A Realistic Evaluation of Semi-Supervised Learning for Fine-Grained Classification

Jong-Chyi Su Zezhou Cheng Subhransu Maji
University of Massachusetts Amherst
{jcsu, zezhoucheng, smaji}@cs.umass.edu

Abstract

We evaluate the effectiveness of semi-supervised learning (SSL) on a realistic benchmark where data exhibits considerable class imbalance and contains images from novel classes. Our benchmark consists of two fine-grained classification datasets obtained by sampling classes from the Aves and Fungi taxonomy. We find that recently proposed SSL methods provide significant benefits, and can effectively use out-of-class data to improve performance when deep networks are trained from scratch. Yet their performance pales in comparison to a transfer learning baseline, an alternative approach for learning from a few examples. Furthermore, in the transfer setting, while existing SSL methods provide improvements, the presence of out-of-class is often detrimental. In this setting, standard fine-tuning followed by distillation-based self-training is the most robust. Our work suggests that semi-supervised learning with experts on realistic datasets may require different strategies than those currently prevalent in the literature.

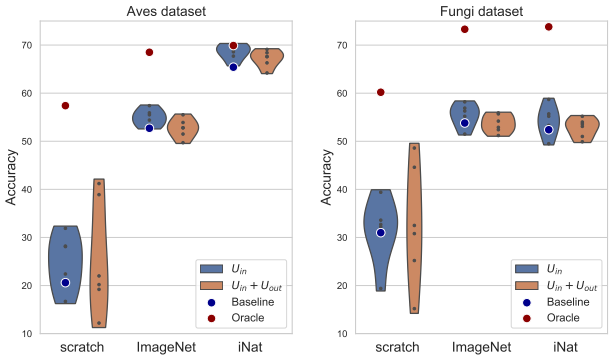

Figure 1. Accuracy of semi-supervised learning (SSL) algorithms on the Semi-Aves and Semi-Fungi datasets (see Fig. 2) using (i) different pre-trained models, and (ii) in-class (U_{in}) and out-of-class ($U_{in} + U_{out}$) unlabeled data. The performances of the supervised baseline and supervised oracle are also shown. Transfer learning from experts is far more effective than SSL from *scratch*, while in the transfer setting SSL provides modest gains. Though out-of-class data (U_{out}) is valuable when training from scratch, it is not the case when training from experts (details in Tab. 2 and 3).

1. Introduction

Semi-supervised learning (SSL) aims to exploit unlabeled data to train models from a few labels, making them practical for applications where labels are a bottleneck. Yet, the current literature on SSL with deep networks for image classification has two main shortcomings. First, most methods are evaluated on curated datasets such as CIFAR, SVHN, or ImageNet, where class distribution is or is close to uniform and unlabeled data contains no novel classes. This is implicit in methods that rely on the assumption that the data is uniformly clustered, use a uniform instead of class-balanced loss, or categorize unlabeled data into one of the labeled classes. In practice, however, class distribution can be highly unbalanced or even unknown, and the unlabeled data may contain novel classes. How effective is SSL in these situations?

Second, most literature has focused on training models from scratch. However, a practical approach for few-shot learning is to use expert models trained on large labeled datasets such as ImageNet [36] or iNaturalist [46]. What gains does SSL provide in this setting, especially since many SSL methods are based on learning invariances from data based on transformations which might have already been learned by the experts during supervised training? Moreover, is out-of-domain data beneficial when experts are available?

Our paper aims to answer these questions by conducting a systematic study of SSL techniques (Fig. 1) on two fine-grained classification datasets that exhibit a long-tailed distribution of classes and contain a large number of out-of-class images (Fig. 2). These datasets are obtained by sampling classes under the Aves (birds) and Fungi taxonomy. The out-of-class images are other Aves (or Fungi) images not belonging to the classes within the labeled set. The first dataset was part of the semi-supervised challenge at FGVC7 workshop [41], while the second one is constructed from

