The Appendix for Disentangled Partial Label Learning

Wei-Xuan Bao[1, 2], **Yong Rui**[3], **Min-Ling Zhang**[1, 2*]

[1]School of Computer Science and Engineering, Southeast University, Nanjing, China
[2]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
[3]Lenovo Research, Lenovo Group Ltd., Beijing, China
baowx@seu.edu.cn, yongrui@lenovo.com, zhangml@seu.edu.cn

A The Proposed TERIAL Approach

A.1 The Pseudo-Code of TERIAL

The complete procedure of TERIAL is summarized in Table 4.

A.2 Theoretical Analysis for the Neighborhood Routing Mechanism

In this section, we analyse two essential issues about the proposed neighborhood routing mechanism: (1) Is it convergent? (2) If it is, then what solution will it converge to? We simultaneously address the above problems based on the von Mises-Fisher (vMF) mixture model from an expectation-maximization (EM) perspective.

Given the observation $\{\mathbf{w}_{u_k}\}_{k=1}^K$ and $\{\mathbf{w}_{v_k} : v \in \mathcal{N}_u\}_{k=1}^K$, the vMF mixture model with parameters $\{\mathbf{c}_{u_k}\}_{k=1}^K$ is defined as:

$$\begin{aligned}\mathbf{w}_{u_k} &\sim \text{vMF}(\mathbf{c}_{u_k}, 1),\\ r_v &\sim \text{Categorical}(\underbrace{[\frac{1}{K}, ..., \frac{1}{K}]}_{K}),\\ \mathbf{w}_{v_{r_v}}|r_v &\sim \text{vMF}(\mathbf{c}_{u_{r_v}}, \frac{1}{\tau}),\\ \mathbf{w}_{v_{k'}}|r_v &\sim \text{vMF}(\boldsymbol{\mu}, 0), k' \neq r_v.\end{aligned}$$

The variable r_v denotes the latent factor that contributes to the connection between node u and node v. The mixture model's parameters $\{\mathbf{c}_{u_k}\}_{k=1}^K$ are viewed as K true aspects of node u to be estimated and the observed features $\{\mathbf{w}_{u_k}\}_{k=1}^K$ are treated as their noisy observation. We assume that $\mathbf{w}_{u_{r_v}}$ should be similar with $\mathbf{w}_{v_{r_v}}$ if node u and its neighbor v are connected due to the latent factor r_v. For chunks associated with non-explanatory factors $k' \neq r_v$, we have limited information about their representation. As a result, we assume that $\mathbf{w}_{v_{k'}}$ are sampled uniformly, i.e., sampled from vMF$(\boldsymbol{\mu}, 0)$. [1]

Based on the above vMF mixture model, we deduce Theorem 1 in the main paper and provide its proof here.

Proof of Theorem 1. Let $\Theta = \{\mathbf{c}_{u_k}\}_{k=1}^K$, $R = \{r_v : v \in \mathcal{N}_u\}$, and $Z = \{\mathbf{w}_{i_k} : i \in \{u\} \cup \mathcal{N}_u, 1 \leqslant k \leqslant K\}$. The objective of an EM algorithm is to maximize $p(Z;\Theta) = \sum_R p(R, Z;\Theta)$. Usually, it is required to introduce an auxiliary distribution $q(R)$ over R to approximate the posterior $p(R|Z;\Theta)$. Then the log-likelihood function is formulated as $\ln p(Z;\Theta) = \sum_R q(R) \ln \frac{p(R,Z;\Theta)}{q(R)} + \sum_R q(R) \ln \frac{q(R)}{p(R|Z;\Theta)}$. Let $L(\Theta, q) = \sum_R q(R) \ln \frac{p(R,Z;\Theta)}{q(R)}$ and $D_{\text{KL}}(q\|p_\Theta)$ denote the Kullback-Leibler (KL) divergence from $p(R|Z;\Theta)$ to the auxiliary distribution $q(R)$. Then the objective function is simplified as $\ln p(Z;\Theta) = L(\Theta, q) + D_{\text{KL}}(q\|p_\Theta)$. Since the KL divergence is non-negative, $L(\Theta, q)$ is a lower bound of $\ln p(Z;\Theta)$.

