Appendix

A.1 Proof of Proposition 1.

As $\gamma = \frac{\eta \max_j(\mathbf{T}^n_j(\boldsymbol{x}))}{\eta \max_j(\mathbf{T}^n_j(\boldsymbol{x}))+\min_i(\mathbf{T}^c_i(\boldsymbol{x}))}$, for $\eta > 1$, we can obtain the following equation:

$$\min_i(\gamma \mathbf{T}^c_i(\boldsymbol{x})) = \eta \max_j((1-\gamma)\mathbf{T}^n_j(\boldsymbol{x})),$$

where $i \in \mathcal{S}$ and $j \notin \mathcal{S}$. Based on the Eq. (4), it is clear that

$$\min_i(\gamma \mathbf{T}^c_i(\boldsymbol{x})) = \min_i \widetilde{\mathbf{T}}_i(\boldsymbol{x}), \text{ and}$$

$$\max_j((1-\gamma)\mathbf{T}^n_j(\boldsymbol{x})) = \max_j \widetilde{\mathbf{T}}_j(\boldsymbol{x}).$$

Then for $\eta > 1$, we have the following iniequality:

$$\min_i \widetilde{\mathbf{T}}_i(\boldsymbol{x}) = \eta \max_j \widetilde{\mathbf{T}}_j(\boldsymbol{x}) > \max_j \widetilde{\mathbf{T}}_j(\boldsymbol{x})$$

A.2 Derivation of Proposition 2.

We first consider the gradient of $\mathcal{L}_{\texttt{DIRK}}$ with respect to a logit value δ_i, which is denoted as:

$$\frac{\partial \mathcal{L}_{\texttt{DIRK}}}{\partial \delta_i} = -\widetilde{\mathbf{T}}_i(\boldsymbol{x}) \cdot \frac{1}{\mathbf{S}_i(\boldsymbol{x})} \frac{\partial \mathbf{S}_i(\boldsymbol{x})}{\partial \delta_i} + \sum_{j=1, j\neq i}^{c} (-\widetilde{\mathbf{T}}_j(\boldsymbol{x}) \cdot \frac{1}{\mathbf{S}_j(\boldsymbol{x})} \frac{\partial \mathbf{S}_j(\boldsymbol{x})}{\partial \delta_i}),$$

where $\partial \mathbf{S}_i(\boldsymbol{x})/\partial \delta_i = \mathbf{S}_i(\boldsymbol{x})(1-\mathbf{S}_i(\boldsymbol{x}))$ and $\partial \mathbf{S}_j(\boldsymbol{x})/\partial \delta_i = -\mathbf{S}_i(\boldsymbol{x})\mathbf{S}_j(\boldsymbol{x})$. Then we have

$$\begin{aligned} \frac{\partial \mathcal{L}_{\texttt{DIRK}}}{\partial \delta_i} &= \mathbf{S}_i(\boldsymbol{x}) - \widetilde{\mathbf{T}}_i(\boldsymbol{x}) \\ &= \begin{cases} \mathbf{S}_i(\boldsymbol{x}) - \gamma \mathbf{T}^c_i(\boldsymbol{x}), & \text{if } i \in \mathcal{S}, \\ \mathbf{S}_i(\boldsymbol{x}) - (1-\gamma)\mathbf{T}^n_i(\boldsymbol{x}), & \text{otherwise.} \end{cases} \end{aligned}$$

Then, under the assumption that $\mathbf{S}_i(\boldsymbol{x}) - \widetilde{\mathrm{T}}_i(\boldsymbol{x}) > 0$ for $i \notin \mathcal{S}$, the L_1 norm of the gradient of $\mathcal{L}_{\texttt{DIRK}}$ is obtained as follows:

$$\begin{aligned} \sum_i \left| \frac{\partial \mathcal{L}_{\texttt{DIRK}}}{\partial \delta_i} \right| &= \sum_{i\in\mathcal{S}} \left| \frac{\partial \mathcal{L}_{\texttt{DIRK}}}{\partial \delta_i} \right| + \sum_{i\notin\mathcal{S}} \left| \frac{\partial \mathcal{L}_{\texttt{DIRK}}}{\partial \delta_i} \right| \\ &= \sum_{i\in\mathcal{S}'} \left| \frac{\partial \mathcal{L}_{\texttt{DIRK}}}{\partial \delta_i} \right| + \sum_{i\in\mathcal{S}''} \left| \frac{\partial \mathcal{L}_{\texttt{DIRK}}}{\partial \delta_i} \right| + \sum_{i\notin\mathcal{S}} \left| \frac{\partial \mathcal{L}_{\texttt{DIRK}}}{\partial \delta_i} \right| \\ &= \sum_{i\in\mathcal{S}'} \left(\gamma \mathrm{T}^c_i(\boldsymbol{x}) - \mathrm{S}_i(\boldsymbol{x}) \right) + \sum_{i\in\mathcal{S}''} \left(\mathrm{S}_i(\boldsymbol{x}) - \gamma \mathrm{T}^c_i(\boldsymbol{x}) \right) + \sum_{i\notin\mathcal{S}} \left(\mathrm{S}_i(\boldsymbol{x}) - (1-\gamma)\mathrm{T}^n_i(\boldsymbol{x}) \right), \end{aligned}$$

