Progressive Feature Adjustment for Semi-supervised Learning from Pretrained Models

Hai-Ming Xu[1] Lingqiao Liu[1✉] Hao Chen[2] Ehsan Abbasnejad[1] Rafael Felix[1]
[1]Australian Institute for Machine Learning, University of Adelaide [2]Zhejiang University
{hai-ming.xu, lingqiao.liu, ehsan.abbasnejad, rafael.felixalves}@adelaide.edu.au
haochen.cad@zju.edu.cn

Abstract

As an effective way to alleviate the burden of data annotation, semi-supervised learning (SSL) provides an attractive solution due to its ability to leverage both labeled and unlabeled data to build a predictive model. While significant progress has been made recently, SSL algorithms are often evaluated and developed under the assumption that the network is randomly initialized. This is in sharp contrast to most vision recognition systems that are built from fine-tuning a pretrained network for better performance. While the marriage of SSL and a pretrained model seems to be straightforward, recent literature suggests that naively applying state-of-the-art SSL with a pretrained model fails to unleash the full potential of training data. In this paper, we postulate the underlying reason is that the pretrained feature representation could bring a bias inherited from the source data, and the bias tends to be magnified through the self-training process in a typical SSL algorithm. To overcome this issue, we propose to use pseudo-labels from the unlabelled data to update the feature extractor that is less sensitive to incorrect labels and only allow the classifier to be trained from the labeled data. More specifically, we progressively adjust the feature extractor to ensure its induced feature distribution maintains a good class separability even under strong input perturbation. Through extensive experimental studies, we show that the proposed approach achieves superior performance over existing solutions.

1. Introduction

Semi-supervised learning (SSL) is considered one of the most practical learning paradigms which can leverage both labeled and unlabeled samples to build a prediction model [4]. With the rapid development of deep neural networks (DNNs), extensive research on deep SSL methods [18, 28, 23, 30, 3, 35, 26] have been studied. Among those studies, most of them are evaluated and developed based on randomly initialized parameters. In recent years, the release and re-use of pretrained DNNs to alleviate training costs are becoming common practice for computer vision research and applications [42]. It seems that applying the existing SSL method to a pretrained model is straightforward. However, recent literatures [45, 32] suggest that such a naive solution fails to unleash the full potential of training data and there seems to be a big room to improve the performance of SSL when a pretrained model is used.

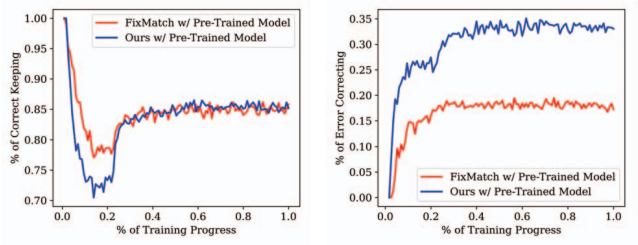

Figure 1. Visualization of the correct keeping and error correcting ability of FixMatch and ours approach with a pretrained model. The y-axis of the left figure denotes the percentage of the initially correctly labeled data keeping their correct class at a given iteration. The y-axis of the right figure denotes the percentage of the initially incorrectly labeled data being predicted to the correct class at a given iteration. The experiment is conducted on *FGVC Aircraft* dataset with 15% labels. As seen, FixMatch with a pretrained model shows weaker error correcting ability than the ours approach. Please refer to Section 4.1 for more details.

In this study, we postulate the key issue preventing existing SSL solutions from attaining their full potential with pretrained models is due to the bias of the pretrained feature extractor: trained from the source domain data, e.g., ImageNet, the feature extractor may not be optimal for the target problem, e.g., fine-grained visual recognition. Such a bias could be a much more severe issue for SSL than for supervised learning. This is because the self-training (or pseudo-labeling) procedure commonly used in SSL tends to magnify the bias. For example, at the beginning of the training, a biased feature extractor and few labeled data could

make the classifier vulnerable to spurious correlated patterns, resulting in a wrong prediction on the unlabeled data. The wrong prediction, however, will be fed back to the classifier again as pseudo labels and reinforce the bias. While SSL from randomly initialized network also suffer from the incorrect pseudo-labels, their feature extractors do not have the bias inherited from the source domain and thus could be easier adjusted through standard SSL methods towards the target problem. In reality, the bias of the feature extractor leads to the phenomenon that the SSL process from a pretrained model is less likely to correct its wrong prediction at the early training stage, as shown in Figure 1.

In this work, we find that a surprisingly simple solution can largely resolve such an issue: we do not use unlabeled data and the corresponding pseudo-label to update the classifier but the feature extractor. Only labeled data are used to train the classifier. The rationale for this strategy is that the feature extractor is more tolerant to the label noise since it does not directly produce the final prediction, and with a good feature representation, it is possible to achieve good performance with only a few labeled data. More specifically, we require the feature representation should become closer to its class center while being pushed away from other class centers. Inspired by FixMatch [26], we also expect the above property holds for strongly-augmented data. The proposed method only modifies existing FixMatch by changing a few lines of code but has demonstrated a dramatic performance boost and even outperforms other carefully designed contrastive-learning-based approaches by a large margin. In summary, the main contribution of this paper are as follows:

- We provide an insight into the issue that standard SSL methods perform unsatisfactory with pretrained models and provide empirical evidence for a better understanding.
- We discover that a simple solution that can significantly improve SSL from a pretrained model. Note that we do not claim the operation of aligning feature to its class-wise embedding is our novelty but the discovery that such a simple strategy can be a good solution to our studied issue.
- We establish a strong baseline for SSL with pretrained model, providing practioner a simple-to-use solution for practical semi-supervised classification.

