Towards Discovering the Effectiveness of Moderately Confident Samples for Semi-Supervised Learning

Hui Tang and Kui Jia[*]
South China University of Technology
eehuitang@mail.scut.edu.cn, kuijia@scut.edu.cn

Abstract

Semi-supervised learning (SSL) has been studied for a long time to solve vision tasks in data-efficient application scenarios. SSL aims to learn a good classification model using a few labeled data together with large-scale unlabeled data. Recent advances achieve the goal by combining multiple SSL techniques, e.g., self-training and consistency regularization. From unlabeled samples, they usually adopt a confidence filter (CF) to select reliable ones with high prediction confidence. In this work, we study whether the moderately confident samples are useless and how to select the useful ones to improve model optimization. To answer these problems, we propose a novel Taylor expansion inspired filtration (TEIF) framework, which admits the samples of moderate confidence with similar feature or gradient to the respective one averaged over the labeled and highly confident unlabeled data. It can produce a stable and new information induced network update, leading to better generalization. Two novel filters are derived from this framework and can be naturally explained in two perspectives. One is gradient synchronization filter (GSF), which strengthens the optimization dynamic of fully-supervised learning; it selects the samples whose gradients are similar to class-wise majority gradients. The other is prototype proximity filter (PPF), which involves more prototypical samples in training to learn better semantic representations; it selects the samples near class-wise prototypes. They can be integrated into SSL methods with CF. We use the state-of-the-art FixMatch as the baseline. Experiments on popular SSL benchmarks show that we achieve the new state of the art.

1. Introduction

Deep learning has achieved great success in computer vision tasks, with image classification [9] as one of the prominent examples. The success can be mainly attributed to large-scale labeled data. However, annotating enormous training data for all tasks of interest is practically infeasible. To reduce the labeling cost, the topic of semi-supervised learning (SSL) has been proposed and there are already a large number of research works in SSL [40]. The goal of SSL is to achieve good model generalization using limited labeled data and many unlabeled data that are assumed to follow the same distribution. In this work, we investigate the classical topic, aiming to push the limit of SSL.

Recent SSL methods rely on deep models [47] to learn feature representations that can facilitate the subsequent classification. A common strategy is self-training [12, 17, 31], where the pseudo labels are iteratively generated and then used as supervision to guide the model training on unlabeled samples. Another popular paradigm is consistency regularization [29, 33], which constrains the model to produce consistent predictions for two different duplicates of the same unlabeled sample. The difference between the two duplicates can be made by random data augmentation [33] or perturbation of network parameters [29]. After, a lot of extensions have been proposed [2, 16, 21, 24, 38, 49]. The two techniques are effective but not optimal on their own, as suggested in [26]. The current best practice in SSL is technique combination, *e.g.*, combining self-training and consistency regularization [4, 5, 18, 37, 44, 48, 48]. The cluster and smoothness assumptions are enforced simultaneously. The former [7] assumes that the decision boundaries are located in low-density regions and the latter [40] assumes that the adjacent samples have similar labels. Such a combination can progressively improve the model performance, as verified in the theoretical work [42]. Note that the two techniques would be uninformative if the model predicts a uniform distribution over classes for unlabeled samples. To address it, existing methods adopt confidence filtering [10, 18, 37], which abandons the samples whose prediction confidences (ranged in $[0, 1]$) are lower than a predefined high threshold (*e.g.*, 0.95 [37]). It is reasonable that the least confident samples are extremely unreliable. But are all the moderately confident samples useless, *e.g.*, ranged in $(0, 75, 0.95)$? Is there any way to pick out the useful ones to enhance the optimization power applied to the model?

[*] Corresponding author.

In this work, we solve the questions by introducing a novel framework of Taylor expansion inspired filtration (TEIF). The Tayor formula of the cross-entropy loss function w.r.t. the feature of one sample with true or pseudo label mainly includes terms of the multiplication of gradient and feature of finite orders. To make the change of loss consistent in the neighborhood of the feature, this framework selects the samples of moderate confidence, whose feature or gradient is similar to the respective one averaged over the labeled and highly confident unlabeled data, which are the most reliable. Hence, the final network update is still close to the one determined by the most reliable samples and further incorporates the new information contained in the selected samples of moderate confidence, such that the model optimization could be steady and improved.

From this framework, two novel filters are derived to select the helpful samples from the moderately confident unlabeled data. The selected samples together with the highly confident ones are then used to train the classification model. The first filter based on gradients assumes that one moderately confident sample is useful if it follows the optimization dynamic of fully-supervised learning [1, 50]. The previous research [1] has verified that deep neuron networks learn simple patterns first that are better fitted by easy examples. The fact implies that pattern learning could be improved if such an optimization dynamic is strengthened. On the other hand, the recent approach [14] relies on the sample feature gradients to characterize the optimization dynamic, *i.e.*, constraining the local and global alignments to be consistent. By nature of the gradient-based filter, we can thus approximate the optimization dynamic by class-wise majority gradients, which are computed on features of the labeled and highly confident unlabeled samples, *i.e.*, easy examples. From those moderately confident samples, we select the ones that have similar feature gradients to the corresponding majority gradient. We thus term this method as gradient synchronization filter (GSF). The second filter based on features assumes that one moderately confident sample is useful if it has a certain level of prototypicality [36]. Specifically, the class-wise prototypical representations, which best characterize specific semantic classes (as suggested in [36]), are computed by taking an average over sample features of each class. The samples near prototypes are selected from those moderately confident unlabeled data. We thus term this method as prototype proximity filter (PPF). Our methods can be naturally integrated into SSL frameworks with confidence filter. To challenge the current state of the art, we choose FixMatch [37] as the baseline. Experiments on commonly used SSL benchmarks show that our methods outperform FixMatch. The empirical study also answers the previously raised questions: some moderately confident samples are useful and there are ways to pick them out. Our main contributions are summarized below. **(1)** We introduce new and significant questions for SSL and provide preliminary answers for them, *i.e.*, whether all the moderately confident samples are useless and how to select the useful ones from them. **(2)** To solve the questions, we propose a novel Taylor expansion inspired filtration (TEIF) framework, which relies on the Taylor expansion of the loss function to inspire the key measurement index of sample filtration, *i.e.*, gradient and feature of finite orders. The principle wherein is to make the network update stable and improved after adding the selected moderately confident samples. **(3)** Two novel filters are derived from this framework and make sense from different perspectives of optimization dynamic and prototype proximity, leading to gradient synchronization filter (GSF) and prototype proximity filter (PPF) respectively. The moderately confident samples selected by GSF or PPF are then involved in model training, which helps learn decision boundaries closer to the ground-truth ones.

