{0.12}}$	$\mathbf{3.98}_{\mathbf{0.02}}$	$\mathbf{36.97}_{\mathbf{1.32}}$	$\mathbf{23.71}_{\mathbf{0.13}}$	$\mathbf{21.39}_{\mathbf{0.12}}$	$\mathbf{3.75}_{\mathbf{1.20}}$	$3.14_{1.17}$	$\mathbf{1.93}_{\mathbf{0.01}}$	$\mathbf{25.37}_{\mathbf{3.58}}$	$\mathbf{7.31}_{\mathbf{0.35}}$	$\mathbf{5.52}_{\mathbf{0.15}}$

Table 1. Test error rates on CIFAR-10, CIFAR-100, SVHN, and STL-10 datasets. Best results are shown in **blue**.

Exponential Moving Average of Pseudo-Margins The current definition of APM weighs the pseudo-margin at iteration t identical to the pseudo-margin at a much earlier iteration p ($t >> p$). This is problematic since very old pseudo-margins eventually become deprecated (especially due to the large number of iterations through unlabeled data in consistency training ($\sim 9K$)), and hence, the old margins are no longer indicative of the current learning status of the model. To this end, instead of averaging all pseudo-margins (from the beginning of training to the current iteration), we propose to use an exponential moving average to place more importance on recent iterations. Formally, APM becomes:

$$\text{APM}_c^t(\hat{x}) = \text{PM}_c^t(\hat{x}) * \frac{\delta}{1+t} + \text{APM}_c^{t-1}(\hat{x}) * (1 - \frac{\delta}{1+t}) \tag{12}$$

We set the smoothing parameter δ to 0.997 in experiments.

3. Experiments

We evaluate the performance of our MarginMatch on a wide range of SSL benchmark datasets. Specifically, we perform experiments with various numbers of labeled examples on CIFAR-10, CIFAR-100 [21], SVHN [31], STL-10 [8], ImageNet [10], and WebVision [26]. For smaller scale datasets such as CIFAR-10, CIFAR-100, SVHN, and STL-10 we randomly sample a small number of labeled examples per class (ranging from 4 labels per class to 400 labels per class) and treat them as labeled data, whereas the remaining labeled examples are treated as unlabeled data, except for STL-10 [8], which provides its own set of unlabeled examples. On ImageNet and WebVision, we use $\sim 10\%$ of the available labeled examples as labeled data, with the remaining (90%) being treated as unlabeled data. In all our experiments, we sample 5% of the unlabeled data and place it in the set of erroneous examples.

We report the mean and standard deviation of error rates from five runs with different parameter initializations. Similar to FixMatch [41], we use Wide Residual Networks [48]: WRN-28-2 for CIFAR-10 and SVHN; WRN-28-8 for CIFAR-100; and WRN-37-2 for STL-10. We use ResNet-50 [14] for both ImageNet and WebVision.

In our experiments, we adopt the same hyperparameters as FixMatch [41]. Specifically, we use stochastic gradient descent (SGD) with a momentum of 0.9. We start with a learning rate of 0.03 and employ a cosine learning rate; at iteration k, our learning rate is $\eta(k) = cos(\frac{7k\pi}{16K})$, where K is the maximum number of iterations and is set to 2^{20}. We also leverage the same data augmentations as in FlexMatch [49]. Specifically, for weak augmentations we employ a standard flip-and-shift augmentation and use RandAugment [9] for strong augmentations. We set the batch size $B = 64$, ratio of unlabeled to labeled data in a batch to $\nu = 7$ and weigh the supervised and unsupervised losses equally (i.e., $\lambda = 1$). We set our initial confidence threshold to $\tau = 0.95$ and our average pseudo-margin (APM) threshold to the APM of the 95^{th} percentile threshold sample. To report the error rates, we compare all the approaches using the model at the end of training as in FixMatch [41].