The EM algorithm typically consists of an expectation (E) step and a maximization (M) step. In the E-step, the auxiliary distribution $q(R)$ is set to $p(R|Z;\Theta)$ using the current estimate of the parameters to tight the lower bound of $L(\Theta, q)$. Note that $p(R|Z;\Theta) = \prod_v p(r_v|Z;\Theta)$ and $p(r_v = k|Z;\Theta) \propto p(r_v = k, Z;\Theta) \propto \exp(\mathbf{w}_{v_k}^\top \mathbf{c}_{u_k}/\tau)$. As a result, the optimal $q(R)$ in each iteration is set as $q(r_v = k) \propto \exp(\mathbf{w}_{v_k}^\top \mathbf{c}_{u_k}/\tau)$, which proves that Eq.(5) is actually equivalent to the E-step in the EM algorithm.

In the M-step, with $q(R)$ fixed to the values determined in the E-step, the lower bound $L(\Theta, q)$ is maximized w.r.t parameters Θ under the constrains that $\mathbf{c}_{u_k}^\top \mathbf{c}_{u_k} = 1 (1 \leqslant k \leqslant K)$. We construct the Lagrange function $\mathcal{L} = L(\Theta, q) + \sum_{k=1}^K \lambda_k (1 - \mathbf{c}_{u_k}^\top \mathbf{c}_{u_k})$ with Lagrange multipliers $\lambda_k (1 \leqslant k \leqslant K)$. Taking partial derivatives of $\mathcal{L}$ with respect to $\{\mathbf{c}_{u_k}\}_{k=1}^K$ and setting them to zero, we get that $\mathbf{c}_{u_k} = \frac{\mathbf{w}_{u_k} + \sum_{v \in \mathcal{N}_u} p_{u,v}^{k(t-1)} \mathbf{w}_{v_k}}{2\lambda_k}$ (Banerjee et al. 2005). Accordingly, considering $\mathbf{c}_{u_k}^\top \mathbf{c}_{u_k} = 1$, the optimal $\mathbf{c}_{u_k}$ is derived exactly as Eq.(4). This proves that Eq.(4) is actually performing the M-step in the EM algorithm.

Let $q^{(t)}(R)$ denote the refined distribution in the tth E-step and $\Theta^{(t)}$ denote the updated parameters in the tth M-step respectively. Then we have that $\ln p(Z;\Theta^{(t-1)}) = L(\Theta^{(t-1)}, q) + D_{\text{KL}}(q\|p_{\Theta^{(t-1)}}) = L(\Theta^{(t-1)}, q^{(t)}) \leqslant L(\Theta^{(t)}, q^{(t)}) \leqslant L(\Theta^{(t)}, q^{(t)}) + D_{\text{KL}}(q^{(t)}\|p_{\Theta^{(t)}}) = \ln p(Z;\Theta^{(t)})$. This proves that the likelihood increases

*Corresponding author

[1]The probability density function of vMF$(\boldsymbol{\mu}, \kappa)$ is defined as $p_{\text{vMF}}(\mathbf{x};\boldsymbol{\mu},\kappa) \propto \exp(\kappa \cdot \boldsymbol{\mu}^\top \mathbf{x})$. Since $p_{\text{vMF}}(\mathbf{x};\boldsymbol{\mu},0)$ is constant, vMF$(\boldsymbol{\mu}, 0)$ could be viewed as a uniform distribution.

Inputs:

$\mathcal{D}$	:	the PL training set $\{(\mathbf{x}_i, S_i) \mid 1 \leqslant i \leqslant n\}$ ($\mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{l_1, l_2, \ldots, l_q\}, \mathbf{x}_i \in \mathcal{X}, S_i \subseteq \mathcal{Y}$)
L	:	the number of disentangling layers
K	:	the assumed number of latent factors
$f_k(\cdot)$	:	the mapping functions ($1 \leqslant k \leqslant K$)
τ	:	the smooth factor in Eq.(3)
T	:	the maximum number of iterations for the clustering procedure
α	:	the balancing factor in Eq.(6)
E_{max}	:	the number of epochs
I_{max}	:	the number of iterations in each epoch
$\mathbf{x}'$	:	the unseen instance

Outputs:

l^*	:	the predicted label for $\mathbf{x}'$

Process:

Initialize the $n \times q$ labeling confidence matrix $\mathbf{Y}$ according to Eq.(1);
Initialize embeddings of labels;
for $ep = 1$ to E_{max} **do**
  for $i = 1$ to I_{max} **do**
    Fetch a random batch $\mathcal{D}'$ from $\mathcal{D}$;
    Formulate the batch $\mathcal{D}'$ as an undirected bipartite graph, where an instance is only connected to its candidate labels;
    Derive the representation of instance nodes according to Eq.(2);
    for $l = 1$ to L **do**
      Initialize correlation coefficients $p_{u,v}^k$ according to Eq.(3);
      for $t = 1$ to T **do**
        For nodes in the graph, respectively derive the temporary clustering centers $\mathbf{c}_{u_k}^{(t)}$ according to Eq.(4);
        Update the correlation coefficients $p_{u,v}^{k(t)}$ according to Eq.(5);
      end for
      Assign the representation of nodes in the graph according to Eq.(6);
    end for
    Alternatively minimize the empirical loss $\mathcal{L}_1 = \mathcal{L}_{\text{ce}}$ or $\mathcal{L}_2 = \mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{ind}}$ according to Eq.(7) and Eq.(9);
    Update the mapping functions and label embeddings;
    Update the labeling confidence matrix $\mathbf{Y}$ according to Eq.(10);
  end for
end for
Derive the disentangled representation of unseen instance $\mathbf{x}'$ according to obtained mapping functions;
Make the prediction l^* by computing the inner product between derived disentangled representation of the unseen instance and label embeddings;

Table 4: The pseudo-code of TERIAL.

monotonically during iterations, which is upper-bounded by zero. Accordingly, we could conclude that the algorithm converges.

B Experiments on Benchmark Datasets

In this paper, all algorithms are implemented with PyTorch (Paszke et al. 2019) and trained on 1 NVIDIA Tesla V100 GPU (32GB).

B.1 Descriptions of Datasets

In this paper, five popular benchmark datasets are used to generate synthetic PL datasets:

- MNIST (LeCun et al. 1998): It is a 10-class dataset of handwritten digits, where each instance is a 28×28 grayscale image and the classes vary from 0 to 9. It has 60000 training examples and 10000 testing examples.
- Kuzushiji-MNIST (Clanuwat et al. 2018): It is a 10-class dataset of Japanese characters, where each instance is a 28×28 grayscale image. It has 60000 training examples and 10000 testing examples.
- Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017): It is a 10-class dataset of fashion items, where each instance is a 28×28 grayscale image and the classes include T-shirt/top, trouser, pullover, dress, sandal, coat, shirt, sneaker, bag, and ankle boot. It has 60000 training examples and 10000 testing examples.
- SVHN (Netzer et al. 2011): It is a 10-class dataset from Google Street View images, where each instance is a $32 \times 32 \times 3$ colored image in RGB format and the classes vary from 0 to 9. It has 73257 training examples and 26032 testing examples.
- CIFAR-10 (Krizhevsky, Hinton et al. 2009): It is a 10-class dataset, where each instance is a $32 \times 32 \times 3$ colored image in RGB format and the classes include airplane, bird, automobile, cat, deer, frog, dog, horse, ship, and truck. It has 50000 training examples and 10000 testing examples.

B.2 Descriptions of Comparing Methods

For benchmark datasets, the comparing algorithms include:

- PRODEN (Lv et al. 2020): It progressively identifies the true label from candidate labels through approximately minimizing a risk estimator.
- RC (Feng et al. 2020): It is a novel risk-consistent partial label learning approach based on the generation model.
- CC (Feng et al. 2020): It is a novel classifier-consistent partial label learning approach based on the generation model.
- LW (Wen et al. 2021): It proposes a family of loss functions and introduces the leverage parameter β to consider the trade-off between losses on partial labels and non-partial labels.
- VALEN (Xu et al. 2021): It makes the first attempt towards instance-dependent PLL and applies the probabilistic model to iteratively recover label distribution for each instance.

	TERIAL against comparing algorithms					
	PRODEN	RC	CC	LW	VALEN	CAVL
$r = 3$	4/1/0	4/1/0	5/0/0	5/0/0	5/0/0	5/0/0
$r = 5$	5/0/0	5/0/0	5/0/0	5/0/0	5/0/0	5/0/0
$r = 7$	4/0/1	4/0/1	5/0/0	5/0/0	5/0/0	5/0/0
In Total	**13/1/1**	**13/1/1**	**15/0/0**	**15/0/0**	**15/0/0**	**15/0/0**

Table 5: Win/tie/loss counts (pairwise *t*-test at 0.05 significance level) between TERIAL and comparing algorithms in terms of different number of false positive labels ($r = 3, 5, 7$).

