Disentangled Partial Label Learning

Wei-Xuan Bao¹,², Yong Rui³, Min-Ling Zhang¹,²*¹

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

Abstract
Partial label learning (PLL) induces a multi-class classifier from training examples each associated with a set of candidate labels, among which only one is valid. The formation of real-world data typically arises from heterogeneous entanglement of series latent explanatory factors, which are considered intrinsic properties for discriminating between different patterns. Though learning disentangled representation is expected to facilitate label disambiguation for partial-label (PL) examples, few existing works were dedicated to addressing this issue. In this paper, we make the first attempt towards disentangled PLL and propose a novel approach named Terial, which makes predictions according to derived disentangled representation of instances and label embeddings. The Terial approach formulates the PL examples as an undirected bipartite graph where instances are only connected with their candidate labels, and employs a tailored neighborhood routing mechanism to yield disentangled representation of nodes in the graph. Specifically, the proposed routing mechanism progressively infers the explanatory factors that contribute to the edge between adjacent nodes and augments the representation of the central node with factor-aware embedding information propagated from specific neighbors simultaneously via iteratively analyzing the promising subspace clusters formed by the node and its neighbors. The estimated labeling confidence matrix is also introduced to accommodate unreliable links owing to the inherent ambiguity of PLL. Moreover, we theoretically prove that the neighborhood routing mechanism will converge to the point estimate that maximizes the marginal likelihood of observed PL training examples. Comprehensive experiments over various datasets demonstrate that our approach outperforms the state-of-the-art counterparts.

Introduction
Data-driven deep learning (LeCun, Bengio, and Hinton 2015) has achieved remarkable success in numerous application scenarios. Its superiority could be primarily attributed to the accessibility of vast amount of supervised training data. Nevertheless, constrained by expertise and efforts, extensive data annotation could inevitably induce ambiguity and label noise, which might impose detrimental effects on model training (Wei et al. 2022; Hu et al. 2022; Yang, Liu, and Yin 2022). It is desirable to explore endowing modern learning systems with the power to deal with imperfect supervision. This realistic topic is referred to as weakly supervised learning (Zhou 2018).

In this paper, we focus on a critical weakly supervised learning framework called partial label learning (PLL) (Cour, Sapp, and Taskar 2011; Wu, Wang, and Zhang 2022). Specifically, PLL aims to learn a multi-class classifier from ambiguous examples where each instance is associated with a set of candidate labels, among which only one is valid. The problem of PLL naturally arises in many real-world application domains such as web mining (Luo and Orabona 2010), multimedia content analysis (Chen, Patel, and Chellappa 2017; Zeng et al. 2013), ecoinformatics (Briggs, Fern, and Raich 2012; Wang, Zhang, and Li 2022), natural language processing (Zhou et al. 2018), etc.

PLL has been extensively studied in past decades. A common thread that runs through the progress in this field is the idea of disambiguating in instances’ candidate label sets. Specifically, there are two main categories of disambiguation strategies, namely identification-based disambiguation strategies and averaging-based disambiguation strategies. Identification-based strategies (Jin and Ghahramani 2002; Nguyen and Caruana 2008) treat the ground-truth label as latent variable and assume certain parametric model to estimate the confidence of each candidate label. Averaging-based strategies (Cour, Sapp, and Taskar 2011; Gong et al. 2018) treat all candidate labels equally in the training phase and yield the final predictions via modifying their modeling outputs according to different averaging strategies. In recent years, deep learning technologies have been dedicated to reinvigorating the research of PLL (Lv et al. 2020; Zhang et al. 2022; Lyu, Wu, and Feng 2022). Deep partial-label (PL) models’ powerful capability of data representation helps to set new state-of-the-art performance for PLL algorithms. It has been empirically and theoretically proved that learning favourable representation could promote exploring potential association between instances and labels (Zhang, Wu, and Bao 2022; Bao, Hang, and Zhang 2021, 2022; Lv et al. 2020; Feng et al. 2020; Wen et al. 2021; Wang et al. 2022), which is beneficial to recovering the ground-truth label from candidate label set.

Recently, disentangled representation learning has re-
A disentangled representation prehends information about the salient factors of variation in the data, isolating information about each specific factor in only a few (or a group of) dimensions. These explanatory factors are considered to be intrinsic properties of the entities and are of fundamental importance for distinguishing between different patterns (Peng et al. 2019; Locatello et al. 2019a; Zhang et al. 2023; Wang et al. 2023a). Commonly, in PLL, the confusing candidate labels are erroneously selected for their potential relationship with the instance in terms of certain latent factors (Xu et al. 2021; Qiao, Xu, and Geng 2023). For example, wings and feathers are two representative factors for depicting flying objects such as birds and planes. The label plane could be accidentally picked as a candidate label for bird instances due to their potential similarity in terms of the latent factor wings. Nonetheless, if we could explicitly disentangle the feathers-related information from original input data, then it will be straightforward to recognize the candidate label plane as a false positive label. Though intuitively learning disentangled representation is expected to facilitate label disambiguation for PL examples, few existing works were dedicated to addressing this issue.