where $\mathcal{S}'' = \mathcal{Y}/\mathcal{S}/\mathcal{S}'$ could be an empty set or satisfies that $\mathbf{S}_i(\boldsymbol{x}) - \widetilde{\mathbf{T}}_i(\boldsymbol{x}) > 0$ for $i \in \mathcal{S}''$. Then We obtain the simplified form of the L_1 norm as

$$\sum_i \left| \frac{\partial \mathcal{L}_{\texttt{DIRK}}}{\partial \delta_i} \right| = 2 \sum_{i\in\mathcal{S}'} \left(\gamma \mathrm{T}^c_i(\boldsymbol{x}) - \mathrm{S}_i(\boldsymbol{x}) \right).$$

A.3 The pseudo-code of DIRK.

We summarize the pseudo-code of our DIRK method in Algorithm 1.

Algorithm 1: Pseudo-code of DIRK (one epoch).

```
Input: The PLL training set $\mathcal{D}$; Initial parameters $\theta_T, \theta_S$ of the teacher mdoel and the student model.
for iter $= 1, 2, ...,$ do
    Sample a mini-batch $B = \{\boldsymbol{x}_i, \mathcal{S}_i\}_{i=1}^n$ from $\mathcal{D}$
    Obtain the teacher's rectificated label confidence $\widetilde{\mathbf{T}}(\boldsymbol{x}_i)$ and student's output $\mathbf{S}(\boldsymbol{x}_i)$
    Caculate the rectificated distillation loss Eq. (5)
    Update the student network's parameters $\theta_S$ via gradient descent
    Update the teacher network's parameters $\theta_T$ via exponential momentum average
end
Output: The student model's parameters $\theta_S$.
```

A.4 The pseudo-code and achitecture illustration of DIRK-REF.

We provide the pseudo-code of our DIRK-REF method in Algorithm 2. Additionally, for clarity and ease of understanding, Figure 1 illustrates the details of DIRK-REF, encompassing both the label distillation module and the representation refinement module.