	MNIST	KMNIST	FMNIST	SVHN	CIFAR-10
Ours	**92.89±0.16%**	**67.03±0.19%**	**81.62±0.08%**	**92.95±0.17%**	49.50±0.34%
PRODEN	92.31±0.22% •	63.98±0.27% •	77.09±0.14% •	92.13±0.31% •	**52.95±0.37%** ○
RC	92.42±0.32% •	64.28±0.14% •	77.35±0.24% •	92.42±0.25%	52.75±0.27% ○
CC	92.22±0.25% •	63.74±0.22% •	76.84±0.12% •	89.99±0.16% •	49.11±0.13%
LW	91.76±0.17% •	62.92±0.13% •	75.94±0.28% •	87.97±0.33% •	45.19±0.26% •
VALEN	89.32±0.45% •	59.37±0.43% •	72.12±0.43% •	83.27±0.13% •	37.49±0.56% •
CAVL	92.58±0.12% •	63.10±0.38% •	79.26±0.19% •	91.69±0.27% •	45.57±0.13% •

Table 6: Classification accuracy (mean±std) of each comparing algorithm on instance-dependent benchmark datasets, where •/○ indicates whether TERIAL is statistically superior/inferior to the comparing approach on each dataset (pairwise *t*-test at 0.05 significance level). The best result among methods is highlighted in bold.

- CAVL (Zhang et al. 2022): It identifies the true label by the class with the maximum class activation value.

B.3 Results of Pairwise *t*-test on Corrupted Benchmark Datasets

The pairwise *t*-test at 0.05 significance level is conducted to show whether the performance difference between TERIAL and comparing algorithms is significant, where the resulting win/tie/lose counts are reported in Table 5.

B.4 Results on Instance-dependent Benchmark Datasets

Instance-dependent PL datasets (Qiao, Xu, and Geng 2023; Wu, Wang, and Zhang 2022) are generated according to the same strategy utilized in (Xu et al. 2021), which made the first attempt towards instance-dependent PLL. Specifically, given an instance $\mathbf{x}$, the flipping probability of each incorrect label is derived from $q_j(\mathbf{x}) = \frac{\hat{h}_j(\mathbf{x})}{\sum_{k \in \mathcal{Y}} \hat{h}_k(\mathbf{x})}$, where $\hat{h}(\cdot)$ denotes a pre-trained neural network. The predictive performance (mean±std) of comparing algorithms on instance-dependent benchmark datasets are reported in Table 6, where •/○ indicates whether TERIAL is statistically superior/inferior to the comparing approach on each dataset (pairwise *t*-test at 0.05 significance level). The best result among methods is highlighted in bold.

C Experiments on Real-World Datasets

Datasets. In this paper, four real-world PL datasets are utilized to evaluate the effectiveness of our proposed approach, including:

- Lost (Cour, Sapp, and Taskar 2011): It is a dataset for automatic face naming from images or videos, where instances correspond to faces cropped from an image or video frame while candidate labels correspond to names extracted from the associated captions or subtitles.
- English (Zhou et al. 2018), Malagasy (Garrette and Baldridge 2013) and Italian (Zhou et al. 2018): They are the datasets for part-of-speech (POS) tagging, where instances correspond to the target words with contextual features while candidate labels correspond to the part-of-speech tags that the target words may have.

Characteristics of these datasets are shown in Table 7.

Comparing methods. Aforementioned DNN based methods are also applied here on real-world datasets employing linear model as backbones. Other settings are the same as before. In addition, we add five classical PLL approaches for comparison, each configured with parameters suggested in respective literatures:

- PL-kNN (Hüllermeier and Beringer 2006): An averaging-based partial label learning algorithm. It makes prediction on unseen instance by hiring weighted kNN voting strategy [suggested configuration: k=10].
- PL-SVM (Nguyen and Caruana 2008): An identification-based partial label learning algorithm. It learns the predictive model by maximizing the classification margin over candidate label set and non-candidate label set [suggested configuration: regularization parameter pool with $\{10^{-3}, \ldots, 10^{3}\}$].
- PL-ECOC (Zhang, Yu, and Tang 2017): A transformation-based partial label learning algorithm. It learns the predictive model by decomposing the PL learning problem into a group of binary learning problems through adapting the error-correcting output codes (ECOC) techniques [suggested configuration: ECOC coding length $\lceil 10 \cdot \log_2(q) \rceil$].