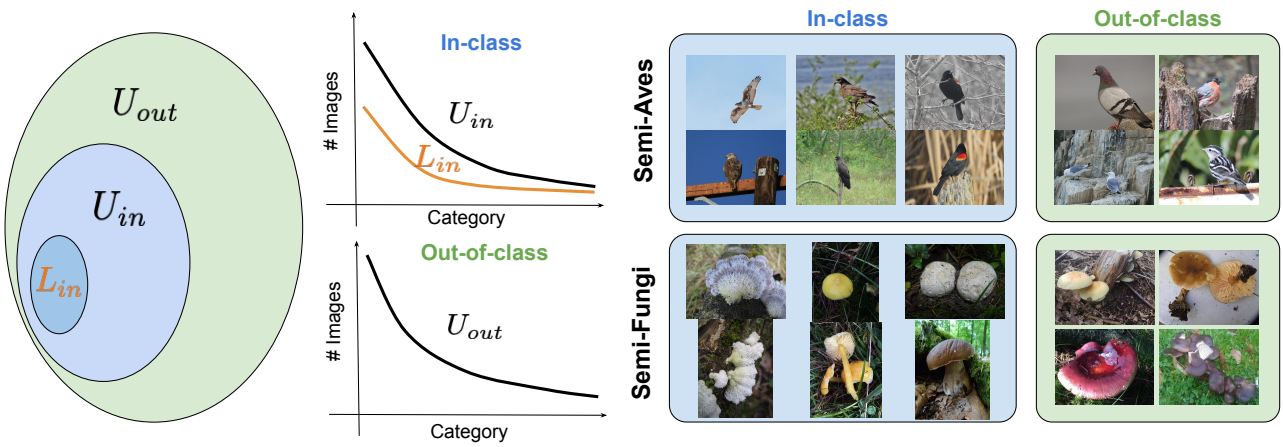

Figure 2. **The proposed benchmark for semi-supervised learning.** The benchmark contains two datasets, with classes from the Aves and Fungi taxa respectively. Each represents a 200-way classification task and the training set contains (i) labeled images from these classes L_{in}, (ii) unlabeled images from these classes U_{in}, and (iii) unlabeled images from related classes U_{out}, as seen on figures to the right. Moreover, the classes exhibit a long-tailed distribution with an imbalance ratio of 8 to 10. The benchmark captures conditions observed in some realistic applications that are not present in existing datasets used to evaluate semi-supervised learning. See § 3 and Tab. 1 for details.

the FGVC fungi challenge [1] following a similar scheme, details of which are described in § 3. We also provide a benchmark on the CUB dataset [48] in the supp. material.

On these datasets, we conduct a systematic study of existing deep-learning-based semi-supervised learning approaches for image classification. We perform experiments on SSL methods including Pseudo-Label [25], Curriculum Pseudo-Label [8], FixMatch [39], self-training using distillation [52], self-supervised learning (MoCo [18]), as well as their combinations when effective. We investigate strategies for using unlabeled data when models are initialized from experts. We also evaluate the performance of methods that use unlabeled data from the same classes as the labeled dataset (U_{in}) and a practical setting where the unlabeled data includes out-of-class images ($U_{in} + U_{out}$). The high-level summary of the experiments reported in Fig. 1, Tab. 2, 3, and Fig. 3 are as follows:

- Some of the SSL methods are effective when models are trained from scratch, especially those with self-supervised pre-training can significantly benefit from out-of-class data (long blue whiskers and longer orange whiskers above the *baseline* for *scratch* in Fig. 1). In this setting, self-supervised learning followed by distillation-based self-training performs the best (Tab. 2 and 3).
- The best SSL approach significantly under-performs the supervised fine-tuning model trained on the labeled portion of the datasets (the *baseline* performance of *ImageNet* and *iNat* is higher than any SSL model trained from scratch in Fig. 1).
- Picking the right expert provides further gains in this few-shot setting but not when training using the entire labeled dataset (*oracle* performance in Fig. 1).
- When training with experts, FixMatch gives the most improvements when having U_{in} only. However, the presence of out-of-class unlabeled data often hurts performance. Self-Training was the most robust to the presence of out-of-class data (Tab. 2, 3 and Fig. 3).
- Surprisingly, we found that no method was able to reliably use out-of-class data even though the domain shift is relatively small (the orange group is not higher than the blue groups for *ImageNet* and *iNat* unlike *scratch* in Fig. 1), echoing the experience of participants in the FGVC7 challenge [41].
- The performance of SSL is far below the model trained using labels of the entire in-class data suggesting that there is significant room for improvement (*oracle* performance in Fig. 1).

In summary, we conduct a systematic evaluation of several recently proposed SSL techniques on two challenging datasets representing a long-tailed distribution of fine-grained categories. We vary the initialization and the domain of the unlabeled data and analyze the robustness of various SSL approaches. Our experiments indicate that SSL does not work out-of-the-box in a transfer learning setting, especially in the presence of out-of-domain data. These results are in a similar vein to prior work on the evaluation of SSL approaches that have analyzed the robustness of SSL techniques to the choice of hyper-parameters [30], network architectures [9, 51], and domain shifts [30, 42, 49], *etc*. However, the evaluation in a transfer learning setting on the proposed benchmarks reveals additional insights. We hope these experiments inspire practical methods that combine the benefits of supervised learning and task-specific learning on partially labeled datasets.

2. Related Works

Semi-supervised learning has a long history in machine learning. In this section, we describe the trends in recent techniques based on deep learning and refer the reader to surveys on SSL for a comprehensive view [31, 45, 56, 57].

Self-Training. These techniques use the model's prediction to automatically generate labels for the unlabeled data [27, 38]. Pseudo-Labeling [25] includes confident predictions, *i.e.*, those greater than a threshold for training. The pseudo-labels can be added iteratively to induce a "curriculum" [4, 8, 17]. Alternatively, one can add an entropy penalty to encourage confident predictions on the unlabeled data [15]. Other methods [9, 52, 53, 58] involve re-training a "student model" from a "teacher model" using its prediction computed in different ways. For example, adding noise and using a larger student model [52, 58], selecting k-most confident pseudo-labels [53], or using a distillation loss which softens the predictions [9, 52]. While these methods have been shown to be successful in various datasets, the effectiveness of the approach is critically dependent on the initial performance of the model and the data distribution. Our experiments show that the presence of out-of-class data negatively impacts some of these methods while using expert initialization provides a significant benefit.

