Evidential Deep Partial Label Learning to Quantify Disambiguation Uncertainty

Supplementary Material

A. Fundamental Derivations

A.1. Expectations of Cross Entropy on Dirichlet Distribution

According to Eq.(3), the cross entropy loss is,

$$\mathcal{L}_{PLL}(\Theta) = \mathbb{E}_{\boldsymbol{p}_i \sim \mathrm{Dir}(\boldsymbol{\alpha}_i)} \left[-\sum_{j=1}^{Q} r_{ij} y_{ij} \log(p_{ij}) \right] \tag{12}$$

where $y_{ij} \in \{0, 1\}$ denotes whether the label y_j is present in the candidate labels ('1') or the non-candidate labels ('0'). r_{ij} represents y_j as the weight of the true label. Therefore, we can simplify the above equation,

$$\mathcal{L}_{PLL}(\Theta) = -\sum_{j=1}^{Q} r_{ij} y_{ij} \cdot \mathbb{E}_{\boldsymbol{p}_i \sim \mathrm{Dir}(\boldsymbol{\alpha}_i)} \left[\log(p_{ij}) \right] \tag{13}$$

Based on the logarithmic expectation property of Dirichlet, it can be inferred that,

$$\mathbb{E}\left[\log(p_{ij}) \right] = \psi(\alpha_{ij}) - \psi(K_i) \tag{14}$$

Meanwhile, the y_{ij} corresponding to non-candidate label is 0. Therefore, we can obtain,

$$\begin{aligned} \mathcal{L}_{PLL}(\Theta) &= \mathbb{E}_{\boldsymbol{p}_i \sim \mathrm{Dir}(\boldsymbol{\alpha}_i)} \left[-\sum_{j=1}^{Q} r_{ij} y_{ij} \log(p_{ij}) \right] \\ &= \int \left[\sum_{j=1}^{Q} -r_{ij} y_{ij} \log\left(p_{ij}\right) \right] \frac{1}{B\left(\boldsymbol{\alpha}_i\right)} \prod_{j=1}^{Q} p_{ij}^{\alpha_{ij}-1} d\boldsymbol{p}_i \\ &= \sum_{j \in S_i} r_{ij} y_{ij} \left(\psi\left(S_i\right) - \psi\left(\alpha_{ij}\right) \right) \end{aligned} \tag{15}$$

A.2. Expectation of The Non-Candidate Labeled Dirichlet Distribution

First, we need to calculate the expectation of the non-candidate labels under the Dirichlet distribution, as follows,

$$\mathcal{L}_{non}(\Theta) = -\sum_{j \notin S_i} \mathbb{E}_{\boldsymbol{p}_j \sim \mathrm{Dir}(\boldsymbol{\alpha}_j)} \left[\log(1 - \boldsymbol{p}_j) \right] \tag{16}$$

Due to $\boldsymbol{p}_j \sim \mathrm{Dir}(\boldsymbol{\alpha}_j)$, the expectation in the above equation is non-standard and cannot be directly written in a closed form. However, Taylor's formula can be used to approximate $\log(1 - \boldsymbol{p}_j) \approx -\boldsymbol{p}_j - \boldsymbol{p}_j^2/2$, then

$$\mathbb{E}[\log(1 - \boldsymbol{p}_j)] \approx -\mathbb{E}[\boldsymbol{p}_j] - \frac{1}{2}\mathbb{E}[\boldsymbol{p}_j^2] \tag{17}$$

where $\mathbb{E}[\mathbf{p}_j] = \boldsymbol{\alpha}_j / K$ and $\mathbb{E}[\mathbf{p}_j^2] = \boldsymbol{\alpha}_j(\boldsymbol{\alpha}_j + 1)/K(K+1)$. Then, we can get,

$$\begin{aligned} \mathcal{L}_{non}(\Theta) &= -\sum_{j \notin S_i} \mathbb{E}_{\boldsymbol{p}_j \sim \mathrm{Dir}(\boldsymbol{\alpha}_j)} \left[\log(1 - \boldsymbol{p}_j) \right] \\ &\approx \sum_{j \notin S_i} \left[\frac{\boldsymbol{\alpha}_j}{K} + \frac{1}{2} \cdot \frac{\boldsymbol{\alpha}_j(\boldsymbol{\alpha}_j + 1)}{K(K+1)} \right] \end{aligned} \tag{18}$$