3.1. CIFAR-10, CIFAR-100, SVHN, and STL-10

We compare MarginMatch against strong baselines and prior works: Pseudo-Labeling [1], Unsupervised Data Augmentation (UDA) [46], MixMatch [5], ReMixMatch [4], FixMatch [41], and FlexMatch [49]. We show in Table 1 the error rates obtained by our MarginMatch and the baselines on the CIFAR-10, CIFAR-100, SVHN and STL-10 datasets. First, we observe that our approach improves the performance on both CIFAR-10 and CIFAR-100. On CIFAR-10, MarginMatch improves performance in all data regimes upon FlexMatch [49], which is the current state-of-the-art, while mantaining a good error rate standard deviation. On CIFAR-100, which is significantly more challenging than CIFAR-10, we observe that MarginMatch bring substantially larger improvements. Notably, we see 3.02% improvement over FlexMatch in error rate using only 4 labels per class, and 3.25% improvement using 25 examples per class. These results on CIFAR-100 emphasize the effectiveness of MarginMatch, which performs well on a more challenging dataset.

On SVHN, our approach performs better than FixMatch using 4 labels per class and performs similarly with FixMatch using 25 and 100 labels per class. However, on this dataset, MarginMatch performs much better compared with FlexMatch. For example, MarginMatch achieves 3.75% error rate using 4 labels per class, whereas FlexMatch obtains an error rate of 8.19% with the same labels per class,

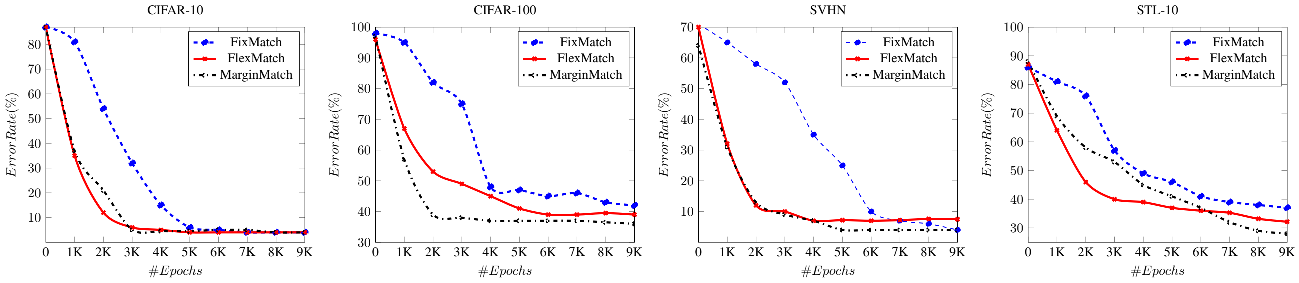

Figure 2. Convergence speed of MarginMatch against FixMatch and FlexMatch with 4 labels per class.

Dataset	ImageNet		WebVision	
	TOP-1	TOP-5	TOP-1	TOP-5
Supervised	48.39	25.49	49.58	26.78
FixMatch	43.66	21.80	44.76	22.65
FlexMatch	42.02	19.49	43.87	22.07
MarginMatch	**41.05**	**18.28**	**43.08**	**21.13**

Table 2. Test error rates on the ImageNet and WebVision datasets. Best results are shown in **blue**.

yielding an improvement of MarginMatch of 4.44% over FlexMatch. We hypothesize that the low performance of FlexMatch is due to its limitation in handling unbalanced class distributions [49]. On STL-10, MarginMatch as well outperforms all the other approaches both in error rates and error rate standard deviation. Notably, on this dataset, our approach pushes the performance of FlexMatch by 3.78% in error rate using only 4 labels per class and by 0.92% using 25 labels per class.

Next, we compare MarginMatch with FixMatch and FlexMatch in terms of convergence speed in the extremely label-scarce setting of 4 labels per class and show these results in Figure 2. Notably, we observe that MarginMatch has a similar convergence speed (or even better on CIFAR-100) compared with FlexMatch while achieving a lower test error rate than FlexMatch on all datasets with 4 labels per class (see Table 1). Even more strikingly, compared with FixMatch, MarginMatch has a much superior convergence speed for a much better test error rate with 4 labels per class. This is because the rigid thresholds in FixMatch allow access only to a small amount of unlabeled data for training at each iteration and it takes a lot longer for the model to train.