In this paper, we pioneer the research of disentangled partial label learning and propose a novel partial label learning algorithm named TERIAL, i.e. disenTangEd parTial Label Learning, which makes predictions according to derived disentangled representation of instances and label embeddings. In order to make full use of the topological information of input data, an undirected bipartite graph is constructed with PL examples, where edges only exist between the instances and their candidate labels. Based on the above data structure, TERIAL implements a tailored neighborhood routing mechanism to simultaneously infer the explanatory factors that cause the edge between adjacent nodes and augment the representation of the central node with factor-aware embedding information from related neighbors via iteratively analyzing the promising subspace clusters formed by the node and its neighbors. The estimated labeling confidence matrix is also introduced to help evaluate the contribution of each factor to the links and updated in every epoch to accommodate unreliable links owing to the inherent ambiguity of PLL. We theoretically prove that the neighborhood routing mechanism will converge to the point estimate that maximizes the marginal likelihood of observed PL training examples. Moreover, statistical distance correlation is employed to encourage independence between representation related with different latent explanatory factors. Comprehensive experiments over benchmark as well as real-world PL datasets validate the superiority of our proposed approach.

The rest of this paper is organized as follows. Section 2 briefly reviews related works on PLL and disentangled representation learning. Section 3 presents technical details of the proposed TERIAL approach. Section 4 reports experimental results over a broad range of PL datasets. Finally, section 5 concludes this paper.

**Related Works**

**Partial Label Learning**

As an emerging weakly-supervised learning framework, partial label learning considers inaccurate supervision where each training example is associated with multiple candidate labels among which only one corresponds to the ground-truth label (Cour, Sapp, and Taskar 2011; Wu, Wang, and Zhang 2022). Disambiguating in label space is a prevalent approach to reveal concealed labeling information for PLL. Generally, disambiguation strategies can be divided into two categories, namely identification-based strategies and averaging-based strategies. For identification-based strategies, the unknown ground-truth label is treated as latent variable whose value is estimated by the assumed parametric model which is optimized with an iterative procedure (Jin and Ghahramani 2002; Liu and Dietterich 2012; Lv et al. 2020; Chai, Tsang, and Chen 2020). For averaging-based strategies, all candidate labels of PL training examples are treated equally in the training phase while the modeling outputs are averaged with proper schemes to yield the final predictions (Cour, Sapp, and Taskar 2011; Tang and Zhang 2017; Gong et al. 2018; Zhang and Yu 2015). In recent years, efficient deep neural networks compatible with stochastic optimizers have been introduced into PLL framework to handle large-scale datasets. (Lv et al. 2020; Feng et al. 2020; Wen et al. 2021) theoretically analyse the consistency and convergency of the proposed minimal loss. (Xu et al. 2021; Qiao, Xu, and Geng 2023) make the first attempt towards instance-dependent PLL and apply probabilistic models to iteratively recover label distribution for each instance. Besides, some sophisticated techniques are borrowed from other domains to improve the generalization ability of PL learning systems, such as class activation map (Zhang et al. 2022), contrastive learning (Wang et al. 2022) and graph matching (Lyu, Wu, and Feng 2022).

Despite the progress that has been made in the study of PLL, existing learning algorithms could only perceive coarse-grained correlation between instances and labels from abstract entangled representation. In this paper, we first attempt to learn disentangled representation from PL training examples to unearth the correlation at a finer granularity of latent semantic factors, which facilitate efficient discrimination between the ground-truth label and false positive labels.

**Disentangled Representation Learning**

The purpose of disentangled representation learning is to identify the explanatory factors of variations behind the data (Bengio, Courville, and Vincent 2013; Locatello et al. 2019b). Specifically, the learned representation are expected to isolate information about each specific factor in only a few (or a group of) dimensions. Benefiting from separating out the underlying structure of the data into disjoint parts, disentangled representation is inherently more interpretable, robust to adversarial attack and capable of enhancing the gen-
eralization ability of learning systems (Wang et al. 2023b; Steenkiste et al. 2019; Reddy, Godfrey, and Balasubramanian 2022; Ma et al. 2019).

Disentangled representation learning has been widely studied in past years. (Kingma and Welling 2014) employs bayesian posterior inference and variational estimation to learn the latent generative factors of observed data. (Higgins et al. 2017) disproves the ensembling performance by setting a weight $\beta$ to aggressively penalize the KL divergence term in the variational auto-encoder. Moreover, some works further explore the roles of the information bottleneck term (Luo et al. 2019; Wu et al. 2020) and the total correlation term (Chen et al. 2018; Kim and Mnih 2018) respectively to refine the objective function of likelihood. Disentangled representation learning has been successfully applied in computer vision tasks (Higgins et al. 2017; Gidaris, Singh, and Komodakis 2018; Ma et al. 2018). In addition, the progress of learning disentangled representation on relational data, such as graph-structured data, has also been made in recent years (Wang et al. 2020; Zhang et al. 2020).

Nevertheless, the task of learning disentangled representation from PL examples to facilitate inducing ameliorative abilities such as graph-structured data, has also been made in recent years (Wang et al. 2020; Zhang et al. 2020).