A.3. Theoretical analysis of ED-PLL from the EM perspective

The algorithm proposed in this article follows the *Expectation Maximization* (EM) algorithm. In **E-step**, assign a belief mass to each class through learning to measure evidence of disambiguation. In **M-step**, use the label confidence weights learned from the previous E-step to maximize likelihood. The detailed proof is as follows,

In **E-step**, we compute the posterior expectation of latent variables (the label confidence weight $\mathbf{r}^t$) given current parameters Θ^t. Specifically, the expected based on EDL for PLL can be expressed as,

$$\begin{aligned} &\arg\min_{\Theta} \sum_{i=1}^{N} \sum_{j \in S_i} y_{ij} \left(\psi\left(K_i\right) - \psi\left(\alpha_{ij}\right) \right) \geq \\ &\arg\min_{\Theta} \sum_{i=1}^{N} \sum_{j \in S_i} r_{ij} y_{ij} \left(\psi\left(K_i\right) - \psi\left(\alpha_{ij}\right) \right) \end{aligned} \tag{19}$$

Therefore, Eq. (3) can be optimized more closely to the true loss by directly using candidate labels, which is beneficial for updating model parameters. Given current model parameters $\Theta^{(t-1)}$, we compute the label confidence weights $r_{ij}^{(t)}$ for $y_j \in S_i$ as an estimation of $P(y_i = j \mid x_i, \Theta^{(t-1)})$. This is done by a neighborhood-smoothing strategy:

$$\mathbf{r}_i^t = \begin{cases} \frac{1}{k+1}\left(\mathbf{y}_i + \sum_{j \in \mathcal{N}_i} \mathbf{y}_j \right) & \text{when } t = 0 \\ \mathrm{softmax}(\mathbf{r}_i^{t-1} \boldsymbol{\alpha}_i / K + \mathbf{y}_i) & \text{otherwise} \end{cases} \tag{20}$$

where r_{ij} represents the responsibility of candidate label y_j being the true label at iteration t.

Table 4. Detail Information of the Real-World datasets

Data	Features	Examples	MEA	Labels	Scenarios
MSRCv2	48	1758	3.16	23	*Object Classification*
BirdSong	38	4998	2.18	13	*Bird Song Classification*
Lost	108	1122	2.33	16	*Automatic Face Naming*
Yahoo! News	163	22991	1.91	219	*Automatic Face Naming*
Soccer Player	279	17472	2.09	171	*Automatic Face Naming*

In **M-step**, we update the model parameters Θ^t by minimizing the evidential deep partial label loss in Eq. (11), which corresponds to maximizing the expected complete-data log-likelihood under the current label confidence weight $\mathbf{r}^t$. According to Eq.(4), we update $\mathbf{r}^t$. When $t = 0$, we perform preliminary disambiguation on candidate labels based on the relationship between similar instances. When $t > 0$, due to the memory performance of the neural network, we can learn simple instances to guide label disambiguation. Therefore, through the update of $\mathbf{r}^t$, the model learning can be guided. Therefore, we can get,

$$\begin{aligned} &\arg\min_{\Theta} \sum_{i=1}^{N} \sum_{j \in S_i} r_{ij}^{t-1} y_{ij} \left(\psi\left(K_i\right) - \psi\left(\alpha_{ij}\right)\right) \geq \\ &\arg\min_{\Theta} \sum_{i=1}^{N} \sum_{j \in S_i} r_{ij}^{t} y_{ij} \left(\psi\left(K_i\right) - \psi\left(\alpha_{ij}\right)\right) \end{aligned} \tag{21}$$

$$\mathcal{L}_{PLL}(\Theta^{t-1}) \geq \mathcal{L}_{PLL}(\Theta^{t}) \tag{22}$$

Therefore, during the training process of ED-PLL, the true labels are gradually recognized, and the refined labels in turn help to improve the classification ability, guiding the model to gradually approach the ground-true labels.