3.2. ImageNet and WebVision

To showcase the effectiveness of our approach in a large-scale setup, we test our MarginMatch on ImageNet [10] and WebVision [26] using 10% labeled examples in total. We show the results obtained in Table 2. We observe that our MarginMatch outperforms FixMatch and FlexMatch on both datasets. It is worth noting that large-scale self-supervised approaches such as SimCLR [7] achieve high performance on ImageNet but at a much higher computational cost. MarginMatch outperforms other SSL methods using the same ResNet-50 architecture at the same computational cost. We emphasize MarginMatch is most successful and relevant in low data regimes on smaller datasets.

δ	0.95	0.99	0.995	0.997	0.999	1
ERR RATE	38.13	38.05	37.92	**37.91**	39.12	39.72

Table 3. Error rates obtained on CIFAR-100 with four examples per class and various smoothing values δ. Best result is in **blue**.

4. Ablation Study

Exponential Moving Average Smoothing for APM Computation In our approach, we employ an exponential moving average (EMA) of the pseudo-margin values with a smoothing value of $\delta = 0.997$ to compute the APM. We now analyze how our approach performs with different EMA smoothing values or with no EMA at all. Table 3 shows these results on CIFAR-100 with 4 labels per class. First, we observe that employing a simple average of pseudo-margin values for the APM computation (i.e., $\delta = 1$) performs extremely poorly, obtaining a 39.72% error rate. This result emphasizes that margins eventually become deprecated and it is essential to scale them down in time. Using a low smoothing factor of $\delta = 0.95$ is not effective either, denoting that abruptly forgetting margin values does not work either. Our chosen $\delta = 0.997$ strikes a balance between the two by eliminating the harmful effects of very old margins while keeping track of a good amount of previous estimates (e.g., a margin value computed 200 epochs previously is scaled down by 0.55, while a margin value computed 1000 epochs previously is scaled by 0.05).

Pseudo-Margin vs. Other Measures for Pseudo-Label Correctness Our MarginMatch monitors the pseudo-margins of a model's predictions across training iterations to ensure the quality of pseudo-labels. However, other measures such as confidence or entropy exist that can assess the pseudo-label correctness. Hence, we perform an ablation where we replace the pseudo-margins in our MarginMatch

Dataset	CIFAR-10			CIFAR-100			SVHN			STL-10		
#Labels/Class	4	25	400	4	25	100	4	25	100	4	25	100
Avg Confidence	$23.87_{2.73}$	$14.21_{1.37}$	$7.54_{0.78}$	$41.23_{2.15}$	$31.49_{1.48}$	$24.11_{2.36}$	$8.99_{4.27}$	$6.54_{0.39}$	$4.73_{0.01}$	$31.67_{8.44}$	$14.87_{1.15}$	$7.59_{0.17}$
Avg Entropy	$8.58_{0.41}$	$6.18_{0.15}$	$5.85_{0.12}$	$45.10_{0.91}$	$26.02_{1.11}$	$22.13_{0.25}$	$15.69_{1.25}$	$12.74_{0.78}$	$9.33_{0.05}$	$29.54_{3.51}$	$10.63_{1.35}$	$10.84_{0.47}$
Avg Margin	$7.25_{0.29}$	$5.38_{0.76}$	$4.73_{0.09}$	$39.72_{1.52}$	$25.21_{0.52}$	$23.18_{0.17}$	$18.45_{1.36}$	$11.29_{0.93}$	$8.40_{0.04}$	$28.45_{4.28}$	$9.34_{1.34}$	$7.59_{0.21}$
EMA Confidence	$4.91_{0.45}$	$4.74_{0.09}$	$3.99_{0.06}$	$38.67_{0.74}$	$25.61_{0.12}$	$21.48_{0.17}$	$3.84_{0.23}$	$3.25_{0.03}$	$1.93_{0.09}$	$25.9_{0.81}$	$7.6_{0.42}$	$5.74_{0.57}$
EMA Entropy	$6.4_{0.43}$	$8.34_{0.12}$	$4.21_{0.09}$	$41.63_{0.76}$	$36.84_{0.13}$	$22.52_{0.07}$	$3.81_{1.26}$	$3.17_{0.87}$	$2.14_{0.04}$	$27.21_{4.05}$	$8.28_{1.01}$	$6.79_{0.27}$
EMA Margin	$\mathbf{4.91_{0.07}}$	$\mathbf{4.73_{0.12}}$	$\mathbf{3.98_{0.02}}$	$\mathbf{36.97_{1.32}}$	$\mathbf{23.71_{0.13}}$	$\mathbf{21.39_{0.12}}$	$\mathbf{3.75_{1.20}}$	$\mathbf{3.14_{1.17}}$	$\mathbf{1.93_{0.01}}$	$\mathbf{25.37_{3.58}}$	$\mathbf{7.31_{0.35}}$	$\mathbf{5.52_{0.15}}$