The Proposed Terial Approach

**Preliminaries**

**Partial Label Learning.** Let $\mathcal{X} = \mathbb{R}^d$ denote the $d$-dimensional input space and $\mathcal{Y} = \{l_1, l_2, \ldots, l_q\}$ denote the label space with $q$ class labels. Given the PL training set $\mathcal{D} = \{(\mathbf{x}_i, S_i)\}_{1 \leq i \leq n}$, where $\mathbf{x}_i \in \mathcal{X}$ is a $d$-dimensional feature vector $[x_{i1}, x_{i2}, \ldots, x_{id}]^\top$ and $S_i \subseteq \mathcal{Y}$ is the candidate label set associated with $\mathbf{x}_i$ among which only one is the ground-truth label, PLL aims to derive a multi-class classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$ from the training set $\mathcal{D}$. In this paper, Terial is fulfilled with the labeling confidence matrix $\mathbf{Y} = [\mathbf{Y}(i,j)]_{n \times q}$ where each element $\mathbf{Y}(i,j)$ represents the estimated confidence of $l_j$ being the ground-truth label for $\mathbf{x}_i$. The matrix is initialized as Eq.(1) and the constraints $\sum_{j=1}^{q} \mathbf{Y}(i,j) = 1 (1 \leq i \leq n)$ always hold during the learning process.

$$\forall \ 1 \leq i \leq n, \ 1 \leq j \leq q : \ \mathbf{Y}(i,j) = \begin{cases} 1/|S_i|, & \text{if } l_j \in S_i \\ 0, & \text{otherwise} \end{cases}$$

**Disentangled Representation Learning.** Disentangled representation learning aims to identify the explanatory factors of variations behind the data and isolate information about each specific factor in only a few dimensions. Assuming that there are $K$ latent factors to be disentangled, for an input feature vector $\mathbf{x}_i$, its learned disentangled representation is formulated as $\mathbf{o}_i = [o_{i1}, \ldots, o_{ik}]^\top \in \mathbb{R}^{K \times d}$, where the $k$th chunked representation $o_{ik} \in \mathbb{R}^{d}$ is for describing the aspect that is pertinent to factor $k$. Benefiting from explicitly characterizing the intrinsic properties of entities, learning disentangled representation is capable of enhancing the generalization ability of learning systems.

**Overview.** To deal with the PLL problem, the proposed Terial approach makes prediction about the category to which an instance belong by computing the inner product between derived disentangled representation of the instance and label embeddings. In the training phase, PL examples are stored in an undirected bipartite graph to rigorously model the correlation between instances and labels, where the edge only exists between an instance and its candidate labels. The representation of instance nodes are derived from factor-specific mapping functions and the representation of label nodes are instantiated with learnable label embeddings. The constructed relational graph is then fed into the pipeline of stacked disentangling layers, which derive the disentangled representation of nodes in the graph via implementing a tailored neighborhood routing mechanism. In PLL, candidate labels are typically erroneously selected for their potential relationship with the instance in terms of certain latent factors. Accordingly, the proposed routing mechanism progressively infers the latent explanatory factors that cause the link between adjacent nodes and augments the representation of the central node with factor-aware information from specific neighbors simultaneously via iteratively analyzing the promising subspace clusters formed by the node and its neighbors. In this process, the estimated labeling confidences are introduced to help evaluate the contribution of each factor to the links and updated in every epoch to accommodate unreliable links owing to the inherent ambiguity of PLL. Finally predictions are made based on the inner product operation and the classification errors are backpropagated, allowing the mapping functions to better perceive and speculate factor-specific representation of instances. The label embeddings are also encoded to accurately capture each label’s own discriminative properties in a disentangled form. As a result, the predictions of unseen instances are made barely relying on the obtained mapping functions and label embeddings. The complete procedure of Terial is summarized in Appendix A.1.

**Bipartite Graph Construction**

Towards fully leveraging the structural information to rigorously model correlation between instances and labels, PL training examples are formulated as an undirected bipartite graph $G = (V, E)$, where the set of nodes $V$ is composed of a set of instance nodes $V_\mathcal{X}$ and a set of label nodes $V_\mathcal{Y}$, i.e., $V = V_\mathcal{X} \cup V_\mathcal{Y}, |V| = n + q$, and $E$ denotes the set of edges. If label $l_v$ is a candidate label of instance $x_u$, then there will exist an edge $e_{uv} \in E$ between node $u$ and node $v$.

Assuming that there are $K$ latent factors to be disentangled, then $K$ mapping functions are employed to extract factor-specific feature information from instances. Particularly, the instance $\mathbf{x}_i (1 \leq i \leq n)$ is projected into $K$ different subspaces according to Eq.(2):

$$\mathbf{z}_{ik} = \frac{f_k(\mathbf{x}_i)}{\|f_k(\mathbf{x}_i)\|_2} \in \mathbb{R}^{d}, \ 1 \leq k \leq K,$$

where the mapping function $f_k(\cdot)$ could be specified with different deep models and the $l_2$ normalization is employed.
The representation of instance node \( i \) is constructed by concatenating \( K \) factor-aware chunked representation and is denoted as \( z_i = [z_{i1}, \ldots, z_{iq}]^\top \in \mathbb{R}^{K \cdot \Delta d} \). Since labels do not have inherent feature vectors, the representation of label nodes are similarly instantiated by embeddings with learnable parameters \( y_j = [y_{j1}, \ldots, y_{jq}]^\top \in \mathbb{R}^{K \cdot \Delta d} (1 \leq j \leq q) \).

