A. Proof of propositions

Proposition 3. *Compared with CCE, LS and CP penalise entropy minimisation while LC reward it.*
Proof. We can rewrite CCE, LS, CP, and LC from the viewpoint of KL divergence:

$$L_{\mathrm{CCE}}(\mathbf{q},\mathbf{p}) = \mathrm{H}(\mathbf{q},\mathbf{p}) = \mathrm{KL}(\mathbf{q}||\mathbf{p}) + \mathrm{H}(\mathbf{q},\mathbf{q}) = \mathrm{KL}(\mathbf{q}||\mathbf{p}), \tag{8}$$

where we have $\mathrm{H}(\mathbf{q},\mathbf{q}) = 0$ because $\mathbf{q}$ is a one-hot distribution.

$$\begin{aligned} L_{\mathrm{CCE+LS}}(\mathbf{q},\mathbf{p};\epsilon) &= (1-\epsilon)\mathrm{KL}(\mathbf{q}||\mathbf{p}) + \epsilon\mathrm{KL}(\mathbf{u}||\mathbf{p}) + \epsilon\mathrm{H}(\mathbf{u},\mathbf{u}) \\ &= (1-\epsilon)\mathrm{KL}(\mathbf{q}||\mathbf{p}) + \epsilon\mathrm{KL}(\mathbf{u}||\mathbf{p}) + \epsilon\cdot\mathrm{constant}, \end{aligned} \tag{9}$$

$$\begin{aligned} L_{\mathrm{CCE+CP}}(\mathbf{q},\mathbf{p};\epsilon) &= (1-\epsilon)\mathrm{KL}(\mathbf{q}||\mathbf{p}) - \epsilon(\mathrm{H}(\mathbf{p},\mathbf{u}) - \mathrm{KL}(\mathbf{p}||\mathbf{u})) \\ &= (1-\epsilon)\mathrm{KL}(\mathbf{q}||\mathbf{p}) + \epsilon\mathrm{KL}(\mathbf{p}||\mathbf{u}) - \epsilon\cdot\mathrm{constant}, \end{aligned} \tag{10}$$

where $\mathrm{H}(\mathbf{p},\mathbf{u}) = \mathrm{H}(\mathbf{u},\mathbf{u}) = \mathrm{constant}$. Analogously, LC in Eq (5) can also be rewritten:

$$L_{\mathrm{CCE+LC}}(\mathbf{q},\mathbf{p};\epsilon) = (1-\epsilon)\mathrm{KL}(\mathbf{q}||\mathbf{p}) - \epsilon\mathrm{KL}(\mathbf{p}||\mathbf{u}) + \epsilon\cdot\mathrm{constant}. \tag{11}$$

In LS and CP, both $+\mathrm{KL}(\mathbf{u}||\mathbf{p})$ and $+\mathrm{KL}(\mathbf{p}||\mathbf{u})$ pulls $\mathbf{p}$ towards $\mathbf{u}$. While in LC, the term $-\mathrm{KL}(\mathbf{p}||\mathbf{u})$ pushes $\mathbf{p}$ away from $\mathbf{u}$. □

Proposition 4. *In CCE, LS and CP, a data point* $\mathbf{x}$ *has the same semantic class. In addition,* $\mathbf{x}$ *has an identical probability of belonging to other classes except for its semantic class.*
Proof. In LS, the target is $\tilde{\mathbf{q}}_{\mathrm{LS}} = (1-\epsilon)\mathbf{q}+\epsilon\mathbf{u}$. For any $0 \le \epsilon < 1$, the semantic class is not changed, because $1-\epsilon+\epsilon*\frac{1}{C} > \epsilon*\frac{1}{C}$. In addition, $j_1 \neq y, j_2 \neq y \Rightarrow \tilde{\mathbf{q}}_{\mathrm{LS}}(j_1|\mathbf{x}) = \tilde{\mathbf{q}}_{\mathrm{LS}}(j_2|\mathbf{x}) = \frac{\epsilon}{C}$.