Table 4. Test error rates comparing pseudo-margin with confidence and entropy. Best results are shown in **blue**.

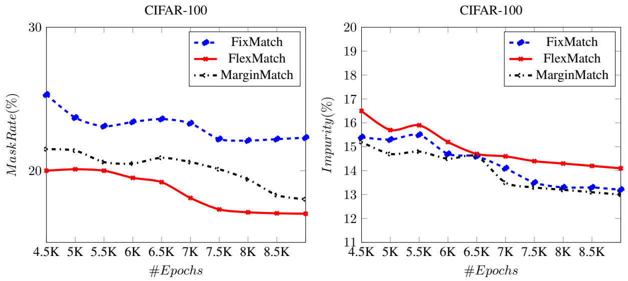

Figure 3. Mask rate and impurity on CIFAR-100 with 4 labeled examples per class.

with average confidence and entropy and compare their performance. Specifically, we design the following approaches: **1) Avg Confidence** monitors the confidence of the prediction for each unlabeled example and takes the average over the training iterations; **2) Avg Entropy** monitors the entropy of the class probability distribution for each unlabeled example and takes the average across the training iterations. In addition, we also consider **3) EMA Confidence** and **4) EMA Entropy** which are similar to Avg Confidence and Avg Entropy, respectively, but use an exponential moving average (EMA) instead of the simple averaging. The estimation of the threshold for each of these approaches is done in a similar manner as for pseudo-margins, using erroneous samples and considering the value of the 95th percentile erroneous sample as the threshold.

We show the results obtained using these approaches in Table 4. First, we observe that all measures (pseudo-margin, confidence and entropy) with EMA perform better than their counterpart with simple averaging. Second, EMA Margin achieves the lowest test error rates compared with EMA Confidence and EMA Entropy. Thus, we conclude that pseudo-margins provide an excellent measure for pseudo-label correctness. See Appendix B for some additional insights into why EMA Margin outperforms EMA confidence and entropy.

5. Analysis

5.1. Mask Rate and Impurity

We now contrast MarginMatch with FixMatch and FlexMatch in terms of the quality of pseudo-labels using two metrics: *mask rate* and *impurity* and show these results in Figure 3, respectively, using CIFAR-100 with 4 labels per class. *Mask rate* is defined as the fraction of pseudo-labeled examples that *do not* participate in the training at epoch t due to confidence masking or pseudo-margin masking (or both). *Impurity* in contrast is defined as the fraction of pseudo-labeled examples that *do* participate in the training at epoch t but with a wrong label. An effective SSL model minimizes both metrics: a low mask rate indicates that the model has access to more unlabeled examples during training (otherwise a low percentage and less diverse set of unlabeled examples are seen during training) while low impurity indicates that the pseudo-labels of these examples are of high quality. Note that we can compute impurity on these two datasets because our unlabeled data comes from the labeled training set of each of these datasets (thus we compare the pseudo-labels against the gold labels of each dataset).

As can be seen from the figures, FixMatch has a significantly larger mask rate due to the rigid confidence threshold set to a high value of 0.95. In contrast, FlexMatch lowers the mask rate by 5% with the introduction of flexible thresholds, but has a much higher impurity compared with FixMatch. Notably, our MarginMatch has only a slightly higher mask rate compared with FlexMatch and at the same time achieves a much lower impurity than FlexMatch and even FixMatch despite that FixMatch employs a very high confidence threshold. These results show that MarginMatch that enforces an additional measure for pseudo-labeled data quality maintains a low mask rate without compromising the quality of the pseudo-labels (i.e., low mask rate and low impurity).