2. Related Works

We briefly review the recent deep semi-supervised learning (SSL) methods, focusing on the components of FixMatch on which we base our work. A comprehensive survey is provided in [40] and pseudo label generation via learning to learn is also a hot topic [13, 19, 20, 28, 30, 41].

Self-Training. The work [12] aims to enforce the cluster assumption that the decision boundaries should be located in low-density regions; technically, it minimizes the information entropy of label distributions predicted by the model for the unlabeled data. Lee [17] picks up the class of maximum predicted probability as a hard target for each unlabeled sample; then, the pseudo labels are used to fine-tune the model. To improve, UPS [31] selects more accurate pseudo labels by considering both uncertainty and confidence of network predictions. Many other applications also adopt self-training, *e.g.*, natural language processing [22], object detection [32], and domain adaptation [6].

Consistency Regularization. This technique aims to implement the smoothness assumption by enforcing consistency between the model's outputs of the original and perturbed versions of the same input [42]. It is first introduced in [3], which regularizes the behavior of a pseudo-ensemble to be robust to the noise process generating it. Rasmus *et al.* [29] and Sajjadi *et al.* [33] develop it by perturbing network parameters, applying stochastic transformations on images, or utilizing the randomness involved in some components of the network, *e.g.*, dropout and random max-pooling. In [16], the model prediction for an unlabeled sample is matched to its temporal ensembling counterpart. Differently, the work [38] turns the match object to the prediction of mean teacher, *i.e.*, the moving average of previous models. In [49], the worst-case perturbations are imposed on network weights and structures, which can be derived by solving respective optimization problems with spectral methods. PAWS [2] enforces consistent non-parametrical predictions between the anchor and positive views.

Technique Combination. Individual SSL techniques on their own are unable to achieve higher levels of performance, as revealed in recent researches [26, 44, 48]. Therefore, a better strategy is to combine these simple but effective techniques. In [4], multiple strongly-augmented versions of an unlabeled sample are involved in mixup training, where they use the sharpened model prediction of the sample's weakly-augmented version as the pseudo label; a distribution alignment component is introduced to make the predicted label distribution close to the ground truth one. Recently, a simplified version [37] trains the model with the strongly-augmented version of any unlabeled sample and uses as supervision the class of maximum prediction probability of its weakly-augmented version, where only the high-confidence samples are selected. CoMatch [18] enforces the relation consistency between pseudo label and feature embedding graphs. Differently, we take the first step towards discovering the effectiveness of moderately confident samples by proposing a novel framework of Taylor expansion inspired filtration, which derives the sample filters based on optimization dynamic or prototype proximity.

3. Method

The task at hand is assumed to distinguish between K classes, given a small batch of B labeled examples $\mathcal{X}^l = \{(\mathbf{x}_i^l, y_i^l)\}_{i=1}^B$ and a large batch of μB unlabeled examples $\mathcal{X}^u = \{\mathbf{x}_i^u\}_{i=1}^{\mu B}$, where $y \in \{1, 2, \dots, K\}$ and $\mu \in \mathbb{N}_+$. The objective of semi-supervised learning (SSL) is to learn a feature extractor $E(\cdot)$ that lifts any input image $\mathbf{x}$ to the feature space, *i.e.*, $\mathbf{z} = E(\mathbf{x})$, and a classifier $F(\cdot)$ that maps the learned feature to a probability vector $\mathbf{p} = F(\mathbf{z})$, where each element p_k ($k \in \{1, 2, \dots, K\}$) represents the possibility of assigning the sample to a class k. We also write $\mathbf{p}$ (*resp.* p_k) as $p(\mathbf{x})$ (*resp.* $p_k(\mathbf{x})$) when the contexts require. The classification model $F(E(\cdot))$ is trained with $\mathcal{X}^l$ and $\mathcal{X}^u$, aiming to generalize well on unseen test samples. Following [37], we apply to unlabeled training samples the two types of strategies, *i.e.*, the weak and strong augmentations, which are denoted by $\alpha(\cdot)$ and $\mathcal{A}(\cdot)$ respectively.

3.1. Preliminaries

We start by introducing the popular SSL techniques, *i.e.*, self-training and consistency regularization. Self-training uses the model prediction itself as the artificial label to guide the model training on unlabeled data [23, 34]. The idea is instantiated by two representative approaches, *i.e.*, pseudo-labeling [17] and entropy minimization [12], which differ in the way of label generation. The former takes the class of highest score as the label (of one-hot form). Let $\hat{y}^u = \arg\max_k p_k^u$ be the pseudo label for an unlabeled example $\mathbf{x}^u$. The objective of self-training in [17] is

$$\mathcal{J}_{ST} = -\frac{1}{\mu B} \sum_{i=1}^{\mu B} \log p_{\hat{y}_i^u}(\mathbf{x}_i^u), \tag{1}$$

which is a standard cross-entropy loss. Here, $p_{\hat{y}_i^u}(\mathbf{x}_i^u)$ indicates the element of the probability vector at the predicted class $\hat{y}_i^u$ for the example $\mathbf{x}_i^u$. Optimizing Eq. (1) reduces the prediction uncertainty for unlabeled data, similar to [12].

Consistency regularization is based on the smoothness assumption [40]. It is typically implemented by matching the model predictions of two different augmented versions of the same unlabeled example. The following objective of consistency regularization is used in [33]

$$\mathcal{J}_{CR} = \sum_{i=1}^{\mu B} ||p(\alpha(\mathbf{x}_i^u)) - p(\alpha(\mathbf{x}_i^u))||_2^2, \tag{2}$$

where we note that $\alpha(\cdot)$ is a stochastic function and thus produces varying data augmentations. Minimizing $\mathcal{J}_{CR}$ encourages the smoothness and stability of model prediction over the entire data manifold [40]. Other variants of consistency regularization include matching predictions of an example and its virtual adversarial counterpart [24], or matching the current and temporal ensemble predictions [16], to name a few. Recent works [4, 24, 37, 44] also use cross-entropy to measure the prediction divergence.

