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
(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 \(L = \{(x_1, y_1), \ldots, (x_B, y_B)\}\) be a batch of size \(B\) of labeled examples and \(U = \{\hat{x}_1, \ldots, \hat{x}_{\nu B}\}\) be a batch of size \(\nu B\) of unlabeled examples, where \(\nu\) is the batch-wise ratio of unlabeled to labeled examples. Let \(p_\theta(y|x)\) denote the class distribution produced by model \(\theta\) 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 pseudo-labeling. In pseudo-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:
\[ \mathcal{L}_u = \sum_{i=1}^{n_B} \mathbb{1}(\max(p_\theta(y|\pi(\hat{x}_i))) > \tau) \times H(\hat{p}_\theta(y|\pi(\hat{x}_i)), p_\theta(y|\Pi(\hat{x}_i))) \quad (1) \]

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

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

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

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 \( c \):

\[ T_c = \frac{\alpha_c}{\max(\alpha_c)} \times \tau \]

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

\[ \mathcal{L}_u = \sum_{i=1}^{n_B} \mathbb{1}(\max(p_\theta(y|\pi(\hat{x}_i))) > T^t_c) \mathbb{1}(\hat{p}_\theta(y|\pi(\hat{x}_i)) = c) \times H(\hat{p}_\theta(y|\pi(\hat{x}_i)), p_\theta(y|\Pi(\hat{x}_i))) \quad (4) \]

The aforementioned works use the confidence of the model solely at the current iteration to enforce quality of pseudo-labels. We believe that 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 images of examples 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 \( \hat{c} \) 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:

\[ PM^t_c(\hat{x}) = z_c - \max_{c' \neq c}(z_c) \]

where \( z_c \) is the logit corresponding to the assigned pseudo-label \( c \) and \( \max_{c' \neq c}(z_c) \) 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:

\[ APM^t_c(\hat{x}) = \frac{1}{t} \sum_{j=1}^{t} PM^j_c(\hat{x}) \]

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 \( APM^t_c \) at iteration \( t \).

Intuitively, if \( c \) is the pseudo-label of \( \hat{x} \) at iteration \( t \), then \( PM^t_c \) 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 \( PM^{t'}_c \) 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; II strong augmentations.

1: Initialize the Average Pseudo-Margin (APM) threshold $\gamma^1$ at the first iteration to a small value (e.g., $\gamma^1 = -\infty$).
2: for $t = 1$ to $T$ do
3:   Estimate learning status $\alpha_c$ (using Eq. 2) and calculate the class-wise flexible thresholds $T^t$ (using Eq. 3) for each class $c$.
4:   while $U$ not exhausted do
5:     Labeled batch $L_b = \{(x_1, y_1), ..., (x_B, y_B)\}$, unlabeled batch $U_b = \{(\tilde{x}_1, \tilde{c}_1), ..., (\tilde{x}_B, \tilde{c}_B)\}$, erroneous (or mislabeled) batch $E_b = \{(\tilde{x}_1, C + 1), ..., (\tilde{x}_B, C + 1)\}$
6:     for $x \in U_b \cup E_b$ do
7:       Compute logits $z_c$ for each class $c$ after applying weak augmentations when $x \in U_b$ and strong augmentations when $x \in E_b$.
8:     Calculate pseudo-margin $PM_t^c$ (using Eq. 5) and update Average Pseudo-Margin $\hat{APM}^{t+1}_c$ for each $c = 1$ to $C + 1$.
9:   end while
10:  Minimize $\mathcal{L} = \mathcal{L}_s + \lambda(\mathcal{L}_u + \mathcal{L}_e)$
11:  $\mathcal{L}_s = \frac{1}{B} \sum_{i=1}^{B} H(y_i, p_{\theta}(y|\pi(x_i)))$
12:  $\mathcal{L}_u = \sum_{i=1}^{B} \mathbb{1}(\hat{A}M^t_{\theta}(y|\pi(\hat{x}_i)) (\hat{x}_i) > \gamma^t) \times \mathbb{1}(\max(p_{\theta}(y|\pi(\hat{x}_i))) > T^t_{\theta}(y|\pi(\hat{x}_i))) \times H(p_{\theta}(y|\pi(\hat{x}_i)), p_{\theta}(y|\Pi(\hat{x}_i)))$
13:  $\mathcal{L}_e = \sum_{i=1}^{B} H(C + 1, p_{\theta}(y|\Pi(\hat{x}_i)))$
14: end while
15: Update $\gamma^{t+1}$ as the $95^{th}$ percentile erroneous sample $\hat{APM}^{t+1}_{C+1}$.
16: 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 $\gamma$ at iteration $t$, estimated as explained below, and $T^t_{\theta}(y|\pi(\hat{x}_i))$ 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:

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

where the supervised loss is given by:

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

Average Pseudo-Margin Threshold Estimation Inspired by Pleiss et al. [33], we propose to estimate the average pseudo-margin threshold $\gamma^t$ 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 $APM_t^c$ 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:

