Boosting Semi-Supervised Learning by Exploiting All Unlabeled Data

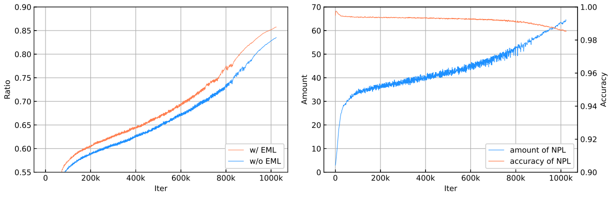
Yuhao Chen[1] Xin Tan[2*] Borui Zhao[1] Zhaowei Chen[1] Renjie Song[1] Jiajun Liang[1] Xuequan Lu[3]
[1]MEGVII Technology [2]East China Normal University [3]Deakin University
{yhao.chen0617, zhaoborui.gm, chaowechan}@gmail.com, xtan@cs.ecnu.edu.cn
{songrenjie, liangjiajun}@megvii.com, xuequan.lu@deakin.edu.au

Abstract

Semi-supervised learning (SSL) has attracted enormous attention due to its vast potential of mitigating the dependence on large labeled datasets. The latest methods (e.g., FixMatch) use a combination of consistency regularization and pseudo-labeling to achieve remarkable successes. However, these methods all suffer from the waste of complicated examples since all pseudo-labels have to be selected by a high threshold to filter out noisy ones. Hence, the examples with ambiguous predictions will not contribute to the training phase. For better leveraging all unlabeled examples, we propose two novel techniques: Entropy Meaning Loss (EML) and Adaptive Negative Learning (ANL). EML incorporates the prediction distribution of non-target classes into the optimization objective to avoid competition with target class, and thus generating more high-confidence predictions for selecting pseudo-label. ANL introduces the additional negative pseudo-label for all unlabeled data to leverage low-confidence examples. It adaptively allocates this label by dynamically evaluating the top-k performance of the model. EML and ANL do not introduce any additional parameter and hyperparameter. We integrate these techniques with FixMatch, and develop a simple yet powerful framework called FullMatch. Extensive experiments on several common SSL benchmarks (CIFAR-10/100, SVHN, STL-10 and ImageNet) demonstrate that FullMatch exceeds FixMatch by a large margin. Integrated with FlexMatch (an advanced FixMatch-based framework), we achieve state-of-the-art performance. Source code is available at https://github.com/megvii-research/FullMatch.

1. Introduction

Semi-supervised learning (SSL) is proposed to leverage an abundance of unlabeled data to enhance the model's performance when labeled data is limited [44]. Consistency regularization [1, 3, 23] and pseudo labeling [10, 12, 25] have shown significant ability for leveraging unlabeled data and thus they are widely used in SSL frameworks. Recently, FixMatch-based methods [21, 26, 31, 38] that combine the two technologies in a unified framework have achieved noticeable successes. Specifically, they apply weak-augmentation (e.g., only random flip and shift) to unlabeled data and obtain their predictions, and then corresponding one-hot pseudo-label is generated if the largest prediction confidence is beyond the predefined threshold (e.g., 0.95), finally it is used as a training target when inputting the strongly-augmented examples (e.g., RandAugment [6], Cutout [8]).



(a)

(b)

Figure 1. Visualization of the experimental results on CIFAR-100 with 10000 labeled images. Evaluations are done every 1K iterations. (a) The increasing proportion of examples with pseudo-label when applying EML to FixMatch. (b) The number of negative pseudo-labels per sample and accuracy during the whole training process. "NPL" denotes negative pseudo-labels.

However, the FixMatch-based methods still have a significant drawback that they rely on an extremely high threshold to produce accurate pseudo-labels, which results in ignoring a large number of unlabeled examples with ambiguous predictions, especially on the early and middle training stages. We can easily observe this phenomenon according to the blue curve in Fig. 1(a), which visualizes the proportion of samples with pseudo-labels at different training iterations when applying FixMatch [26] to CIFAR-100 [16] with 10,000 labels. It shows that the ratio of selected examples with pseudo-label is around 58% after 200k iterations and merely reaches 84% in the end. This moti-

[*]Corresponding author. He is sponsored by Shanghai Sailing Program (23YF1410500).

vates us to exploit more unlabeled data to boost the overall performance.

One intuitive solution is to *assign pseudo-label for potential examples* (i.e., the maximum confidence is close to the predefined threshold). We argue the competition between partial classes leads to failure to produce high-confidence prediction, while the unsupervised loss of FixMatch (i.e, cross-entropy) only focus on the target class when training the examples with pseudo-label. Therefore, we propose a novel scheme to enhance confidence on target class, namely Entropy Meaning Loss (EML). For examples with pseudo-label, EML imposes additional supervision on all non-target classes (i.e., classes which specify the absence of a specific label) to push their prediction close to a uniform distribution, thus preventing any class competition with the target class. Fig. 1(a) illustrates that FixMatch equipped with EML can select more examples with the pseudo-label during the whole training process. Since EML attempts to yield more low-entropy predictions to select more examples with pseudo-label rather than tuning the threshold, it can be also applied to any dynamic-threshold methods (e.g., FlexMatch [38], Dash [34]).