Consistency-based learning. These methods learn by encouraging the consistency of the model's predictions on the unlabeled data. These could be across different augmentations of the data [3, 24, 34, 37], including adversarial versions [28]. Alternatively, consistency can be enforced across time, *e.g.*, using moving average of the predictions (*temporal ensembling* [24]), using the moving average of model parameters (*mean teacher* [43]), or using a stochastic averaging of model parameters [2]. A number of methods for *data augmentation* have been proposed which has generally improved both supervised and semi-supervised learning. These include the variety of image augmentations proposed in RandAugment [11], the CutOut scheme [12], linear combinations of images used in MixUp [55], and even augmentations in the feature space [23]. These augmentations have been incorporated in methods such as MixMatch [6], ReMixMatch [5], FixMatch [39], UDA [51], and ICT [47] in different ways for consistency-based learning. We choose FixMatch as the candidate approach which has shown state-of-the-art results on existing SSL benchmarks, which we describe in detail in § 4. While consistency via data-augmentation is effective when a model is trained from scratch, it is unclear if this is effective when using a pre-trained model, as invariance to these transformations may have been acquired during supervised pre-training.

Self-supervised learning. Another line of work has explored using self-supervised (or unsupervised) learning objectives to improve semi-supervised learning. These include incorporating pre-text tasks such as predicting image rotations [14], the order of patches (jigsaw puzzle task) [29] during semi-supervised learning [35, 42, 54]. Alternatively, self-supervised learning can be used as an initialization before training with labels. The recent success of self-supervised learning based on contrastive learning [9, 18, 21, 32, 44] has been incorporated by several approaches leading to promising results on ImageNet [9]. We also find that contrastive learning followed by self-training is the best performing when trained from scratch on our benchmark. However, the value of out-of-class data is diminished when using expert models.

Analysis of semi-supervised learning. The most related to our work is that of Oliver *et al*. [30] who provide a benchmark for comparing deep-learning-based SSL methods for image classification. While only CIFAR10 and SVHN datasets were used, the paper pointed out that hyper-parameters can have a significant impact on performance. Yet these are hard to tune without access to large amounts of labeled data, which is precisely the setting in semi-supervised learning. In our analysis, we pay careful attention to hyper-parameter optimization (see § 5.1). Their work also showed that transfer learning from experts can be more effective, but did not explore if their combination with semi-supervised learning can be helpful. In addition, they showed out-of-class unlabeled data may be harmful, but the classes in U_{in} and U_{out} are widely different. Last, their work as well as most SSL methods have been presented on well-curated datasets. Our work focuses on the evaluation in a realistic setting and includes an analysis of methods proposed since Oliver *et al*.'s paper. These include methods based on self-training, self-supervised training, and FixMatch which outperform the consistency-based approach [28] analyzed in their paper. More comparisons between [30] are provided in the supplementary material.

3. A Realistic Benchmark

In SSL, we are provided with labeled training data $(x_i, y_i) \in L_{in}$ and unlabeled training data $(u_i, \cdot) \in U$. The unlabeled data can either belong to the same classes as the labeled data (U_{in}), or to novel classes (U_{out}). In a realistic setting, one may expect that the unlabeled data contains novel classes. In many applications it is easy to acquire images from related domains through coarse labeling, *e.g.*, it is easier to label an image as a bird than a "yellow bunting". Such images could be potentially used to learn better representations. Thus we evaluate SSL methods in two settings, one when the unlabeled data contains no novel images, and another when it does, *i.e.*, U_{in} and $U_{in} + U_{out}$ respectively.

Dataset	Classes L_{in} / U_{in} / U_{out}	Images L_{in} / U_{in} / U_{out}	Unlabeled Class Domain	Image Resolution	Class Distribution	Imbalance Ratio
CIFAR-10	10 / 10 / 0	4K / 40K / 0	$L = U$	32×32	uniform	1
CIFAR-100	100 / 100 / 0	10K / 50K / 0	$L = U$	32×32	uniform	1
SVHN	10 / 10 / 0	1K / 65K / 0	$L = U$	32×32	uniform	1
STL-10	10 / 0 / -	5K / 0 / 100K	$L \neq U$	96×96	uniform	1
ImageNet	1000 / 1000 / 0	140K / 1.26M / 0	$L = U$	224×224	≈ uniform	1.8
Semi-Aves	200 / 200 / 800	6K / 27K / 122K	$L = U_{in} \neq U_{out}$	224×224	long-tailed	7.9
Semi-Fungi	200 / 200 / 1194	4K / 13K / 65K	$L = U_{in} \neq U_{out}$	224×224	long-tailed	10.1

Table 1. A comparison of Semi-Aves and Semi-Fungi datasets with existing SSL benchmarks. The Semi-Aves and Semi-Fungi present a challenge due to the large number of classes, presence of novel images in the unlabeled set, long-tailed distribution of classes as indicated by the class imbalance ratio (maximum / minimum images per class) in the training set.

We use two datasets by sampling classes from the natural domains for our benchmark. As shown in Fig. 2, the classes belong to the Aves and Fungi taxonomy and contain a long-tailed distribution of classes, as commonly observed in fine-grained domains. Tab. 1 shows a comparison with other benchmarks. Larger image sizes, significant class imbalance, fine-grained categories, and a large number of out-of-class images allow a more realistic evaluation of SSL techniques. Below we describe each dataset.

Semi-Aves. We use the dataset from the semi-supervised challenge at the FGVC7 workshop at CVPR 2020 [41]. The dataset includes a subset of bird species from the Aves kingdom of iNaturalist 2018 dataset [46]. However, there are no overlapping images since the images were collected from recent years. There are 200 in-class and 800 out-of-class categories. The training and validation set has a total of 5959 labeled images, 26,640 and 122,208 in-class and out-of-class unlabeled images, and 8000 test images. The training data in L_{in}, U_{in}, and U_{out} is long-tail distributed, specifically L_{in} has 15 to 53 images and U_{in} has 16 to 229 images per class. The test data has a uniform distribution with 40 images per class.