2. Related Work

Semi-supervised learning (SSL) has experienced rapid progress with the development of deep neural networks (DNNs) [3, 2, 26, 44, 33]. The current state-of-the-art SSL approaches [18, 28, 23, 3, 26, 20, 37] usually depend on the consistency regularization [3] and pseudo-labeling [20]. A common framework is to employ two processes: one process generates a prediction target, usually in the form of pseudo-labeling [26, 20], but could also be logits [28] or other supervision forms [3, 38, 36]. Then the generated pseudo supervision will be used to update the network with a different input, e.g., a different augmentation of the original input image [26, 7], mixed image [3], or a network with different parameters [28].

However, existing SSL approaches are primarily optimized from randomly initialized weights, and recent works [45, 32] show that the impressive performance improvement of these standard SSL methods (includes the state-of-the-art method FixMatch [26]) will disappear when models are training from a pretrained model. Despite the initial findings reported in [45], it still lacks a clear picture of why existing SSL methods perform unsatisfactory when pretrained models are used. In this paper, we investigate the optimization procedure of SSL from pretrained models and provide some empirical evidences to reveal barriers that limit performance. Based on the analysis, we further propose a feature adjustment module to progressively adjust the feature extractor and achieve great performance improvement on multiple vision benchmarks.

3. Preliminary of Semi-supervised Learning

In semi-supervised learning, two sets of samples are normally provided: $\{x_l^1, x_l^2, \cdots, x_l^{N_l}\} \in \mathcal{X}_\mathcal{L}$ whose annotations $\{y_l^1, y_l^2, \cdots, y_l^{N_l}\} \in \mathcal{Y}_\mathcal{L}$ are available and $\{x_u^1, x_u^2, \cdots, x_u^{N_u}\} \in \mathcal{X}_\mathcal{U}$ where $N_u \gg N_l$ but without accessing label information.

Although there are many existing SSL methods in the literature, this work mainly takes one of the state-of-the-art approaches FixMatch [26] as an example. It is because FixMatch successfully integrates two popular techniques in SSL together, i.e., consistency regularization and pseudo-labeling, through decoupling the artificial label generation and model update with weak and strong data augmentations. Specifically, for samples from $\mathcal{X}_\mathcal{L}$, the model $\mathcal{M}$ is trained via a standard classification loss. For each unlabeled data $x_u^i \in \mathcal{X}_\mathcal{U}$, "hard" pseudo label is firstly produced on the weakly augmented image

$$p(y|x_u^i) = \mathcal{M}\left(A_0(x_u^i)\right);\ \tilde{y} = \underset{c}{\operatorname{argmax}}\, p(y=c|x_u^i), \quad (1)$$

where $A_0(\cdot)$ denotes weak data augmentation. Then, the model is optimized to have a consistent prediction on the strongly augmented image

$$\mathcal{L}_u = \frac{1}{|\mathcal{B}_u|}\sum_{i\in\mathcal{B}_u} \mathbb{1}\big(p(\tilde{y}|x_u^i) \geq \tau\big)\mathrm{CE}\Big(\mathcal{M}\big(A_1(x_u^i)\big),\ \tilde{y}\Big), \quad (2)$$

where $p(\tilde{y}|x_u^i)$ means the $\tilde{y}$-th output probability on weakly augmented x_u^i. Here, $\mathbb{1}(\cdot)$ is the indicator function to select

samples whose predicting confidence is greater than a predefined confidence threshold τ. Additionally, $\mathrm{CE}(\cdot,\cdot)$ is the standard cross-entropy loss, $A_1(\cdot)$ is the strong augmentation and $\mathcal{B}_u$ denotes the size of unlabeled samples in one mini-batch.

4. Semi-supervised Learning from Pretrained Models

In this section, we first investigate the bias inherented in pretrained models and how existing SSL methods are troubled to achieve satisfactory performance. Then, we propose a feature adjustment module for SSL to resist the bias.

4.1. Pretrained Models as a Double-edged Sword for SSL

With the growth of data and the enrichment of computing resources, developing powerful pretrained models have attracted increasing attention from both the academia and industry [10, 13, 5, 24, 40, 43]. On the one hand, pretraining on large-scale images gives models the general feature extraction ability for various kinds of downstream applications [46, 22, 39]. On the other hand, the generated feature representations will inevitably bring a bias inherited from the source data, and thus it is usually necessary to fine-tune the model on target datasets for effective usage of pretrained models [34, 12, 15].

For the SSL scenario, it is natural to expect that state-of-the-art performance can be achieved by applying the state-of-the-art SSL method to a pretrained model, i.e., semi-supervised fine-tuning. However, evidence from recent literatures [45, 32] show that this solution is far from the best, and there is a big room to improve SSL when a pretrained model is used. This motivates us to revisit SSL and understand what hinders it from achieving its full potential. We postulate the major issue is that the use of pretrained model is a double-edged sword for SSL: on the one hand, it brings the prior knowledge learned from the source data and boosts the performance. On the other hand, it also introduces a strong prediction bias inherited from the source data. After all, the feature learned from the source domain data may not be optimal for the target task. In effect, the prediction bias will encourage the classifier to use certain features or visual patterns for prediction, especially when the labeled data is limited in quantity and diversity. However, not all those visual patterns are true causal factors to determine the class and the spurious correlation might be mistakenly identified during training, e.g., prediction could rely on the clue from background [25].