5.2. Anecdotal Evidence

We show in Figure 4 anecdotal evidence of the effectiveness of MarginMatch. To this end, we extract two *bird* images from our unlabeled portion of CIFAR-10 [21] of various learning difficulties that resemble characteristics of *plane* images (e.g., the background). The top part of the figure illustrates the progression over the training iterations of the confidence and the confidence thresholds of FlexMatch for the classes *bird* and *plane*, whereas the bottom part of the figure illustrates the progression of the APM threshold of MarginMatch along with its APMs of *bird* and *plane* classes over the training iterations. In the rightmost image, for MarginMatch we can observe that the APM of the *bird* class becomes stronger and stronger as the training progresses and

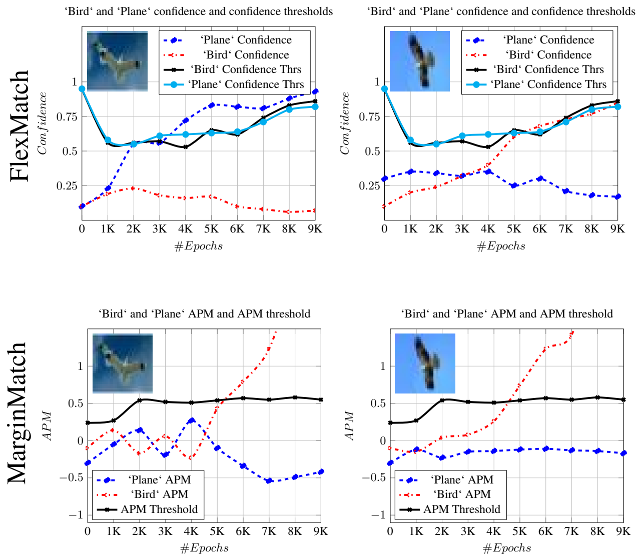

Figure 4. Confidence thresholding vs. APM Thresholding on two images from the CIFAR-10 dataset.

eventually exceeds the APM threshold of MarginMatch, and hence, the image is included in the training until the end with the correct *bird* label. Interestingly, for the same image, in FlexMatch the confidence for the *bird* class is very close to the *bird* confidence threshold and eventually falls below this threshold and exits the training set. In contrast, the leftmost image is significantly more challenging than the rightmost image since it is more similar to *plane* images, which makes it an easily confusable example. Here, we observe that the confidence of FlexMatch exceeds the flexible threshold with the incorrect argmax class *plane* starting from iteration 3000. Moreover, Flexmatch continues to use this image with the wrong *plane* label for the remaining of the training process. Critically, in MarginMatch the APM value for the *plane* class does not exceed the APM threshold, and the model eventually learns to classify this image correctly and includes it in training with the correct *bird* pseudo-label.

6. Related Work

Here, we focus on various SSL approaches that our MarginMatch directly builds upon although there are other SSL techniques that are not presented in this review, such as approaches based on generative models [17, 24], graph-based approaches [19, 20] and robust SSL [32, 37] (see Appendix C for a comparison between MarginMatch and Robust SSL approaches).

Self-training [28, 36, 40, 47] is a popular SSL method where the predictions of a model on unlabeled data are used as artificial labels to train against. Noisy student training [47] is a popular self-training approach that also leverages knowledge distillation [16] and iteratively jointly trains two models in a teacher-student framework. Noisy student uses a larger model size and noised inputs, exposing the student to more difficult learning environments, leading to an increased performance compared to the teacher.

Pseudo-labeling is a variant of self-training where these predictions are sharpened to obtain hard labels [25]. The use of hard labels can be seen as a means of entropy minimization [12] and nowadays is a valuable component in most successful SSL approaches [4, 5, 49]. These hard labels are usually used along a confidence threshold, where unconfident unlabeled examples are completely disregarded (e.g., [5]) to avoid using noisy pseudo-labels. Recently, approaches such as Curriculum Labeling (CL) [6] or FlexMatch [49] started to explore curriculum learning in the SSL context. CL proposes a self-pacing strategy of identifying easy and hard examples to ensure that the model first uses easy and progressively moves towards harder examples. Similarly, MarginMatch uses curriculum learning and pseudo-labeling, but the focus of our approach is placed on producing better thresholds for assessing the quality of pseudo-labels.