B. Experimental setup and details

B.1. Experiments Details of Partially Labeled Datasets

Tab. 4 summarizes the details of the above real-world datasets, where MEA represents the mean number of ambiguous labels in the candidate label set.

B.2. Compared Methods

To demonstrate the superiority of the proposed ED-PLL algorithm, we conducted comparisons with state-of-the-art PLL methods, including deep learning based on PLL methods PRODEN [24], LW [30], RC [14], CC [14], CAVL [40], PiCO [28] and DIRK [31]. According to the corresponding literature, the parameters of these comparable methods are set as follows:

- PRODEN [24]: Model updating and label disambiguation are done step-by-step using an incremental recognition approach. [suggested configuration: Training batch(200), learning rate (10^{-3}), weight decay (10^{-5})].
- LW [30]: The trade-off between candidate label loss and non-candidate label loss is considered using leverage weighted loss. [suggested configuration: Training batch(200), learning rate (10^{-3}), weight decay (10^{-5})].
- RC [14]: A risk-consistent method using importance reweighting strategy. [suggested configuration: Training batch(200), learning rate (10^{-3}), weight decay (10^{-5})].
- CC [14]: a classifer-consistent method applying the cross entropy loss and transition matrix to form an empirical risk estimator. [suggested configuration: Training batch(200), learning rate (10^{-3}), weight decay (10^{-5})].
- CAVL [40]: Truth labels are identified using class activation graphs and their promotion form class activation values to improve partial label learning, and ultimately real labels are progressively identified through the intrinsic representation of the model. [suggested configuration: Learning rate (10^{-3}), training batch(250), weight decay (10^{-3})].
- PiCO [28]: Utilizing the idea of learning by contrast for representation learning and ultimately disambiguation through prototype updating. [suggested configuration: Learning rate (10^{-2}), training batch(500), weight decay (10^{-3})].
- DIRK [31]: A partial label learning algorithm synergizing knowledge distillation and contrastive learning to improve model robustness. [suggested configuration: Learning rate (10^{-2}), training batch(200), weight decay (10^{-3})].

In order to further verify the effectiveness of the ED-PLL, this paper replaces the loss function in PiCO and DIRK with the ED-PLL loss proposed in this paper to generate algorithms PiCO-ED and DIRK-ED. The parameter settings in PiCO-ED and DIRK-ED are the same as in the original paper.

B.3. Analysis of Disambiguation Uncertainty

To more intuitively measure the uncertainty of disambiguation, we analyzed the **CIFAR-10**, **MNIST**, **K-MNIST**, and

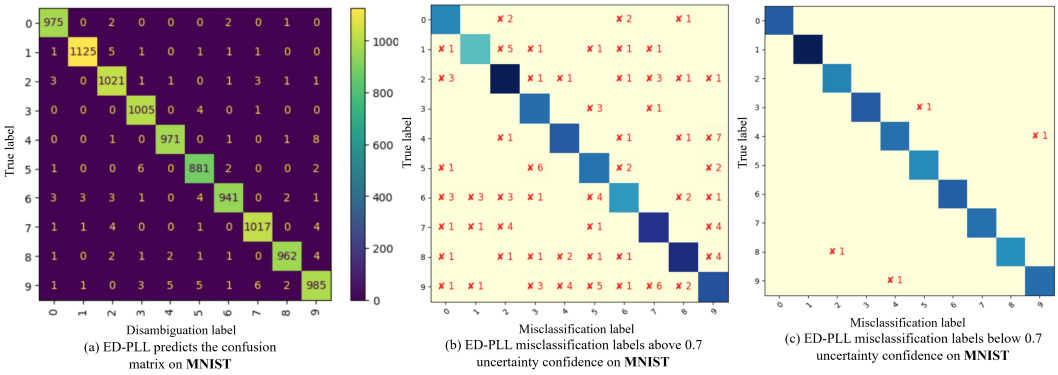


Figure 7. The impact of uncertainty analysis on disambiguation accuracy **MNIST** dataset.