Provided that the instance \( x_i \) does contain meaningful information about the explanatory factor \( k \), we assume that \( z_{ik} \) approximately characterizes the \( k \)-th aspect of node \( i \). Nonetheless, \( z_i \) could not be straightforwardly employed to serve as \( \Theta \) since the raw input data is typically insufficient to completely depict an entity in the real world (Bengio, Courville, and Vincent 2013; Locatello et al. 2019b). Accordingly, considering the prospective correlation between the instance and its candidate labels, TERIAL takes advantage of comprehensive information propagated from neighboring nodes to augment the preliminary representation \( z_\text{u} \) of the central node \( u \). The constructed graph-structure data are then fed into a pipeline consisting of stacked disentangling layers, which progressively enrich the node representation through a tailored neighborhood routing mechanism.

The Neighborhood Routing Mechanism

The disentangling layer is deployed to yield rich and accurate disentangled representation of instances. It augments the fed nodes’ representation through a tailored neighborhood routing mechanism \( g(\cdot) \). Let \( w_u \) denote the input representation of node \( u (u \in V) \) for a disentangling layer. Next we will elaborate how the proposed routing mechanism derives the enriched representation \( c_u = g(w_u, \{ w_v | v \in \mathcal{N}_u \}) \), where \( \mathcal{N}_u \) denotes the set of neighboring nodes of node \( u \). Without loss of generality, we will focus on the message-passing process which takes the instance node as the center. The refined representation of label nodes could be achieved in the same manner.

For the central (instance) node \( u (u \in V_u) \), its links with candidate label nodes could be attributed to their potential relationship associated with certain explanatory factors. Moreover, owing to the inherent ambiguity of PLL, there could exist unreliable edges in the graph, i.e., the degree of correlation between one node and its neighbors could vary a lot. As a result, the proposed routing mechanism is required to simultaneously identify the true latent explanatory factors that cause the link between adjacent nodes and accurately quantify the bonds between the central node and its neighbors.

The first-order and the second-order proximity are widely accepted explanations for the existence of a link in the graph (Granovetter 1973). They are also the essential ingredients of many graph-based algorithms (Wu et al. 2020; Wang et al. 2020). Based on them, we propose two rational hypotheses, which are the foundations of the induced neighborhood routing mechanism.

**Hypothesis 1.** If a large subset of neighbors of node \( u \) have similar representation w.r.t the latent factor \( k \), i.e., they form a cluster in the \( k \)-th subspace, then factor \( k \) is likely to be a clue to the connections between node \( u \) and these neighboring nodes.

**Hypothesis 2.** If the representation of node \( u \) is similar to that of its neighboring node \( v \) in terms of aspect \( k \), then the latent factor \( k \) is likely to be the reason why the two nodes are connected.

To propagate factor-specific information from neighbors to the central node, Hypothesis 1 inspires us to search for the largest cluster in each of the \( K \) projected subspaces. Since the central node \( u \) is not involved in the clustering procedure, Hypothesis 1 is robust under the scenario where \( w_v \) is noisy or incomplete. In addition, when performing clustering in the \( k \)-th subspace, irrelevant neighboring nodes will be automatically pruned, because their projected representation could be noises and will not form a large enough cluster. Though the clustering procedure is usually time-consuming due to the requirement of extensive iterations for convergence, Hypothesis 2 indicates that the value of \( w^{\prime}_{uk} w^{\prime}_{vk} \) could be a hint on the factors that cause the link between nodes. Therefore, serving as a strong prior, Hypothesis 2 is adopted to guide the clustering process for faster convergence. Based on the above consideration, we introduce the proposed neighborhood routing mechanism.

Let \( p_{u,v}^{k}(1 \leq k \leq K) \) quantify the influence of factor \( k \) to the link between the central node \( u \) and its neighboring node \( v \). According to Hypothesis 2, \( p_{u,v}^{k} \) is initialized as:

\[
p_{u,v}^{k(0)} = \frac{\exp(w^{\prime}_{uk} w^{\prime}_{vk})}{\sum_{k'=1}^{K} \exp(w^{\prime}_{uk'} w^{\prime}_{vk'})} \quad (1 \leq k \leq K),
\]

where \( \tau \) is the smoothing factor which controls the hardness of the assignment and is set as \( \tau = 1 \) in this paper.

Inspired by Hypothesis 1, we then iteratively search for the largest cluster in \( K \) subspaces. Reasonably, the augmented representation of the central node \( u \) is set to be the clustering center of its neighborhoods, and the routing mechanism is formulated as follows:

\[
c_{uk}^{(t)} = \frac{w^{\prime}_{uk} + \sum_{v \in \mathcal{N}_u} p_{u,v}^{k(t-1)} w^{\prime}_{vk}}{\|w^{\prime}_{uk} + \sum_{v \in \mathcal{N}_u} p_{u,v}^{k(t-1)} w^{\prime}_{vk}\|_2},
\]

where \( c_{uk}^{(t)} \) denotes the temporary clustering center corresponding to the \( k \)-th subspace in the \( t \)-th iteration. Furthermore, in order to alleviate the detrimental effect of unreliable links corresponding to false positive candidate labels in the graph, estimated labeling confidences are introduced to help update the correlation coefficient \( p_{u,v}^{k(t)} \) in each iteration:

\[
p_{u,v}^{k(t)} = Y(u,v) \frac{\exp(c_{uk}^{(t)} \cdot w^{\prime}_{vk})}{\sum_{k'=1}^{K} \exp(c_{uk}^{(t)} \cdot w^{\prime}_{vk'})} \quad (1 \leq k \leq K),
\]

The clustering center and correlation coefficients are iteratively updated in an alternative manner. After \( T \) iterations\(^1\), we finally obtain the constructed clustering center \( c_{uk}^{(T)} \) in each subspace and the derived representation of

\[^1\text{In this paper, the maximum number of iterations is set to be } T = 6, \text{ which suffices to yield stable performance for the proposed approach.}\]
central node $u$ is set as $c_u = g(w_u, \{w_v|v \in \mathcal{N}_u\}) = \{c_{u_1}^{(T)} \ldots , c_{u_k}^{(T)}\}^T$. 