Algorithm 2: Pseudo-code of DIRK-REF (one epoch).

```
Input: The PLL training set $\mathcal{D}$; Encoder network $f_{\mathrm{T}}(\cdot)$, $f_{\mathrm{S}}(\cdot)$; Projection network $g_{\mathrm{T}}(\cdot)$, $g_{\mathrm{S}}(\cdot)$; Classifier $h_{\mathrm{T}}(\cdot)$, $h_{\mathrm{S}}(\cdot)$; An
       embedding queue $\boldsymbol{E}$ and a label confidence queue $\boldsymbol{I}$; Three different data augmentations $\mathrm{Aug}_1(\cdot), \mathrm{Aug}_2(\cdot), \mathrm{Aug}_3(\cdot)$.
for iter $= 1, 2, ...,$ do
    Sample a mini-batch $B = \{\boldsymbol{x}_i, \mathcal{S}_i\}_{i=1}^n$ from $\mathcal{D}$;
    // embeddings generation
    $\mathcal{B}_T^E = \{g_{\mathrm{T}}(f_{\mathrm{T}}(\mathrm{Aug}_1(\boldsymbol{x}_i)))\,|\boldsymbol{x}_i \in B\}, \mathcal{B}_S^E = \{g_{\mathrm{S}}(f_{\mathrm{S}}(\mathrm{Aug}_2(\boldsymbol{x}_i)))\,|\boldsymbol{x}_i \in B\};$
    $\mathcal{E} = \mathcal{B}_T^E \cup \mathcal{B}_S^E \cup \boldsymbol{E};$
    // rectificated label confidences generation
    $\mathcal{B}_T^L = \{\mathit{Rectification}(h_{\mathrm{T}}(f_{\mathrm{T}}(\mathrm{Aug}_3(\boldsymbol{x}_i))))\,|\boldsymbol{x}_i \in B\}$
    $\mathcal{I} = \mathcal{B}_T^L \cup \mathcal{B}_T^L \cup \boldsymbol{I}$
    Set $\widetilde{\mathbf{T}}(\cdot) = \mathit{Rectification}(h_{\mathrm{T}}(f_{\mathrm{T}}(\mathrm{Aug}_3(\cdot))))$
    // index set of instances generation
    for $\boldsymbol{x}_i \in B$ do
        $\tilde{y}_i = \arg\max_{j \in \mathcal{S}_i} h_{\mathrm{T}}(f_{\mathrm{T}}(\mathrm{Aug}_3(\boldsymbol{x}_i)))$
        $P(\boldsymbol{x}_i) = \{k|\arg\max_k \mathcal{I}_k = \tilde{y}_i, \mathcal{S}_k \cap \mathcal{S}_i \neq \emptyset\}$
        $\mathcal{E}(\boldsymbol{x}_i) = \{k|\mathcal{E}_k \neq \boldsymbol{z}_i\}$
    end
    // representation refinement loss calculation
    $w_j = \frac{\exp(\mathrm{sim}(\widetilde{\mathbf{T}}(\boldsymbol{x}_i), \mathcal{I}_j)/\tau_1)}{\sum_{k \in P(\boldsymbol{x}_i)} \exp(\mathrm{sim}(\widetilde{\mathbf{T}}(\boldsymbol{x}_i), \mathcal{I}_k)/\tau_1)}$
    $\mathcal{L}_{\mathtt{REF}}(f, g; \tau_1, \tau_2, \mathcal{B}_S^E) = \frac{1}{|\mathcal{B}_S^E|} \sum_{\boldsymbol{z}_i \in \mathcal{B}_S^E} \left\{ -\sum_{j \in P(\boldsymbol{x}_i)} w_j \log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \mathcal{E}_j)/\tau_2)}{\sum_{k \in \mathcal{E}(\boldsymbol{x}_i)} \exp(\mathrm{sim}(\boldsymbol{z}_i, \mathcal{E}_k))/\tau_2)} \right\},$
    // label distillation loss calculation
    $\mathcal{L}_{\mathtt{DIRK}}(f, h; B) = \frac{1}{|B|} \sum_{\boldsymbol{x}_i \in B} \sum_{j=1}^c -\widetilde{\mathbf{T}}(\boldsymbol{x}_i) \log h_{\mathrm{S}}(f_{\mathrm{S}}(\mathrm{Aug}_3(\boldsymbol{x}_i)))$
    // network updating
    Minimize loss $\mathcal{L}_{\mathtt{DIRK\text{-}REF}} = \mathcal{L}_{\mathtt{DIRK}} + \lambda \mathcal{L}_{\mathtt{REF}}$
    Update the student network's parameters via gradient descent
    Update the teacher network's parameters via exponential momentum average
    Update the Embedding pool $\boldsymbol{E}$ and label distribution pool $\boldsymbol{I}$
end
```