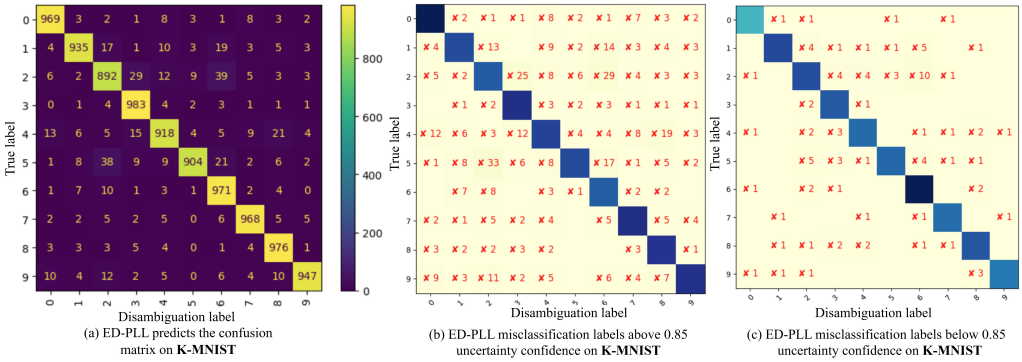


Figure 8. The impact of uncertainty analysis on disambiguation accuracy **K-MNIST** dataset.

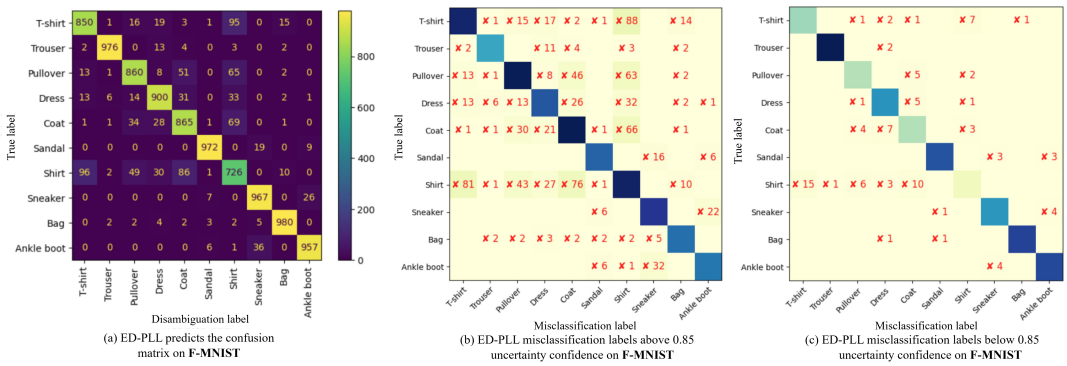
F-MNIST datasets with $q = 0.3$. As shown in Fig. 7(b), the results of false disambiguation can be effectively selected, and the number of misclassifications in Fig. 2(c) can be significantly reduced to improve the effectiveness of the model. Meanwhile, as shown in Fig. 10, Fig. 11 and Fig. 12, we further visualize the disambiguation results of ED-PLL, where the first row represents the low uncertainty confidence disambiguation results. It can be observed that the model can achieve accurate classification when faced with given ambiguous labels. For instance, with high uncertainty confidence in the second row, it can be seen that the data itself is very vague, making it difficult to distinguish the original features of the instance. Meanwhile, in the datasets of **CIFAR-10**, **K-MNIST**, and **F-MNIST** datasets, we can further verify the superiority of ED-PLL, which can not only effectively disambiguate, but also screen out a large number of false disambiguation labels through uncertainty analysis.

B.4. Uncertainty Quantification of ED-PLL.