**Theoretical Analysis.** We theoretically analyze the convergence property of the proposed neighborhood routing mechanism and deduce the following Theorem 1. Its proof is provided in Appendix A.2.

*Theorem 1.* For the vMF mixture model, the proposed neighborhood routing mechanism could be interpreted from the expectation-maximization perspective. Particularly, it converges to a point estimate of $\{c_{u_k}\}_{k=1}^K$ that maximizes the marginal likelihood $p(\{w_{i_k} : i \in \{u\} \cup \mathcal{N}_u, 1 \leq k \leq K\}; \{c_{u_k}\}_{k=1}^K)$.

**Multi-Layer Stacking.** Above we elaborate how to utilize the proposed routing mechanism to aggregate factor-specific embedding information in the disentangling layer. Furthermore, we argue that it is feasible to stack $L$ disentangling layers to explore rich semantics from multi-hop neighbors when producing a node’s representation. Specifically, let $w_u^{(\beta)}$ and $o_u^{(\beta)}$ respectively denote the input and output representation of node $u$ for the $\beta$th disentangling layer, where $u \in V$ and $\beta \in \{1, \ldots , L\}$. We demand that $w_u^{(\beta)} = o_u^{(\beta-1)}(\beta \in \{2, \ldots , L\})$ among multiple layers. In order to avoid the over-smoothing issue, which is a common problem in graph learning (Chen et al. 2020), the nodes’ representation in each layer are progressively assigned according to Eq.(6):

$$o_u^{(\beta)} = \alpha \cdot w_u^{(\beta)} + (1 - \alpha) \cdot g(w_u^{(\beta)}, \{w_v^{(\beta)}|v \in \mathcal{N}_u\}),$$

where the balancing factor is set as $\alpha = 0.6$ in this paper.

**Independence Modeling**

Though the proposed routing mechanism encourages representation conditioned on different explanatory factors to be different from each other, there still might exist redundancy among them. Accordingly, we employ the distance correlation (Székely, Rizzo, and Bakirov 2007; Wang et al. 2020) as a regularizer to encourage representation associated with different latent factors to be independent. Specifically, distance correlation is a statistical measure that is capable of characterizing independence of any two paired vectors, from their both linear and nonlinear relationships. The derived loss function is formulated as:

$$L_{ind} = \sum_{k=1}^K \sum_{k'=k+1}^K \text{dCor}(E_k, E_{k'}),$$

where $E_k = \{o_1^{(L)}; \ldots , o_{n+k}^{(L)}\} \in \mathbb{R}^{(n+q)\times \Delta d}$. The function of distance correlation $\text{dCor}(\cdot, \cdot)$ is defined as:

$$\text{dCor}(E_k, E_{k'}) = \frac{\text{dCov}(E_k, E_{k'})}{\sqrt{\text{dVar}(E_k) \cdot \text{dVar}(E_{k'})}},$$

where $\text{dCov}(\cdot, \cdot)$ denotes the distance covariance between two matrices and $\text{dVar}(\cdot)$ denotes the distance variance of the matrix. We refer readers to (Székely, Rizzo, and Bakirov 2007) for the details of calculation.

---

**Model Optimization and Prediction**

Through the processing of stacked disentangling layers, we eventually obtain the disentangled representation $o_i^{(L)}(1 \leq i \leq n)$ of instances. Besides, the label embeddings $y_j^{(1 \leq j \leq q)}$ are also expected to be disentangled through model training. As a result, we simply use inner product $s_{i,j} = o_i^{(L)} \cdot y_j$ as the score of $l_j$ being the ground-truth label of $x_i$. The classification loss is defined as:

$$L_{ce} = \sum_{i=1}^n \sum_{l_j \in S_i} Y(i,j)l_i(s_{i,j}, l_j),$$

where $l_i(\cdot, \cdot)$ denotes the cross-entropy loss function.

During training, the empirical losses $L_1 = L_{ce}$ and $L_2 = L_{ce} + L_{ind}$ are optimized alternatively to prevent the training process from falling into local minimas (Goodfellow et al. 2014; Ren et al. 2015; Wang et al. 2020). The labeling confidence matrix is re-estimated at the end of each epoch according to Eq.(10):

$$Y(i,j) = \begin{cases} \frac{s_{i,j}}{\sum_{i,j' \in S_i} s_{i,j'}}, & \text{if } l_j \in S_i \\ 0, & \text{otherwise.} \end{cases}$$

After training, mapping functions $f_k(1 \leq k \leq K)$ are allowed to better perceive and speculative factor-specific information from the raw input data. The learned embeddings of label nodes are also encoded with each label’s own discriminative properties in a disentangled form. In the testing phase, the disentangled representation of unseen instance $x'$ is derived as $x'^{out} = [f_1(x')^\top, \ldots , f_K(x')^\top]^\top$ and its score about label $l_j$ is set as $s'_{i,j} = x'^{out} \cdot y_j$. The final prediction is made by $l^* = \arg \max_{l_j \in Y} s'_{i,j}$.