Nevertheless, it is still impossible to leverage all unlabeled data by generating pseudo-labels with a threshold strategy. This motivates us to further consider *how to utilize the low-confidence unlabeled examples without pseudo-label* (i.e., the maximum confidence is far from the predefined threshold). Intuitively, the prediction may get confused among the top classes, but it will be confident that the input does not belong to the categories ranked after these classes. Fig. 2 shows an inference result of FixMatch. The ground truth is "cat", FixMatch is confused by several top classes (e.g., "dog", "frog") and make low-confidence prediction, however it shows highly confidence that some low-rank classes (e.g.,"airplane", "horse") are not ground truth class, thus we can safely assign *negative pseudo-labels* to these classes. Based on this insight, we propose a novel method named Adaptive Negative Learning (ANL). Specifically, ANL first calculate a k *adaptively* based on the prediction consistency, so that the accuracy of top-k is close to 1, and then regard the classes ranked after k as negative pseudo-labels. Furthermore, if the example is selected a pseudo-label, ANL will shrink the range of non-target classes (i.e., EML only needs constrain the top-k classes except target class). Note that ANL is a threshold-independent scheme and thus can be applied on *all* unlabeled data. As shown in Fig. 1(b), our ANL's rendered negative pseudo-labels are increasing as the model is optimized while keeping high accuracy. In summary, our method **makes full use** of the unlabeled dataset, which is hardly ever seen in modern SSL algorithms.

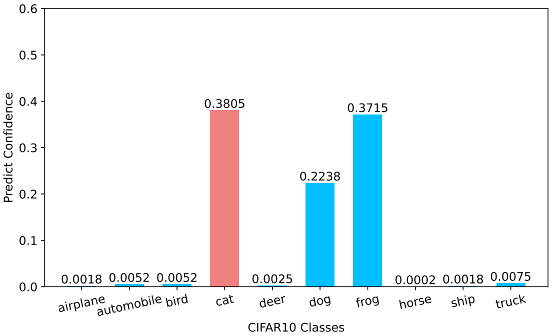


Figure 2. An example of inference result of FixMatch. It can conclude that the input does not belong to these low-rank classes, such as airplanes, horse.

To demonstrate the effectiveness of the proposed EML and ANL, we simply integrate EML and ANL to FixMatch and exploit a new framework named FullMatch. We conduct various experiments on CIFAR-10/100 [16], SVHN [22], STL-10 [5] and ImageNet [7]. The results demonstrate that FullMatch surpasses the performance of FixMatch with a large margin while the training cost remains similar to FixMatch. Moreover, our method can be easily adapted to other FixMatch-based algorithms and obtain further improvement. For example, by combining it with FlexMatch [38], we achieve state-of-the-art performance. To summarize, our key contributions include:

1) We introduce an additional supervision namely Entropy Meaning Loss (EML) when training examples with pseudo-label, which enforces a uniform distribution of non-target classes to avoid them competition with target class and thus producing more high-confidence predictions.

2) We propose the Adaptive Negative Learning (ANL), a dynamic negative pseudo-labels allocation scheme, which renders negative pseudo-labels with very limited extra computational overhead for all unlabeled data, including the low-confidence ones.

3) We design a simple yet effective framework named FullMatch by simply integrating FixMatch with the proposed EML and ANL, which leverages **all unlabeled data** and thus achieving remarkable gains on five benchmarks. Furthermore, our method is shown to be orthogonal to other FixMatch-based frameworks. Specifically, FlexMatch with our method, achieves state-of-the-art results.

2. Related Work

Semi-supervised learning is one of the fundamental tasks in machine learning and computer vision. The goal of SSL is to learn from unlabeled samples with the guidance of limited labeled samples. In this section, we focus only on approaches closely relevant to our method.

Entropy minimization. It has been proved effective in SSL. [12] pointed out that unlabeled data should be used to facilitate a network for generating predictions far from the decision boundary. We can achieve this by promoting the network to make low entropy output distribution (i.e., high confidence prediction) on unlabeled data. [12, 20] added an explicit loss term to constrain the prediction entropy across C classes on all unlabeled data: $-\sum_{c=1}^{C}\mu_c log(\mu_c)$. Pseudo-labeling [17] implicitly minimized the prediction entropy by converting model predictions to one-hot label and only retaining those when the highest class prediction probability is above on predefined threshold. Nowadays, pseudo-labeling has always been used in modern SSL algorithms [15, 32, 35, 42] as a component of their pipeline to produce better performance. Inspired by them, our method attempts to obtain separable classification boundaries but imposes additional supervision on non-target classes.

Consistency regularization. It has been extensively used in SSL [3, 20, 40] to hold similar output distribution when input was perturbed. Since data augmentation [13,36,39] shows huge superiority in supervised learning, recent works have begun to give more attention to data augmentation perturbation and have achieved significant successes. For example, UDA [33], RemixMatch [2] and FixMatch [26] all employed weakly-augmented strategies to produce the training target for unlabeled data and enforce prediction consistency against strongly-augmented version. The difference between FixMatch and UDA/RemixMatch is that FixMatch adopts pseudo-labeling instead of employing a “soft” label by sharpening the predicted distributions, which is beneficial to entropy minimization. Therefore, FixMatch obtains better performance and is a milestone algorithm in SSL. However, the predefined threshold used in FixMatch causes certain low-confidence examples to make no contributions to the model learning, and FlexMatch introduces Curriculum Pseudo Labeling to dynamically adjust the confidence threshold and remarkably boosts performance. By contrast, the proposed EML enhances the prediction confidence of the model itself by enlarging the distinction between the target and non-target classes, thus generating more low entropy predictions under the same threshold. This reveals that our method can obtain better improvement either with a fixed threshold or dynamic threshold.

Negative learning. It is an indirect learning strategy where the category of the inputs is not the same as the supervised learning. Compared with the “positive label” (i.e., the image belongs to this category), the superiority of negative labels is less-cost and more accurate. UPS [24] and NS^{3}L [4] select negative labels for the classes whose probability values fall below a fixed small threshold (e.g., 0.01). In other words, these methods still utilize the “high-confidence” prediction based on the (low) probability value. Obviously, these methods can not label samples with negative labels when given ambiguous predictions (e.g., all probability values are between 0.01 and 0.95). By contrast, our proposed ANL focus on the rank of categories rather than the probability value, it adaptively assigns negative pseudo-labels for *all* unlabeled data while maintaining simplicity (i.e., no extra threshold hyperparameter).