The state-of-the-art method of FixMatch [37] combines self-training and consistency regularization as follows. It uses the pseudo label of a weakly-augmented version $\alpha(\mathbf{x}^u)$, *i.e.*, $\hat{y}^u = \arg\max_k p_k(\alpha(\mathbf{x}^u))$, as the supervision of a strongly-augmented counterpart $\mathcal{A}(\mathbf{x}^u)$. Collectively, the model is trained with the weakly-augmented labeled data $\{(\alpha(\mathbf{x}_i^l), y_i^l)\}_{i=1}^B$ and strongly-augmented unlabeled data $\{(\mathcal{A}(\mathbf{x}_i^u), \hat{y}_i^u)\}_{i=1}^{\mu B}$. Considering that the two techniques would be uninformative if model predictions follow the uniform distribution, FixMatch adopts the typical confidence filter to discard the unlabeled examples whose confidence (*i.e.*, maximum class probability) is lower than a pre-defined high threshold τ. Thus, the selected set $\{(\mathcal{A}(\mathbf{x}_i^u), \hat{y}_i^u) | p_{\hat{y}_i^u}(\alpha(\mathbf{x}_i^u)) \geq \tau\}_{i=1}^{\mu B}$ is in fact involved in training. The overall objective of FixMatch is composed of a supervised loss $\mathcal{J}_{sup}$ and an unsupervised loss $\mathcal{J}_{uns}$, as

$$\min_{E,F} \mathcal{J}_{sup} + \lambda_u \mathcal{J}_{uns}, \tag{3}$$

where λ_u is a trade-off hyperparameter and

$$\mathcal{J}_{sup} = -\frac{1}{B} \sum_{i=1}^{B} \log p_{y_i^l}(\alpha(\mathbf{x}_i^l)),$$

$$\mathcal{J}_{uns} = -\frac{1}{\mu B} \sum_{i=1}^{\mu B} \mathrm{I}[p_{\hat{y}_i^u}(\alpha(\mathbf{x}_i^u)) \geq \tau] \log p_{\hat{y}_i^u}(\mathcal{A}(\mathbf{x}_i^u)).$$

Here, I[$\cdot$] is an indicator function. In FixMatch, the threshold $\tau \in [0, 1)$ of confidence filter is set to a high value, *e.g.*, 0.95, since the pseudo labels of the most confident samples are assumed to be mostly correct [51]. Put it another way, FixMatch abandons all the samples with $p_{\hat{y}^u}(\alpha(\mathbf{x}^u)) < \tau$ at each training iteration. The least confident samples are indeed too unreliable to use. However, are all the samples with moderate confidence useless, *e.g.*, $0.75 < p_{\hat{y}^u}(\alpha(\mathbf{x}^u)) < 0.95$? We in this work struggle to answer the question by introducing a novel Taylor expansion inspired filtration (TEIF) framework, as detailed shortly. The filters derived from this framework can select the helpful ones from the moderately confident samples to improve optimization over the model, as empirically verified in Sec. 4.

3.2. Taylor Expansion Inspired Filtration

We first define the Taylor formula by borrowing from the established knowledge of function approximation in the advanced-math community [11, 46], as follows

$$\begin{aligned}\mathcal{J}(\mathbf{z}') = &\mathcal{J}(\mathbf{z}) + [\nabla\mathcal{J}(\mathbf{z})]^T(\mathbf{z}' - \mathbf{z}) + \\ &\frac{1}{2}(\mathbf{z}' - \mathbf{z})^T[H\mathcal{J}(\mathbf{z})](\mathbf{z}' - \mathbf{z}) + R_2,\end{aligned} \tag{4}$$

where $\mathcal{J}(\mathbf{z})$ is the cross-entropy loss function at the current feature $\mathbf{z}$ of a sample $\mathbf{x}$ with true or pseudo label, $\mathbf{z}'$ is in the neighborhood of $\mathbf{z}$, $\nabla\mathcal{J}(\mathbf{z})$ is the first-order gradient, $H\mathcal{J}(\mathbf{z})$ is the Hessian matrix, and R_2 is the infinitesimal remainder of the second order expansion. In Eq. (4), the loss difference between $\mathbf{z}'$ and $\mathbf{z}$, *i.e.*, $\mathcal{J}(\mathbf{z}') - \mathcal{J}(\mathbf{z})$, mainly comprises two terms that are the multiplication of gradient and feature of finite orders.

Recall that existing SSL methods [37, 44] compute the loss by using the labeled and highly confident unlabeled samples only, which are commonly believed to be the most reliable ones. On this basis, the selected samples of moderate confidence should satisfy the condition that the change direction of the loss in the neighborhood of $\mathbf{z}$, *i.e.*, rise or fall, is similar to the one averaged over the most reliable samples. Hence, the final network update does not deviate too much from the one determined by the labeled and highly confident samples; meanwhile, the new knowledge is introduced by the selected moderately confident samples, and thus the volumes of information contained in the update are increased. Therefore, the model optimization could be improved, leading to better generalization.

To this end, we do the sample filtration for the moderately confident unlabeled data by selecting the samples whose gradient or feature is similar to the respective one averaged over the labeled and highly confident samples, termed the Taylor expansion inspired filtration (TEIF) framework. Due to limited computation overhead, we only consider the first-order quantities in this work and will study the higher-order information in future work by designing algorithms to reduce the computational complexity.

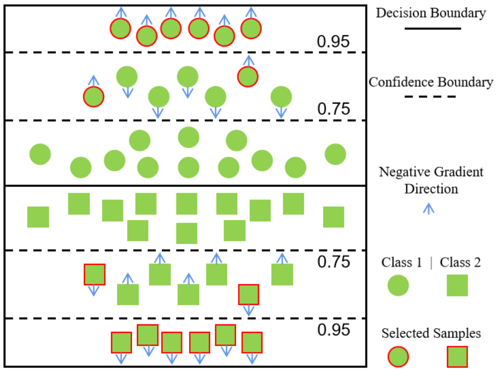


Figure 1. An illustrative example of sample selection in GSF.

