ProSelfLC: Progressive Self Label Correction for Training Robust Deep Neural Networks

We thank all reviewers for their detailed comments.

Reviewer #1: Weak accept

1. "There aren't really any very original theoretical contribution, but rather a theoretically consistent dissertation."
Personally, we see two novelties: (a) The balance control between predictions and annotations; (b) Re-defining low-entropy status and defending entropy minimisation.

2. "(i) Everything has to be written in a relative way ...
(ii) What are the drawbacks of ProSelfLC? ..."
No problem. (i) We will check and write everything in a relative way. (ii) A subsection for discussing the drawbacks of ProSelfLC will be added. For example, one hyper-parameter B is designed to control the growth speed of self trust score. The decision of B is task-dependent and searched according to validation performance.

Reviewer #2: Borderline

1. "(i) It is unclear whether some works have studied trust adaption instead of using the fixed trust (via epsilon). If so, these works (more relevant to the proposed method) should be elaborated and experimentally compared.
(ii) In Section 2, self knowledge distillation methods were mentioned. However, none of them have been included in experiments. The current description of existing works (especially those used in experiments) is less self- contained."
(i) As far as we know, Joint-soft did trust adaption via stage-wise training, which is compared in Table 6.
(ii) First, self KD methods [49, 54, 55] maximise the consistency of intraclass images' predictions or the consistency of different classifiers. In our view, they do not modify labels, thus being less relevant for comparison. While Boot-soft and Joint-soft belong to self KD methods and are compared. Second, the two-stage self KD method [53] can be an add-on (brother) other than a competitor. E.g., in real-world practice, we think that the first stage can be ProSelfLC instead of CCE with early stopping. We keep it for future work to exploit ProSelfLC in stage-wise algorithms.

2. "Similarity structure is frequently mentioned but less understandable because it was described in too abstract way."
We will make it more explicit and clear. A probability vector $\mathbf{p} \in \mathbb{R}^C$ is treated as an instance-to-classes similarity vector, i.e., $\mathbf{p}(j|\mathbf{x})$ is a measurement of how much a data point $\mathbf{x}$ is similar with (analogously, likely to be) j-th class. Consequently, $\mathbf{p}$ should not be exactly one-hot, and is proposed to be corrected at training, so that it can define a more informative and structured learning target.

3. "Table 2 is hard to understand and there lacks enough text description on it. Also, the meanings of horizontal and vertical axes of Figures 2 and 3 were not clearly described."
We will address these and make them clear in the revision.

4. "(i) It is more convincing to compare performance of merely using global trust or local trust without using both.
(ii) How to search/specify optimal parameters in the proposed method should be more clearly described.
(iii) It is necessary to have multiple runs and report the mean and standard deviation of the results. "
(i) Figure 2 displays the experimental results: When only $g(t)$ is used, both label noise correction and testing generalisation are worse than the full version, but better than the fixed ϵ; When only $l(\mathbf{p})$ is used without considering training time, the model cannot learn, i.e., the convergence fails.
(ii) Table 5 shows the results of different Bs. Here, we note that when noise rate is higher, a smaller B performs better. We will describe these more clearly in the revision.
(iii) First, we have runs of multiple Bs, e.g., in Table 3. Generally, the result deviation of random seeds would not be higher than that of Bs. Second, we show the training dynamics in Figures 2, 3 and 4, where we see that the convergence is stable across methods and scenarios. Third, when $B = 10$, we run 3 times on ImageNet 2012. Their results are 76.0, 76.0 and 75.9. We can add them in the revision.

Reviewer #3: Weak accept

1. "The approach may suffer from a positive-feedback amplification of early errors. ... It is likely that some of these highly confident predictions are wrong."
Great question! First of all, ProSelfLC alleviates this issue a lot and makes a model confident in correct predictions. Please look at Fig. 3(e) together with 3(b) and 3(c). ***Fig. 3(e) shows the confidence of predictions, whose majority are correct according to Fig. 3(b) and 3(c).*** In Fig. 3(b), ProSelfLC fits noisy labels least, i.e., around 12% so that the correction rate of noisy labels is about 88% in Fig. 3(c). Nonetheless, we agree that ProSelfLC is non-perfect. A few noisy labels are memorised with high confidence.

2. "It is good to report model calibration, e.g. ECE ... "
On CIFAR-100 test set, we can report ECE (%, #bins=10) of ours (and CCE in brackets), as a complement of Fig. 3. For confidence metrics (CMs), we try probability, entropy, and their scaled variants using different Ts. Though ECE is sensitive to CM and T, ours are smaller than CCE.

Scaling: logits/T	$T = 1$	$T = 1/4$	$T = 1/8$
CM = $\max_j \mathbf{p}(j\|\mathbf{x})$	15.71 (40.98)	4.24 (18.27)	**2.39** (9.94)
CM = $1 - \mathrm{H}(\mathbf{p})/\mathrm{H}(\mathbf{u})$	17.38 (42.83)	5.22 (17.84)	**2.66** (9.53)

3. "Approach is heuristically motivated, not theoretically."
We agree the proposed target modification is heuristic and human-inspired. Unfortunately, the "target" concept is intrinsically heuristic. E.g., human-annotated targets are heuristic too. Being heuristic makes them easy to interpret.

4. "The method introduces additional hyperparameters..."
As in Tables 3 and 5, other baselines have ϵ while we have B. Therefore, our hyperparameters are not more than them.