3. Method

As discussed above, we will concentrate on two questions in this section: 1) how to allocate more examples with pseudo-label; 2) how to learn knowledge from unlabeled examples with ambiguous predictions. Correspondingly, we propose two novel and efficient techniques: Entropy Meaning Loss (EML) and Adaptive Negative Learning (ANL). EML constrains the output distribution of non-target classes to obtain more separable decision boundaries, thus generating more high-confidence predictions. ANL dynamically assigns negative pseudo-label based on the model’s optimization status to leverage examples with ambiguous prediction. By applying these two key components to FixMatch [26], we can employ the total unlabeled dataset and bring improvements for various baselines. In this section, we first review the key components of FixMatch. Then, we explain the proposed Entropy Meaning Loss (EML) and Adaptive Negative Learning (ANL), respectively. Finally, we introduce the FullMatch algorithm by integrating our method with FixMatch.

3.1. Preliminaries

Consistency regularization is proved very useful in SSL. The original consistency loss in SSL is a $L-2$ loss.

$$\sum_{i=1}^{B}(\left\|p_m(y|\omega(\mu^{(i)}) - p_m(y|\phi(\mu^{(i)}))\right\|_2^2) \tag{1}$$

where p_m denotes the prediction distribution of the model. ω and ϕ are different perturbations imposed on the unlabeled examples $\mu^{(i)}$. B represents the batch size of unlabeled examples. FixMatch introduces Pseudo-Labeling techniques related to entropy minimization in the consistency regularization process. The improved consistency loss function in FixMatch can be formulated as:

$$\frac{1}{B}\sum_{i=1}^{B}\mathbb{1}(max(Q^{(i)}) \geq \tau)H(\hat{Q^{(i)}}, P^{(i)}) \tag{2}$$

where $Q^{(i)} = p_m(y|\omega(\mu^{(i)}))$ and $P^{(i)} = p_m(y|\phi(\mu^{(i)}))$ represents the prediction distribution of the weakly-augmented version and strongly-augmented version, respectively. ω and ϕ denote weakly and strongly augmentations. $\hat{Q^{(i)}} = argmax(Q^{(i)})$ is the hard target. τ is a confidence threshold and H represents the cross-entropy loss function. FixMatch generates the pseudo-label according to the output distribution with weak-augmented inputs, and then calculates the difference from strongly-augmented inputs.

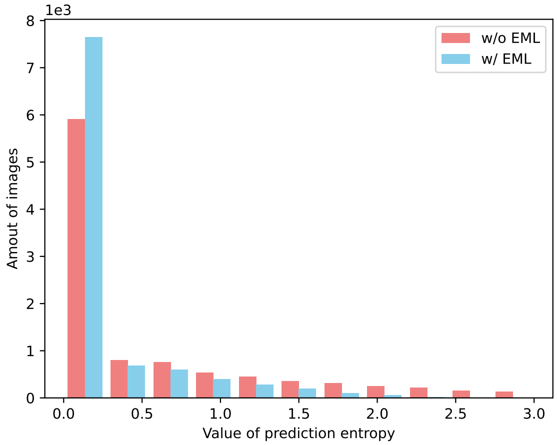


Figure 3. Visualization the distribution of prediction entropy when adopting EML to FixMatch on CIFAR-100 testset . The model supervised by EML can generate more low-entropy predictions and thus select more examples with pseudo-label.

As suggested by previous research, a high confidence threshold τ generates accurate pseudo-label but filters out lots of unlabeled data with low-confidence predictions, thus causing the under-exploration of the unlabeled data (see Fig. 1(a)). We will propose two simple yet effective schemes to address the dilemma below.

3.2. Entropy Meaning Loss

We propose Entropy Meaning Loss to allocate more samples with pseudo-labels. Most current works focus on dynamically adjusting the threshold (e.g., FlexMatch, Dash). Unlike them, we aim to strengthen the representation ability of the model itself to produce more predictions far from the decision boundary (i.e., high-confidence predictions), which means it is orthogonal to those dynamic thresholding works.

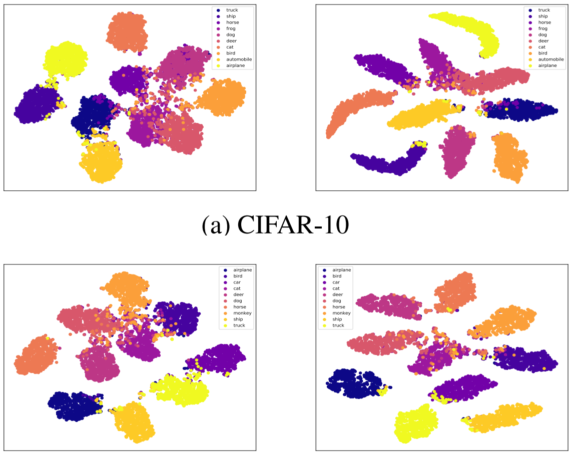
We assume $Q^{(i)} = [q_1^{(i)}, ..., q_C^{(i)}]$ represents the prediction vector for the weakly-augmented version of sample i. Let $S^{(i)} = [s_1^{(i)}, ..., s_C^{(i)}] \subseteq \{0, 1\}^C$ be a binary vector denoting the selected labels, where $s_c^{(i)} = 1$ represents the class c that is selected as a target class (e.g., pseudo-label class) and $s_c^{(i)} = 0$ when this class is absence of a specific label. The vector can be computed as:

$$s_c^{(i)} = \mathbb{1}(q_c^{(i)} \geq \tau) \tag{3}$$

where τ is the selection threshold. Furthermore, we can calculate the vector $U^{(i)} = [u_1^{(i)}, ..., u_C^{(i)}]$, where $u_c^{(i)} = 1$ denotes class c is a non-target class and sample i is assigned a pseudo-label, which is formulated as:

$$u_c^{(i)} = \mathbb{1}(max(Q^{(i)}) \geq \tau) \cdot \mathbb{1}(s_c^{(i)} = 0) \tag{4}$$