We note that the gradients are closely related to the optimization dynamic [1, 14, 50] and the features characterize a certain level of semantics of the specific class [27, 36, 37], *i.e.*, semantic prototypicality; we accordingly depict the two filters derived from our TEIF framework in perspectives of optimization dynamic and prototype proximity later.

Remarks. It is obvious that additional correct predictions would exist for unlabeled examples whose confidence of pseudo labels is less than τ; self-training with these additional examples would improve performance of the learned model. The theoretical result in [42] also tells that given enough correctly pseudo-labeled examples, self-training with consistency regularization is guaranteed to achieve high accuracy on the true labels for the unlabeled data. The following introduced filters aim to identify more correctly pseudo-labeled examples from those with moderately confident predictions, as empirically demonstrated in Fig. 4c.

3.3. Gradient Synchronization Filter

The optimization dynamic of fully-supervised learning is that deep models learn simple patterns first, which are better fitted by easy examples, as suggested in [1]. The optimization over deep models is content-aware, *i.e.*, multiple training samples that share patterns are utilized first. Existing methods characterize the optimization dynamic by gradients of the loss function w.r.t. the network parameters [50] or sample features [14]. The latter scheme naturally corresponds to the gradient-based filter derived from the proposed TEIF framework and has low computation and memory overheads. In other words, from the moderately confident samples, the filter selects the ones that follow the optimization dynamic of fully-supervised learning, which can facilitate pattern learning. Specifically, we first compute the majority gradient by taking the sum over normalized feature gradients of labeled samples and highly confident unlabeled ones that are assumed to be mostly correct. To capture the optimization dynamic exactly, we define the majority gradient at the finer class level, as follows

$$\mathbf{g}_k^m = \sum_{i\in\mathcal{D}_k^l} \frac{\mathbf{g}_i^l}{||\mathbf{g}_i^l||_2} + \sum_{i\in\hat{\mathcal{D}}_k^u} \mathrm{I}[p_{\hat{y}_i^u}(\alpha(\mathbf{x}_i^u)) \geq \tau] \frac{\mathbf{g}_i^u}{||\mathbf{g}_i^u||_2}, \tag{5}$$

where the instance indexes of the k-th class in labeled and unlabeled batches are respetively $\mathcal{D}_k^l = \{i|(\mathbf{x}_i^l, y_i^l) \in \mathcal{X}^l \wedge y_i^l = k\}$ and $\hat{\mathcal{D}}_k^u = \{i|\mathbf{x}_i^u \in \mathcal{X}^u \wedge \hat{y}_i^u = k\}$, $k \in \{1, 2, \dots, K\}$, and the feature gradients of labeled and unlabeled samples are respectively

$$\mathbf{g}_i^l = \frac{\partial - \log p_{y_i^l}(\alpha(\mathbf{x}_i^l))}{\partial \mathbf{z}_i^l}, \mathbf{g}_i^u = \frac{\partial - \log p_{\hat{y}_i^u}(\alpha(\mathbf{x}_i^u))}{\partial \mathbf{z}_i^u}.$$

We note that in each training iteration, samples in the labeled and unlabeled batches are taken at random (since we use stochastic gradient descent (SGD)), which may provide incomplete knowledge on the optimization dynamic, leading to biased sample selection. To avoid it, we apply the exponential moving average [45] to each majority gradient over the past batches in the same training epoch, as follows

$$\mathbf{g}_k^m \leftarrow \eta_m \mathbf{g}_k^m + (1 - \eta_m)\mathbf{g}_k^m[t], \tag{6}$$

where $t \in \{1, 2, \dots, T\}$, T is the number of iterations in one training epoch, $\mathbf{g}_k^m[t]$ is the majority gradient at the current iteration t, $\mathbf{g}_k^m = \mathbf{g}_k^m[t]$ when $t = 1$, and $\eta_m \in [0, 1]$ is the moving average coefficient. Then, we employ the cosine similarity to measure the degree of synchronization between the feature gradient of one moderately confident unlabeled sample $\mathbf{x}_i^u$ and its corresponding majority gradient $\mathbf{g}_{\hat{y}_i^u}^m$ as

$$s_i^u = 0.5(1 + \frac{\mathbf{g}_i^u \cdot \mathbf{g}_{\hat{y}_i^u}^m}{||\mathbf{g}_i^u||_2 \, ||\mathbf{g}_{\hat{y}_i^u}^m||_2}), \tag{7}$$

which is scaled to be in $[0, 1]$. From the samples whose confidence $p_{\hat{y}^u}^u$ is in $(0.75, 0.95)$, we select the ones with the gradient synchronization degree $s^u \geq \tau_s$. Here, τ_s is a threshold hyperparameter for our proposed gradient synchronization filter (GSF). The moderately confident samples selected by GSF together with the highly confident ones are used to compute the unsupervised loss $\mathcal{J}_{uns}$ in each training iteration. An illustration is given in Fig. 1.

3.4. Prototype Proximity Filter

The barely supervised study from FixMatch [37] has shown that the samples in one class differ in their prototypicality, *i.e.*, to what extent they can characterize semantics of the specific class. The most representative samples are most suitable for data-efficient learning in that they yield a significant performance gain under the same low-label protocol. Inspired by it, the feature-based filter derived from the proposed TEIF framework is in fact to select the most prototypical examples that are close to the corresponding prototypes, from the moderately confident unlabeled data. We follow [27, 36] to define each class-wise prototype, *i.e.*, prototypical representation, by the mean over sample features of one particular semantic class, which is written as

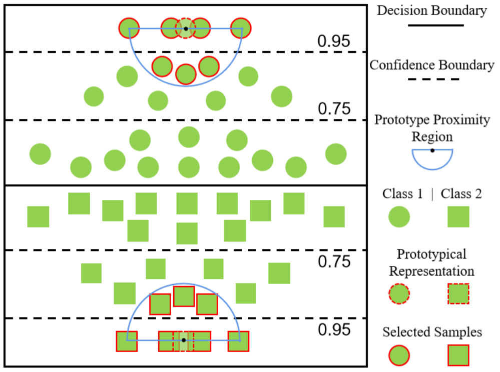


Figure 2. An illustrative example of sample selection in PPF.