Semi-Fungi. We create a Semi-Fungi dataset following the similar strategy of the Semi-Aves dataset. We use the train-val set of images from the FGVCx Fungi challenge at the FGVC5 workshop at CVPR 2018 [1]. The dataset was collected from the "Svampe Atlas"[1] website, thus the image domain is different from iNaturalist. The original dataset has 1394 fungi species with a long-tailed distribution. We first sort the classes by frequency and randomly select 200 of the top 600 classes as in-class categories. We then select 20 images per class as the test set, and randomly select 4141 images as labeled data and the rest 13,166 images as in-class unlabeled data. The rest 1194 species are used as out-of-class unlabeled images, which has a total of 64,871 images. In Semi-Fungi, there are 6 to 78 images per class in L_{in}, and 16 to 276 images in U_{in}. The test set is uniformly distributed with 20 images per class.

[1] https://svampe.databasen.org

4. Methods

In this section, we describe the details of the SSL methods we compared in our benchmark.

(1) Supervised baseline / oracle: We train the model only using labeled data L_{in} with a cross-entropy loss. For the oracle, we include the ground-truth labels of U_{in} for training.

(2) Pseudo-Labeling [25]: The approach uses a base model's confident predictions on unlabeled images as labels. Concretely, if the maximum probability of a class is greater than a threshold τ, we then take the class as the target label. Following the implementation of Oliver *et al.* [30], we sample half of the batch from L_{in} and half from unlabeled data U during training. Denote (x_i, y_i) as a labeled sample, the predictions on unlabeled data u_i of the model f as $q_i = f(u_i)$, pseudo-label as $\hat{q}_i = \text{argmax}(q_i)$, and cross-entropy function as $H(p, q) = -\sum_r p(r) \log q(r)$. Then, the objective for each batch is:

$$\mathcal{L} = \sum_{j=1}^{n} H(y_i, f(x_i)) + \sum_{i=1}^{n} \mathbb{1}\big[\max(q_i) \geq \tau\big] H(\hat{q}_i, q_i). \tag{1}$$

(3) Curriculum Pseudo-Labeling [8]: Unlike pseudo-labeling where labels are generated in an online manner, curriculum labeling generates pseudo-labels after the training is finished on the current labeled set before retraining. We first train a supervised model on labeled data L_{in}, then select images with the highest predictions from all the unlabeled data $u \in U$, and add them with their pseudo-labels to the labeled dataset. In the next iteration, we retrain a model *from scratch* using the new set of labeled data. We repeat

this process 5 times and select {20, 40, 60, 80, 100}% of the unlabeled data from the original pool of unlabeled data U. The steps are as the following:

(i) Initialize $L = L_{in}$, $\beta = 20$.

(ii) Supervised training on L .

(iii) Generate predictions $q = f_\theta(x)$ for every $u \in U$.

(iv) From U select β% examples with highest prediction scores and their pseudo-labels as L_{top}.

(v) Add selected unlabeled data with their pseudo-labels to the labeled dataset $L = L_{in} \cup L_{top}$.

(vi) If β <100, $\beta = \beta$+20 and repeat from step (ii) .

(4) FixMatch: FixMatch combines pseudo-labeling and consistency regularization. For each unlabeled image, it minimizes the cross-entropy between the pseudo-label (thresholded prediction) of the weakly-augmented image and the predictions of the strong-augmented image. For labeled data, only weak augmentations are applied. Specifically, let α be a weak augmentation (image flipping in our case) and $\mathcal{A}$ be a strong augmentation (RandAugment [11] in our case). Let the predictions under strong and weak augmentations are $Q_i = f(\mathcal{A}(u_i))$, $q_i = f(\alpha(u_i))$. The total loss for labeled and unlabeled data is

$$\mathcal{L} = \sum_{j=1}^{m} H(y_j, f(\alpha(x_j))) + \sum_{i=1}^{km} \mathbb{1}\big[\max(q_i) \geq \tau\big] H(\hat{q_i}, Q_i). \tag{2}$$

In the original implementation, each batch uses m labeled and km unlabeled data with a total batch size $n = (k+1)m$, where the sampling ratio k is a hyper-parameter.

(5) Self-Training: While the term of "Self-Training" is general, we use this to refer to the following procedure using distillation [20]. We first train a supervised model f^t on the labeled data which we call the teacher model, then train a student model f^s with scaled cross-entropy loss on the unlabeled data and cross-entropy loss on labeled data. Distillation was originally used for model compression [7], but has been shown to improve the performance when training the student model with the same architecture [13] or across different modalities [16, 40, 44]. Given unlabeled data $(u, \cdot)$, let the *logits* from teacher and student model as z^t and z^s, and the prediction of labeled data (x, y) from the student model is y^s. The objective includes the cross-entropy loss for labeled data (x, y), and the distillation loss for unlabeled data:

$$\mathcal{L} = (1-\lambda) \sum_{j=1}^{n} H(y_j, y_j^s) + \lambda \sum_{i=1}^{n} H\left(\sigma\left(\frac{z_i^t}{T}\right), \sigma\left(\frac{z_i^s}{T}\right)\right), \tag{3}$$

where λ is the weight between supervised and distillation losses, σ is the softmax function, and T is a temperature (scaling) parameter.