We assume $P^{(i)} = [p_1^{(i)}, ..., p_C^{(i)}]$ represents the prediction confidence vector on the strongly-augmented of sample i, and $p_{tc}^{(i)}$ denotes the confidence of the target class (i.e., pseudo-label class). With the optimization by the unsupervised loss function (i.e, cross-entropy), $p_{tc}^{(i)}$ will gradually converge to the given label (i.e., the confidence of pseudo-label class will gradually increase to 1 with model learning). But for certain challenging examples, the competition between confusion classes and target class always leads to a failure in generating high-confidence prediction. To tackle this issue, we impose an additional constraint on the rest of the categories (i.e., all non-target classes) to allow them to share the remaining confidence $1 - p_{tc}^{(i)}$ equally to avoid any class competition with the target class. This can be formulated as:



(a) CIFAR-10

(b) STL-10

Figure 4. T-SNE visualization when adopting EML to FixMatch. The model supervised by EML (right) produces more clean and separable decision boundaries.

$$y_c^{(i)} = \frac{1 - \mathbb{1}(u_c^{(i)} = 0) \cdot p_c^{(i)}}{\sum_c \mathbb{1}(u_c^{(i)} = 1)} \tag{5}$$

where $y_c^{(i)}$ is the label of the non-target classes. It indicates that once the predicted probability of the target class is determined, other non-target classes should share the remaining confidence scores equally. Note that EML is just applied on the pseudo-label samples, which means $\sum_c \mathbb{1}(u_c^{(i)} = 1)$ is always larger than 0 due to $max(Q^{(i)}) \geq \tau$. Since $y_c^{(i)}$ ranges from 0 to 1, the model can be trained with a binary cross entropy (BCE) loss. Thus, our proposed Entropy Meaning Loss (EML) can be defined as:

$$\mathcal{L}_{eml} = -\frac{1}{BC} \sum_{i=1}^{B} \sum_{c=1}^{C} u_c^{(i)} \cdot [y_c^{(i)} log(p_c^{(i)}) \\ +(1 - y_c^{(i)})(log(1 - p_c^{(i)}))] \tag{6}$$

Note that $y_c^{(i)}$ is calculated by the score of target class, therefore EML will also produce gradient to target class, which can be computed as:

$$g_{tc}^{(i)} = -\frac{1}{BC(C-1)} log(\frac{\prod\limits_{c=0,c\neq tc}^{C}(1-p_c^{(i)})}{\prod\limits_{c=0,c\neq tc}^{C} p_c^{(i)}}) \tag{7}$$

the gradient directions of EML and unsupervised loss $\mathcal{L}_{us}$ (Eq.(2)) (i.e., cross-entropy loss) are the same, which indicates EML can cooperate with $\mathcal{L}_{us}$ to further promote the confidence of target class while constraining the distribution of the non-target classes. For detailed proof, please refer to *Supplementary Material*, Section A.

To intuitively illustrate the effectiveness of EML, Fig. 3 compares the distributions of prediction entropy on CIFAR-100 testset with or without introducing EML. The total images is 10000. Obviously, the amount of low-entropy prediction (e.g., the value of prediction entropy is less than 0.25) will increase about 18% (78% vs 60%) when introducing EML. We further show the effectiveness of EML by using t-SNE [28] on CIFAR-10 and STL-10 [5]. Fig. 4 demonstrates that EML can produce more clean and separable boundaries, hence giving more high-confidence predictions.

3.3. Adaptive Negative Learning

Since it is easy to produce ambiguous predictions on complicated scenarios (e.g., the largest confidence is only 0.3 while the threshold is 0.95), these examples are difficult to be assigned pseudo-label (filtered by the threshold or incorrect predictions), resulting in no contributions to the model optimization. To address this, we allocate an additional label with less noise to leverage these examples.

Fig. 6 illustrates the top-k (e.g., top-5 and top-9) accuracy curves of FixMatch on CIFAR-10 when the amount of labeled data is only 40. The top-5 prediction can reach a promising accuracy after 100k iterations, which means all unlabeled data in CIFAR-10 do not belong to the $last$-5 prediction classes (i.e., categories after top-5) with a high chance when iterations are greater than 100k. This phenomenon motivates us to render negative pseudo-labels for unlabeled data.

Consequently, an ideal approach is to exploit an additional dataset to evaluate the top-k performance, thereby calculating a suitable k value so that the top-k error rate is close to zero. Since we cannot employ the test set in the model training procedure, an optional strategy is to separate an additional validation set from the labeled training data. Nevertheless, this brings two severe defects: 1) separating a validation set from a labeled training set is expensive, especially when the amount of labeled data is particularly limited (e.g., each class only has 4 labels). 2) An extra forward propagation is essential to dynamically regulate k at each iteration, leading to a dramatic drop in the efficiency of model training.

