Self-Relaxed Joint Training: Sample Selection for Severity Estimation with Ordinal Noisy Labels –Supplementary Materials–

Algorithm 2 Proposed framework with JoCor [34]

1: **Input:** Dataset $\tilde{\mathcal{D}}$, two networks f_1 and f_2 with initialized weights $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, learning rate η, noise rate ϵ, epoch T' and $T_{\max}$, iteration $t_{\max}$, temperature τ;
for $T = 1, 2, \dots, T_{\max}$ **do**
  2: **Shuffle** training set $\tilde{\mathcal{D}}$;
  for $t = 1, \dots, t_{\max}$ **do**
    3: **Fetch** mini-batch $\tilde{\mathcal{B}}$ from $\tilde{\mathcal{D}}$;
    4: **Select** clean samples from $\tilde{\mathcal{B}}$ by $\mathcal{L}_{\mathrm{h}}^{\mathrm{JoCor}}$ (with τ);
      $\mathcal{B} \leftarrow \arg\min_{\mathcal{B}':|\mathcal{B}'| \geq R(T)|\tilde{\mathcal{B}}|} \mathcal{L}_{\mathrm{h}}^{\mathrm{JoCor}}(f_1, f_2, \mathcal{B}')$;
    5: **Derive** soft labels $\boldsymbol{l}_{\mathrm{s}}$ from $\boldsymbol{l}_{\mathrm{h}}$ for $\mathcal{B}_1, \mathcal{B}_2$ by Eq.(3);
    6: **Update** networks;
      $\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}_1 - \eta \nabla \mathcal{L}_{\mathrm{s}}^{\mathrm{JoCor}}(f_1, f_2, \mathcal{B})$;
      $\boldsymbol{\theta}_2 \leftarrow \boldsymbol{\theta}_2 - \eta \nabla \mathcal{L}_{\mathrm{s}}^{\mathrm{JoCor}}(f_1, f_2, \mathcal{B})$;
  end
  7: **Update** $R(T) \leftarrow 1 - \min\left\{\frac{T}{T'}\epsilon, \epsilon\right\}$;
end
8: **Output:** two trained networks with $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$.

A. Applying our framework to other joint-training methods

In our paper, we detailed Algorihtm 1, where our framework is applied to Co-teaching [8]. However, our framework is versatile and can be applicable to other joint-training methods, such as JoCor [34] and CoDis [36]. Algorithms 2 and 3 show the entire training procedure of "JoCor + Ours" and "CoDis + Ours," respectively.

Algorithm 2 of "JoCor + Ours" has a very similar structure as Algorihtm 1; however, its loss functions $\mathcal{L}_{\mathrm{h}}^{\mathrm{JoCor}}$ and $\mathcal{L}_{\mathrm{s}}^{\mathrm{JoCor}}$ are different from $\mathcal{L}_{\mathrm{h}}$ and $\mathcal{L}_{\mathrm{s}}$, respectively. JoCor [34] uses the common clean sample set $\mathcal{B}$ for the two networks and introduces co-regularization to reduce divergence between the networks. Consequently, $\mathcal{L}_{\mathrm{h}}^{\mathrm{JoCor}}$ becomes:

$$\mathcal{L}_{\mathrm{h}}^{\mathrm{JoCor}}(f_1, f_2, \tilde{\mathcal{B}}) = (\mathcal{L}_{\mathrm{h}}(f_1, \tilde{\mathcal{B}}) + \mathcal{L}_{\mathrm{h}}(f_2, \tilde{\mathcal{B}})) + \lambda \mathcal{L}_{\mathrm{reg}}(f_1, f_2, \tilde{\mathcal{B}}), \quad (6)$$

Algorithm 3 Proposed framework with CoDis [36]