Worse, as most of the state-of-the-art SSL methods [26, 3, 20, 2, 44] are built upon self-training, a.k.a., pseudo-labeling framework, which generates pseudo labels from the prediction on the unlabeled data, such a process tends to further magnify the prediction bias. For example, we can consider the scenario of applying FixMatch, one of the most commonly used SSL methods, with a pretrained model. At the beginning of training, due to the limited amount of labeled (and pseudo-labeled) samples and biased feature representation, the learned classifier tends to be affected by the spurious correlation between features and class labels. Then if the classifier generates incorrect pseudo labels from unlabeled data, the bias will thus reinforce itself by further training with such pseudo labels. Consequently, this will make the SSL less prone to correct its wrong prediction made during the training process.

In order to verify our assumption, we conduct an empirical analysis on FixMatch with pretrained models. Specifically, we introduce two measurements called correct-keeping rate (CKR) and error-correcting rate(ECR). The former is defined as a percentage of the initially[1] correctly labeled data keeping their correct class at a given iteration, while the latter one is defined as the percentage of the initially incorrectly labeled data being predicted to the correct class at a given iteration. The statistical results are shown in Figure 1. Then we can observe that FixMatch (with a pretrained model) has a descent CKR metric when training converges. However, the ECR of FixMatch (with a pretrained model) quickly reaches a plateau and has a poor ECR metric. This issue becomes more evident by comparing its ECR with the proposed method.

4.2. Progressive Feature Adjustment

The above analysis suggests that the standard SSL approach could suffer more from the biased feature extractor and we may need a special process to alleviate the impact of bias. In this work, we propose to only use the labeled data to train the classifier, while pseudo-labels generated from the larger amount of unlabeled samples will only be used to update the feature extractor. In this way, unlabeled data influences the classifier indirectly by producing better feature representations. As the classifier is always trained on noise-free labeled data, even if the feature representation is imperfect, the classifier can suppress the noisy dimensions and identify the discriminative patterns in the feature representation. Thus the feature extractor can be more tolerant to the noise in pseudo-labels. It seems that one drawback of the above method is the lack of training examples for the classifier. However, since a pretrained feature extractor has already been able to provide a reasonable starting point and will be further refined by the proposed progressive adjustment method, training the classifier on a limited number of samples can guarantee a good performance. The overall architecture is shown in Figure 2.

Specifically, given a batch of labeled samples $(x_l, y_l) \in$

[1] "Initially" here means an unlabeled sample was falsely labeled for the first time in the whole training process.

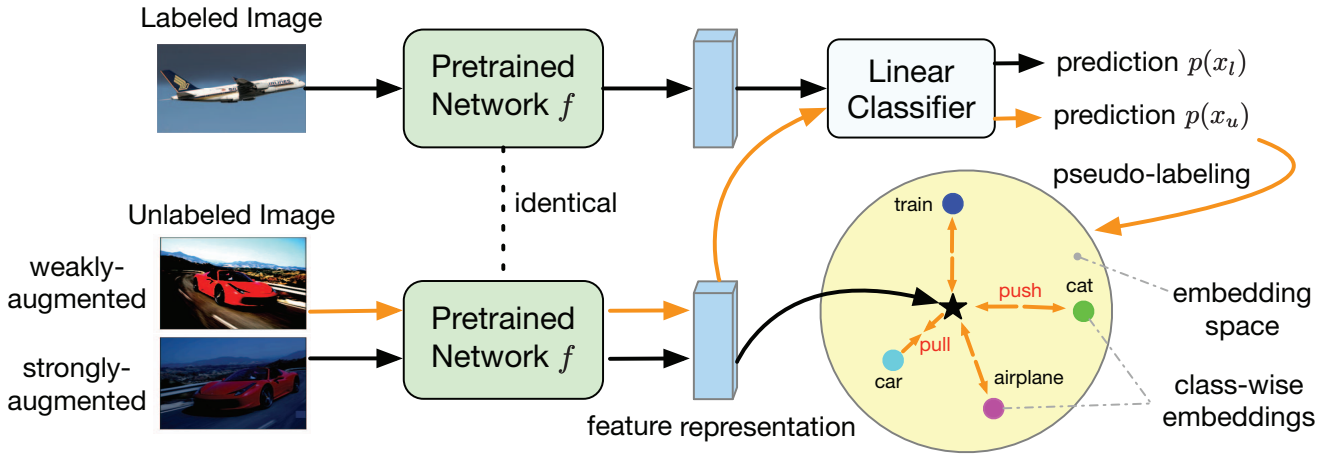

Figure 2. Overview of our approach. The feature extractor is initialized with a pretrained model and the classifier is random initialized for the target dataset. In order to alleviate the bias as presented in Section 4.1, we let the classifier only trained on the labeled samples and use the large amount of unlabeled samples to adjust the feature extractor alone through pulling the immediate feature representation to its corresponding class embedding and pushing away to other class embeddings. The class-wise embeddings are progressively updated along the model optimization as presented in Eq. 6 and Eq. 7.

$\mathcal{B}_l$, both of the feature extractor f and the linear classifier $\mathbf{W} := \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_c, \cdots\}$ will be optimized together

$$p(y = c|x_l^i) = \frac{\exp\left(\mathbf{w}_c^T f(x_l^i)\right)}{\sum_j \exp\left(\mathbf{w}_j^T f(x_l^i)\right)},$$

$$\mathcal{L}_l = \frac{1}{|\mathcal{B}_l|} \sum_{i \in \mathcal{B}_l} \mathrm{CE}\left(p(y|x_l^i), y_l^i\right) \tag{3}$$

where $\mathbf{w}_c$ denotes the classifier for class c. $\mathrm{CE}(\cdot, \cdot)$ is the standard cross-entropy loss.

