Supplemental Materials: Towards Professional Level Crowd Annotation of Expert Domain Data

Pei Wang
UC San Diego
pew062@eng.ucsd.edu

Nuno Vasconcelos
UC San Diego
nuno@ucsd.edu

In this supplement, we show the details and additional experiment results that are not presented in the main paper due to the page limitation

A. Experimental Implementation Details

Dataset: Various fine-grained vision datasets are used: CUB [22], Fungi [16], Butterflies [11] and Gulls [19]. In [16], CUB [22] with 200 bird species is re-organized for Semi-supervised Learning (SSL). The labeled training set has 500 examples from 100 classes (5 examples per class). The unlabeled set has $3,885$ in-class examples [1] and 5903 out-class examples by considering the remaining 100 classes of CUB as novel. Fungi has 200 classes, consisting of $4,141$ labelled and $13,166$ in-class and $64,871$ out-class unlabeled images which has $1,193$ novel classes[2]. This dataset is more difficult because of its long-tailed property. Butterflies and Gulls are two datasets of small class cardinality, with only 5 classes, and 300 (150) labeled images, $1,244$ (431) unlabeled images for Butterflies (Gulls). Our results are based on the test sets of [16, 19] with thrice repeated experiments. Both datasets were subject to standard normalizations. Training images were first randomly resized to 224×224 and then randomly flipped, whereas testing images were first resized to 256×256 and then center-cropped to 224×224. All images were also first converted to $[0.0, 1.0]$ from $[0, 255]$ and then normalized by subtracting the mean $[0.485, 0.456, 0.406]$ and dividing by the standard deviation $[0.229, 0.224, 0.225]$ of each RGB color channel.

Network: For fair comparison with [16, 19], we use ResNet-18 on Butterflies and Gulls, and ResNet-50 on CUB and Fungi if not otherwise stated. The models are pretrained on ImageNet [2], except for Butterflies where training is from scratch. This follows the setting of [17, 19] because two of the butterfly categories are in ImageNet. We used the training setups of [16] on CUB and Fungi[3] and [19] on Butterflies and Gulls[4]. The deliberative explanations and compared Grad-CAM are generated using [14, 18]. We tuned the threshold on the heat map such that 5% image size is remained for visualization, which follows the setting of [18–20].

Crowd-sourcing: Amazon Mechanical Turk is used[5]. The interface is given in Figure 2 of the paper. The per image reward is $0.01 across all our experiments. We did not limit the maximum number that per turker can work on. Statistically, each worker completed 21.1 query image annotation tasks on average and the maximum is 135.

In our budget-aware experiments, the cost of an expert is harder to determine and can vary significantly with the application area, e.g. doctors tend to be more expensive than botanists. We tried to identify a lower bound for the cost, in a domain of mild expertise. For this, we asked MTurkers to take a survey, declaring if they were specialists on birds or fungi. To answer the survey, they were shown 3 images of birds or fungi. Those who felt confident about their ability to do the classification, were then asked the expected per image reward, for labeling images from 100 candidate classes. Four options were given: $< \$0.1$, $\$0.1 - \$0.5$, $\$0.5 - \$1.0$, and $> \$1$. We gathered 5 results for birds and 3 for fungus. One person chose $\$0.5 - \$1.0$ and all others chose $> \$1$, showing that the task is considered difficult. We thus use $1 as cost estimate for expert labeling. This can be thought as a lower bound, although it is unrealistically low for many image domains.

B. Support Set Ablations

Sample choice of the positive support sets We consider four strategies to select the examples of support set $\mathcal{S}_{\hat{y}} \subset \mathcal{D}^l_{\hat{y}}$, based on the predicted posterior probability $f_{\hat{y}}(\mathbf{x})$ of class $\hat{y}$ given example $\mathbf{x}$. Strategy S1 is to choose the

[1] This number is from the data released on project link https://github.com/cvl-umass/ssl-evaluation, which is slightly different from the paper $(3,853)$

[2] This number is different from 1194 on the project page of https://github.com/cvl-umass/ssl-evaluation, because the class 'Inocybe rimosa' is repetitively indexed and we fixed this problem.

[3] https://github.com/cvl-umass/ssl-evaluation

[4] https://github.com/peiwang062/MEMORABLE

[5] https://www.mturk.com/

	Lab. Acc.	Cla. Acc.
Softmax	68.7	65.9
Softmax+attributive	71.4	67.1
Softmax+deliberative	74.3	68.6

Table A. Ablation study for support sets.

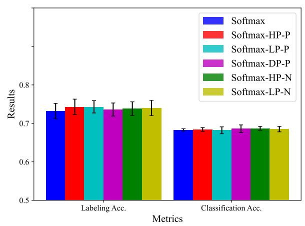


Figure A. Result comparison of different support set sample choices.