1: **Input:** Dataset $\tilde{\mathcal{D}}$, two networks f_1 and f_2 with initialized weights $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, learning rate η, noise rate ϵ, epoch T' and $T_{\max}$, iteration $t_{\max}$, temperature τ;
for $T = 1, 2, \dots, T_{\max}$ **do**
  2: **Shuffle** training set $\tilde{\mathcal{D}}$;
  for $t = 1, \dots, t_{\max}$ **do**
    3: **Fetch** mini-batch $\tilde{\mathcal{B}}$ from $\tilde{\mathcal{D}}$;
    4: **Select** clean samples from $\tilde{\mathcal{B}}$ by $\mathcal{L}_{\mathrm{h}}^{\mathrm{CoDis}}$ (with τ);
      $\mathcal{B}_1 \leftarrow \arg\min_{\mathcal{B}':|\mathcal{B}'| \geq R(T)|\tilde{\mathcal{B}}|} \mathcal{L}_{\mathrm{h}}^{\mathrm{CoDis}}(f_1, f_2, \mathcal{B}')$;
      $\mathcal{B}_2 \leftarrow \arg\min_{\mathcal{B}':|\mathcal{B}'| \geq R(T)|\tilde{\mathcal{B}}|} \mathcal{L}_{\mathrm{h}}^{\mathrm{CoDis}}(f_2, f_1, \mathcal{B}')$;
    5: **Derive** soft labels $\boldsymbol{l}_{\mathrm{s}}$ from $\boldsymbol{l}_{\mathrm{h}}$ for $\mathcal{B}_1, \mathcal{B}_2$ by Eq.(3);
    6: **Update** networks;
      $\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}_1 - \eta \nabla \mathcal{L}_{\mathrm{s}}(f_1, \mathcal{B}_2)$;
      $\boldsymbol{\theta}_2 \leftarrow \boldsymbol{\theta}_2 - \eta \nabla \mathcal{L}_{\mathrm{s}}(f_2, \mathcal{B}_1)$;
  end
  7: **Update** $R(T) \leftarrow 1 - \min\left\{\frac{T}{T'}\epsilon, \epsilon\right\}$;
end
8: **Output:** two trained networks with $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$.

where $\mathcal{L}_{\mathrm{reg}}$ is a regularization term:

$$\mathcal{L}_{\mathrm{reg}}(f_1, f_2, \tilde{\mathcal{B}}) = \sum_{\{\boldsymbol{x}_i, \tilde{y}_i\} \in \tilde{\mathcal{B}}} J(\boldsymbol{p}_1(\boldsymbol{x}_i), \boldsymbol{p}_2(\boldsymbol{x}_i)), \quad (7)$$

and $J(\cdot,\cdot)$ denotes the Jeffrey divergence (i.e., the symmetrized Kullback-Leibler (KL) divergence). For updating the models with soft labels, "JoCor+Ours" uses the loss function $\mathcal{L}_{\mathrm{s}}^{\mathrm{JoCor}}$ obtained by replacing $\mathcal{L}_{\mathrm{h}}$ with $\mathcal{L}_{\mathrm{s}}$ in Eq. (6).

Algorithm 3 of "CoDis + Ours" has a more elaborated structure than Algorihtm 1; CoDis [36] uses possibly clean samples that have high discrepancy prediction probabilities between two networks, f_1 and f_2. The proposed framework with CoDis selects small loss samples with the loss

Table 7. Classification results on LIMUC with Truncated-Gaussian noise. Following tradition, the test accuracy (Acc.), mean absolute error (MAE), and macro F1 (mF1) are averaged over the last ten epochs. The mean and standard deviations of five-fold cross-validation are shown. The best and second-best results are highlighted in red and blue, respectively. For plugin settings, improved results are shown by **bold**.