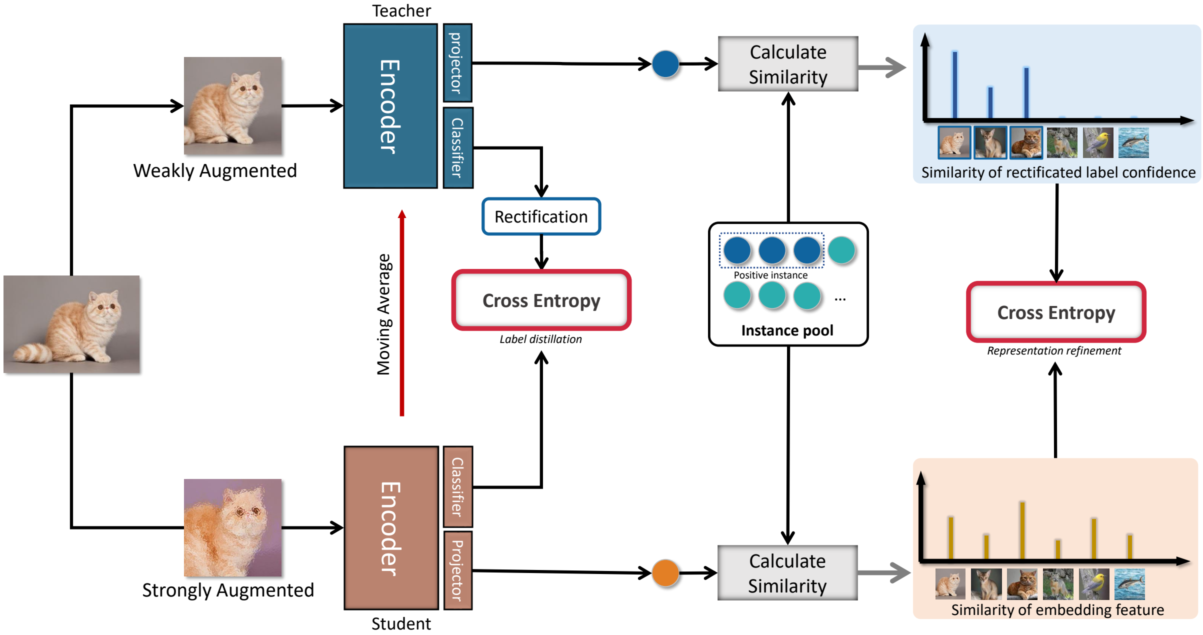


Figure 1: **Illustration of DIRK-REF**. DIRK-REF consists of two modules: label distillation and representation refinement. The label distillation module transfers the teacher's rectificated label confidence to the student. The representation refinement module computes similarity in label space using the teacher's rectificated outputs and confidence queue. This similarity knowledge then refines embedding features in the representation space.

A.5 Experiments Details of datasets.

We conduct experiments on seven benchmark datasets: Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), Kuzushiji-MNIST (Clanuwat et al. 2018), CIFAR-10 (Krizhevsky, Hinton et al. 2009), CIFAR-100 (Krizhevsky, Hinton et al. 2009), CUB-200 (Welinder et al. 2010), Flower (Nilsback and Zisserman 2008) and Oxford-IIIT Pet (Parkhi et al. 2012). To construct IDPLL datasets, we manually corrupt these benchmarks using an instance-dependent generation process (Xu et al. 2021), where the "*rate*" parameter controls the ambiguity level in code and its size is determined by the class space. Specifically, Table 1 presents the variable "*rate*" settings and corresponding average number of candidate labels (avg.#CLs). As an example, the PyTorch code for the generation process is as follows (Xu et al. 2021):

```
def ID_partialize(train_X, train_Y, model, batch_size, rate=0.4):
    # Obtain mdoel output and correct label
    _, outputs = model(train_X)
    train_p_Y = train_Y.clone().detach()
    # Sets the flipping probability of each labe per sample
    partial_rate_array = F.softmax(outputs, dim=1).clone().detach()
    partial_rate_array[torch.where(train_Y==1)] = 0
    partial_rate_array = partial_rate_array / torch.max(partial_rate_array, dim=1, keepdim=True)[0]
    partial_rate_array = partial_rate_array / partial_rate_array.mean(dim=1, keepdim=True) * rate
    partial_rate_array[partial_rate_array > 1.0] = 1.0
    # Binomial distribution sampling
    m = torch.distributions.binomial.Binomial(total_count=1, probs=partial_rate_array)
    z = m.sample()
    # Generating partial labels
    train_p_Y[torch.where(z == 1)] = 1.0
    return train_p_Y
```

Several real-world partially labeled datasets have been collected across diverse tasks and domains, including Lost (Cour,

Dataset	#Train	#Test	#Dims	#Labels	avg. #CLs
Fashion-MNIST	60,000	10,000	28×28	10	7.40 (*rate* = 1.00)
Kuzushiji-MNIST	60,000	10,000	28×28	10	7.95 (*rate* = 0.90)
CIFAR-10	50,000	10,000	32×32	10	5.89 (*rate* = 1.00)
CIFAR-100	50,000	10,000	32×32	100	9.40 (*rate* = 0.10)
CUB-200	5,994	5,794	224×224	200	6.17 (*rate* = 0.03)
Flower	1,020	6,149	224×224	102	5.49 (*rate* = 0.05)
Oxford-IIIT Pet	3,680	3,669	224×224	37	3.73 (rate = 0.10)

Table 1: Characteristic of the benchmark data sets corrupted instance-dependently.

Sapp, and Taskar 2011), Soccer Player (Zeng et al. 2013), Yahoo! News (Guillaumin, Verbeek, and Schmid 2010) from automatic face naming, MSRCv2 (Liu and Dietterich 2012) from object classification, BirdSong (Briggs, Fern, and Raich 2012) from bird song classification. Table 2 summarizes the characteristics of these real-world partial label datasets. For the automatic face naming task, faces cropped from an image are represented as instances while names extracted from the associated captions or subtitles are regarded as the corresponding candidate labels. For the task of object classification, image segmentations are regarded as instances while objects appearing within the same image are the corresponding candidate labels. For the bird song classification, singing syllables of the birds are represented as instances while bird species jointly singing during a 10-seconds period are regarded as candidate labels.