**Experiments**

In this section, comprehensive experiments are conducted to verify the effectiveness of our proposed TERRIAL approach.

**Classification Performance**

*Datasets.* Five popular benchmark datasets are employed to generate synthetic PL datasets, including MNIST (LeCun et al. 1998), Kuzushiji-MNIST (abbreviated as KMNIST) (Cinun et al. 2018), Fashion-MNIST (abbreviated as FMNIST) (Xiao, Rasul, and Vollgraf 2017), SVHN (Netzer et al. 2011) and CIFAR-10 (Krizhevsky, Hinton et al. 2009). More details about these benchmark datasets are shown in Appendix B.1. Following the conventional experimental protocol in PLL (Hüllermeier and Beringer 2006; Cour, Sapp, and Taskar 2011; Gong et al. 2018; Liu and Dietterich 2012), the benchmark datasets are corrupted to PL datasets with the parameter $r$. Specifically, for each instance, $r$ false positive class labels are randomly selected to construct the candidate label set along with the ground-truth label. In this subsection, the number of false positive class labels is set as $r \in \{3, 5, 7\}$. Moreover, instance-dependent PL datasets (Qiao, Xu, and Geng 2023; Wu, Wang, and Zhang 2022) are also generated according to the same strategy utilized in (Xu et al. 2021), which made the first attempt towards instance-dependent PLL.
<table>
<thead>
<tr>
<th>Datasets</th>
<th>Method</th>
<th>$r = 3$</th>
<th>$r = 5$</th>
<th>$r = 7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>Ours</td>
<td>98.10% ±0.12%</td>
<td>97.71% ±0.09%</td>
<td>96.57% ±0.06%</td>
</tr>
<tr>
<td></td>
<td>PRODEN</td>
<td>97.85% ±0.03%</td>
<td>97.39% ±0.13%</td>
<td>95.77% ±0.24%</td>
</tr>
<tr>
<td></td>
<td>RC</td>
<td>97.97% ±0.16%</td>
<td>97.49% ±0.17%</td>
<td>96.19% ±0.19%</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>97.80% ±0.12%</td>
<td>97.43% ±0.19%</td>
<td>96.03% ±0.15%</td>
</tr>
<tr>
<td></td>
<td>LW</td>
<td>97.26% ±0.09%</td>
<td>97.19% ±0.22%</td>
<td>95.34% ±0.20%</td>
</tr>
<tr>
<td></td>
<td>VALEN</td>
<td>96.54% ±0.19%</td>
<td>96.21% ±0.27%</td>
<td>94.72% ±0.26%</td>
</tr>
<tr>
<td></td>
<td>CAVL</td>
<td>97.79% ±0.08%</td>
<td>96.69% ±0.12%</td>
<td>95.52% ±0.16%</td>
</tr>
<tr>
<td>KMNST</td>
<td>Ours</td>
<td>89.24% ±0.11%</td>
<td>87.53% ±0.19%</td>
<td>82.97% ±0.16%</td>
</tr>
<tr>
<td></td>
<td>PRODEN</td>
<td>88.32% ±0.03%</td>
<td>88.91% ±0.11%</td>
<td>76.32% ±0.14%</td>
</tr>
<tr>
<td></td>
<td>RC</td>
<td>88.78% ±0.07%</td>
<td>88.98% ±0.11%</td>
<td>80.55% ±0.13%</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>88.48% ±0.05%</td>
<td>86.84% ±0.17%</td>
<td>79.89% ±0.10%</td>
</tr>
<tr>
<td></td>
<td>LW</td>
<td>87.82% ±0.09%</td>
<td>84.66% ±0.22%</td>
<td>78.41% ±0.19%</td>
</tr>
<tr>
<td></td>
<td>VALEN</td>
<td>86.71% ±0.23%</td>
<td>80.79% ±0.35%</td>
<td>73.26% ±0.38%</td>
</tr>
<tr>
<td></td>
<td>CAVL</td>
<td>87.14% ±0.16%</td>
<td>82.64% ±0.26%</td>
<td>77.63% ±0.28%</td>
</tr>
<tr>
<td>FMNST</td>
<td>Ours</td>
<td>88.68% ±0.08%</td>
<td>87.95% ±0.09%</td>
<td>86.69% ±0.05%</td>
</tr>
<tr>
<td></td>
<td>PRODEN</td>
<td>87.51% ±0.11%</td>
<td>87.42% ±0.16%</td>
<td>86.02% ±0.23%</td>
</tr>
<tr>
<td></td>
<td>RC</td>
<td>88.17% ±0.10%</td>
<td>87.42% ±0.16%</td>
<td>86.02% ±0.23%</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>87.89% ±0.23%</td>
<td>87.23% ±0.13%</td>
<td>85.87% ±0.12%</td>
</tr>
<tr>
<td></td>
<td>LW</td>
<td>87.93% ±0.04%</td>
<td>86.34% ±0.08%</td>
<td>84.57% ±0.13%</td>
</tr>
<tr>
<td></td>
<td>VALEN</td>
<td>83.24% ±0.12%</td>
<td>80.21% ±0.13%</td>
<td>83.86% ±0.22%</td>
</tr>
<tr>
<td></td>
<td>CAVL</td>
<td>88.11% ±0.08%</td>
<td>87.48% ±0.10%</td>
<td>83.86% ±0.16%</td>
</tr>
<tr>
<td>SVHN</td>
<td>Ours</td>
<td>95.24% ±0.09%</td>
<td>94.77% ±0.12%</td>
<td>93.92% ±0.07%</td>
</tr>
<tr>
<td></td>
<td>PRODEN</td>
<td>94.54% ±0.17%</td>
<td>94.12% ±0.21%</td>
<td>92.78% ±0.26%</td>
</tr>
<tr>
<td></td>
<td>RC</td>
<td>94.76% ±0.12%</td>
<td>94.25% ±0.11%</td>
<td>92.92% ±0.16%</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>94.52% ±0.12%</td>
<td>93.83% ±0.11%</td>
<td>92.61% ±0.16%</td>
</tr>
<tr>
<td></td>
<td>LW</td>
<td>94.44% ±0.28%</td>
<td>94.19% ±0.09%</td>
<td>92.66% ±0.14%</td>
</tr>
<tr>
<td></td>
<td>VALEN</td>
<td>88.93% ±0.25%</td>
<td>85.36% ±0.28%</td>
<td>80.86% ±0.17%</td>
</tr>
<tr>
<td></td>
<td>CAVL</td>
<td>72.05% ±0.13%</td>
<td>49.96% ±0.19%</td>
<td>35.65% ±0.26%</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>Ours</td>
<td>79.01% ±0.23%</td>
<td>76.61% ±0.25%</td>
<td>65.07% ±0.37%</td>
</tr>
<tr>
<td></td>
<td>PRODEN</td>
<td>78.87% ±0.29%</td>
<td>75.59% ±0.34%</td>
<td>72.34% ±0.46%</td>
</tr>
<tr>
<td></td>
<td>RC</td>
<td>78.57% ±0.32%</td>
<td>74.92% ±0.44%</td>
<td>71.16% ±0.38%</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>77.88% ±0.29%</td>
<td>73.12% ±0.35%</td>
<td>58.91% ±0.28%</td>
</tr>
<tr>
<td></td>
<td>LW</td>
<td>77.92% ±0.26%</td>
<td>74.65% ±0.28%</td>
<td>63.02% ±0.29%</td>
</tr>
<tr>
<td></td>
<td>VALEN</td>
<td>71.31% ±0.38%</td>
<td>42.17% ±0.39%</td>
<td>31.66% ±0.31%</td>
</tr>
<tr>
<td></td>
<td>CAVL</td>
<td>77.93% ±0.25%</td>
<td>50.94% ±0.26%</td>
<td>26.30% ±0.45%</td>
</tr>
</tbody>
</table>