$$\mathbf{z}_k^p = \frac{1}{n_k}\left(\sum_{i\in\mathcal{D}_k^l} \mathbf{z}_i^l + \sum_{i\in\hat{\mathcal{D}}_k^u} \mathrm{I}[p_{\hat{y}_i^u}(\alpha(\mathbf{x}_i^u)) \geq \tau]\mathbf{z}_i^u\right), \tag{8}$$

where n_k is the total number of instances of the k-th class in labeled and highly confident unlabeled sets. The categorical information contained in $\mathbf{z}_k^p$ is usually insufficient to represent the semantic meaning of the k-th class since it only considers training data in the current iteration. For instance, it is possible that some classes are missing in the current labeled and unlabeled batches since the two batches are randomly sampled. To address the issue, we also follow [45] to conduct the exponential moving average of each class-wise prototype over all previous iterations, as follows

$$\mathbf{z}_k^p \leftarrow \eta_p \mathbf{z}_k^p + (1 - \eta_p)\mathbf{z}_k^p[t], \tag{9}$$

where $\mathbf{z}_k^p = \mathbf{z}_k^p[t]$ when $t = 1$ and $\eta_p \in [0, 1]$ is the moving average coefficient. Then, we use a variant of Student t-distribution [39, 43] to measure the extent of vicinity between the feature of one moderately confident unlabeled sample $\mathbf{x}_i^u$ and its correpsonding prototype $\mathbf{z}_{\hat{y}_i^u}^p$ as

$$v_i^u = \frac{1}{1 + ||\mathbf{z}_i^u - \mathbf{z}_{\hat{y}_i^u}^p||^2}, \tag{10}$$

which is ranged from 0 to 1. From the moderately confident samples with $p_{\hat{y}^u}^u \in (0.75, 0.95)$, we select the ones whose prototype vicinity extent $v_i^u \geq \tau_v$. Here, τ_v is the threshold of our proposed prototype proximity filter (PPF). During feature learning, the extent of instance-to-center vicinity could be in an arbitrary scale since the Euclidean distance is in a range of $[0, +\infty)$, which is the base of Eq. (10). To solve it, we propose to use an adaptive threshold, which is

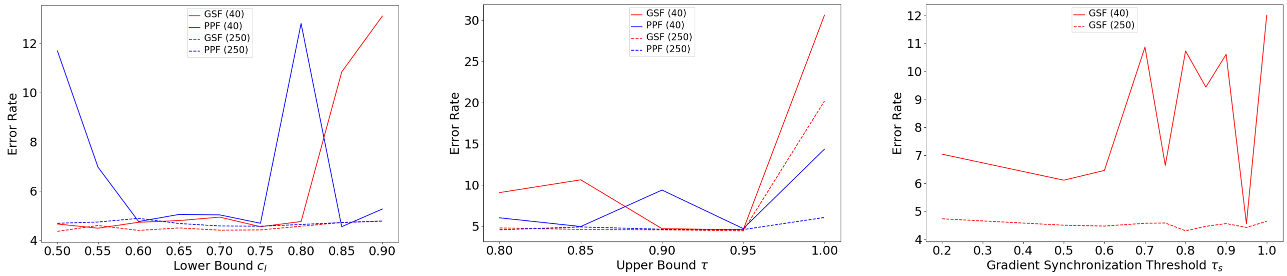


Figure 3. Error rates w.r.t. the lower (**left**) and upper (**center**) bounds of moderate confidence, and the gradient synchronization threshold of our proposed GSF (**right**) on 40- and 250-label settings.

the moving average of minimal vicinity extent over labeled and highly confident unlabeled samples, as follows

$$\tau_v[t] = \min\left(\min_{i\in\mathcal{D}^l} v_i^l, \min_{i\in\hat{\mathcal{D}}^u, p_{\hat{y}_i^u}(\alpha(\mathbf{x}_i^u))\geq\tau} v_i^u\right), \quad (11)$$

$$\tau_v \leftarrow \eta_p\tau_v + (1-\eta_p)\tau_v[t], \quad (12)$$

where $\mathcal{D}^l$ and $\hat{\mathcal{D}}^u$ collect the indexes of instances in the current labeled and unlabeled batches respectively. The unsupervised loss $\mathcal{J}_{uns}$ in Eq. (3) is then computed using the moderately confident samples selected by PPF and the highly confident ones. An illustration is given in Fig. 2.

4. Experiments

We choose the state-of-the-art FixMatch [37] as the baseline and examine our proposed gradient synchronization filter (GSF) and prototype proximity filter (PPF) on three commonly used SSL benchmarks. Specifically, we conduct experiments with various number of labeled samples on the datasets of CIFAR-10 [15], CIFAR-100 [15], and SVHN [25]. CIFAR-10 has 10 classes, $50,000$ training images, and $10,000$ test ones; CIFAR-100 has 100 classes, $50,000$ training images, and $10,000$ test ones, which is very challenging due to much more classes; SVHN has 10 classes, $73,257$ training images, and $26,032$ test ones.

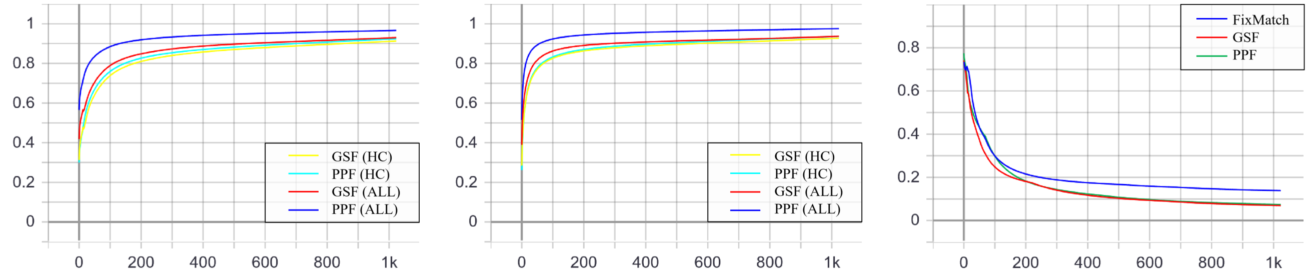
We follow the evaluation protocols of FixMatch and use the training set with a few labels for training and the test set for inference. We use a WRN-28-2 [47] as the backbone except for CIFAR-100 that uses a wider WRN-28-8, where the last FC layer is the instantiation of $F(\cdot)$. We set the confidence threshold τ as 0.95, which is the higher bound of the moderate confidence; we denote its lower bound by c_l and set c_l as 0.75. We set τ_s as 0.95. τ_v is dynamically updated by Eqs. (11) and (12). We empirically set η_m and η_p as 0.7. The hyperparameters λ_u, μ, and B are set as 1, 7, and 64 respectively. The weak augmentation $\alpha(\cdot)$ is implemented by random flipping and cropping; the strong one $\mathcal{A}(\cdot)$ performs additional image processing by RandAugment [8]. Other training details are the same as [37].

