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

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

1. Results on Semi-Aves with a different split

Since the original split of Semi-Aves was used for hyper-parameter selection, we create another split of the dataset for additional evaluation, which can be seen as a two-fold cross-validation. We first merge the images from L_{in}, U_{in}, and test sets, then randomly redistribute. The classes and the number of images of each class are kept the same in each set. The U_{out} set is also unchanged. Thus this split has roughly the same difficulty as the original split but contains new in-domain training and test images. We show the results of this split in Tab. 1, using the same hyper-parameters of the main paper. The trends are similar — Self-Training and FixMatch are both effective, but FixMatch is affected negatively by the out-of-class data.

2. SSL benchmark on the CUB dataset

Since Semi-Aves and Semi-Fungi are new datasets, here we provide another benchmark based on the widely-used Caltech-UCSD Birds-200-2011 (CUB) dataset [7]. The original class labels are sorted by the species name. Hence, we select the 100 *odd* classes as in-class species and 100 *even* classes as out-of-class species, to ensure a low domain mismatch between U_{in} and U_{out}. There are 41-60 images per class originally (disregard the original training/test split). For each in-class species, we select 5/5/10 images for L_{in}, validation set, and test set. The rest (21-40 images) are used for unlabeled data U_{in}. For out-of-class species, all the images are included in U_{out}. The statistics of the dataset split is shown in Tab. 2. Since the dataset is quite small, we did not use self-supervised learning (MoCo) for pre-training, and we use the validation set to select the best model. The results on the CUB dataset are shown in Tab. 3. In this benchmark, we can see that both Curriculum Pseudo-Label and Self-Training are helpful, even when having U_{out} included. This is potentially due to the small domain mismatch between the two sets of unlabeled data.

3. Related prior work on SSL analysis

On out-of-class unlabeled data. Oliver *et al*. [5] showed that out-of-class unlabeled data negatively impacts performance, but analysis was done on CIFAR-10 with images from 6 labeled and 4 unlabeled classes. The classes are quite different making the problem of selecting in-domain images relatively easy in comparison to fine-grained domains — in our benchmarks the out-of-class data U_{out} are other species of birds or fungi. In fact, we show that more out-of-class data helps when using self-supervised and self-training methods trained from scratch. However, the additional data does not seem to help when initialized with experts.

On transfer learning. Oliver *et al*. showed a transfer learning accuracy of 87.9% on CIFAR-10 with 4k labels, outperforming many SSL methods including PL [3] and VAT+EM [4]. Although recent results are better, the low resolution of CIFAR-10 (32×32 pixels) makes transfer learning from ImageNet less effective. On STL-10 that has a higher resolution (96×96 pixels), fine-tuning a ImageNet pre-trained ResNet-50 model on 5k labels provides 97.2% accuracy, while that trained on iNaturalist provides 95.0% accuracy. This beats 94.8% of FixMatch using 5k labeled examples when trained from scratch. Note that the iNaturalist dataset has no overlap with STL-10, yet transfer learning is effective.

4. Analysis on out-of-class unlabeled data

The effect of threshold parameter for Pseudo-Label. We found Pseudo-Label method is sensitive to the threshold parameter τ. Fig. 1 plots the accuracy as a function of τ with different unlabeled data and experts on Semi-Aves. A higher threshold performs better, especially in the presence of out-of-class data U_{out} as this excludes novel class images where the confidence of prediction is likely to be low. On the other hand, lower values work just as well when unlabeled data is in-domain U_{in}. However, this scheme only appears to work when using strong experts (*e.g*., iNat) whose confidence is likely calibrated, unlike random or ImageNet pre-trained model, where the presence of out-of-class data reduces per-