In CP, $\tilde{\mathbf{q}}_{\mathrm{CP}} = (1-\epsilon)\mathbf{q} - \epsilon\mathbf{p}$. In terms of label definition, *CP is against intuition because these zero-value positions in* $\mathbf{q}$ *are filled with negative values in* $\tilde{\mathbf{q}}_{\mathrm{CP}}$. A probability has to be not smaller than zero. So we rephrase $\tilde{\mathbf{q}}_{\mathrm{CP}}(y|\mathbf{x}) = (1-\epsilon) - \epsilon*\mathbf{p}(y|\mathbf{x})$, and $\forall j \neq y, \tilde{\mathbf{q}}_{\mathrm{CP}}(j|\mathbf{x}) = 0$ by replacing negative values with zeros, as illustrated in Figure 1a. □

B. Discussions on wrongly confident predictions and model calibration

1. *It is likely that some highly confident predictions are wrong. Will ProSelfLC suffer from an amplification of those errors?*

First of all, ProSelfLC alleviates this issue a lot and makes a model confident in correct predictions, according to Figure 3e together with 3b and 3c. ***Figure 3e shows the confidence of predictions, whose majority are correct according to Figure 3b and 3c.*** In Figure 3b, ProSelfLC fits noisy labels least, i.e., around 12% so that the correction rate of noisy labels is about 88% in Figure 3c. Nonetheless, ProSelfLC is non-perfect. A few noisy labels are memorised with high confidence.

2. *How about the results of model calibration using a computational evaluation metric: Expected Calibration Error (ECE) [30, 11]?*

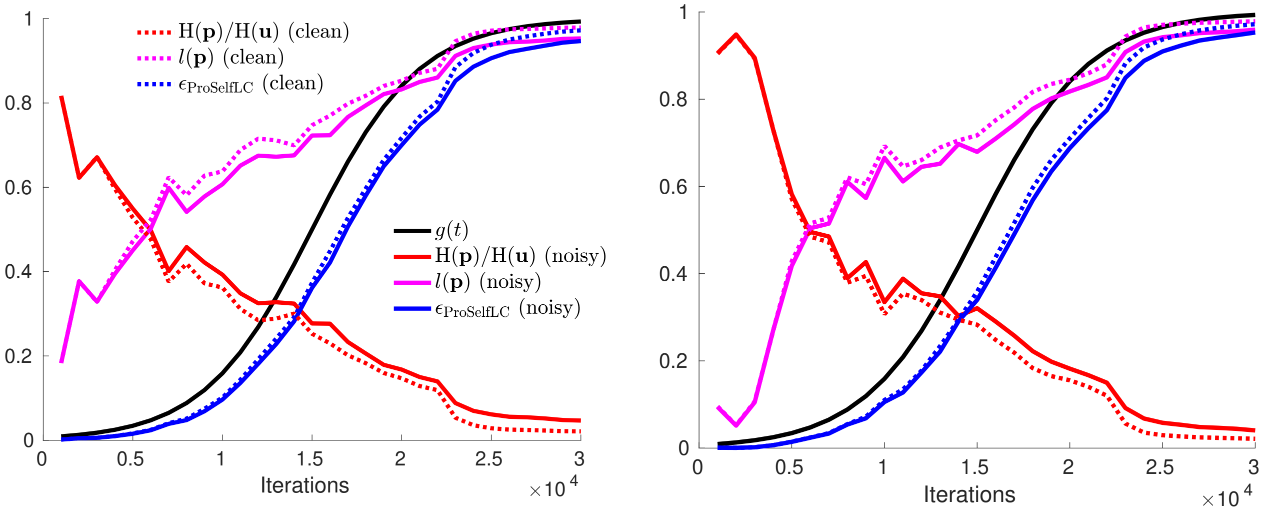
Following the practice of [11], on the CIFAR-100 test set, we report the ECE (%, #bins=10) of ProSelfLC versus CCE, as a complement of Figure 3. For a comparison, CCE's results are shown in corresponding brackets. We try several confidence metrics (CMs), including probability, entropy, and their temperature-scaled variants using a parameter T. Though the ECE metric is sensitive to CM and T, ProSelfLC's ECEs are smaller than CCE's.

Table 7: ECE results of multiple combinations of logits scaling (logits$/T$) and confidence metrics (probability and entropy).

Scaling logits with a temperature parameter T: 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)

C. The changes of entropy statistics and $\epsilon_{\mathrm{ProSelfLC}}$ at training

In Figure 5, we visualise how the entropies of noisy and clean subsets change at training.



(a) Asymmetric label noise rate = 20%.

(b) Asymmetric label noise rate = 40%.

Figure 5: The changes of entropy statistics and $\epsilon_{\mathrm{ProSelfLC}}$ at training. We store a model every 1000 iterations to monitor the learning process. For data-dependent metrics, after training, we split the corrupted training data into clean and noisy subsets according to the information about how the training data is corrupted before training. Finally, we report the mean results of each subset.