4.1. Ablation Study

We by convention conduct the ablation study on a single 250-label setting from CIFAR-10. For a more thorough examination, we also do the study on a single 40-label setting.

Moderate Confidence Bounding. To examine the range of moderate confidence used in our methods, we provide the classification accuracy as **(1)** changing its lower bound c_l in $[0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9]$, with its upper bound τ fixed as 0.95, and **(2)** altering τ in $[0.8, 0.85, 0.9, 0.95, 1]$ with c_l fixed as 0.75. The results of the two cases are shown in Fig. 3. It is observed that the performance is almost always benign and stable in the respective small-value ranges of the lower and upper bounds. A possible reason is that the increase in the number of selected unlabeled samples (*i.e.*, more samples are selected at smaller values) could compensate the negative effects caused by including low-quality samples in training. As the lower and upper bounds increase, the performance stays almost the same on the 250-label setting whereas it does not on the 40-label setting, revealing that the sensitivity of one SSL method to the hyperparameters is increased with fewer labels per class due to less reliable pattern learning. Note that when the upper bound has a value of 1, all unlabeled samples for training are selected by our GSF or PPF, and thus all highly confident samples may be discarded. In this case, the error rates go up, suggesting that the use of highly confident samples is indispensable for learning a good model. Our methods achieve minimal error rates when the lower and upper bounds are 0.75 and 0.95 respectively.

Gradient Synchronization Thresholding. To inspect the influence of the gradient synchronization threshold τ_s (cf. Sec. 3.3), we perform the sensitivity analysis by varying τ_s in $[0.2, 0.5, 0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1]$ with the moderate confidence ranged in $[0.75, 0.95]$. The results are shown in Fig. 3. We can observe that with 25 labels per class, the error rate fluctuates very slightly with the increase of τ_s, indicating that more highly confident samples (on account of more labels) can build a better foundation of model optimization for pattern learning. Hence, the selected low-quality samples have less adverse impact on the



(a) CIFAR-10 with 40 labels

(b) CIFAR-10 with 250 labels

(c) Mislabeled ratio

Figure 4. (**a**)-(**b**): Ratios of all selected samples (denoted by "ALL") and highly confident ones (denoted by "HC") in the unlabeled batch. The margin between a pair of ALL and HC is the ratio of selected samples of moderate confidence. (**c**): Ratios of mislabeled samples in the selected pseudo-labeled set on the 40-label setting of CIFAR-10. In each figure, the horizontal axis represents the training epoch.

learned model's classification behavior. On the 40-label setting, the performance changes considerably as varying the value of τ_s, confirming that the hyperparameter sensitivity is inversely proportional to the number of labels available. As we can see, at the value of 0.95, the error rates are at their lowest. Particularly, when τ_s is 1, our GSF degenerates into the baseline FixMatch, whose error rates are higher.

Sample Selection Ratio. Recall that we train the classification model by using the moderately confident samples filtered by GSF or PPF, together with the highly confident ones. To know how the ratios of all selected samples and highly confident ones in the unlabeled batch evolve during the model training of our methods, we plot the change curves of these ratios in Fig. 4, where each point represents the averaged ratio over all previous and the current training iterations. As the training process proceeds, these ratios first increase and then stabilize at the level close to 1, indicating that an increasing number of unlabeled samples participate in training; on the 250-label setting, these ratios are consistently higher than those on the 40-label setting, manifesting that more experience, more confidence in making decisions; the gap between the paired ratios of all selected samples and highly confident ones, *i.e*., the ratio of selected moderately confident samples, is big in the earlier stage and decreases in the later stage since the instinct of self-training is to produce more and more samples of high confidence. We also show the mislabeled ratio for different methods in Fig. 4c, where we observe that our GSF and PPF enjoy a lower mislabeled ratio, suggesting that our methods can learn better decision boundaries closer to the ground-truth ones.

Saliency Map Visualization. To intuitively understand what the network has learned, we utilize the typical Grad-CAM [35] to visualize the saliency maps from FixMatch and our GSF and PPF. The saliency map for an example is obtained by weighting the feature maps with the gradients w.r.t. the features. The results are shown in Fig. 5, where the examples are randomly sampled from each class. We find that all methods attend to the image regions that are semantically related to classification in most examples; with 25 labels per class, they produce more accurate visualizations, implying that more supervisory information is conducive to pattern learning and thus the SSL technique self-training is in urgent need of improvement to better generate and utilize the massive amounts of pseudo-labeled data. Notably, our methods learn better feature representations that capture more complete semantic patterns, *e.g*., first example.

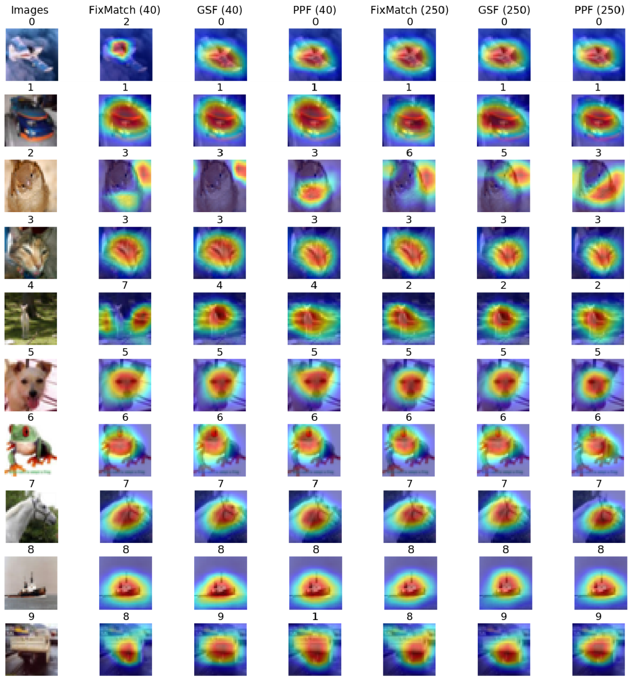


Figure 5. Visualizing the Grad-CAM saliency maps from the baseline FixMatch and our proposed GSF and PPF on 40- and 250-label settings. Note that the number on top of each picture means the ground-truth (first column) or predicted labels (other columns).