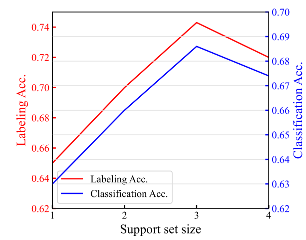


Figure B. Result comparison of different support set sizes.

examples of K highest probabilities $f_{\hat{y}}(\mathbf{x})$ ('Softmax-HP-P'). These are the easiest to assign to class $\hat{y}$ and include the most representative class features. Strategy S2 is to choose examples with the K lowest top-probability $f_{\hat{y}}(\mathbf{x})$ ('Softmax-LP-P'). These are harder and more likely to be outliers for class $\hat{y}$, including features that are rarely visible, occlusions, or other variations. Strategy S3 is to select a set of examples with diverse probability $f_{\hat{y}}(\mathbf{x})$ ('Softmax-DP-P'). This means the selected examples have more diverse features. Finally, Strategy S4 is to select the examples randomly ('Softmax'), which is used as baseline. Figure A compares the results. As we mentioned in the paper, we have found no big difference between these strategies and just used randomly selection.

Sample choice of the negative support sets For $\mathcal{S}^c_{\hat{y}}$, similarly to $\mathcal{S}_{\hat{y}}$, we experimented with the highest-probability ('Softmax-HP-N'), lowest top-probability ('Softmax-LP-N'), and random example, again finding that these strategies make no big difference. Figure A shows the results as well.

The size of support sets The support set size K is ablated from 1 to 4. Figures B shows that with just one image both annotation and classification accuracies are weak. Both accuracies improve for larger K saturating at about $K = 3$. This likely reflects the fact that too many images can be distracting or even confusing.

Explanations We investigate the importance of explanations, comparing attributive explanations based on Grad-CAM [14], ('w Grad-CAM')[6] and the proposed deliberative explanations ('w deliberative'), with results on Table A. The baseline 'Softmax' is the setting only having the support set but no explanations, corresponding to the D in Table 1 of the paper. Overall, although Grad-CAM enables a clear improvement, the proposed deliberative explanations have the largest benefit.

[6] On Grad-CAM experiments, a slightly different description for circled regions for turkers is given, "The circle regions may have some class-specific features, which might be helpful for your identification."

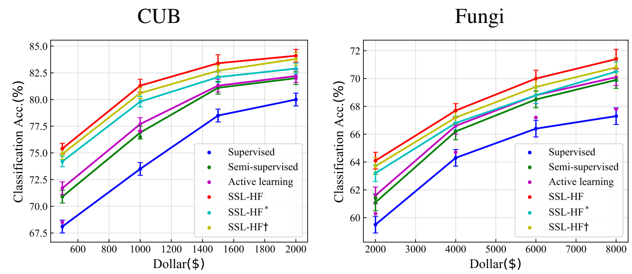


Figure C. The trade-off comparison of supervised/SSL/SSL-HF.

C. Comparison to Other Implementations of SSL-HF

We also experimented on other two different settings of SSL-HF. (1) SSL-HF* where three annotations are collected per image and majority voting is used to decide on the final label. (2) SSL-HF† where the "agree" examples are recyclable and with replacement like SSL methods [8, 15], i.e., in Algorithm 1, eliminating step 14 and 15 and the classifier is updated by $f^t \leftarrow \arg\min_f \mathcal{R}_{\mathcal{D}^l \cup \mathcal{L}^t}(f)$. The budget-aware results are compared in Figure C. Obviously, they are inferior to the original SSL-HF because repetitive annotations lead to great cost increase even if there are some slightly accuracy improvements.

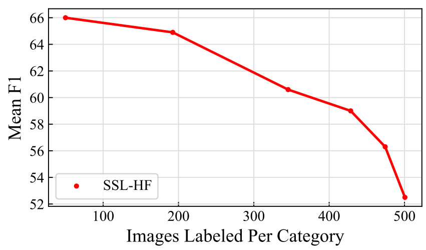
D. Comparison to Crowd Source Methods

As discussed in the related work section, SSL-HF is partly inspired by Tropel [13] which was proposed for binary detection. Following the setup of [12], we compare to Tropel and Mullapudi *et al* [12] in Figure D. Here, since the source code and experimental details of [12] are not available online, it is difficult to reprodcue their results. Because the results in [12] are reported graphically, we do not have the original data even we have tried to reach out to the authors, but did not get the response. We are unable to compare the result on the same figure and have to attach the screenshotted Figure 3(a) of [12] lower to our result for reference. We are not to make any conclusion because the comparison might be unfair because of some of implementation details of [12] are unknown.

E. Enhancement by Other Techniques

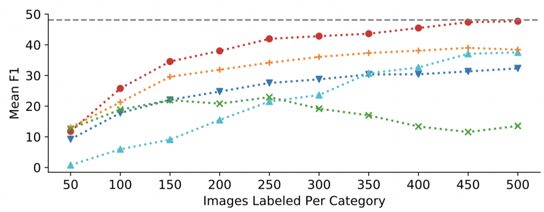
Since SSL-HF is a general solution for SSL problem, it is versatile to a variety of machine learning techniques. In this section, we take some of them as an example to investigate if SSL-HF can benefit from them. The results are summarized in Table B. The benchmark of Table 2 of the paper is used.