Method	Noise rate: $\epsilon = 0.2$			Noise rate: $\epsilon = 0.4$		
	Acc.↑	MAE↓	mF1↑	Acc.↑	MAE↓	mF1↑
Standard	0.665±0.010	0.373±0.007	0.573±0.007	0.566±0.018	0.489±0.018	0.479±0.011
Sord [5]	0.708±0.009	0.309±0.010	0.632±0.015	0.632±0.016	0.389±0.019	0.564±0.024
Label-smooth [23]	0.690±0.010	0.339±0.010	0.601±0.016	0.609±0.016	0.432±0.017	0.511±0.007
F-correction [25]	0.670±0.009	0.362±0.010	0.585±0.010	0.609±0.008	0.430±0.008	0.529±0.010
Reweight [18]	0.667±0.006	0.371±0.008	0.573±0.013	0.575±0.008	0.477±0.008	0.494±0.013
Mixup [9]	0.676±0.008	0.359±0.005	0.583±0.011	0.605±0.012	0.449±0.015	0.490±0.013
CDR [38]	0.674±0.012	0.362±0.007	0.582±0.016	0.571±0.027	0.482±0.027	0.481±0.015
Garg [7]	0.657±0.054	0.433±0.146	0.447±0.128	0.525±0.040	0.786±0.121	0.267±0.015
Co-teaching [8]	0.698±0.002	0.332±0.004	0.610±0.012	0.646±0.020	0.393±0.023	0.544±0.023
Co-teaching + Ours	**0.731±0.005**	**0.289±0.005**	**0.646±0.014**	**0.677±0.019**	**0.356±0.019**	**0.545±0.011**
JoCor [34]	0.720±0.006	0.306±0.006	0.633±0.008	0.690±0.015	0.345±0.017	0.573±0.007
JoCor + Ours	**0.731±0.009**	**0.287±0.010**	**0.642±0.018**	0.678±0.016	0.353±0.017	0.549±0.009
CoDis [36]	0.694±0.004	0.336±0.005	0.609±0.013	0.622±0.012	0.418±0.013	0.530±0.014
CoDis + Ours	**0.723±0.005**	**0.294±0.006**	**0.639±0.017**	**0.684±0.012**	**0.342±0.015**	**0.581±0.028**

Table 8. Results of LIMUC dataset with Truncated-Gaussian noise under different loss usages for sample selection and updating. The best and second-best results are highlighted in red and blue, respectively.

Selection	Updating	Noise rate: $\epsilon = 0.2$			Noise rate: $\epsilon = 0.4$		
		Acc.↑	MAE↓	mF1↑	Acc.↑	MAE↓	mF1↑
hard	hard	0.698±0.002	0.332±0.004	0.610±0.012	0.646±0.020	0.393±0.023	0.544±0.023
soft	soft	0.722±0.006	0.300±0.008	0.628±0.019	0.661±0.021	0.382±0.024	0.489±0.021
hard	soft	0.731±0.005	0.289±0.005	0.646±0.014	0.677±0.019	0.356±0.019	0.545±0.011

Table 9. Results of private UC dataset with Truncated-Gaussian noise under different loss usages for sample selection and updating.

Selection	Updating	Noise rate: $\epsilon = 0.2$			Noise rate: $\epsilon = 0.4$		
		Acc.↑	MAE↓	mF1↑	Acc.↑	MAE↓	mF1↑
hard	hard	0.788±0.009	0.236±0.008	0.599±0.031	0.702±0.022	0.328±0.020	0.490±0.036
soft	soft	0.809±0.007	0.209±0.006	0.611±0.028	0.721±0.030	0.318±0.030	0.442±0.038
hard	soft	0.815±0.010	0.202±0.008	0.621±0.035	0.748±0.031	0.282±0.033	0.491±0.032

function:

$$\mathcal{L}_{\mathrm{h}}^{\mathrm{CoDis}}(f_1, f_2, \tilde{\mathcal{B}}) = \mathcal{L}_{\mathrm{h}}(f_1, \tilde{\mathcal{B}}) - \lambda \mathcal{L}_{\mathrm{reg}}(f_1, f_2, \tilde{\mathcal{B}}). \quad (8)$$

For updating the models with soft labels, "CoDis + Ours" uses the loss function $\mathcal{L}_{\mathrm{s}}$.

B. Experimental evaluations under the Truncated-Gaussian noise

Table 7 shows the results on LIMUC [26] under the Truncated-Gaussian noise, simulating the case that experts make the mis-labelings between the neighboring labels. (Specifically, the i,jth element of the label transition matrix, P_{ij}, takes $1-\rho$ for $|i-j| = 1$ and $P_{ij} = 0$ for $|i-j| > 1$.) Our methods ("$*$ + Ours") outperform the others. Compared to the results under the Quasi-Gaussian noise, the individual accuracies in Table 7 are slightly lower, which is the same trend seen in the results on our private dataset in Section 4.2.