To further demonstrate the effectiveness of the ED-PLL algorithm proposed in this article, we conducted reliability measurements on the **K-MNIST** dataset. Therefore, we conducted a preliminary analysis of the output uncertainty associated with various PLL methods, as illustrated in Fig. 13. The results indicate the ED-PLL not only effectively reduces the impact of label ambiguity, but also makes the model more robust and reliable, thereby improving the credibility of the model output and achieving competitive *expected calibration error* (ECE) scores. Specifically, our method achieves the lowest ECE score among the comparison methods on **K-MNIST**. The analysis reveals a near-perfect alignment between actual accuracy and average consistency, both of which closely follow the ideal calibration line. The extremely low expected calibration error (ECE=0.0039) demonstrates excellent calibration performance. Meanwhile, compared with PiCO and DIRK, PiCO-ED and DIRK-ED obtained lower ECE scores, which further demonstrates the reliability of ED-PLL.

B.5. Convergence analysis.

To prove the convergence of ED-PLL more intuitively, this section analyzes the convergence curves of all algorithms on the **MNIST**, **K-MNIST**, and **F-MNIST** datasets with $q = 0.3$. Fig. 14 and Fig. 15 demonstrate the effective disambiguation capability of ED-PLL. As learning iterations increase, the model asymptotically converges to a steady-state value, experimentally validating its convergence. Comprehensive results further confirm that ED-PLL combines excellent disambiguation performance with ro-



(a) ED-PLL predicts the confusion matrix on **F-MNIST**

(b) ED-PLL misclassification labels above 0.85 uncertainty confidence on **F-MNIST**

(c) ED-PLL misclassification labels below 0.85 uncertainty confidence on **F-MNIST**

Figure 9. The impact of uncertainty analysis on disambiguation accuracy **F-MNIST** dataset.

Figure 10. Visualization of disambiguation results of ED-PLL algorithm on MNIST dataset, $q = 0.3$.

bust stability.

B.6. Parameter Sensitivity Analysis.

In Algorithm 1, the ED-PLL involves main parameters. i.e. the balance coefficient α, the balance coefficient β, and the number of epoch. As shown in Fig. 16, the performance of ED-PLL first gets better and better as the number of iterations increases, and then does not change anymore when it approaches a certain value. Therefore, we set the epoch to 200 in the comparative algorithm based on PLL loss improvement, and set the epoch to 500 in the process of improving the structure based on the PLL model. Figure 12 demonstrates how ED-PLL performs under other different parameter configurations. For simplification, only the **MNIST** and **MSRCv2** datasets are picked out to show the performance of ED-PLL as the parameters change. As one parameter varies, the other two parameters are set as the default value, i.e, $\alpha = 0.8$ and $\beta = 0.5$. As α varies from 0 to 1 in Figure 12, the best performance of ED-PLL occurs at $\alpha = 0.8$. It is seen that the performance of ED-PLL reaches its best when $\beta = 0.5$. It also shows that the conflict-aware regularization can further improve disambiguation performance by reducing the impact of unreliable results within the intra-class.

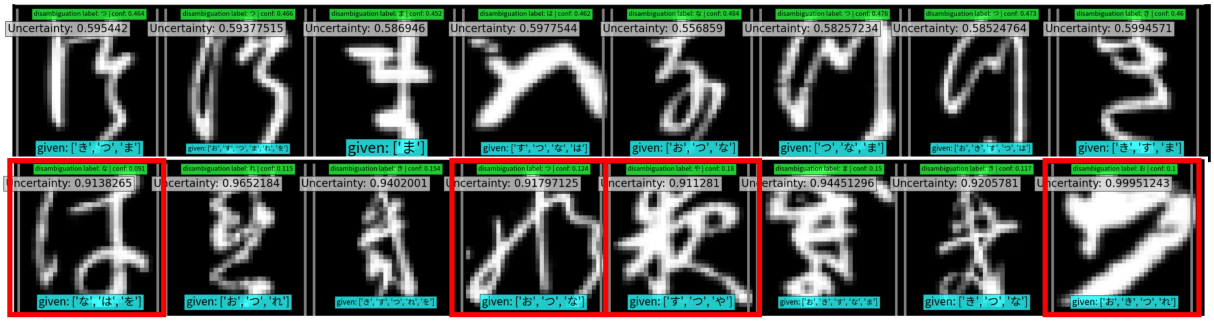


Figure 11. Visualization of disambiguation results of ED-PLL algorithm on Kuzushiji-MNIST dataset, $q = 0.3$.