Consistency regularization [2] is a method that applies random perturbations when generating the artificial label, such as data augmentation [4, 5, 41], dropout [39], or adversarial perturbations [29]. Current state-of-the-art approaches [41, 49] exploit a combination of weak and strong data augmentations, which were shown to be extremely beneficial in SSL. The most popular strong augmentations used in the SSL literature are RandAugment [9] and CTAugment [4]. The approaches based on these methods first generate a hard label using pseudo-labeling on a weakly augmented image (i.e., using a low noise transformation such as a flip-and-shift augmentation), then optimize the predictions of the model on a strongly augmented version of the same image towards this hard label. Similar to these approaches, MarginMatch uses the same combination of weak and strong data augmentations.

7. Conclusion

In this paper, we proposed a novel semi-supervised learning method that improves the pseudo-label quality using training dynamics. Our new method is lightweight and achieves state-of-the-art performance on four computer vision SSL datasets in low data regimes and on two large-scale benchmarks. MarginMatch takes into consideration not only a flexible confidence threshold to account for the difficulty of each class, but also a measure of quality for each unlabeled example using training dynamics. In addition, MarginMatch is a general approach that can be leveraged in most SSL frameworks and we hope that it can attract future research in analyzing the effectiveness of SSL approaches focused on data quality. As future work, we aim to further explore our method in settings when there is a mismatch between the labeled and unlabeled data distributions (i.e., making use of out-of-domain unlabeled data).

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *CoRR*, abs/1908.02983, 2019. 1, 5

[2] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 2, 8

[3] Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. *CoRR*, abs/1706.08498, 2017. 1, 3

[4] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 1, 5, 8

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019. 1, 5, 8

[6] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *AAAI*, 2021. 8

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 6

[8] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 2, 5

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 5, 8

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 5, 6

[11] Gamaleldin Fathy Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. Large margin deep networks for classification. 2018. 1, 3

[12] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004. 8

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017. 3

[14] 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. 1, 5

[15] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017. 1

[16] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 8

[17] Geoffrey E Hinton and Russ R Salakhutdinov. Using deep belief nets to learn covariance kernels for gaussian processes. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. 8

[18] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions, 2018. 1, 3

[19] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. 8

[20] Thorsten Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 290–297. AAAI Press, 2003. 8

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 5, 7

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc. 1

[23] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR (Poster)*. OpenReview.net, 2017. 1, 2

[24] J.A. Lasserre, C.M. Bishop, and T.P. Minka. Principled hybrids of generative and discriminative models. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 87–94, 2006. 8

[25] Dong-Hyun 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. 1, 2, 8

[26] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *CoRR*, abs/1708.02862, 2017. 2, 5, 6

[27] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018. 1

[28] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA, June 2006. Association for Computational Linguistics. 8

[29] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016. 8

[30] 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

[31] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 2, 5

[32] Jongjin Park, Sukmin Yun, Jongheon Jeong, and Jinwoo Shin. Opencos: Contrastive semi-supervised learning for handling open-set unlabeled data. In *ECCV Workshops*, 2021. 8

[33] Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17044–17056. Curran Associates, Inc., 2020. 1, 3, 4

[34] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 1

[36] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, volume 1, pages 29–36, 2005. 1, 8

[37] Kuniaki Saito, Donghyun Kim, and Kate Saenko. Openmatch: Open-set semi-supervised learning with open-set consistency regularization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25956–25967. Curran Associates, Inc., 2021. 8

[38] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Mutual exclusivity loss for semi-supervised deep learning. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1908–1912. IEEE, 2016. 1

[39] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171, 2016. 1, 2, 8

[40] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965. 8

[41] 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. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 596–608. Curran Associates, Inc., 2020. 1, 2, 4, 5, 8

[42] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1

[43] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 1

[44] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017. 1

[45] Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *Neural Networks*, 2021. 1

[46] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc., 2020. 1, 5

[47] 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, 8

[48] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017. 5

[49] 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. 1, 2, 3, 4, 5, 6, 8

[50] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016. 3