Data Set	# Examples	# Features	# Class Labels	average # Candidate Labels	Task Domain
Lost	1,122	108	16	2.23	*automatic face naming*
English	24,000	300	45	1.19	*POS tagging*
Malagasy	5,303	384	44	8.35	*POS tagging*
Italian	21,878	518	90	1.60	*POS tagging*

Table 7: Characteristics of the real-world PL datasets.

- IPAL (Zhang and Yu 2015): An instance-based partial label learning algorithm. It learns the predictive model by adapting label propagation for graph-based disambiguation [suggested configuration: balancing parameter $\alpha = 0.95$].
- SURE (Feng and An 2019): A self-training partial label learning algorithm. It learns the desired model and performs pseudo-labeling jointly by solving a tailored convex-concave optimization problem [suggested configuration: regularization parameters $\lambda = 0.3, \beta = 0.05$].

For TERIAL, the assumed number of latent factors K are set as: $K = 9$ on Lost, $K = 10$ on English, $K = 12$ on Malagasy, $K = 14$ on Italian. We perform five-fold cross-validation on real-world datasets and report the average accuracy with the standard deviation for each comparing algorithm.

Empirical Results. The predictive performance (mean±std) of comparing algorithms on real-world datasets are reported in Table 8. In addition, •/◦ indicates whether TERIAL is statistically superior/inferior to the comparing approach on each dataset (pairwise *t*-test at 0.05 significance level). It could be observed that TERIAL achieves superior performance against other comparing methods in most cases. The only losses occur on the dataset of Malagasy, which not only has a small amount of data, but also has the largest average number of candidate labels, making it difficult for TERIAL to give full play to its skills.

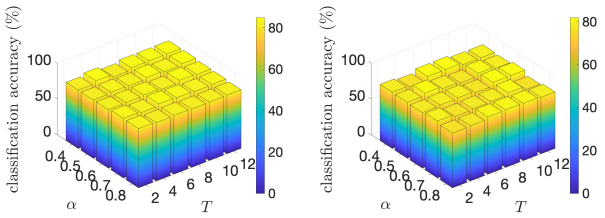
D Further Studies

D.1 Impact of Parameters α and T

The learning rate α and the maximum number of iterations T are key hyper-parameters for the routing mechanism. Here we investigate their impact on TERIAL's predictive performance. The results on the datasets of KMNIST ($r = 7$) and CIFAR-10 ($r = 3$) are illustrated in Fig. 2. Overall, the classification accuracy fluctuates moderately as the values of α and T change. For different datasets, fine-tuning these two parameters might lead to performance improvement, while $\alpha = 0.6$ and $T = 6$ is a reasonable default setting in this paper.

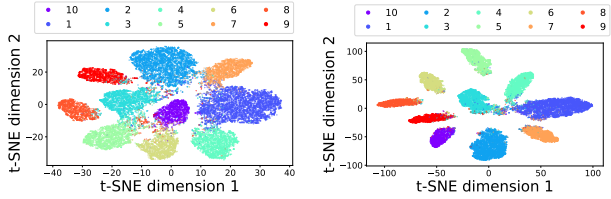
D.2 Impact of Independence Modeling

The performance of TERIAL with(w/) or without(w/o) the independence modeling module on benchmark datasets corrupted by the instance-dependent strategy is reported in Table 9.



(a) KMNIST ($r = 7$) (b) CIFAR-10 ($r = 3$)

Figure 2: Impact of learning rate α and the number of iterations T on classification performance of TERIAL on datasets of KMNIST ($r = 7$) and CIFAR-10 ($r = 3$).



(a) RC (b) TERIAL

Figure 3: t-SNE visualization of representation produced by RC and TERIAL on the test dataset of SVHN($r = 5$).

E Visualization

In Fig. 3, we visualize the representation produced by TERIAL and RC on the test dataset of SVHN($r = 5$).

References

Banerjee, A.; Dhillon, I. S.; Ghosh, J.; and Sra, S. 2005. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6: 1345–1382.

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

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

Feng, L.; and An, B. 2019. Partial label learning with self-guided retraining. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 3542–3549. Honolulu, HI.