Dataset	#Train	#Test	#Dims	#Labels	avg. #CLs†
Lost	898	224	108	16	2.23
MSRCv2	1,406	352	48	23	3.16
BirdSong	3,998	1,000	38	13	2.18
Soccer Player	13,978	3,494	79	171	2.09
Yahoo! News	18,393	4,598	163	219	1.91

Table 2: Characteristic of the real-world experimental data sets.

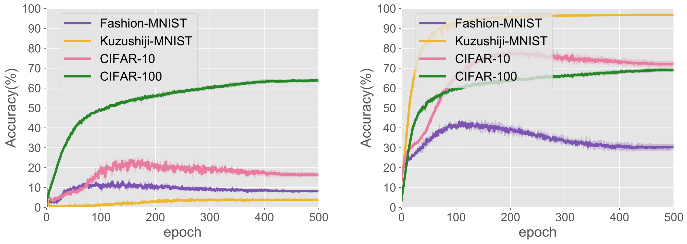
A.6 Ablation Experiment

Impact of the scaling factor γ of DIRK. We study the scaling factor γ, which dynamically combines two components into the rectificated label confidence to satisfy the assumption partial label knowledge. Table 3 and Figure 2 report performance with fixed $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. On the whole, larger γ generally improves accuracy on Fashion-MNIST and CIFAR-10 datasets, but this trend does not always hold for Kuzushiji-MNIST and CIFAR-100 datasets. Importantly, fixed γ risks overfitting to false positives, as evidenced in Figures 2. For instance, γ = 0.5 or 0.7 can overfit Fashion-MNIST and CIFAR10. Therefore, it is crucial to carefully select the optimal γ for each dataset, avoiding such overfitting. The adaptive γ in our method addresses this through dynamic adjustment during training.

	Fashion-MNIST	Kuzushiji-MNIST	CIFAR-10	CIFAR-100
$\gamma = 0.1$	0.04 ± 0.01%	0.18 ± 0.04%	0.03 ± 0.04%	69.12 ± 0.54%
$\gamma = 0.3$	0.45 ± 0.12%	0.49 ± 0.02%	0.14 ± 0.02%	50.26 ± 1.26%
$\gamma = 0.5$	8.27 ± 0.54%	2.98 ± 0.43%	16.42 ± 1.46%	62.45 ± 1.03%
$\gamma = 0.7$	31.83 ± 3.23%	97.84 ± 0.72%	71.28 ± 1.34%	68.02 ± 1.28%
$\gamma = 0.9$	90.14 ± 0.23%	95.23 ± 1.34%	89.28 ± 0.12%	63.81 ± 1.13%

Table 3: Accuracy comparison (mean ± std) with different fixed γ values on four benchmark datasets with low ambiguity level. The best result in each column is underlined.

Influence of mini-batch size of DIRK-REF. Figure 3 presents how mini-batch size impacts DIRK-REF training over epochs. We observe smaller batches of 64-128 samples yield optimal results. In the crucial early training stage with fewer epochs, these delicate batch sizes outperform larger batches. We posit this stems from providing fewer yet more accurate and informative

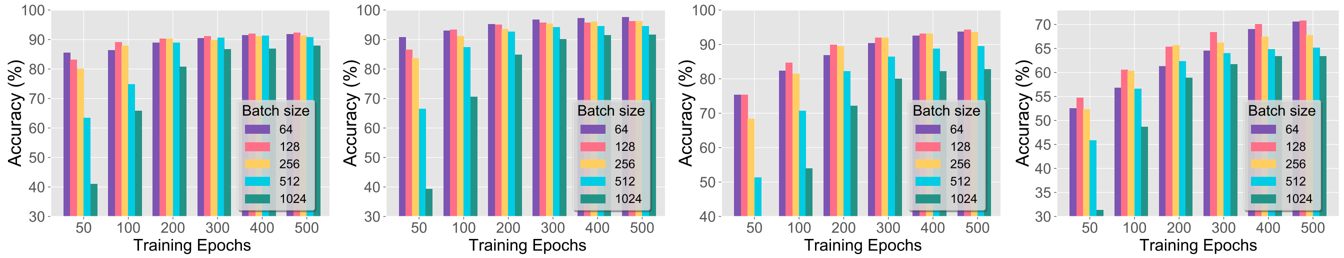


(a) $\gamma = 0.5$

(b) $\gamma = 0.7$

Figure 2: (a-b): Performance of Dirk with varing fixed γ values. The vertical axis on the left displays the accuracies of Fashion-MNIST, Kuzushiji-MNIST, and CIFAR-10, while the vertical axis on the right shows the accuracies of CIFAR-100.

knowledge-based positive pairs at each iteration. The limited samples prevent overfitting and allow the model to extract useful similarities. As training progresses with more epochs, performance gaps between different batch sizes decrease. This implies the representation benefits less from carefully chosen pairs as it matures. Overall, deliberate mini-batch sizes offer advantages in early training, while the effect diminishes with more epochs as the model converges.