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

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

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

Our full MarginMatch algorithm is shown in Algorithm 1.
Table 1. Test error rates on CIFAR-10, CIFAR-100, SVHN, and STL-10 datasets. Best results are shown in blue.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
<th>SVHN</th>
<th>STL-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Labels/Class</td>
<td>4</td>
<td>25</td>
<td>400</td>
<td>4</td>
</tr>
<tr>
<td>Pseudo-Labeling</td>
<td>74.6%</td>
<td>46.49%</td>
<td>15.08%</td>
<td>87.45%</td>
</tr>
<tr>
<td>UDA</td>
<td>10.79%</td>
<td>5.32%</td>
<td>4.41%</td>
<td>48.95%</td>
</tr>
<tr>
<td>MixMatch</td>
<td>45.24%</td>
<td>12.76%</td>
<td>7.13%</td>
<td>62.15%</td>
</tr>
<tr>
<td>ReMixMatch</td>
<td>5.37%</td>
<td>4.85%</td>
<td>4.04%</td>
<td>47.15%</td>
</tr>
<tr>
<td>FixMatch</td>
<td>7.8%</td>
<td>4.91%</td>
<td>2.5%</td>
<td>48.21%</td>
</tr>
<tr>
<td>FlexMatch</td>
<td>5.04%</td>
<td>5.04%</td>
<td>4.19%</td>
<td>39.99%</td>
</tr>
<tr>
<td>MarginMatch</td>
<td>4.91%</td>
<td>4.73%</td>
<td>3.98%</td>
<td>36.97%</td>
</tr>
</tbody>
</table>

3.3. 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 maintaining a good error rate standard deviation. On CIFAR-100, which is significantly more challenging than CIFAR-10, we observe that MarginMatch brings 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.
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

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ImageNet</th>
<th>WebVision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TOP-1</td>
<td>TOP-5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervised</td>
<td>48.39</td>
<td>25.49</td>
</tr>
<tr>
<td>FixMatch</td>
<td>43.66</td>
<td>21.80</td>
</tr>
<tr>
<td>FlexMatch</td>
<td>42.02</td>
<td>19.49</td>
</tr>
<tr>
<td>MarginMatch</td>
<td>41.05</td>
<td>18.28</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>0.95</th>
<th>0.99</th>
<th>0.995</th>
<th>0.997</th>
<th>0.999</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR RATE</td>
<td>38.13</td>
<td>38.05</td>
<td>37.92</td>
<td><strong>37.91</strong></td>
<td>39.12</td>
<td>39.72</td>
</tr>
</tbody>
</table>

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.

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
<th>SVHN</th>
<th>STL-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Labels/Class</td>
<td>4</td>
<td>25</td>
<td>400</td>
<td>4</td>
</tr>
<tr>
<td>Avg Confidence</td>
<td>23.87%</td>
<td>14.21%</td>
<td>7.54%</td>
<td>7.84%</td>
</tr>
<tr>
<td>Avg Entropy</td>
<td>8.58%</td>
<td>4.01</td>
<td>5.86%</td>
<td>5.12%</td>
</tr>
<tr>
<td>Avg Margin</td>
<td>7.41%</td>
<td>2.20%</td>
<td>5.38%</td>
<td>7.07%</td>
</tr>
</tbody>
</table>

EMMA Accuracy | 4.91% | 4.5% | 4.74% | 0.09 | 3.99% | 0.26 | 38.67% | 0.74 | 25.64% | 0.12 | 21.48% | 0.17 | 3.84% | 0.23 | 3.25% | 0.63 | 1.95% | 0.09 | 25.92% | 0.61 | 7.6% | 0.42 | 5.74% | 0.57 |
| EMA Entropy | 6.42% | 4.01 | 8.34% | 1.22 | 4.22% | 0.04 | 41.63% | 0.76 | 36.84% | 0.11 | 32.52% | 0.07 | 3.81% | 0.26 | 3.17% | 0.00 | 2.14% | 0.00 | 27.21% | 0.05 | 8.28% | 0.01 | 6.7% | 0.27 |
| EMA Margin | 4.91% | 4.07 | 4.74% | 0.12 | 3.98% | 0.02 | 36.97% | 0.22 | 23.71% | 0.12 | 21.39% | 0.12 | 3.75% | 0.03 | 3.14% | 0.14 | 1.95% | 0.01 | 25.37% | 0.54 | 7.31% | 0.35 | 5.5% | 0.15 |

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
where the predictions of a model on unlabeled data are used with the incorrect argmax class. 

Related Work

models in a teacher-student framework. Noisy student uses a knowledge distillation [16] and iteratively jointly trains two [47] is a popular self-training approach that also leverages other SSL techniques that are not presented in this review, MarginMatch directly builds upon although there are approaches based on these methods [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 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 show 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


Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adver-


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

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


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


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