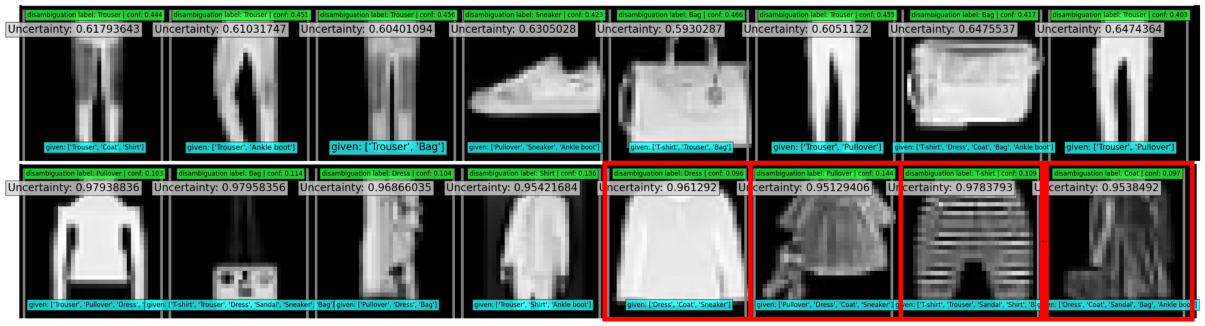


Figure 12. Visualization of disambiguation results of ED-PLL algorithm on Fashion-MNIST dataset, $q = 0.3$.

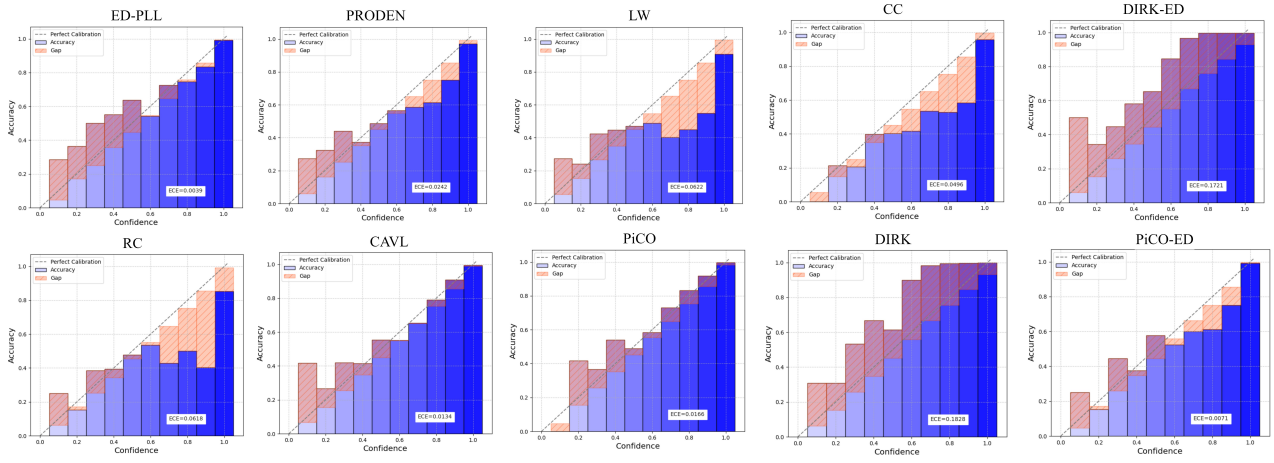


Figure 13. Reliability diagram and expected calibration error on **K-MNIST**.