(a) Fashion-MNIST.

(b) Kuzushiji-MNIST.

(c) CIFAR-10.

(d) CIFAR-100.

Figure 3: Performance of Dirk-ref with differenet mini-batch size and training epoch on Fahison-MNIST, Kuzushiji-MNIST, CIFAR-10, CIFAR-100 datasets.

A.7 Uncertainty quantification of Dirk.

Figures 4 and Figure 5 present reliability histograms illustrating model calibration on Fashion-MNIST, Kuzushiji-MNIST, and CIFAR-10 datasets. We observe differing tendencies between methods. Proden and Rc exhibit significant overconfidence, which we attribute to their likely over-dependence on pseudo-labels from self-training strategies. Without sufficient regularization, they accumulate errors that lead to over-estimating output confidence. In contrast, CC shows underfitting tendencies, possibly because transition matrix-based methods emphasize holistic dataset consistency rather than relying heavily on individual pseudo-labeled samples. This more conservative approach prevents overconfidence but underestimates certainty. Additionally, while Able is overconfident, Pico is not. We hypothesize this results from Pico's use of similarity comparisons between learned embeddings and prototypical features during training and inference. This intrinsic technique may provide enhanced robustness against error accumulation and overconfidence. The prototypes serve as stable anchors to calibrate predictions.

A.8 Computation cost of Dirk.

Compared to other SOTA PLL methods, Dirk only involves label distillation, which is more efficient. As shown in Table 4, Dirk has the shortest mean runtimes compared to other PLL methods.

	Dirk	Pop	Idgp	Able	Valen	Cr-dpll	Pico
CIFAR10	**5.80**	16.99	6.10	11.39	6.12	26.23	26.93
CIFAR100	**5.70**	16.59	6.85	13.35	10.57	19.03	26.88
CUB-200	**20.98**	39.25	21.33	21.01	26.53	29.80	25.66

Table 4: Comparison of mean runtime (in hours) on datasets, with the shortest time among each row highlighted in bold.

A.9 Complete experiments of DIRK-REF.

Complete comparison results of DIRK-REF on all benchmark datasets and real-world datasets are presented in Table 5. Consistent with the conclusions in the main text, DIRK-REF equipped with the knowledge-based representation refinement module demonstrates the best performance overall. Notably, the real-world PLL datasets used in our experiments do not contain natural features like those found in images. Therefore, the representation refinement module can only offer limited improvements on these datasets. This is validated by the results in Table 6 which shows DIRK-REF yields only slight gains over DIRK on these datasets.

	Fashion-MNIST	Kuzushiji-MNIST	CIFAR-10	CIFAR-100	CUB-200	Flower	Oxford-IIIT Pet
$\lambda = 0$ (DIRK)	91.48 ± 0.21%	96.80 ± 0.52%	90.87 ± 0.25%	68.77 ± 0.49%	49.29 ± 1.00%	44.03 ± 0.02%	64.95 ± 2.11%
$\lambda = 0.1$	92.01 ± 0.24%	**98.31 ± 0.20%**	93.50 ± 0.16%	70.94 ± 1.17%	50.91 ± 0.24%	47.66 ± 0.74%	68.28 ± 0.14%
$\lambda = 0.3$	**92.10 ± 0.08%**	98.14 ± 0.20%	94.00 ± 0.13%	70.72 ± 0.54%	52.78 ± 0.15%	50.24 ± 0.31%	68.80 ± 0.33%
$\lambda = 0.5$	91.88 ± 0.32%	98.09 ± 0.17%	94.24 ± 0.03%	71.53 ± 1.35%	51.89 ± 0.11%	**52.18 ± 0.28%**	68.27 ± 0.13%
$\lambda = 0.7$	92.03 ± 0.58%	97.84 ± 0.18%	**94.25 ± 0.26%**	**71.72 ± 0.63%**	52.91 ± 0.24%	48.06 ± 0.16%	**68.95 ± 0.12%**
$\lambda = 1.0$	91.88 ± 0.36%	97.58 ± 0.32%	93.73 ± 0.31%	70.61 ± 0.85%	**52.93 ± 0.31%**	48.26 ± 0.38%	68.78 ± 0.42%

Table 5: Accuracy comparison (mean ± std) with diffrent λ of the representation refinement module.