For a batch of unlabeled samples $x_u \in \mathcal{B}_u$, we utilize the up-to-date classifier to generate posterior probability estimation $p(y|x_u^i)$

$$p(y = c|x_u^i) = \frac{\exp\left(\mathbf{w}_c^T f\left(A_0(x_u^i)\right)\right)}{\sum_j \exp\left(\mathbf{w}_j^T f\left(A_0(x_u^i)\right)\right)}, \tag{4}$$

where $f(A_0(x_u^i))$ denotes the feature extracted by first going through the weak data augmentation module A_0 and then the feature extractor f (same as that in FixMatch). The class corresponding to the maximal posterior probability is the predicted class of the given unlabeled sample, that is, $\tilde{y}^i = argmax_c\ p(y = c|x_u^i)$. If $p\left(\tilde{y}^i|x_u^i\right) \geq \tau$ where τ is the confidence threshold which can be a fixed scalar, e.g., 0.95 as in FixMatch [26] or a dynamic generated scalar as in FlexMatch [44], then $\tilde{y}^i$ will be used as a pseudo-label for the corresponding unlabeled sample.

Instead of using the pseudo-labeled unlabeled samples to train on the feature extractor and classifier altogether, as in FixMatch, we propose to use them to adjust the feature extractor only. Without introducing an additional linear connected layer, we maintain a set of class-wise embeddings $\{\mu_c\}_0^C$ and try to minimize the following loss:

$$\hat{\mathcal{L}}_f = \frac{1}{|\mathcal{B}_u|} \sum_{i \in \mathcal{B}_u} \mathbb{1}\left(p(\tilde{y}^i|x_u^i) \geq \tau\right) \mathcal{L}_f(x_u^i, \tilde{y}^i) + \frac{1}{|\mathcal{B}_l|} \sum_{i \in \mathcal{B}_l} \mathcal{L}_f(x_l^i, y_l^i).$$

$$where, \mathcal{L}_f(x, y) = -\log \frac{\exp\left(\cos\left(\mu_y, f\left(A_1(x)\right)\right)/T\right)}{\sum_j \exp\left(\cos\left(\mu_j, f\left(A_1(x)\right)\right)/T\right)}, \tag{5}$$

where $cos(\cdot, \cdot)$ denotes the cosine similarity and T is a temperature hyperparameter. We empirically set $T = 0.1$ in our study. A_1 denotes a different type of data augmentation to A_0 and we use RandAugment [8] followed by Cutout [11] as the strong data augmentation A_1. μ_c is the running class mean vector for the c-th class.

In effect, the above loss function will pull features from the same class closer while push features from different classes far apart. For labeled data, we could use the ground-truth class label for assigning samples to their corresponding class embeddings. For unlabeled data, we use pseudo-labels instead and only apply the loss to samples that can generate pseudo-labels. Also, motivated by FixMatch, we propose to apply this loss on strongly augmented data to further avoid the confirmation bias. Note that the above loss also implicitly encourages same-class features from the labeled and unlabeled data move closer relative to the distance to other class samples. Thus it tends to make the classifier learned from the labeled data more generalizable to unlabeled data.

To sum up, we train the classifier with labeled data only and both labeled and unlabeled data with the loss in Eq. 5. The overall loss function $\mathcal{L}$ is the weighted summation of both: $\mathcal{L} = \mathcal{L}_l + \lambda \cdot \hat{\mathcal{L}}_f$, where λ is the fixed weight hyper-

parameter.

In order to adjust the feature extractor efficiently, oracle class-wise embeddings of target dataset should be an ideal choice. However, it is unrealistic due to the lack of annotations for unlabeled samples and the inherent bias in the pretrained feature extractor. Thus, we propose to progressively update these class-wise embeddings from both labeled and unlabeled data. For labeled data, μ_c is updated via

$$\mu_c^{new} = \beta\mu_c^{old} + (1-\beta)f\big(A_1(x_l^i)\big)\mathbb{1}(y_l^i = c), \quad (6)$$

and for unlabeled data, μ_c is updated via

$$\mu_c^{new} = \beta\mu_c^{old} + (1-\beta)f\big(A_1(x_u^i)\big)\mathbb{1}\Big(p\big(c|A_0(x_u^i)\big) \geq \tau\Big), \quad (7)$$

where the indicator function $\mathbb{1}(y_l^i = c)$ selects samples from the c-th class from the labeled data, the indicator function $\mathbb{1}\Big(p^t\big(c|A_0(x_u^i)\big) \geq \tau\Big)$ selects unlabeled samples that are confidently classified into the c-th class by the classifier. This is identical to the criterion of generating pseudo-labels. β is a momentum term that controls how far the class-wise feature reaches into embedding history.

5. Experimental results

In this section, we compare our approach with several SSL methods with pretrained models.

5.1. Experimental details

We strictly follow [32] to design our experiment, including the evaluation datasets and pretrained model choices. We made such a choice since Self-Tuning [32] has demonstrated the state-of-the-art performance and its experimental evaluation is comprehensive and realistic. Our code will be released after the anonymity period. Some experimental details are as follows:

Datasets: Following the protocol of [32], which addressed the same research problem as this paper, four vision benchmarks are evaluated, i.e., *FGVC Aircraft* [21], *Stanford Cars* [16], *CUB-200-2011* [31], and *CIFAR-100* [17]. Specifically, the first three are challenging fine-grained classification datasets and label proportions ranging from 15% to 50% are tested. Label partition of *CIFAR-100* follows the standard SSL protocol: 4/25/100 labeled images per class.