Confidence calibration Because the unlabeled samples are added progressively based on their confidence estimation in Algorithm 1, to calibrate the confidence is important and potentially able to lead to better performance. Temperature



(a) SSL-HF

Approach		Pseudo Positives	Pseudo Negatives	BG Splitting	Train Features	Active Learning
Ours	○	✗	✓	✓	✓	✓(adaptive)
KD-Semi	+	✓	✓	✓	✓	✓(adaptive)
DeepProp	×	✓	✓	✓	✓	✓(adaptive)
Tropel-Deep	△	✗	✗	✗	✓	✓(most-likely-positive)
Tropel	▽	✗	✗	✗	✗	✓(most-likely-positive)
Fully supervised		✗	✗	✓	✓	✗



(b) Other methods

Figure D. F1 score comparison to state of the arts crowd source methods. The bottom figure is a screenshot from [12] because source codes and implementation details are not available. Because of a screenshot, 'Ours' refers to [12].

scaling [4] is adopt to validate this idea. The temperature of softmax is tuned to be an optimal value of 1.2. We found there is a stable improvement.

Architecture SSL-HF can also benefit from more advanced architectures. Vision transformer (ViT) 'vit_b_16' from [3] is used as an example. The supervised baseline of training only on the expert-labeled set is $82.4(0.3)$. When training with additional unlabeled examples, SSL-HF can still result in a significant improvement, 3.3% on CUB.

Noisy labeling training Since the pseudo-labels are noisy, further performance improvements can in principle be accrued by training with noisy label learning algorithms [1, 9, 21]. However, as shown in the table, things went contrary to our wishes. This is opposite to the observation in [19] where noisy labeling training is found to able to enhance the classifier performance trained on the machine teaching annotated labels. We think there might be two reasons. First, on [19], the datasets, Butterflies and Gulls, are relatively easier in the sense that only five classes exist. These algorithms are easily to fit the noisy labels, but not true on more complicated datasets. Second, there is a gap of noise mode. On [19], the noisy labels are entirely provided by humans. This matches the evaluation metrics in the literature where noise is generated by randomly replacing the ground truth labels with other possible labels [6, 21] or similar classes defined by humans [1, 10, 23]. However, for SSL-HF, the

Method	CUB
Baseline	68.6 (0.6)
w. Confidence calibration [4]	69.9 (0.4)
w. ViT [3]*	85.7 (0.2)
w. DivideMix [9]	42.6 (0.6)

Table B. The enhancement by different methods. *When using ViT architecture, the supervised baseline of training only on the expert-labeled set is 82.4 (0.3).

Method	Cars
Baseline	30.2 (0.7)
Pseudo-Label [8]	30.9 (0.3)
Self-Training [16]	31.5 (0.4)
AL [5]	33.7 (1.1)
SSL-HF	37.3 (0.6)

Table C. Comparison with SSL and AL on Cars

noise is generated by the classifier. In fact, the drop is sensible because if an improvement exists, it would be a free lunch and can be embedded into the classifier training. But in reality there is no literature on it.

F. Evaluation on More Domains

We also evaluated SSL-HF on Cars [7] following the same setting as in [16]. SSL-HF still has a large gain.

References

[1] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*, pages 1062–1070. PMLR, 2019.

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

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2020.

[4] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

[5] Alex Holub, Pietro Perona, and Michael C Burl. Entropy-based active learning for object recognition. In *CVPR Workshops*, pages 1–8. IEEE, 2008.

[6] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2304–2313. PMLR, 2018.

[7] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*.

[8] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop*, volume 3, page 896, 2013.

[9] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *ICLR*, 2020.

[10] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[11] Oisin Mac Aodha, Shihan Su, Yuxin Chen, Pietro Perona, and Yisong Yue. Teaching categories to human learners with visual explanations. In *CVPR*, pages 3820–3828, 2018.

[12] Ravi Teja Mullapudi, Fait Poms, William R Mark, Deva Ramanan, and Kayvon Fatahalian. Learning rare category classifiers on a tight labeling budget. In *ICCV*, pages 8423–8432, 2021.

[13] Genevieve Patterson, Grant Van Horn, Serge Belongie, Pietro Perona, and James Hays. Tropel: Crowdsourcing detectors with minimal training. In *AAAI*, volume 3, 2015.

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

[15] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *NeurIPS*, 33:596–608, 2020.

[16] Jong-Chyi Su, Zezhou Cheng, and Subhransu Maji. A realistic evaluation of semi-supervised learning for fine-grained classification. In *CVPR*, pages 12966–12975, 2021.

[17] Pei Wang, Kabir Nagrecha, and Nuno Vasconcelos. Gradient-based algorithms for machine teaching. In *CVPR*, pages 1387–1396, 2021.

[18] Pei Wang and Nuno Vasconcelos. Deliberative explanations: visualizing network insecurities. *NeurIPS*, 32, 2019.

[19] Pei Wang and Nuno Vasconcelos. A machine teaching framework for scalable recognition. In *ICCV*, pages 4945–4954, October 2021.

[20] Pei Wang and Nuno Vasconcelos. A generalized explanation framework for visualization of deep learning model predictions. *IEEE T-PAMI*, 2023.

[21] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *CVPR*, pages 8688–8696, 2018.

[22] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[23] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *CVPR*, pages 7017–7025, 2019.