4.2. Results

We compare our methods with existing ones on the standard CIFAR-10, CIFAR-100, and SVHN benchmarks. The compared methods can be divided into two groups. The first

	CIFAR-10			CIFAR-100			SVHN		
Method	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels	40 labels	250 labels	1000 labels
Π-Model [29]	-	54.26±3.97	14.01±0.38	-	57.25±0.48	37.88±0.11	-	18.96±1.92	7.54±0.36
Pseudo-Labeling [17]	-	49.78±0.43	16.09±0.28	-	57.38±0.46	36.21±0.19	-	20.21±1.09	9.94±0.61
Mean Teacher [38]	-	32.32±2.30	9.19±0.19	-	53.91±0.57	35.83±0.24	-	3.57±0.11	3.42±0.07
MixMatch [5]	47.54±11.50	11.05±0.86	6.42±0.10	67.61±1.32	39.94±0.37	28.31±0.33	42.55±14.53	3.98±0.23	3.50±0.28
UPS [31]	-	-	6.42	-	-	-	-	-	-
Meta-Semi [41]	-	-	6.10±0.10	-	-	29.69±0.18	-	-	-
UDA [44]	29.05±5.93	8.82±1.08	4.88±0.18	59.28±0.88	33.13±0.22	24.50±0.25	52.63±20.51	5.69±2.76	2.46±0.24
ReMixMatch [4]	19.10±9.64	5.44±0.05	4.72±0.13	44.28±2.06	27.43±0.31	23.03±0.56	3.34±0.20	2.92±0.48	2.65±0.08
FixMatch [37]	13.81±3.37	5.07±0.65	4.26±0.05	48.85±1.75	28.29±0.11	22.60±0.12	3.96±2.17	**2.48**±0.38	**2.28**±0.11
CoMatch [18]	**6.91**±1.39	4.91±0.33	-	-	-	-	-	-	-
GSF	7.73±2.50	**4.53**±0.11	**3.82**±0.10	**41.96**±1.43	**26.32**±0.33	**21.45**±0.18	**3.16**±1.22	**2.47**±0.18	**2.38**±0.10
PPF	**7.71**±3.06	**4.84**±0.17	**3.94**±0.14	**42.08**±1.10	**26.42**±0.17	**21.34**±0.13	**3.10**±0.59	2.50±0.22	2.39±0.12

Table 1. Error rates (%) for CIFAR-10, CIFAR-100, and SVHN.

group is based on either self-training or consistency regularization, *i.e*., Π-Model, Pseudo-Labeling, Mean Teacher, and UPS; the second one relies on combining the two techniques, *i.e*., MixMatch, UDA, ReMixMatch, FixMatch, and CoMatch. For CIFAR-10, we evaluate on 40, 250, and 4000 label settings; for CIFAR-100 and SVHN, we evaluate on cases of 4, 25, and 100 labels per class. We compute the mean and standard deviation of test accuracy over 5 trials with different sets of labeled data.

The results are reported in Tab. 1, from which we take several interesting observations below. **(1)** The methods in the second group exhibit a clear performance gain over those in the first group mostly, indicating that the design of technique combination is reasonable and effective. **(2)** The performance gain decreases as the number of labels per class increases. For example, on CIFAR-10, MixMatch improves over Meach Teacher by 21.27% with 25 labels per class but only by 2.77% with 400 labels per class. This is reasonable since the goal of SSL is to learn models that perform better in cases of fewer labels. **(3)** By enforcing prediction consistency between weakly- and strongly-augmented samples, UDA outperforms MixMatch by a significant margin on most settings; in particular, with only 4 labels per class, UDA is 18.49% and 8.33% better than MixMatch on CIFAR-10 and CIFAR-100 respectively. **(4)** With distribution alignment and augmentation anchoring, ReMixMatch performs much better than UDA, *e.g*., gains of 9.95%, 15%, and 49.29% on the 40-label setting from CIFAR-10, CIFAR-100, and SVHN respectively. **(5)** Without distribution alignment to encourage predictions to follow the class distribution of labeled data, FixMatch is inferior to ReMixMatch, *i.e*., decreases of 4.57% and 0.62% with 4 labels per class on CIFAR-100 and SVHN respectively, despite the gain of 5.29% on CIFAR-10 with 40 labeled samples. **(6)** Our GSF and PPF are substantially better than FixMatch, *e.g*., gains of 6.89% and 6.77% with 4 labels per class, 1.97% and 1.87% with 25 labels per class, and 1.15% and 1.26% with 100 labels per class on the challenging CIFAR-100, respectively. It is noteworthy that without distribution alignment, our methods still largely exceed ReMixMatch, *e.g*., by 2.32% and 2.2% on the 400-label setting from CIFAR-100 respectively, confirming the effectiveness of our methods on discovering helpful samples of moderate confidence. **(7)** With the number of labels decreased, our methods exhibit an increasing performance gain over FixMatch on all benchmarks and are comparable to the state-of-the-art CoMatch, suggesting that the strategy of discovering the effectiveness of moderately confident samples has the potential to handle label-scarce scenarios.

5. Conlusion and Future Work

In this work, we found that some moderately confident samples are useful to improve recognition accuracy for semi-supervised learning (SSL). We pick them out by gradient synchronization filter or prototype proximity filter, which are derived from the proposed Taylor expansion inspired filtration framework. The former can strengthen the optimization dynamic of fully-supervised learning by selecting samples whose gradients are similar to class-wise majority gradients. The latter can enhance the semantic prototypicality of learned feature representation by selecting samples close to prototypes. Experiments on SSL benchmarks show that our methods based on FixMatch achieve significant improvements in accuracy, verifying their efficacy in filtering samples of moderate confidence.