Method	Lost	BirdSong	MSRCv2	Soccer Player	Yahoo!News
DIRK	79.24 ± 0.63%	74.52 ± 0.23%	48.59 ± 0.28%	55.83 ± 0.35%	67.65 ± 0.32%
DIRK-REF	**79.24 ± 0.72%**	**74.55 ± 0.13%**	**49.91 ± 0.15%**	**55.92 ± 0.30%**	**67.69 ± 0.28%**

Table 6: Accuracy comparison (mean ± std) on real-world partial label datasets.

A.10 Differences with decoupled knowledge distillation (DKD) (Zhao et al. 2022)

Although the decoupling stage in our method looks similar to that in DKD (Zhao et al. 2022), the motivations and implementations are distinct. Our primary goal for decoupling is to obtain reliable label confidences in instance-dependent partial label learning, which comply with partial label knowledge illustrated in pilot experiments. In the implementation, our method decouples the original confidence into candidate and non-candidate distributions, respectively. These distributions are then recombined using an adaptive scaling factor γ. In contrast, DKD separates classical knowledge distillation into target classification knowledge distillation (TCKD) and non-target classification knowledge distillation (NCKD). Then, TCKD and NCKD are reformulated by a fixed coefficient.

A.11 Limitation

In this work, we explored the instance-dependent PLL scenario. However, real-world data contains a mixture of false positive label types, including instance-dependent, class-dependent, and open-set noise. For instance, occlusion can cause instance-specific errors. Handling such composite noise remains an open challenge. Developing more robust models to address diverse false positives requires an integrative approach combining techniques like adaptive ensembling, meta-learning and open-set recognition.

References

Briggs, F.; Fern, X. Z.; and Raich, R. 2012. Rank-loss support instance machines for MIML instance annotation. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 534–542.

Clanuwat, T.; Bober-Irizar, M.; Kitamoto, A.; Lamb, A.; Yamamoto, K.; and Ha, D. 2018. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*.

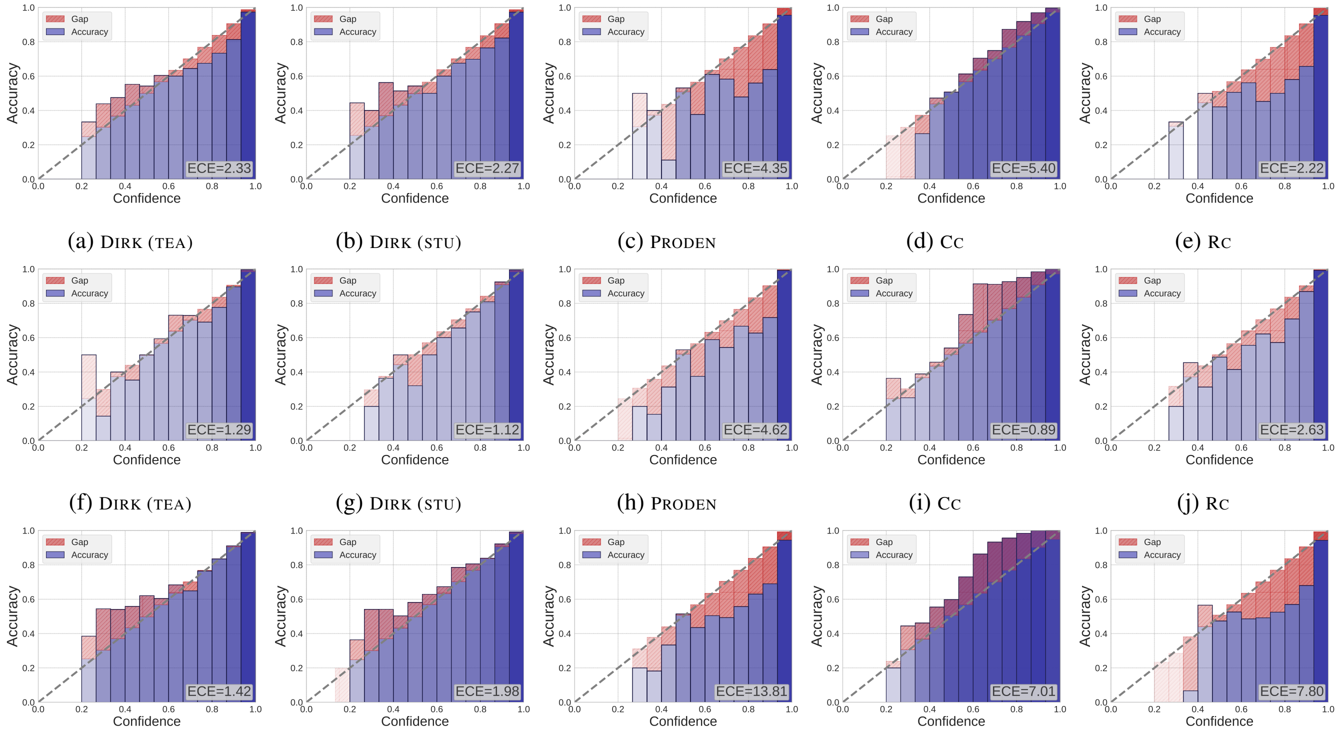
Cour, T.; Sapp, B.; and Taskar, B. 2011. Learning from partial labels. *Journal of Machine Learning Research*, 12: 1501–1536.