Feng, L.; Lv, J.; Han, B.; Xu, M.; Niu, G.; Geng, X.; An, B.; and Sugiyama, M. 2020. Provably consistent partial-

	Lost	English	Malagasy	Italian
Ours	**80.50±2.66%**	**73.48±3.46%**	60.21±1.29%	**69.79±5.18%**
PRODEN	77.23±1.62% •	72.12±2.98% •	62.41±4.20% ○	68.91±3.92%
RC	78.10±2.19% •	72.44±6.18%	62.55±2.06% ○	67.92±2.77%
CC	77.12±2.89% •	71.58±3.93% •	61.72±4.75% ○	67.03±1.97% •
LW	77.32±6.12% •	71.98±3.44% •	62.47±6.10% ○	67.98±5.58%
VALEN	68.76±5.19% •	63.19±4.87% •	52.72±3.28% •	59.13±4.97% •
CAVL	78.25±4.73% •	72.63±2.78%	60.15±3.16%	67.10±1.67% •
PL-kNN	35.91±2.96% •	34.72±1.43% •	59.19±2.19%	45.00±6.28% •
PL-SVM	73.44±3.98% •	70.15±3.72% •	56.52±1.58% •	61.63±4.54% •
PL-ECOC	63.64±1.01% •	69.77±2.52% •	61.41±3.47%	63.27±6.12% •
IPAL	72.61±4.74% •	63.23±3.79% •	**63.72±2.39%** ○	56.92±2.55% •
SURE	77.91±3.31% •	72.33±4.31%	61.07±4.37%	66.57±4.22% •

Table 8: Classification accuracy (mean±std) of each comparing algorithm on real-world datasets, where •/○ indicates whether TERIAL is statistically superior/inferior to the comparing approach on each dataset (pairwise *t*-test at 0.05 significance level). The best result among methods is highlighted in bold.

	MNIST	KMNIST	FMNIST	SVHN	CIFAR-10
w/ $\mathcal{L}_{\text{ind}}$	92.89%	67.03%	81.62%	92.95%	49.50%
w/o $\mathcal{L}_{\text{ind}}$	91.37%	65.82%	79.71%	90.49%	48.42%

Table 9: Impact of independence loss $\mathcal{L}_{\text{ind}}$ on classification performance of TERIAL on benchmark datasets corrupted by the instance-dependent strategy.

label learning. In *Advances in Neural Information Processing Systems 33*, 6–12. Virtual Event.

Garrette, D.; and Baldridge, J. 2013. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 138–147. Atlanta, GA.

Hüllermeier, E.; and Beringer, J. 2006. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5): 419–439.

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

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lv, J.; Xu, M.; Feng, L.; Niu, G.; Geng, X.; and Sugiyama, M. 2020. Progressive identification of true labels for partial-label learning. In *Proceedings of the 37th International Conference on Machine Learning*, 6500–6510. Virtual Event.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Nguyen, N.; and Caruana, R. 2008. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 551–559. Las Vegas, NV.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E. Z.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, 8024–8035. Vancouver, Canada.

Qiao, C.; Xu, N.; and Geng, X. 2023. Decomposition-based generation process for instance-dependent partial label learning. In *Proceedings of the 11th International Conference on Learning Representations*. Kigali, Rwanda.

Wen, H.; Cui, J.; Hang, H.; Liu, J.; Wang, Y.; and Lin, Z. 2021. Leveraged weighted loss for partial label learning. In *Proceedings of the 38th International Conference on Machine Learning*, 11091–11100. Virtual Event.

Wu, D.-D.; Wang, D.-B.; and Zhang, M.-L. 2022. Revisiting consistency regularization for deep partial label learning. In *Proceedings of the 39th International Conference on Machine Learning*, 24212–24225. Baltimore, MD.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *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 34*, 27119–27130. Virtual Event.

Zhang, F.; Feng, L.; Han, B.; Liu, T.; Niu, G.; Qin, T.; and Sugiyama, M. 2022. Exploiting class activation value for partial-label learning. In *Proceedings of the 10th International Conference on Learning Representations*. Virtual Event.

Zhang, M.-L.; and Yu, F. 2015. Solving the partial label learning problem: An instance-based approach. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 4048–4054. Buenos Aires, Argentina.

Zhang, M.-L.; Yu, F.; and Tang, C.-Z. 2017. Disambiguation-free partial label learning. *IEEE Trans-*

actions on Knowledge and Data Engineering, 29(10): 2155–2167.

Zhou, D.; Zhang, Z.; Zhang, M.-L.; and He, Y. 2018. Weakly supervised POS tagging without disambiguation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 17(4): 1–19.