Table 1: Classification accuracy (mean ± std) of each comparing algorithm on corrupted benchmark datasets (# false positive labels $r \in \{3, 5, 7\}$). The best results among methods are highlighted in bold.

We also conduct comparative experiments on real-world PL datasets. The details and empirical results are reported in Appendix C.

**Comparing Methods.** To verify the effectiveness of our proposed approach, [Terial] is compared with six state-of-the-art PLL approaches including PRODEN (Lv et al. 2020), RC, CC (Feng et al. 2020), LW (Wen et al. 2021), VALEN (Xu et al. 2021), CAVL (Zhang et al. 2022). More details about comparing algorithms are shown in Appendix B.2. Their hyper-parameters are specified according to the suggested parameter settings or searched to maximize the accuracy on a validation set containing 10% of the training samples. For [Terial], the assumed number of latent factors is set as $K = 10$ on datasets of MNIST, KMNIST, FMNST and $K = 8$ on SVHN and CIFAR-10. The number of disentangling layers is set as $L = 2$, which is sufficient to achieve state-of-the-art performance for our proposed approach. We use different backbones according to the target datasets. To be more specific, we employ the base model as a 3-layer MLP ($d = 300 − 100 − 10$) on MNIST, KMNIST, FMNST and a 34-layer ResNet on SVHN and CIFAR-10. For the fairness of comparison, the mapping function $f_k(\cdot)$ of [Terial] is set as the base model removing the classification layer. For all DNN based methods, we search the initial learning rate from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ and the weight decay from $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. The mini-batch size is set as 256 and the number of epochs is set as 200. All the models are trained with stochastic gradient descent (SGD) (Robbins and Monro 1951) optimizer with momentum 0.9. All experiments are repeated for 5 times with different random seeds, and the average accuracy and the standard deviation are reported.

**Empirical Results.** The predictive performance (mean ± std) of comparing algorithms on benchmark datasets corrupted by $r \in \{3, 5, 7\}$ are reported in Table 1, where the best results are highlighted in bold. In addition, the corresponding results of pairwise t-test at 0.05 significance level are reported in Appendix B.3. Out of the 90 statistical comparisons (6 comparing algorithms × 5 datasets × 3 settings of $r$), [Terial] outperforms all other state-of-the-art algorithms in 86 cases, with only two losses of comparisons against PRODEN and RC on CIFAR-10. These impressive results suggest that the learned disentangled representation from PL examples could more accurately and profoundly characterize the essential properties of instances thus helping the classifier to disambiguate and predict.