In this work, we present a scheme to approximately evaluate the top-k performance, referred to as Adaptive Negative Learning (ANL). ANL does not require an additional labeled validation set, nor redundant inference processes. This is inspired by UDA [33] that by optimizing the consistency between two augmented versions, the model becomes smoother with respect to changes in the input space and thus, the overall performance can be better. Therefore, our key assumption is that the model performance can be reflected by the consistency of the predictions with different augmented inputs. That is, we first compute the $temp$ labels according to the weakly-augmented prediction regardless of whether the max score is larger than the threshold, then we view the $temp$ labels as the ground truth of the strongly-augmented version and calculate the minimum k so as to its top-k accuracy is 100%. This can be formulated as:

$$k = \arg\min_{\theta\in[2,C]} (Acc(P_t, \hat{Q}_t, \theta) = 100\%) \tag{8}$$

where $\hat{Q}_t = \arg\max(Q, t)$ is $temp$ labels at step t, P_t denotes the prediction vector of strongly-augmented and they are calculated across the total batch samples. Acc and C represent the function of calculating top-k accuracy and the number of categories, respectively. Since there are always certain examples without pseudo-labels at each training step (see Fig. 1(a)) and meanwhile we calculate k on *all* unlabeled data, the over-fitting issue can be alleviated. Finally, we assign negative pseudo-labels to categories ranked after top-k on the prediction distributions of the weakly-augmented version. As a result, the vector $S^{(i)}$ (Sec. 3.2) can be recalculated as:

$$s_c^{(i)} = \mathbb{1}[q_c^{(i)} \geq \tau] + \mathbb{1}[Rank(q_c^{(i)}) > k] \tag{9}$$

where $Rank$ is a category sorting function based on confidence scores in the descending order. In the early stage of the training process, when the model is fed with different augmented versions on the same samples, the output distribution is significantly different, thus the value of k will be enlarged, i.e., ANL will not afford any negative pseudo-labels when $k = C$. With the optimization of consistency loss (i.e., cross-entropy loss), the model has stronger output invariant to input noise, the value of k will turn smaller, and more negative pseudo-labels will be selected. The adaptive negative learning loss $\mathcal{L}_{anl}$ can be formulated as:

$$-\frac{1}{B}\sum_{i=1}^{B}\sum_{c=1}^{C} \mathbb{1}[Rank(q_c^{(i)}) > k]\ log(1-p_c^{(i)}) \tag{10}$$

Note that the expense of employing ANL is almost free. It does not introduce any extra forward propagation pro-

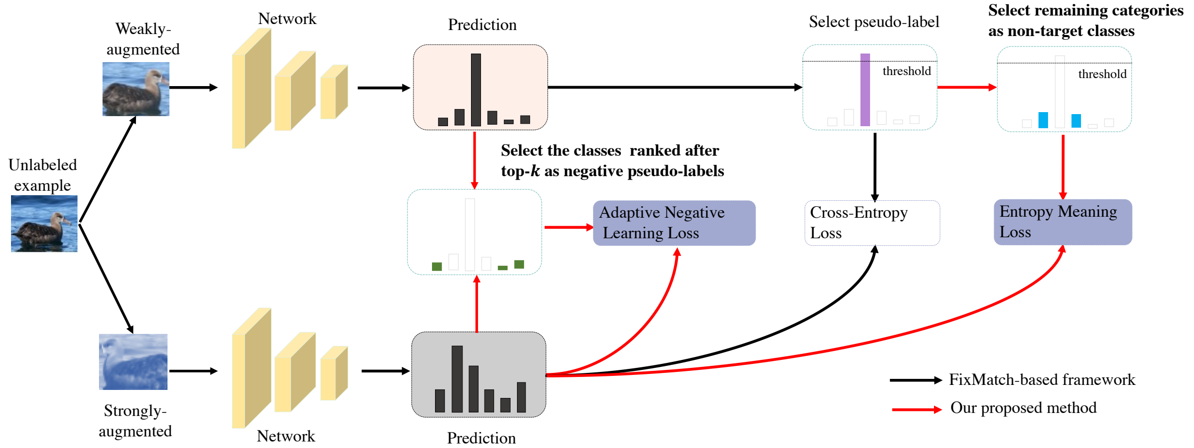


Figure 5. Overview of the proposed FullMatch. First, we allocate the negative pseudo-label (green bar) for all unlabeled data with the proposed Adaptive Negative Learning. Then, if the highest probability is above the predefined threshold (dotted line), we will assign the pseudo-label (purple bar) just like FixMatch, but we optimize further remaining non-target classes (blue bar) via the proposed Entropy Meaning Loss. The black line indicates the existing FixMatch-based methods, and the red line is our proposed method. (Best viewed in color).

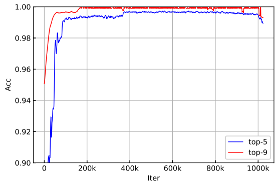


Figure 6. The top-5 and top-9 accuracy curves of FixMatch during training on CIFAR-10 with 40 label samples.

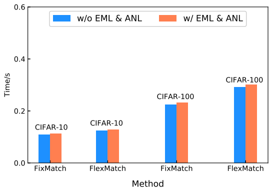


Figure 7. Average training time cost of one iteration on the same Geforce RTX 2080 Ti GPU.

cesses for evaluating the performance, nor any new hyperparameters. Unlike UPS [24] and NS^{3}L [4], ANL do not rely on the confidence values and allow to allocate negative pseudo-labels to examples with ambiguous prediction. For more analysis about ANL (e.g., training with limited labeled examples), please refer to the *Supplementary Material*, Section B.

3.4. FullMatch

By integrating the Entropy Meaning Loss (EML) and Adaptive Negative Learning (ANL) into FixMatch, we propose an advanced SSL algorithm named FullMatch. Since ANL can allocate negative pseudo-labels for all unlabeled data, it encourages us to take negative pseudo-labels as additional targets into account when calculating $y_c^{(i)}$ and $\mathcal{L}_{eml}$. This means the count of non-target class in examples with pseudo-label is $k-1$ instead of $C-1$.