(6) Self-Supervised Learning (MoCo [18]): We use momentum contrastive (MoCo) learning as a strong baseline for self-supervised training. MoCo learns an image encoder $f(x)$ that maps the image x to a representation $q = f(x)$ and uses a contrastive objective that requires positive pairs to be closer than negative pairs in the representation space. The positive pairs are sampled from two geometric or photometric augmented views of a same images while negative images are augmentations from different images. MoCo adapts the InfoNCE [32] loss as the objective function. The loss for each encoded query q is:

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k^+/T)}{\exp(q \cdot k^+/T) + \sum_i^K \exp(q \cdot k_i^-/T)}, \tag{4}$$

where T is the temperature, k^+ and k^- are the positive and negative sample of the query q. The number of negative samples K is limited by the mini-batch size. In order to stabilize the training, MoCo uses the memory bank [50] to store the negative samples and updates the encoder of the keys in the memory bank based on momentum. After the self-supervised pre-training, we remove the MLP layers after the global average pooling layer, add a linear classifier (a fully convolutional layer followed by softmax), and train the entire network with supervised cross-entropy loss. We found that freezing the pre-trained backbone gives worse performance than fine-tuning the entire network.

(7) MoCo + Self-Training: Here we initialize the model using MoCo learning on the unlabeled data before semi-supervised learning using Self-Training. A recent work by Chen *et al*. [9] has shown this to be a strong semi-supervised learning baseline. The procedure is as follows:

(i) Pre-train the model using MoCo on L_{in} and U.

(ii) Fine-tune the model on L_{in} with a cross-entropy loss. Call this the teacher model f^t.

(iii) Train a student model f^s initialized from step (i) with distillation loss (Eq. 3) using the teacher model f^t.

5. Experiments

5.1. Implementation details

Network architecture and pre-training. For a fair comparison, we use a ResNet-50 network [19] on 224×224 images for all our experiments. For transfer learning, we use pre-trained models on ImageNet [36] and iNaturalist 2018 (iNat) [46] dataset, which contains 8142 species including 1248 Aves and 321 Fungi species. Note that there are *no overlapping images* between iNat's training set and Semi-Aves, though there are overlapping categories. The images for Semi-Fungi images do not overlap with iNaturalist, but we do not know how many overlapping classes there are as species names were not provided in the original dataset [1]

	Method	from scratch		from ImageNet		from iNat	
		Top1	Top5	Top1	Top5	Top1	Top5
	Supervised baseline	20.6±0.4	41.7±0.7	52.7±0.2	78.1±0.1	65.4±0.4	86.6±0.2
	Supervised oracle	57.4±0.3	79.2±0.1	68.5±1.4	88.5±0.4	69.9±0.5	89.8±0.7
U_{in}	Pseudo-Label [25]	16.7±0.2	36.5±0.8	54.4±0.3	78.8±0.3	65.8±0.2	86.5±0.2
U_{in}	Curriculum Pseudo-Label [8]	20.5±0.5	41.7±0.5	53.4±0.8	78.3±0.5	69.1±0.3	87.8±0.1
U_{in}	FixMatch [39]	28.1±0.1	51.8±0.6	**57.4±0.8**	78.5±0.5	**70.2±0.6**	87.0±0.1
U_{in}	Self-Training	22.4±0.4	44.1±0.1	55.5±0.1	79.8±0.1	67.7±0.2	87.5±0.2
U_{in}	MoCo [18]	28.2±0.3	53.0±0.1	52.7±0.1	78.7±0.2	68.6±0.1	87.7±0.1
U_{in}	MoCo + Self-Training	**31.9±0.1**	**56.8±0.1**	55.9±0.2	**80.3±0.1**	**70.1±0.2**	**88.1±0.1**
$U_{in} + U_{out}$	Pseudo-Label [25]	12.2±0.8	31.9±1.6	52.8±0.5	77.8±0.1	66.3±0.3	86.4±0.2
$U_{in} + U_{out}$	Curriculum Pseudo-Label [8]	20.2±0.5	41.0±0.9	52.8±0.5	77.8±0.1	**69.1±0.1**	**87.6±0.1**
$U_{in} + U_{out}$	FixMatch [39]	19.2±0.2	42.6±0.6	49.7±0.2	72.8±0.5	64.2±0.2	84.5±0.1
$U_{in} + U_{out}$	Self-Training	22.0±0.5	43.3±0.2	**55.5±0.3**	**79.7±0.2**	67.6±0.2	**87.6±0.1**
$U_{in} + U_{out}$	MoCo [18]	38.9±0.4	65.4±0.3	51.5±0.4	77.9±0.2	67.6±0.1	**87.3±0.2**
$U_{in} + U_{out}$	MoCo + Self-Training	**41.2±0.2**	**65.9±0.3**	53.9±0.2	**79.4±0.3**	68.4±0.2	**87.6±0.2**

Table 2. **Results on Semi-Aves benchmark.** We experiment with six different SSL methods as well as supervised baselines under different settings: (1) using U_{in} or $U_{in} + U_{out}$ as the unlabeled dataset, (2) training from scratch, or using ImageNet or iNat pre-trained model. We show that when training from scratch with U_{in}, MoCo + Self-Training performs the best. When having expert models, transfer learning is a strong baseline, and FixMatch and Self-Training can still give improvements. When adding unlabeled data from U_{out}, the performance pales except for the self-supervised method when training from scratch. The best results and those within the variance are marked in **teal**.