Methods: Nine popular deep SSL approaches are included for comparison, i.e., Π-model [19], Pseudo-Labeling [20], Mean Teacher [28], UDA [35], FixMatch [26], FlexMatch [44], SimCLRv2 [6], FixMatch+AKC+ARC [1] and Self-Tuning [32]. Meanwhile, performance of Fine-Tuning on labeled data is also reported for a reference baseline. For our approach, we also consider a simple extension by incorporating it into the recently proposed consistency-based SSL method FlexMatch [44], which uses dynamically assigned threshold with a FixMatch framework. We call this extension **Ours+**. Note that it shows our approach can still boost the performance even with more advanced SSL algorithms. In our work, all experiments were implemented in PyTorch and run on a GeForce RTX 2080Ti GPU with 11GB memory.

Method	Label Number		
	400	2500	10000
Fine-Tuning (baseline)	60.79	31.69	21.74
Pseudo-Labeling [20]	59.21	–	–
MT [28]	60.68	–	–
UDA [35]	58.32	–	–
FixMatch† [26]	52.88	25.63	18.38
FlexMatch† [44]	40.41	23.19	17.73
Self-Tuning [32]	47.17	24.16	17.57
Ours†	45.48	23.12	16.89
Ours+†	**37.36**	**22.06**	**16.58**

Table 1. Error rates (%) on *CIFAR-100* with EfficientNet-B2. † means ours implementation based on [32].

Pretrained models: Following [32], three models pretrained on ImageNet [9] are chosen for evaluation, i.e., ResNet-50 [14] and EfficientNet [27] which are pretrained in a supervised way, and a ResNet-50 which is trained through an unsupervised learning method MoCo v2 [13].

5.2. Train from Supervised Pretrained Models

In this section, we compare various SSL methods trained from supervised pretrained models.

Fine-grained Classification benchmarks: We use a ResNet-50 network, which is supervised pretrained on ImageNet, to initialize all SSL models. The results are shown in Table 2. It is clear that our proposed method achieves overall significant improvement than other comparing SSL approaches. Specifically, compared with traditional SSL methods, our approach increases the test accuracy by a large margin on all kinds of partitions of three benchmarks. Taking the state-of-the-art method FixMatch [26] as an example, the performance gain of our approach exceeds 10 percent on both *Stanford Cars* and *CUB-200-2011* with 15% labels. This is thanks to the proposed feature adjustment module in our approach which greatly reduces the bias inherented in the pretrained model. Furthermore, our approach is also superior to the recently proposed Self-Tuning method [32] especially when labels are limited, e.g., only 15% training samples are labeled. When the CPL module proposed in FlexMatch [44] is added to our approach, Ours+ leads to a further performance boost.

Standard SSL benchmarks: We choose *CIFAR-100* dataset [17] which is one of the most challenging datasets among standard SSL benchmarks to evaluate SSL methods

Dataset	Method	Label Proportion		
		15 %	30 %	50 %
FGVC Aircraft	Fine-Tuning (baseline)	39.57±0.20	57.46±0.12	67.93±0.28
	Π-model [19]	37.32±0.25	58.49±0.26	65.63±0.36
	Pseudo-Labeling [20]	46.83±0.30	62.77±0.31	73.21±0.39
	Mean Teacher [28]	51.59±0.23	71.62±0.29	80.31±0.32
	UDA† [35]	59.50±0.36	74.08±0.41	81.10±0.42
	FixMatch† [26]	60.19±0.43	75.28±0.39	81.19±0.41
	FlexMatch† [44]	63.21±0.15	77.08±0.34	82.56±0.22
	SimCLRv2 [6]	40.78±0.21	59.03±0.29	68.54±0.30
	FixMatch+AKC+ARC† [1]	63.87±0.41	75.99±0.38	81.24±0.31
	Self-Tuning [32]	64.11±0.32	76.03±0.25	81.22±0.29
	Ours†	69.64±0.41	82.36±0.44	85.02±0.33
	Ours+†	**71.23**±0.26	**82.80**±0.15	**85.53**±0.32
Stanford Cars	Fine-Tuning (baseline)	36.77±0.12	60.63±0.18	75.10±0.21
	Π-model [19]	45.19±0.21	57.29±0.26	64.18±0.29
	Pseudo-Labeling [20]	40.93±0.23	67.02±0.19	78.71±0.30
	Mean Teacher [28]	54.28±0.14	66.02±0.21	74.24±0.23
	UDA† [35]	61.88±0.39	79.16±0.36	86.79±0.31
	FixMatch† [26]	64.97±0.37	81.23±0.31	87.74±0.35
	FlexMatch† [44]	71.96±0.28	83.81±0.26	88.12±0.21
	SimCLRv2 [6]	45.74±0.16	61.70±0.18	77.49±0.24
	FixMatch+AKC+ARC† [1]	68.63±0.38	82.81±0.27	87.98±0.32
	Self-Tuning [32]	72.50±0.45	83.58±0.28	88.11±0.29
	Ours†	77.22±0.42	86.91±0.07	90.38±0.16
	Ours+†	**79.70**±0.31	**87.92**±0.32	**90.71**±0.13
CUB-200-2011	Fine-Tuning (baseline)	45.25±0.12	59.68±0.21	70.12±0.29
	Π-model [19]	45.20±0.23	56.20±0.29	64.07±0.32
	Pseudo-Labeling [20]	45.33±0.24	62.02±0.31	72.30±0.29
	Mean Teacher [28]	53.26±0.19	66.66±0.20	74.37±0.30
	UDA† [35]	52.23±0.23	67.93±0.25	75.63±0.28
	FixMatch† [26]	54.21±0.26	69.28±0.28	77.49±0.31
	FlexMatch† [44]	61.26±0.18	71.62±0.32	78.06±0.21
	SimCLRv2 [6]	45.74±0.15	62.70±0.24	71.01±0.34
	FixMatch+AKC+ARC† [1]	63.21±0.35	73.61±0.32	79.08±0.29
	Self-Tuning [32]	64.17±0.47	75.13±0.35	80.22±0.36
	Ours†	65.55±0.21	74.99±0.33	80.00±0.11
	Ours+†	**68.06**±0.22	**76.09**±0.34	**80.40**±0.21