Tables 8 and 9 show how the combination of $\mathcal{L}_{\mathrm{h}}$ and $\mathcal{L}_{\mathrm{s}}$ is appropriate for learning with ordinal noisy labels un-

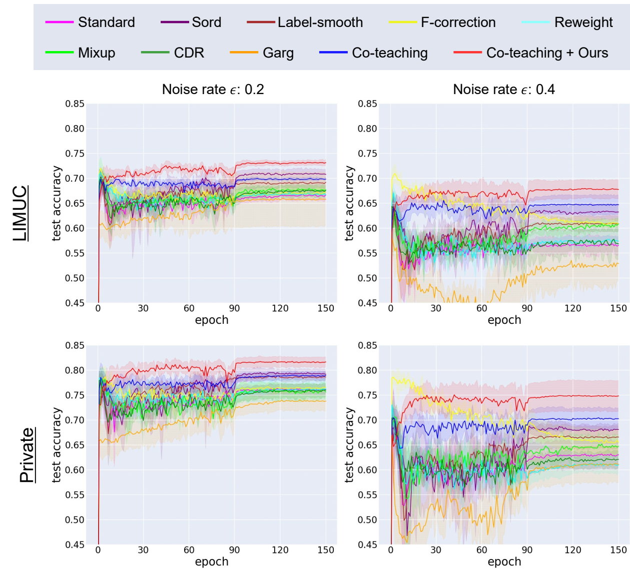

Figure 4. Test accuracy curves. The width of the shading indicates the standard deviation in cross-validation.

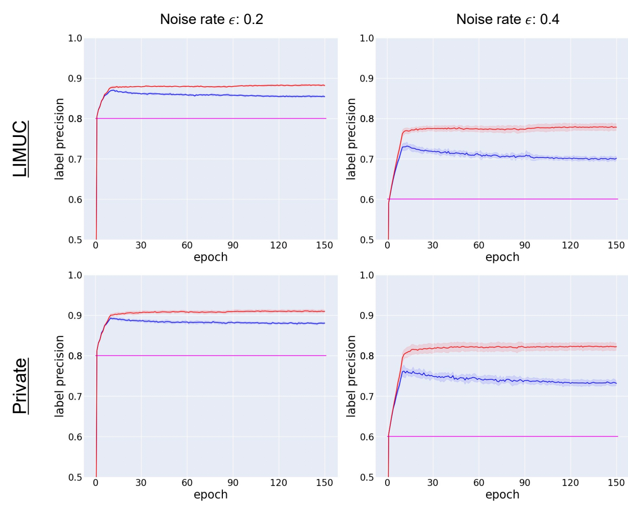

Figure 5. Label precision curves. The blue and red curves show the label precisions by “Co-teaching” and “Co-teaching + Ours,” respectively. The pink horizontal line shows $(1 - \epsilon)$.

der Truncated-Gaussian. These tables show the results for LIMUC and the private dataset, respectively. The tendency of the results is almost the same as those under the Quasi-Gaussian noise, shown in Tables 4 and 5.

Fig. 4 shows the test accuracy curves for the individual methods on two UC datasets with Truncated-Gaussian noise. The comparative methods show a sharp increase in their test accuracy in early epochs. Then, the comparative models often start “memorizing” the samples with incorrect labels. Our method (“Co-teaching + Ours”) could avoid the memorization effect.

Fig. 5 shows the change in label precision on two UC datasets with Truncated-Gaussian noise. The backbone method is Co-teaching. The pink horizontal lines $(1 - \epsilon)$ indicate the label precision under random sample selection. Our method (“Co-teaching + Ours,” the red curve) shows far better label precisions than random selection (pink line) and Co-teaching (the blue curve).

C. Code availability

We share our codes for experiments at `https://github.com/shumpei-takezaki/Self-Relaxed-Joint-Training`.