	Method	from scratch		from ImageNet		from iNat	
		Top1	Top5	Top1	Top5	Top1	Top5
	Supervised baseline	31.0±0.4	54.7±0.8	53.8±0.4	80.0±0.4	52.4±0.6	79.5±0.5
	Supervised oracle	60.2±0.8	83.3±0.9	73.3±0.1	92.5±0.3	73.8±0.3	92.4±0.3
U_{in}	Pseudo-Label [25]	19.4±0.4	43.2±1.5	51.5±1.2	81.2±0.2	49.5±0.4	78.5±0.2
U_{in}	Curriculum Pseudo-Label [8]	31.4±0.6	55.0±0.6	53.7±0.2	80.2±0.1	53.3±0.5	80.0±0.5
U_{in}	FixMatch [39]	32.2±1.0	57.0±1.2	56.3±0.5	80.4±0.5	**58.7±0.7**	81.7±0.2
U_{in}	Self-Training	32.7±0.2	56.9±0.2	56.9±0.3	81.7±0.2	55.7±0.3	82.3±0.2
U_{in}	MoCo [18]	33.6±0.2	59.4±0.3	55.2±0.2	82.9±0.2	52.5±0.4	79.5±0.2
U_{in}	MoCo + Self-Training	**39.4±0.3**	**64.4±0.5**	**58.2±0.5**	**84.4±0.2**	55.2±0.5	**82.9±0.2**
$U_{in} + U_{out}$	Pseudo-Label [25]	15.2±1.0	40.6±1.2	52.4±0.2	80.4±0.5	49.9±0.2	78.5±0.3
$U_{in} + U_{out}$	Curriculum Pseudo-Label [8]	30.8±0.1	54.4±0.3	54.2±0.2	79.9±0.2	53.6±0.3	79.9±0.2
$U_{in} + U_{out}$	FixMatch [39]	25.2±0.3	50.2±0.8	51.2±0.6	77.6±0.3	53.1±0.8	79.9±0.1
$U_{in} + U_{out}$	Self-Training	32.5±0.5	56.3±0.3	**55.7±0.3**	81.0±0.2	**55.2±0.2**	**82.0±0.3**
$U_{in} + U_{out}$	MoCo [18]	44.6±0.4	72.6±0.5	52.9±0.3	81.2±0.1	51.0±0.2	78.5±0.3
$U_{in} + U_{out}$	MoCo + Self-Training	**48.6±0.3**	**74.7±0.2**	**55.9±0.1**	**82.9±0.2**	54.0±0.2	81.3±0.3

Table 3. **Results on Semi-Fungi benchmark.** We experiment on Semi-Fungi using the same hyper-parameters from Semi-Aves in Table 2. We can see similar conclusions: When training from scratch, MoCo + Self-Training performs the best and adding U_{out} can give an extra performance boost. With expert models, FixMatch and Self-Training (with or without MoCo) is often the best performing one, but the latter is more robust to the out-of-class data.

from which it was constructed. However, this is less of a concern as we find that iNat pre-trained model performs worse than an ImageNet pre-trained model on Semi-Fungi, suggesting the class overlap is likely small if any. To obtain an iNat pre-trained model, we train the model using SGD with momentum with a learning rate of 0.0045 and a batch size of 64 for 75 epochs which matches the reported 60%

Top-1 accuracy[2]. We use the ImageNet pre-trained model from torchvision [33].

Data augmentation. For the Semi-Fungi dataset, we first pre-process the images to have a maximum of 300 pixels for each side, while Semi-Aves has a maximum of 500 pixels. We use random-resize-crop to the size of 224×224 and random-flipping for data augmentation, for all the methods except for MoCo and FixMatch. MoCo additionally uses Gaussian blur, color jittering, and random grayscale conversion, while FixMatch uses RandAugment [11].

Hyperparameter search. We found the SSL methods to be sensitive to hyper-parameters such as learning rates, weight decay, etc. As noted in [30], a small validation set poses a risk of picking sub-optimal hyper-parameters. Moreover, labeled data is best used as a source of supervision. While k-fold cross-validation is an alternative, it is expensive. Hence, we use the combined training and validation set for training SSL methods in our experiments and report performance on the test set which is sufficiently large. In particular, hyperparameters for all methods were based on the performance on the Semi-Aves dataset and kept fixed for the Semi-Fungi dataset (Tab. 3). Thus the results in Tab. 2 should be seen as a validation set performance, while those in Tab. 3 represent a novel test set. However, the high-level conclusions are identical across the two benchmarks.