Although our methods as plug-ins incur very little extra cost, FixMatch-like methods themselves are time-consuming. It discourages research groups with limited GPU resources from advancing research. A possible solution is to build an international idle resource dynamic scheduling platform by the conference organizer.

Acknowledgments. This work is supported in part by Program for Guangdong Introducing Innovative and Enterpreneurial Teams (No.: 2017ZT07X183), National Natural Science Foundation of China (No.: 61771201), and Guangdong R&D key project of China (No.: 2019B010155001).

References

[1] Devansh Arpit, Stanisław Jastrzundefinedbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proc. Int. Conf. Mach. Learn.*, page 233–242, 2017. 2, 4

[2] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *ICCV*, pages 8443–8452, October 2021. 1, 3

[3] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *NeurIPS*, page 3365–3373, 2014. 2

[4] 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. 1, 3, 8

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, volume 32, 2019. 1, 8

[6] W. Chang, T. You, S. Seo, S. Kwak, and B. Han. Domain-specific batch normalization for unsupervised domain adaptation. In *CVPR*, pages 7346–7354, 2019. 2

[7] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 57–64, 2005. 1

[8] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, volume 33, pages 18613–18624, 2020. 6

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 1

[10] Geoff French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *ICLR*, 2018. 1

[11] Manfred Gilli, Dietmar Maringer, and Enrico Schumann. Chapter 11 - basic methods. In Manfred Gilli, Dietmar Maringer, and Enrico Schumann, editors, *Numerical Methods and Optimization in Finance (Second Edition)*, pages 229–271. Academic Press, second edition edition, 2019. 4

[12] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, pages 529–536, 2004. 1, 2, 3

[13] Lan-Zhe Guo, Zhen-Yu Zhang, Yuan Jiang, Yu-Feng Li, and Zhi-Hua Zhou. Safe deep semi-supervised learning for unseen-class unlabeled data. In Hal Daumé III and Aarti Singh, editors, *Proc. Int. Conf. Mach. Learn.*, volume 119 of *Proceedings of Machine Learning Research*, pages 3897–3906. PMLR, 13-18 Jul 2020. 2

[14] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Unsupervised domain adaptation with hierarchical gradient synchronization. In *CVPR*, pages 4042–4051, 2020. 2, 4

[15] A. Krizhevsky. Learning multiple layers of features from tiny images. In *Technical report*, 2009. 6

[16] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2016. 1, 2, 3

[17] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. In *Proc. Int. Conf. Mach. Learn. Worksh.*, 07 2013. 1, 2, 3, 8

[18] Junnan Li, Caiming Xiong, and Steven C.H. Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *ICCV*, pages 9475–9484, October 2021. 1, 3, 8

[19] Wei-Hong Li, Chuan-Sheng Foo, and Hakan Bilen. Learning to impute: A general framework for semi-supervised learning. *CoRR*, abs/1912.10364, 2019. 2

[20] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS*, volume 32. Curran Associates, Inc., 2019. 2

[21] Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *CVPR*, pages 8896–8905, 2018. 1

[22] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, page 152–159, 2006. 2

[23] G. J. McLachlan. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70:365–369, 1975. 3

[24] Takeru Miyato, Shin-Ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE TPAMI*, 41:1979–1993, 2019. 1, 3

[25] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Workshop of Proc. Neur. Info. Proc. Sys.*, 2011. 6

[26] Avital Oliver, Augustus Odena, Colin Raffel, Ekin D. Cubuk, and Ian J. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *NeurIPS*, page 3239–3250, 2018. 1, 3

[27] Y. Pan, T. Yao, Y. Li, Y. Wang, C. Ngo, and T. Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *CVPR*, pages 2234–2242, 2019. 4, 5

[28] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. Meta pseudo labels. In *CVPR*, pages 11557–11568, June 2021. 2

[29] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder networks. In *NeurIPS*, page 3546–3554, 2015. 1, 2, 8

[30] Mengye Ren, Sachin Ravi, Eleni Triantafillou, Jake Snell, Kevin Swersky, Josh B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018. 2

[31] 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. 1, 2, 8

[32] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *Seventh IEEE Workshops on Applications of Computer Vision*, volume 1, pages 29–36, 2005. 2

[33] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *NeurIPS*, volume 29, 2016. 1, 2, 3

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

[35] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017. 7

[36] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, page 4080–4090, 2017. 2, 4, 5

[37] 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*, volume 33, pages 596–608, 2020. 1, 2, 3, 4, 5, 6, 8

[38] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, volume 30, 2017. 1, 2, 8

[39] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journ. of Mach. Learn. Res.*, 9:2579–2605, 2008. 5

[40] J.E. van Engelen and H.H Hoos. A survey on semi-supervised learning. *Mach. Learn.*, 109:373–440, 2020. 1, 2, 3

[41] Yulin Wang, Jiayi Guo, Shiji Song, and Gao Huang. Meta-semi: A meta-learning approach for semi-supervised learning. *CoRR*, abs/2007.02394, 2020. 2, 8

[42] Colin Wei, Kendrick Shen, Yi ning Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. In *ICLR*, 2021. 1, 2, 4

[43] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proc. Int. Conf. Mach. Learn.*, pages 478–487, 2016. 5

[44] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, volume 33, pages 6256–6268, 2020. 1, 3, 4, 8

[45] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *Proc. Int. Conf. Mach. Learn.*, pages 5419–5428, 2018. 5

[46] Xin-She Yang. Chapter 1 - introduction to algorithms. In Xin-She Yang, editor, *Nature-Inspired Optimization Algorithms*, pages 1–21. Elsevier, Oxford, 2014. 4

[47] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 1, 6

[48] 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 A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *NeurIPS*, 2021. 1, 3

[49] Liheng Zhang and Guo-Jun Qi. Wcp: Worst-case perturbations for semi-supervised deep learning. In *CVPR*, pages 3911–3920, 2020. 1, 2

[50] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *ICLR*, 2021. 2, 4

[51] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pages 297–313, 2018. 4