As shown in Fig. 5, we first calculate k to assign negative pseudo-labels for *all* unlabeled examples and use Eq.(10) to train the model. This means the network can learn from the low-confidence examples instead of neglecting them directly. Then, we render pseudo-labels based on the prediction of weakly-augmented examples and use cross-entropy as loss function, similar to FixMatch. For examples with pseudo-label, we further view the remaining categories as non-target classes and utilize EML to train the corresponding class outputs in strongly-augmented predictions. Therefore, we can formulate the overall loss in FullMatch as a simple weighted sum of the FixMatch loss (i.e., supervised loss, unsupervised loss), ANL loss and EML:

$$\mathcal{L}_{sum} = \mathcal{L}_s + \mathcal{L}_{us} + \alpha \cdot \mathcal{L}_{anl} + \beta \cdot \mathcal{L}_{eml} \tag{11}$$

For simplicity, we set α and β to 1. $\mathcal{L}_s$ and $\mathcal{L}_{us}$ (Eq. (2)) are respectively the supervision loss for labeled samples and the consistency loss for unlabeled samples:

$$\mathcal{L}_s = \frac{1}{B_l} \sum_{i=1}^{B_l} H(y^{(i)}, p_m(y|\omega(x^{(i)}))) \tag{12}$$

where B_l is the batch size of labeled examples. See *Supplementary Material*, Section D.1 for the full algorithm of FullMatch.

4. Experiments

We evaluate the efficacy of the proposed FullMatch on several popular SSL datasets: CIFAR-10/100 [16], SVHN [22], STL-10 [5] and ImageNet [7], and perform extensive experiments across various amounts of labeled data. In addition, we conduct experiments based on the FlexMatch [38] algorithm to further show our method is orthogonal to FlexMatch, which is a strengthened version of

	CIFAR-10			CIFAR-100			SVHN		STL-10
Label Amount	40	250	4000	400	2500	10000	40	1000	1000
UDA [33]	89.38±3.75	94.84±0.06	95.71±0.07	53.61±1.59	72.27±0.21	77.51±0.23	94.88±4.27	**98.11**±0.01	93.36±0.17
RemixMatch [2]	90.12±1.03	93.7±0.05	95.16±0.01	57.25±1.05	73.97±0.35	**79.98**±0.27	75.96±9.13	94.84±0.31	93.26±0.14
Semco‡ [21]	92.13±0.22	94.88±0.27	96.20±0.08	55.89±1.18	68.07±0.01	75.55±0.12	-	-	92.51±0.29
Dash [34]	86.78±3.75	95.44±0.13	95.92±0.06	55.24±0.96	72.82±0.21	78.03±0.14	96.97±1.59	97.97±0.06	92.74±0.40
UPS [24]	94.74±0.29	94.89±0.08	95.75±0.05	58.93±1.66	72.86±0.24	78.03±0.23	-	-	93.98±0.28
AlphaMatch‡ [11]	91.35±3.38	95.03±0.29	-	61.26±3.13†	**74.98**±0.27†	-	97.03±0.26	-	90.36±0.75
CoMatch [18]	93.12±0.92	95.10±0.35	95.94±0.03	59.98±1.11	72.99±0.21	78.17±0.23	-	-	91.34±0.41
SimMatch†‡ [41]	94.40±1.37	95.16±0.39	96.04±0.01	62.19±2.21	74.93±0.32	79.42±0.11	-	-	-
CR [9]	94.31±0.9	94.96±0.3	95.84±0.13	50.77±0.79	72.42±0.37	78.97±0.23	96.33±1.84	97.61±0.06	93.04±0.42
NP-Match [30]	95.09±0.04	95.04±0.06	95.89±0.02	61.08±0.99	73.97±0.26	78.78±0.13	-	-	94.41±0.24
FixMatch [26]	92.53±0.28	95.14±0.05	95.79±0.08	57.45±1.76	71.97±0.16	77.8±0.12	96.19±1.18	**98.04**±0.03	93.75±0.33
FullMatch (ours)	**94.11**±1.01	**95.36**±0.12	**96.25**±0.08	**59.42**±1.40	**73.06**±0.40	**78.56**±0.10	**97.65**±0.10	98.01±0.03	**94.26**±0.09
FlexMatch [38]	95.03±0.06	95.02±0.09	95.81±0.01	60.06±1.62	73.51±0.2	78.1±0.15	96.08±1.24	97.37±0.06	94.23±0.18
$FullFlex$ (ours)	**95.56**±0.15	**95.61**±0.04	**96.28**±0.03	**62.60**±0.64	**74.60**±0.42	**79.26**±0.21	**97.48**±0.04	**97.58**±0.02	**94.50**±0.12

Table 1. **Top-1 accuracy (%) for CIFAR-10/100, SVHN and STL-10 datasets on 3 different folds.** $FullFlex$ indicates applying our method to FlexMatch. † indicates introducing an additional technique named DA (Distribution Alignment) [2]. ‡ represents the result comes from the original paper.

FixMatch by introducing $Curriculum\ Pseudo\ Labeling$ (CPL) to adjust the threshold dynamically. We also present the ablation study to better understand why our method is effective.

For fair comparisons, we keep the same hyperparameters as FixMatch and FlexMatch. Specifically, we employ a cosine learning rate decay schedule [19] and standard stochastic gradient descent (SGD) with a momentum of 0.9 as optimizer [27] across all amounts of labeled examples and datasets, the initial learning rate is 0.03 and the total iteration number is set to 2^{20}. We use RandAugment [6] as the strong augmentation in all experiments, and ResNet-50 [14] for ImageNet datasets and Wide ResNet [37,43] (e.g., WRN 28-2 and WRN 28-8) for other benchmarks. Our framework is implemented on TorchSSL [38]. See *Supplementary Material*, Section D.2 for details.