Semi-supervised training. For SSL methods except for FixMatch, we use SGD with a momentum of 0.9 and a cosine learning rate decay schedule [26] following [22, 39] for optimization. Learning rate and weight decay were picked from a range of [0.03, 0.0001]. We use a batch size of 64 during training. When there is unlabeled data, we select half of the batch from labeled and another half from unlabeled data (32 each). We train models for 10k and 50k iterations for training from expert models and from scratch. Other hyper-parameters include threshold τ for Pseudo-Labeling, which we select from {0.80, 0.85, 0.90, 0.95}. When training from scratch, we use τ=0.85 and 0.8 for with and without U_{out}; when training from experts we use τ=0.95. For Self-Training, we set T=1 and λ=0.7 for all the experiments. For FixMatch [39], we are able to train the model up to a batch size of 192 (32 labeled and 160 unlabeled images) on 4 GPUs. We find the performance drops significantly with small batch size (*e.g.* 48), however, we are unable to use the same batch size as original paper (*i.e.* 6144) due to limited resources. We use a learning rate of 0.01 and threshold τ=0.80 to train FixMatch for 500 epochs when training from scratch and 250 epochs with pre-trained models. We report the results from the last training epoch for all the methods. We also notice that FixMatch has more overfitting and the results could be further improved. More details are provided in the supplementary material.

[2] https://github.com/macaodha/inat_comp_2018

Self-supervised training. We adopt the default settings of MoCo-v2 [10], including MLP projector, 800 training epochs, *etc*., but adapt the number of negative samples and learning rate to our task. We use a batch size of 256 and 2048 negative samples in all experiments. We find that using a large number of negative samples (*e.g.* 65,536) hurts the performance. When training the MoCo from scratch, we use the default learning rate of 0.03; when training MoCo from ImageNet or iNaturalist pre-trained model, we use a smaller learning rate (0.0003) and fewer training epochs (200) to avoid the potential forgetting problem. In the end, we train a classifier on the global average pooling features of ResNet-50 without freezing the backbone. We find that freezing the feature encoder always leads to worse performance than fine-tuning the entire network.

5.2. Results

Our experimental results on Semi-Aves and Semi-Fungi are shown in Tab. 2 and 3, respectively. To better visualize the results, we plot the relative gain of each SSL method, *i.e.* the differences between supervised baseline in raw accuracy, on both datasets in Fig. 3. We discuss the results of each setting in the following.

Training from scratch. We first discuss the results of training from scratch using only U_{in} on both datasets. Comparing to supervised baseline, Curriculum Pseudo-Label does not give improvements and Pseudo-Label even underperforms the baseline. This is possibly due to the low initial accuracy of the model which gets amplified during pseudo labeling. FixMatch and Self-Training both result in improvements. Self-supervised learning (MoCo) gives a good initialization and the improvements are similar or even more than using FixMatch. Finally, Self-Training using MoCo pre-trained model as the teacher model results in a further 2-3% improvement.

Using expert models. We then consider using an ImageNet or iNat pre-trained model for transfer learning with U_{in} only. The transfer learning baseline from either expert model outperforms the best SSL method (MoCo + Self-Training) trained from scratch by a large margin, showing that transfer learning is more powerful in our realistic datasets. This observation echos Oliver *et al*. [30] who showed transferring from ImageNet to CIFAR10 performs better than SSL methods. Next, we can see that most of the SSL methods, as well as MoCo pre-training, provide improvements over the baselines. The only exception is

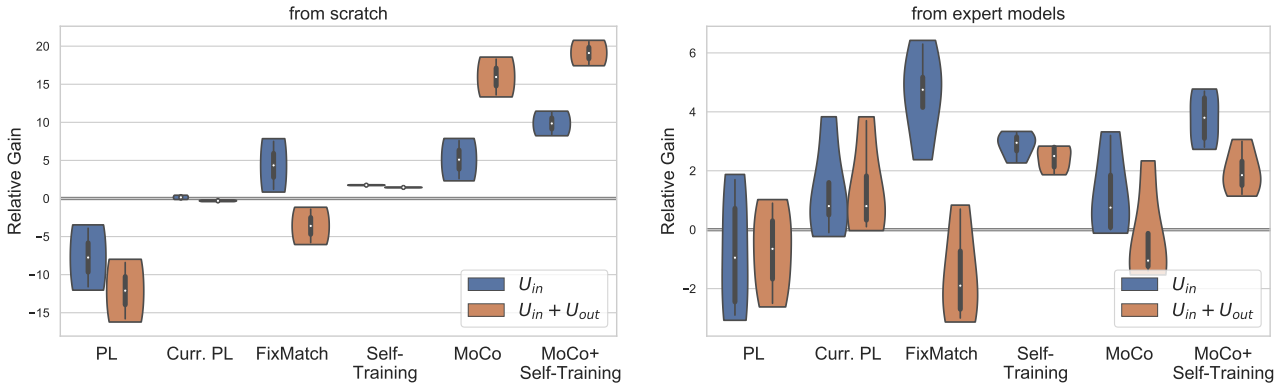

Figure 3. **Relative gains of SSL methods on Semi-Aves and Semi-Fungi. Left:** trained from scratch. **Right:** using expert models. For each SSL method, we plot the relative gain, *i.e.* the difference between the supervised baseline in raw accuracy, from the results in both Tab. 2 and 3. This shows that (1) the presence of out-of-class data U_{out} often hurts the performance, and (2) Self-Training is often the best method when using pre-trained models.

Pseudo-Label on Semi-Fungi. Among SSL methods, FixMatch and MoCo + Self-Training perform the best.