Table 2. Test accuracy (%) ↑ on three fine-grained SSTL benchmarks. We empirically find strong augmentation for labeled data used in Self-Tuning [32] can bring performance gains to other SSL methods. Following the same setting of Self-Tuning, Methods with † are implemented by ourself based on the released codebase of Self-Tuning [32].

from a pretrained model. Due to the lack of open-resourced pretrained checkpoints on WideResNet-28-8 model [41], EfficientNet-B2 model [27] supervised pretrained on ImageNet is adopted in this work. Table 1 presents the error rates of each method. Our proposed method yields the best performance among the comparing methods.

5.3. Train from Unsupervised Pretrained Models

Various semi-supervised learning approaches have been shown to benefit from supervised pretrained models in Section 5.2, we continue to study the transfer effect from MoCov2 [13] which is pretrained on ImageNet without using any annotations. As the test accuracy presented in Figure 4, our best performed model, Ours+, excels to other semi-supervised learning baselines.

5.4. Ablation Study

We are interested in ablating our approach from the following perspective views:

5.4.1 The distribution of feature representation:

In our approach, the progressive feature adjustment module is introduced to update the feature extractor separately for

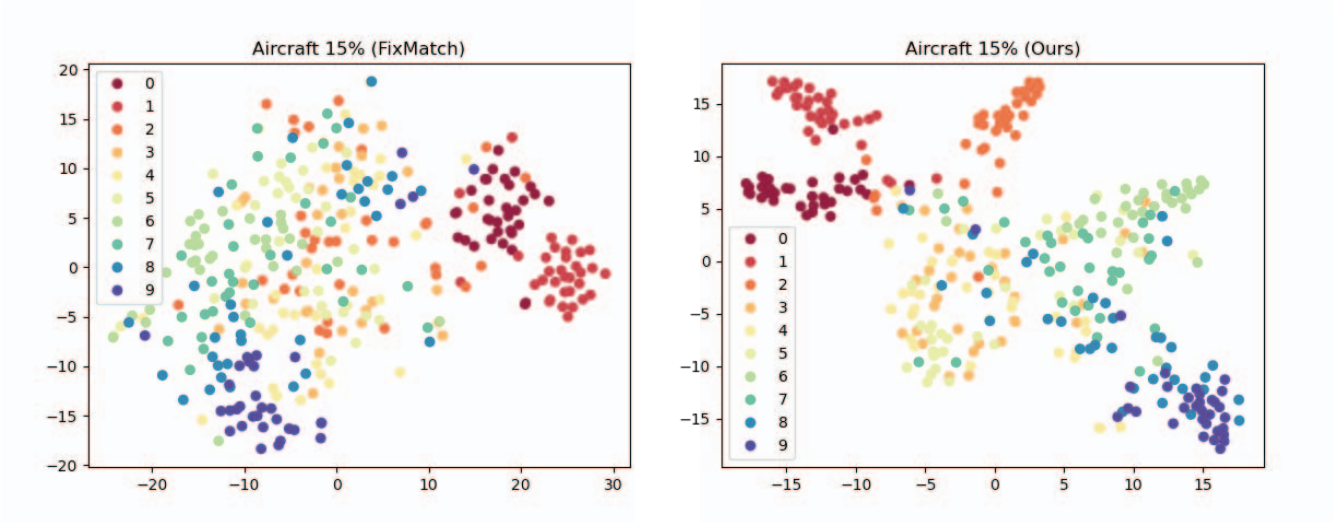

Figure 3. Feature embedding visualizations of (left) FixMatch and (right) Ours approach for the first 10 classes of *FGVC Aircraft* dataset by using t-SNE [29]. Both of the models are initialized with identical ResNet-50 supervised pre-trained on ImageNet.

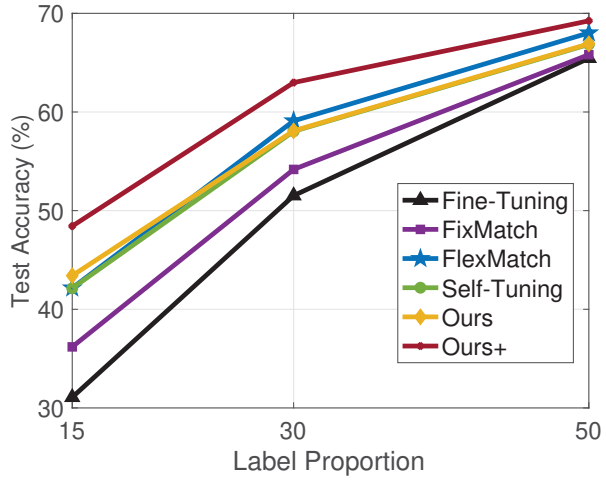

Figure 4. Test accuracy (%) ↑ of comparing methods on *CUB-200-2011* with MoCov2 which is unsupervisedly pre-trained on ImageNet1K [9].