	Method	from scratch		from ImageNet		from iNat	
		Top1	Top5	Top1	Top5	Top1	Top5
	Supervised baseline	21.8	42.9	51.9	76.0	66.7	85.9
	Supervised oracle	56.0	78.7	71.9	89.4	76.7	91.2
U_{in}	Pseudo-Label [3]	18.0	37.4	54.3	79.1	66.3	86.4
U_{in}	Curriculum Pseudo-Label [1]	20.0	41.5	53.4	79.0	70.0	**88.7**
U_{in}	FixMatch [6]	24.0	47.6	**58.2**	79.6	70.4	88.2
U_{in}	Self-Training	23.7	45.1	53.9	76.8	67.6	86.4
U_{in}	MoCo [2]	28.5	54.7	52.8	79.2	69.4	88.3
U_{in}	MoCo + Self-Training	**34.0**	**58.9**	56.6	**80.1**	**71.1**	88.3
$U_{in} + U_{out}$	Pseudo-Label [3]	11.8	30.7	53.6	78.1	66.8	86.4
$U_{in} + U_{out}$	Curriculum Pseudo-Label [1]	21.3	42.1	53.8	79.4	**69.9**	**88.5**
$U_{in} + U_{out}$	FixMatch [6]	17.5	39.7	50.8	74.4	65.1	85.1
$U_{in} + U_{out}$	Self-Training	23.0	45.0	54.3	77.1	68.0	86.1
$U_{in} + U_{out}$	MoCo [2]	37.9	65.4	51.0	78.6	68.5	87.8
$U_{in} + U_{out}$	MoCo + Self-Training	**40.8**	**66.8**	**55.0**	**80.2**	68.9	87.9

Table 1. **Results on Semi-Aves benchmark with a different split.** Using the same hyper-parameters, we can see similar conclusions here as using the original split. Overall, Self-Training and FixMatch are effective but out-of-class data often hurts the performance.

split →	L_{in}	val	test	U_{in}	U_{out}
#images →	500	500	1000	3853	5903

Table 2. **Number of images in the CUB benchmark.**

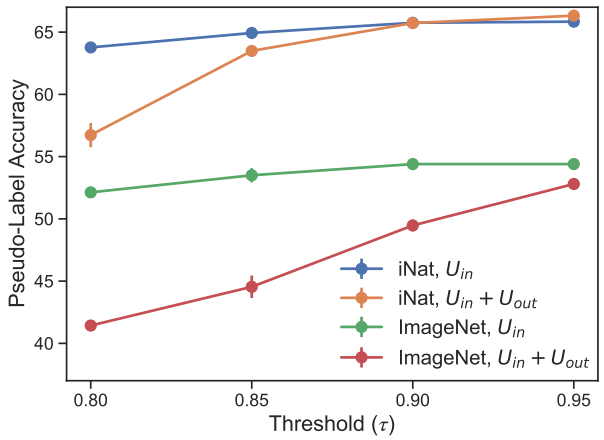

Figure 1. **Pseudo-label with different threshold τ.** Pseudo-label is sensitive to the threshold hyperparameter. The negative impact of out-of-class unlabeled data is reduced by increasing the threshold, yet when the initial performance is low the scheme is not effective as seen by the performance of the ImageNet pre-trained model.

formance. This poses a practical problem for this method — increasing the threshold increases robustness but reduces the amount of unlabeled data that is used during training.

The effect of out-of-class unlabeled data. To see how the domain mismatch between U_{in} and U_{out} can affect SSL methods, we analyze the predictions of the unlabeled data. We use the supervised model trained on L_{in} to compute the predictions of the unlabeled data on the Semi-Aves dataset. We plot the histogram of the maximum probability and the entropy of the predictions of U_{in} and U_{out} in Fig. 2 (left and middle). We also plot the distribution of the distillation loss, which is calculated between the supervised model (teacher) and the ImageNet pre-trained model (student), with a temperature $T = 1$ (Fig. 2 right). This is in the beginning of the self-training process and the last layer of the student model is randomly initialized. Overall, the model is generally more uncertain about the out-of-class data, which often has a higher entropy or a smaller maximum probability. The distillation loss on U_{in} is also often higher than that of U_{out}, suggesting the model focuses more on those from U_{in} during training. However, there is still a good amount of data from U_{out} having a high maximum probability, which has a negative impact for pseudo-label methods.

5. Implementation details of FixMatch

For FixMatch, we used the official Tensorflow code and a PyTorch re-implementation for our experiments. The PyTorch version did reproduce the results on CIFAR-10 with 4000 labels (95.68% vs 95.74% reported in the paper). We found the optimization details such as learning rate, batch size, and number of epochs to be crucial for FixMatch. Due to the limitation of our resources, we can only use a batch size of 64 for labeled data (and 320 for unlabeled data), where the original paper used a batch size of 1024 (and 5120 for unlabeled data). Since there is small discrepancy between the two implementations, we reported the best result among the two for each setting: Tensorflow version for training from scratch and PyTorch version when using expert models.