In addition, the predictive performance (mean ± std) of comparing algorithms on instance-dependent benchmark datasets, are reported in Appendix B.4. We observe that [Terial] could still achieve best results in most cases, with the only exception on CIFAR-10 against PRODEN and RC. This indicates that the disentangled representation could better clarify the dependency between instances and labels.

**Further Studies**

In this section, we investigate the rationality and effectiveness of some designs of our [Terial] approach.

**Impact of Disentangling Layers.** The disentangling layers implemented by proposed neighborhood routing mechanism are the core of [Terial]. Here we investigate how the number of such layers $L \in \{0, 1, 2, 3\}$ affects the model’s classifi-
cification performance. The TERIAL approach with $L$ disentangling layers is denoted as TERIAL-$L$ and the corresponding results on corrupted datasets of KMNIST($r = 3$) and SVHN($r = 7$) are summarized in Table 2. We can observe that the neighborhood routing mechanism really helps capture sophisticated and complete disentangling information since TERIAL-0 is inferior to all of other compared approaches. Moreover, we find that gather information from multi-hop neighbors could lead to better performance though the margin could be small (between TERIAL-2 and TERIAL-3) as $L$ gets larger. As a result, in this paper we set $L = 2$ and further improvement is expected to be achieved through fine-tuning $L$.

For TERIAL, $K$ is an important hyper-parameter which decides the assumed number of latent explanatory factors. Accordingly, we investigate how the predictive performance varies with the parameter $K$ on the corrupted datasets of KMNIST($r = 5$) and FMNIST($r = 5$). As is shown in Table 3, the performance corresponding to $K = 1$ is inferior to all other comparing results. This suggests that learned abstract entangled representation is of limited help to the learning system in distinguishing between different categories. Moreover, we find that increasing $K$ from 1 to 10 could absolutely enhance the model’s classification performance. This strongly justifies that disentangled representation could facilitate disambiguation for PL examples. However, the classification accuracy drops when $K$ gets larger. It’s likely that as $K$ increases, the dimensionality of chunked representation $\Delta d = \frac{d'}{K}$ becomes smaller and thus the disentangled components gradually lose their ability to completely portray the explanatory factors of instances.\(^1\)

In addition, the learning rate $\alpha$ and the maximum number of iterations $T$ are key hyper-parameters for the routing mechanism. The ablation studies on these two parameters are provided in Appendix D.1.

**Impact of Independence Modeling.** In the independence modeling module, the statistical distance correlation is applied to encourage independence between factor-specific chunked representation. The derived loss function $L_{\text{ind}}$ are optimized alternatively to prevent the training process from falling into local minimas. We evaluate the performance of TERIAL with(w/l) or without(w/o) the independence modeling module on benchmark datasets corrupted by the instance-dependent strategy, and the results are reported in Appendix D.2. We can observe that in all five benchmark datasets removing independence modeling module will lead to performance drop.

**Visualization**

In this part, we visualize the disentangled representation learned by TERIAL from multiple perspectives. The corresponding results of the best baseline RC that is theoretically proved risk-consistent are also reported for comparison.\(^4\)

Firstly, in Fig. 1, values of correlation between the elements of the 100-dimensional representation learned by RC and TERIAL with 10 latent factors on the test dataset of KMNIST($r = 5$) are presented in a heat-map. The heat-map for TERIAL exhibits ten clear diagonal blocks while there are not obvious correlation between features learned by RC. This indicates that disentangled representation learned by TERIAL could definitely capture mutually exclusive information associated with different explanatory factors.

In addition, we visualize the representation produced by TERIAL and RC on the test dataset of SVHN($r = 5$) in Appendix E. We can observe that in the t-SNE (Van der Maaten and Hinton 2008) visualization where representation is produced by RC, the class boundaries are not clear while some classes overlap. On the contrary, representation produced by TERIAL are more distinguishable and lead to well-separated clusters. This suggests that TERIAL could capture high-quality representation from PL examples through disentangled representation learning, thus improving the generalization ability of the learned classifier.

**Conclusion**

In this paper, we make the first attempt towards disentangled partial label learning. A novel PLL approach named TERIAL is proposed, which makes predictions based on derived disentangled representation of instances and label embeddings. Specifically, in TERIAL, the partial label examples are stored in an undirected bipartite graph to rigorously model the correlation between instances and labels. Then TERIAL employs a tailored neighborhood routing mechanism to progressively infer the explanatory factors that cause the edge between adjacent nodes and augment the representation of the central node by propagating proper disentangling information from related neighbors. The estimated labeling confidence matrix is also introduced to accommodate unreliable links due to the inherent ambiguity of PLL. In addition, the statistical distance correlation is employed to encourage the independence between representation corresponding to different latent factors. Extensive experiments over various datasets verify the effectiveness of our proposed approach.

---

\(^1\)Here $d' = 100$, which is the number of neurons in the second hidden layer of the base model.

\(^4\)The representation provided by TERIAL denotes the output of mapping functions and the representation provided by RC denotes the output of the base model removing the classification layer.