Method	Label Number		
	400	2500	10000
FixMatch [26]	42.50	27.07	21.88
HCCMatch (ours)	42.69	26.16	21.39

Table 3. Error rates (%) ↓ on *CIFAR-100* with a *randomly initialized* WideResNet-28-8 [41] network. We implement HCCMatch based on the PyTorch implementation[3] of FixMatch which has obtained better performance than the reported ones in [26].

Method (w/ same pre-trained model)	Label Proportion		
	15	30	50
UDA [35]	59.50	74.08	81.10
UDA+Feature Adjustment(ours)	**65.74**	**80.11**	**83.83**

Table 4. Ablation study to the effectiveness of the proposed progressive feature adjustment module to the popular consistency-regularization based SSL method UDA on *FGVC Aircraft*.

alleviating the bias inherented in pretrained models. Therefore, we are interested in the effect of using such module or not on the feature distribution. Figure 3 presents the feature distribution of FixMatch and ours approach for some classes of *FGVC Aircraft* with t-SNE [29]. We can find that our method encourages same class features to be close to each other while staying away from the other class samples and produces a more distinguishable distribution for the target data, while FixMatch suffers from the inherented bias of pretrained model and poorly adapts the feature distribution given the observation whose features from different classes are mixed together, thus its performance is heavily limited.

[3] https://github.com/kekmodel/FixMatch-pytorch (CC BY)

5.4.2 Is our approach effective for randomly initialized network?

In our formulation, the progressive feature adjustment module can be seen as a special SSL method for SSL from a pretrained model. So we are interested to know its effectiveness for randomly initialized network. To investigate this, we train our approach with a randomly initialized WideResNet-28-8 [41] network on CIFAR-100. As the results shown in Table 3, our approach does not produce significant improvement as what we have observed in the SSL with pretrained models task. We postulate that this is because the feature extractor of a randomly initialized network does not inherited the prediction bias from the source domain, and thus the original design in FixMatch algorithm has already been sufficient to adjust the feature extractor for the target problem.

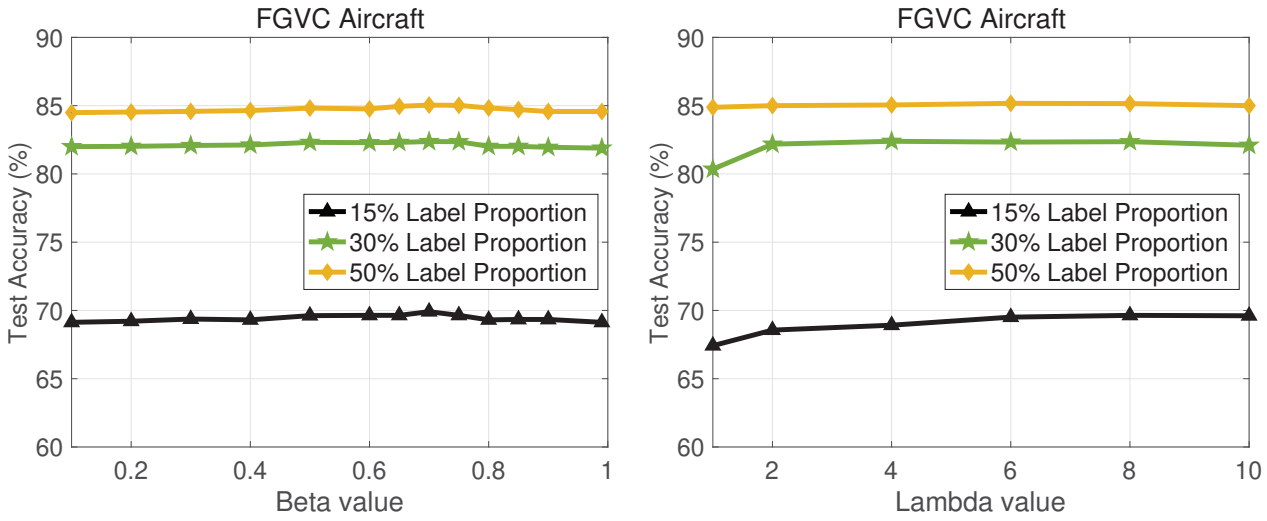

Figure 5. Ablation study to the hyperparameter sensitivity.

5.4.3 Does our approach work for other SSL method?

We are also interested in if the proposed progressive feature adjustment module can be extended to other SSL methods. To investigate this, we apply this module to UDA [35] which is another popular consistency-regularization based SSL method. We conduct experiments on *FGVC Aircraft* dataset and present the results in Table 4. As seen, by incorporating the proposed progressive feature adjustment module, we can significantly improve UDA in the SSL from pretrained models setting. This suggests that the proposed progressive feature adjustment module could be used to upgrade various consistency-regularization based SSL methods when pretrained models are available.

5.4.4 The ways of updating class-wise embeddings

In our approach, the class-wise embedding, i.e., class mean vectors, are dynamically updated from features of strongly augmented labeled images and features of strongly augmented unlabeled samples whose pseudo-supervisions are confident enough. In this section, we investigate two alternative strategies: 1) accumulate features of weakly augmented images to update mean vectors, 2) estimate parameters from features of unlabeled samples without a confidence threshold. As the results shown in Table 5, both of these two alternatives will result in a slight performance drop to our method.