Guillaumin, M.; Verbeek, J.; and Schmid, C. 2010. Multiple instance metric learning from automatically labeled bags of faces. In *European Conference on Computer Vision*, 634–647.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Liu, L.; and Dietterich, T. G. 2012. A conditional multinomial mixture model for superset label learning. In *Advances in neural information processing systems*, 557–565.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 722–729.

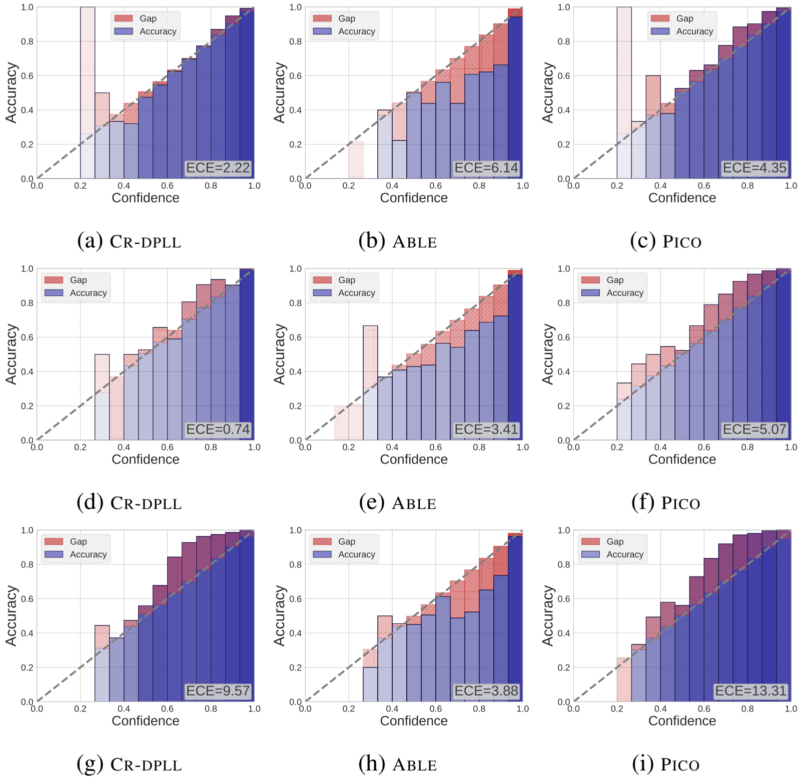


(a) DIRK (TEA) (b) DIRK (STU) (c) PRODEN (d) CC (e) RC

(f) DIRK (TEA) (g) DIRK (STU) (h) PRODEN (i) CC (j) RC

(k) DIRK (TEA) (l) DIRK (STU) (m) PRODEN (n) CC (o) RC

Figure 4: Reliability diagram and expected calibration error. The reliability diagrams for each dataset are presented, with the top row (a-d) showing Fashion-MNIST, mid row (e-j) showing Kuzushiji-MNIST, and bottom row (k-o) showing CIFAR-10.



(a) Cr-DPLL (b) Able (c) Pico

(d) Cr-DPLL (e) Able (f) Pico

(g) Cr-DPLL (h) Able (i) Pico

Figure 5: Reliability diagram and expected calibration error. The reliability diagrams for each dataset are presented, with the top row (a-c) showing Fashion-MNIST, mid row (d-f) showing Kuzushiji-MNIST, and bottom row (g-i) showing CIFAR-10.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. 2012. Cats and dogs. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 3498–3505.

Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; and Perona, P. 2010. Caltech-UCSD birds 200.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xu, N.; Qiao, C.; Geng, X.; and Zhang, M.-L. 2021. Instance-dependent partial label learning. In *Advances in neural information processing systems*, 27119–27130.

Zeng, Z.; Xiao, S.; Jia, K.; Chan, T.-H.; Gao, S.; Xu, D.; and Ma, Y. 2013. Learning by associating ambiguously labeled images. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 708–715.

Zhao, B.; Cui, Q.; Song, R.; Qiu, Y.; and Liang, J. 2022. Decoupled Knowledge Distillation. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 11943–11952.