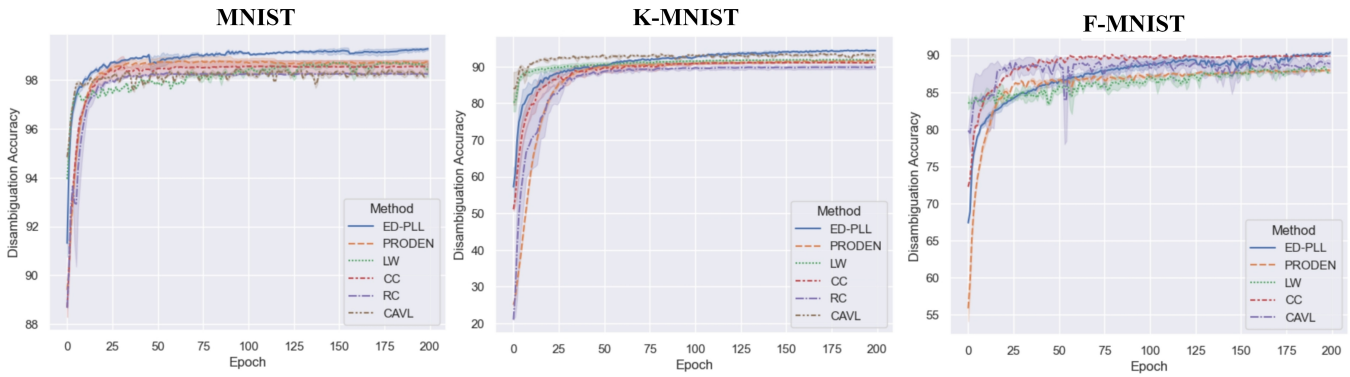


Figure 14. Convergence curves of PLL based on loss improvement.

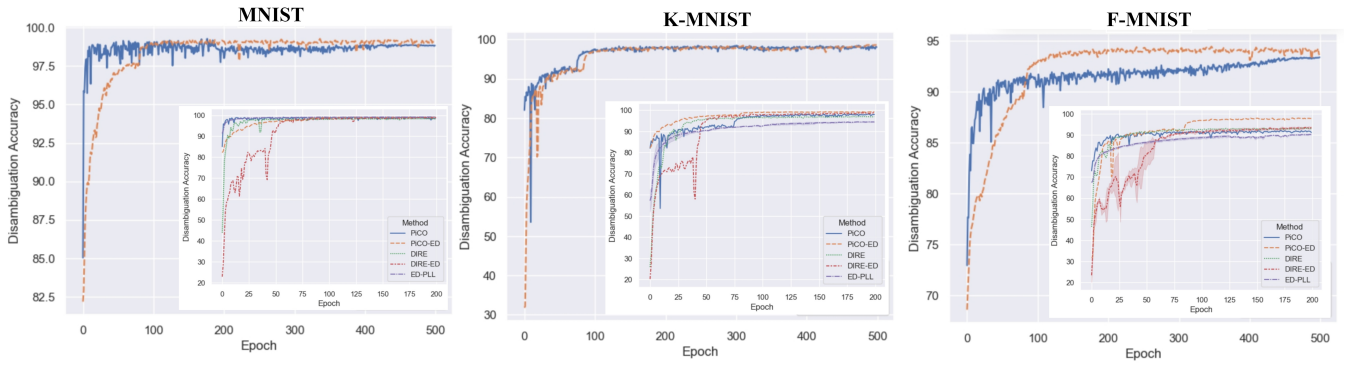


Figure 15. Convergence curves of PLL based on loss improvement.

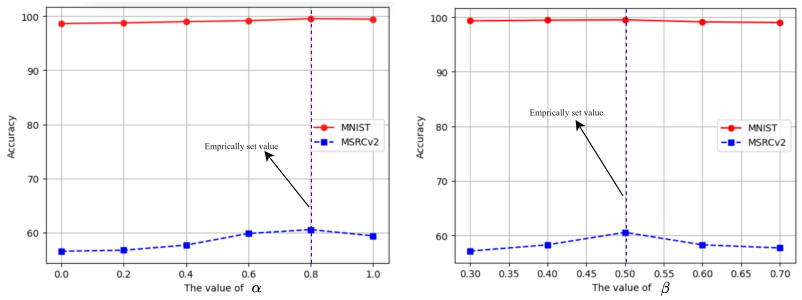


Figure 16. The accuracy of ED-PLL on the **MNIST** and **MSRCv2** datasets under different parameter configurations.