5.4.5 The sensitivity analysis of hyperparameter selection in our approach:

There are two hyperparameters in our method: one is the momentum term beta (i.e., β) for updating the class-wise embedding μ in Eq. 6 and Eq. 7, and the other one is the balance weight lambda (i.e., λ) for the overall loss. As shown in Figure 5, Our method is robust to the selection of both β and λ hyperparameters.

Ways of updating μ in HCCMatch	Label Proportion		
	15	30	50
w/ weakly augmented images	68.38	82.06	84.79
w/o confidence threshold	68.73	81.61	84.49
default (ours)	**69.64**	**82.36**	**85.02**

Table 5. Ablation study to the ways of implementing online generative classifier learning in the proposed HCCMatch approach on *FGVC Aircraft* dataset.

6. Conclusion

Semi-supervised learning from pretrained models is an encouraging research direction, because it combines the advantages of the two learning paradigms to achieve more data-efficient learning. Given the observations in the literature show that existing semi-supervised learning algorithms do not produce a satisfactory performance boost compared to their training-from-scratch version, we investigate the learning procedure of semi-supervised learning from pretrained models and find that the bias inherented in the original pretrained models may be magnified along the semi-supervised training. Empirical evidences are also provided for a better understanding. Based upon the analysis, we propose a progressive feature adjustment module to decouple the process of pseudo-supervision generation and model update and thus alleviate the bias successfully. Extensive experimental results on four vision benchmarks verify the effectiveness of our proposed approach.

Acknowledgement. This work was done in Adelaide Intelligence Research (AIR) Lab and Hai-Ming Xu and Lingqiao Liu are supported by the Centre of Augmented Reasoning (CAR).

References

[1] Abulikemu Abuduweili, Xingjian Li, Humphrey Shi, Cheng-Zhong Xu, and Dejing Dou. Adaptive consistency regularization for semi-supervised transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6923–6932, 2021. 5, 6

[2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *ArXiv preprint*, abs/1911.09785, 2019. 2, 3

[3] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5050–5060, 2019. 1, 2, 3

[4] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. 1

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 3

[6] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 5, 6

[7] Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium, 2018. Association for Computational Linguistics. 2

[8] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 4

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. 5, 7

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[11] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *ArXiv preprint*, abs/1708.04552, 2017. 4

[12] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4805–4814, 2019. 3

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. IEEE, 2020. 3, 5, 6

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 5

[15] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018. 3

[16] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 5

[17] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 5

[18] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 1, 2

[19] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 5, 6

[20] Donghyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, 2013. 2, 3, 5, 6

[21] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *Technical report*, 2013. 5

[22] Christos Matsoukas, Johan Fredin Haslum, Moein Sorkhei, Magnus Söderberg, and Kevin Smith. What makes transfer learning work for medical images: Feature reuse & other factors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9225–9234, 2022. 3

[23] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization

method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 1, 2

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3

[25] Yangyang Shu, Baosheng Yu, Haiming Xu, and Lingqiao Liu. Improving fine-grained visual recognition in low data regimes via self-boosting attention mechanism. In *European Conference on Computer Vision*, pages 449–465. Springer, 2022. 3

[26] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1, 2, 3, 4, 5, 6, 7

[27] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019. 5, 6

[28] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1195–1204, 2017. 1, 2, 5, 6

[29] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 7

[30] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3635–3641. ijcai.org, 2019. 1

[31] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5

[32] Ximei Wang, Jinghan Gao, Mingsheng Long, and Jianmin Wang. Self-tuning for data-efficient deep learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 10738–10748. PMLR, 2021. 1, 2, 3, 5, 6

[33] Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, Heli Qi, Zhen Wu, Yu-Feng Li, Satoshi Nakamura, Wei Ye, Marios Savvides, Bhiksha Raj, Takahiro Shinozaki, Bernt Schiele, Jindong Wang, Xing Xie, and Yue Zhang. Usb: A unified semi-supervised learning benchmark for classification. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2

[34] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Growing a brain: Fine-tuning by increasing model capacity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2471–2480, 2017. 3

[35] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1, 5, 6, 7, 8

[36] Hai-Ming Xu, Lingqiao Liu, and Ehsan Abbasnejad. Progressive class semantic matching for semi-supervised text classification. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3003–3013, Seattle, United States, July 2022. Association for Computational Linguistics. 2

[37] Hai-Ming Xu, Lingqiao Liu, Qiuchen Bian, and Zhen Yang. Semi-supervised semantic segmentation with prototype-based consistency regularization. *Advances in Neural Information Processing Systems*, 35:26007–26020, 2022. 2

[38] Hai-Ming Xu, Lingqiao Liu, and Dong Gong. Semi-supervised learning via conditional rotation angle estimation. In *2021 Digital Image Computing: Techniques and Applications (DICTA)*, pages 01–08. IEEE, 2021. 2

[39] Ning Yang, Sio Hang Pun, Mang I Vai, Yifan Yang, and Qingliang Miao. A unified knowledge extraction method based on bert and handshaking tagging scheme. *Applied Sciences*, 12(13):6543, 2022. 3

[40] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 3

[41] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. BMVA Press, 2016. 6, 7

[42] Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 6241–6245. ijcai.org, 2019. 1

[43] Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, et al. Pangu-alpha: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *arXiv preprint arXiv:2104.12369*, 2021. 3

[44] 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 *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-12, 2021, virtual*, 2021. 2, 3, 4, 5, 6

[45] Hong-Yu Zhou, Avital Oliver, Jianxin Wu, and Yefeng Zheng. When semi-supervised learning meets transfer learning: Training strategies, models and datasets. *Technical report*, 2018. 1, 2, 3

[46] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 3