	Method	from scratch		from ImageNet		from iNat	
		Top1	Top5	Top1	Top5	Top1	Top5
	Supervised baseline	11.1	27.4	58.7	85.8	77.3	94.2
	Supervised oracle	68.5	87.8	84.5	97.1	90.0	98.0
U_{in}	Pseudo-Label [3]	13.5	**32.1**	57.0	85.3	78.3	95.7
U_{in}	Curriculum Pseudo-Label [1]	**14.5**	31.4	57.3	84.7	80.1	**96.7**
U_{in}	FixMatch [6]	10.7	26.6	53.2	79.8	**81.6**	95.2
U_{in}	Self-Training	12.6	29.8	**61.3**	**86.3**	80.6	95.8
$U_{in} + U_{out}$	Pseudo-Label [3]	11.9	30.8	59.1	86.1	77.7	94.8
$U_{in} + U_{out}$	Curriculum Pseudo-Label [1]	**12.9**	**32.3**	59.6	**86.5**	**81.2**	**96.8**
$U_{in} + U_{out}$	FixMatch [6]	10.7	27.1	52.8	81.7	78.6	95.7
$U_{in} + U_{out}$	Self-Training	12.2	29.2	**61.4**	85.9	79.9	96.0

Table 3. **SSL benchmark on the CUB dataset.** In this benchmark, we can see both Curriculum Pseudo-Label and Self-Training are helpful, even with out-of-class unlabeled data.

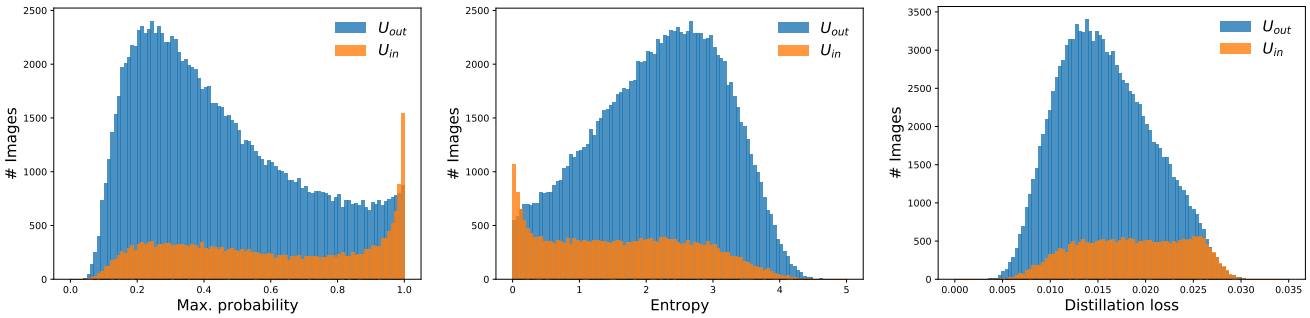

Figure 2. **Predictions of unlabeled data using a supervised model.** We plot the distribution of the predictions of data from U_{in} and U_{out}. Specifically, we plot the maximum probability of the class predictions (left), entropy of the predictions (middle), and the distillation loss between the teacher and student model before the training starts (right). Unlabeled data from the same distribution tend to have a higher maximum probability, a lower entropy, or a higher distillation loss.

6. Value for computing

Among the SSL methods, Pseudo-Label requires the least amount of computation, but it does not uniformly lead to improvements in our benchmark. Curriculum Pseudo-Label trains a model several times (six in our implementation), hence is more expensive, though the performance saturates after the first few iterations. FixMatch requires more training epochs and is the most time-consuming comparing to other SSL methods, but the performance is the best when having expert model with in-class unlabeled data only. Self-training only needs two rounds of training, one for training the teacher model and one for the student. However, it is often the best and is more robust to out-of-class data.

References

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

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

[3] 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. 1, 2, 3

[4] 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

[5] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *NeurIPS*, 2018. 1

[6] 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

[7] 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. 1