Effect of out-of-class unlabeled data. Now we consider the setting where the unlabeled data contains both in-class and out-of-class data ($U_{in} + U_{out}$). This is the trade-off between more unlabeled data at the cost of a distribution shift. This effect can be seen in the orange vs. blue plot in Fig. 3. When training from scratch, the performances of Pseudo-Label and FixMatch drop by 4-9%, while Curriculum Pseudo-Label and Self-Training only drop by less than 1%, showing that they are more robust to the domain shift of unlabeled data. On the other hand, self-supervised pre-training (MoCo) can benefit significantly from U_{out}, providing around 11% improvement over using U_{in} only on both Aves and Fungi datasets. Combining with Self-Training gives another 3-6% improvement, making the gap between transfer learning baseline smaller.

Finally, we consider having $U_{in} + U_{out}$ with expert models. In Fig. 3 we can see the performance often drops in the presence of U_{out}. Curriculum Pseudo-Label and Self-Training are more robust and yield less than 1% decrease in most cases, while FixMatch is less robust whose performance drops by around 6%. The performances of MoCo also drops around 1-3% and are sometimes worse than the supervised baseline. Adding Self-Training however provides a 1-3% boost in performance. Overall, Self-Training from either a supervised or a self-supervised model is the most robust one.

Robustness to hyper-parameters and trends. We found Pseudo-Label to be sensitive to the threshold τ. When using experts higher thresholds worked better. Increasing the threshold also increased the robustness in the presence of novel classes. Curriculum Pseudo-Label was found to be more robust in our benchmark, even when adding U_{out}. Self-Training was the most robust to hyper-parameters, we chose the same temperature T and weight λ for all the experiments and it consistently improved results regardless of using an expert model or using out-of-domain data.

6. Conclusion

There has been a significant interest in self-supervised and semi-supervised learning towards the goal of learning from a few examples. However, these methods should be studied in the broader context of approaches for transfer learning, model selection, active learning, and hyperparameter optimization for it to have an impact on realistic applications. Our benchmark is a step in this direction where we find the strong performance on benchmarks like CIFAR and ImageNet does not always translate to other datasets that violate assumptions implicit in the learning methods.

Self-supervised learning followed by Self-Training is a strong baseline in the absence of experts. Of surprise is the marginal gains some SSL methods provide when experts are available. Of encouragement is that the simple baseline of Self-Training from experts is robust to out-of-domain data. Moreover, no method was able to reliably use a large number of out-of-class examples in either domain despite our extensive search over model hyper-parameters and small domain shifts. Yet, the performances of these methods are far from saturated as indicated by the supervised oracle leaving much room for improvement. We hope our proposed benchmarks and results lead to new innovations in SSL.

Acknowledgements This project is supported in part by NSF #1749833 and was performed using high performance computing equipment obtained under a grant from the Collaborative R&D Fund managed by the Massachusetts Technology Collaborative.

References

[1] *2018 FGVCx Fungi Classification Challenge*. https://github.com/visipedia/fgvcx_fungi_comp. 2, 4, 5

[2] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. *ICLR*, 2019. 3

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

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009. 3

[5] 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. *ICLR*, 2020. 3

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

[7] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006. 5

[8] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Self-paced pseudo-labeling for semi-supervised learning. *AAAI*, 2021. 2, 3, 4, 6

[9] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *NeurIPS*, 2020. 2, 3, 5

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 7

[11] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 3, 5, 7

[12] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[13] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. *ICML*, 2018. 5

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 3

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

[16] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *CVPR*, 2016. 5

[17] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. *ICML*, 2019. 3

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2, 3, 5, 6

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 5

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 5

[21] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019. 3

[22] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *CVPR*, 2019. 7

[23] Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. Featmatch: Feature-based augmentation for semi-supervised learning. *ECCV*, 2020. 3

[24] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *ICLR*, 2017. 3

[25] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013. 2, 3, 4, 6

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

[27] Geoffrey J McLachlan. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70(350):365–369, 1975. 3

[28] 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. 3

[29] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 3

[30] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *NeurIPS*, 2018. 2, 3, 4, 7

[31] Chapelle Olivier, Scholkopf Bernhard, and Zien Alexander. Semi-supervised learning. 20(3):542–542, 2006. 3

[32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3, 5

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 7

[34] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *NeurIPS*, 2015. 3

[35] Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Semi-supervised learning with scarce annotations. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition Workshops, pages 762–763, 2020. 3

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1, 5

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

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

[39] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 2, 3, 6, 7

[40] Jong-Chyi Su and Subhransu Maji. Adapting models to signal degradation using distillation. In *BMVC*, 2017. 5

[41] Jong-Chyi Su and Subhransu Maji. The semi-supervised inaturalist-aves challenge at fgvc7 workshop, 2021. 1, 2, 4

[42] Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. When does self-supervision improve few-shot learning? In *ECCV*, 2020. 2, 3

[43] Antti Tarvainen and Harri Valpola. Weight-averaged, consistency targets improve semi-supervised deep learning results. *CoRR,, vol. abs/1703*, 2017, 1780. 3

[44] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *ICLR*, 2020. 3, 5

[45] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. 3

[46] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, 2018. 1, 4, 5

[47] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *International Joint Conference on Artificial Intelligence*, 2019. 3

[48] 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. 2

[49] Bram Wallace and Bharath Hariharan. Extending and analyzing self-supervised learning across domains. In *ECCV*, 2020. 2

[50] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 5

[51] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *NeurIPS*, 2020. 2, 3

[52] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020. 2, 3

[53] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019. 3

[54] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4L: Self-supervised semi-supervised learning. *ICCV*, 2019. 3

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

[56] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009. 3

[57] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005. 3

[58] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. *NeurIPS*, 2020. 3