4.1. Main Results

We report the performance of FullMatch on the four popular SSL benchmarks: CIFAR-10/100, SVHN and STL-10, as shown in Table 1. We calculate the mean and variance of top-1 accuracy when training on 3 different "folds" of labeled data. The results of other algorithms are mainly from TorchSSL and NP-Match [30]. We use few new results for a fair comparison (e.g., the performance of FlexMatch on SVHN), which are better than their published version. The experiments show that FullMatch outperforms FixMatch under all amounts of labeled data and benchmarks, except the SVHN dataset with 1000 labels. Additionally, since our method can be integrated with any variants based on FixMatch, we also use the state-of-the-art method FlexMatch [38] as the baseline and called the integrated method $FullFlex$. The results demonstrate that our method can significantly boost the baseline models, including FixMatch and FlexMatch for almost all datasets, and meanwhile surpasses the latest methods, e.g., NP-Match [30], SimMatch [41] and DoubleMatch [29]. Our method has the following advantages:

1) FullMatch brings considerable improvements compared with FixMatch, especially when the amount of labeled data is extremely limited. We report the performance of different algorithms with only 4 labels per class, corresponding to the 40 labeled data of CIFAR-10, SVHN and 400 labeled data of CIFAR-100. The results indicate the average accuracy of FullMatch exceeds FixMatch by more than 1%, especially on CIFAR-100 (an increase of 2%).

2) FullMatch is efficient. Our method does not introduce any extra hyperparameters or extra forward propagation process. Fig. 7 compares the average training time cost of a single iteration with and without using our method on Geforce RTX 2080 Ti GPU. It is obvious that while enhancing the performance of existing algorithms, our method introduces negligible computational overhead.

3) The proposed FullMatch is orthogonal to existing popular methods. Namely, our method can further improve the performance of other FixMatch-based methods. For instance, we introduce CPL into FullMatch, named $FullFlex$. The extensive experiments show that FullFlex achieves state-of-the-art performance under almost all benchmarks.

4.2. Results on ImageNet

We also evaluate our method on ImageNet to confirm the effectiveness on a more realistic and larger dataset. Following TorchSSL settings, we select samples with 100 labels for each class, which is less than 8% of the total train set. Table 2 shows the performance of different algorithms after 2^{20} iterations, where the results of different methods come from TorchSSL and NP-Match [30] (it conducts all experiments based on TorchSSL settings). When all hyper-parameters are kept consistently with TorchSSL, the top-1 accuracy of FullMatch and FullFlex are both improved by more than 1%, which further confirms the effectiveness of our method on this complicated dataset. Furthermore, compared with the latest methods, our method still achieve a state-of-the-art performance.

	Top-1	Top-5
UPS [24]	57.31	79.77
NP-Match [30]	58.22	80.67
FixMatch [26]	56.34	78.20
FullMatch (ours)	**57.44** (+1.1)	**79.26** (+1.06)
FlexMatch [38]	58.15	80.52
FullFlex (ours)	**59.58** (+1.43)	**81.38** (+0.86)

Table 2. **Top-1 and Top-5 accuracy (%) on ImageNet.** In green are the values of performance improvement over the baselines.

4.3. Ablation Study

Since FullMatch is essentially a combination of two novel techniques and FixMatch, we present an ablation study to verify the effectiveness of different components.

Entropy Meaning Loss (EML). We conduct an ablation study on EML, as shown in Table 3. It can be seen that FixMatch can obtain obvious improvement when combining EML. We also conduct experiments with different implementations of EML, seeing row.3 vs row.4, we can find that the proposed EML is very effective, regardless of the form of the loss function.

Adaptive Negative Learning (ANL). We study the role of ANL for the model. If we only afford negative pseudo-labels to examples with pseudo-labels, there will only be a slight improvement (57.83 vs 57.68), see Table 3. It is believed that the slight gain mainly comes from alleviating the negative effect of incorrect pseudo-labels because negative pseudo-labels are always less noisy. Moreover, if we assign negative pseudo-labels to the samples that are without positive pseudo-labels, it can bring a considerable performance gain (0.91%). When we assign all data the negative pseudo-labels, the performance is further improved (1%), indicating that our model can extract discriminative knowledge from all unlabeled data. Finally, if we apply EML and ANL to FixMatch, the FullMatch exceeds the baseline with a large margin (1.64%), which shows that the proposed two components are useful and complementary.

	CE	BCE	w PL	w/o PL	Accuracy	Δ
FixMatch					57.68	-
EML	✓				58.35	+0.67
		✓			**58.47**	+0.79
ANL			✓		57.83	+0.15
				✓	58.59	+0.91
			✓	✓	**58.67**	+0.99
FullMatch		✓	✓	✓	**59.32**	+1.64

Table 3. **Ablation study of FullMatch on 400-label split from CIFAR-100.** CE and BCE represent the loss implementation of EML. "w PL" and "w/o PL" means applying ANL on examples with/without pseudo-label, respectively. Δ represents the performance improvement over the baseline.

Different α and β. Table 4 reports the accuracy (%) of different α and β. It reveals FullMatch is not sensitive to α and β. We take untuned weights (i.e. $\alpha, \beta = 1$) for all experiments to show the gains come entirely from our method.

α	0.5			1.0			2.0		
β	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0
acc	78.43	78.36	78.50	78.38	78.46	78.48	78.49	78.31	78.47

Table 4. **Ablation study on α and β.** All experiments are conducted on CIFAR-100 with 10000-label.

5. Conclusion

In this paper, we first analyze the unlabeled data wasting in the FixMatch-based methods, and then we are motivated to propose two novel techniques: Entropy Meaning Loss (EML) and Adaptive Negative Learning (ANL). The EML explicitly constrains the output distribution of non-target classes to produce more high-confidence predictions, thus selecting more examples with pseudo-label under the same threshold. The ANL introduces additional negative pseudo-labels to learn knowledge from low-confidence examples without pseudo-label. Note that ANL assesses the top-k performance dynamically to allocate negative pseudo-labels and does not introduce any extra hyper-parameters. FullMatch, the proposed method based upon FixMatch, achieves significant improvement in most scenarios while being extremely concise and efficient. In addition, we also integrate our method with FlexMatch, which achieves state-of-the-art performance on a variety of SSL benchmarks. Experimental results strongly confirm the effectiveness of our method. We believe this work will provide new insights to explore the low-confidence unlabeled data in SSL.

References

[1] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *NeurIPS*, 2014. 1

[2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring. In *ICLR*, 2020. 3, 7

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. MixMatch: a holistic approach to semi-supervised learning. In *NeurIPS*, 2019. 1, 3

[4] John Chen, Vatsal Shah, and Anastasios Kyrillidis. Negative sampling in semi-supervised learning. In *ICML*, 2020. 3, 6

[5] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011. 2, 5, 6

[6] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, 2020. 1, 7

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 6

[8] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1

[9] Lee Doyup, Sungwoong Kim, Ildoo Kim, Yeongjae Cheon, Minsu Cho, and Wook-Shin Han. Contrastive Regularization for Semi-Supervised Learning. In *CVPR*, 2022. 7

[10] Zhengyang Feng, Qianyu Zhou, Qiqi Gu, Xin Tan, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Dmt: Dynamic mutual training for semi-supervised learning. *PR*, 2022. 1

[11] Chengyue Gong, Dilin Wang, and Qiang Liu. Alphamatch: Improving consistency for semi-supervised learning with alpha-divergence. In *CVPR*, 2021. 7

[12] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, 2005. 1, 2, 3

[13] Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, Adam Prügel-Bennett, and Jonathon Hare. Fmix: Enhancing mixed sample data augmentation. *arXiv preprint arXiv:2002.12047*, 2020. 3

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[15] Byoungjip Kim, Jinho Choo, Yeong-Dae Kwon, Seongho Joe, Seungjai Min, and Youngjune Gwon. Selfmatch: Combining contrastive self-supervision and consistency for semi-supervised learning. *arXiv preprint arXiv:2101.06480*, 2021. 3

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009. 1, 2, 6

[17] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshops*, 2013. 3

[18] Junnan Li, Caiming Xiong, and Steven CH Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *ICCV*, 2021. 7

[19] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 7

[20] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE TPAMI*, 2018. 3

[21] Islam Nassar, Samitha Herath, Ehsan Abbasnejad, Wray Buntine, and Gholamreza Haffari. All labels are not created equal: Enhancing semi-supervision via label grouping and co-training. In *CVPR*, 2021. 1, 7

[22] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop*, 2011. 2, 6

[23] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *NeurIPS*, 2015. 1

[24] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In Defense of Pseudo-Labeling: An Uncertainty-Aware Pseudo-label Selection Framework for Semi-Supervised Learning. In *ICLR*, 2021. 3, 6, 7, 8

[25] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Mutual exclusivity loss for semi-supervised deep learning. In *ICIP*, 2016. 1

[26] 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 *NeurIPS*, 2020. 1, 3, 7, 8

[27] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013. 7

[28] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *JMLR*, 2008. 5

[29] Erik Wallin, Lennart Svensson, Fredrik Kahl, and Lars Hammarstrand. DoubleMatch: Improving Semi-Supervised Learning with Self-Supervision. *arXiv preprint arXiv:2205.05575*, 2022. 7

[30] Jianfeng Wang, Thomas Lukasiewicz, Daniela Massiceti, Xiaolin Hu, Vladimir Pavlovic, and Alexandros Neophytou. NP-Match: When Neural Processes meet Semi-Supervised Learning. In *ICML*, 2022. 7, 8

[31] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Zhen Wu, and Jindong Wang. FreeMatch: Self-adaptive Thresholding for Semi-supervised Learning. *arXiv preprint arXiv:2205.07246*, 2022. 1

[32] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-Supervised Semantic Segmentation Using Unreliable Pseudo-Labels. In *CVPR*, 2022. 3

[33] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, 2020. 3, 5, 7

[34] Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *ICML*, 2021. 2, 7

[35] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. St++: Make self-training work better for semi-supervised semantic segmentation. In *CVPR*, 2022. 3

[36] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 3

[37] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *BMVC*, 2016. 7

[38] 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 *NeurIPS*, 2021. 1, 2, 6, 7, 8

[39] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 3

[40] Wenqiao Zhang, Lei Zhu, James Hallinan, Shengyu Zhang, Andrew Makmur, Qingpeng Cai, and Beng Chin Ooi. Boostmis: Boosting medical image semi-supervised learning with adaptive pseudo labeling and informative active annotation. In *CVPR*, 2022. 3

[41] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. SimMatch: Semi-supervised Learning with Similarity Matching. In *CVPR*, 2022. 7

[42] Hongyu Zhou, Zheng Ge, Songtao Liu, Weixin Mao, Zeming Li, Haiyan Yu, and Jian Sun. Dense Teacher: Dense Pseudo-Labels for Semi-supervised Object Detection. In *ECCV*, 2022. 3

[43] Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Time-consistent self-supervision for semi-supervised learning. In *ICML*, 2020. 7

[44] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 2009. 1