2024 IEEE Conference on Artificial Intelligence (CAI)

Fine-Grained Partial Label Learning

Cheng Chen[1,2,3], Yueming Lyu[2,3], Xingrui Yu[2,3], Jing Li[2,3], Ivor W Tsang[2,3,4]

The Australian Artificial Intelligence Institute (AAII), University of Technology[1]*, Sydney, Australia*

CFAR, Agency for Science, Technology and Research[4]*,Singapore*

IHPC, Agency for Science, Technology and Research[3]*, Singapore*

School of Computer Science and Engineering, College of Computing and Data Science, Nanyang Technological University[4]*, Singapore*

***Abstract*—Partial Label Learning (PLL) involves associating each instance with a candidate label set, which includes the true label along with false positive labels. Traditional PLL approaches typically assume that these false positive labels follow either a uniform or a non-uniform structure, based on the true label of an instance. However, these assumptions do not fully encompass all aspects of real-world scenarios, especially in cases of fine-grained objects with subtle differences in appearance. Therefore, we introduce a fine-grained PLL task specifically designed to handle candidate sets that comprise the true label, partially non-uniform fine-grained labels manifesting in a hierarchical structure, and other false positive labels. Nevertheless, the incorporation of these partially non-uniform, fine-grained labels can result in an intractable posterior, leading to an inconsistent classifier. To address this issue, we propose a Specific Classes-Hierarchical (SCH) regularisation that ensures classifier consistency. Subsequently, the Global Label-Hierarchical-wise Embedding (GLHE) regularisation is introduced, using the refined pseudo labels of positive samples from the obtained consistent classifier. This allows the model to learn more distinctive representations from fine-grained instances. Our extensive experiments on datasets such as CIFAR-10, CIFAR-100, and CUB-200 have demonstrated the effectiveness of our approach.**

Example 1: The Uniform Probability Distribution, as proposed by [16], is represented by $P(\vec{Y} = \vec{y}|y) = \frac{1}{2^{c-1}-1}$ which is equal to $\frac{1}{3}$ for $k = 3$. We introduce the Partially Non-Uniform Probability Distribution, defined as $P(\vec{Y} = \vec{y}|y, y', x) \cdot P(y'|y)$. The variable x represents the level of image blurriness, which can be challenging to quantify; therefore, we have omitted it in the (I). The potential candidate label sets are $\vec{y}_1 = \{1\}$, $\vec{y}_2 = \{2\}$,$\vec{y}_3 = \{3\}$, $\vec{y}_4 = \{1, 2\}$, $\vec{y}_5 = \{1, 3\}$, $\vec{y}_6 = \{2, 3\}$. The probability $p(\vec{Y}|y)$ is transformed to $p(\vec{Y} = \vec{y}|y, y')$ by considering y', with $P(y'|y)$. For instance, $P(y' = 2|y = 1) = 0.8$ can be denoted as given the true label is 1, there is an 80% chance of it being misclassified as the false positive label 2. In addition, $p(\vec{Y} = \vec{y}_4|y = 1, y' = 2)$ can be interpreted as, given a true label $y = 1$ and fine-grained label $y' = 2$ of an instance x, there is a $1/3 \times 80\%$ probability that its candidate label set will be $\vec{y}_4$, and a $1/3 \times 20\%$ chance that it contains only the true label $y = 1$ itself. The 20% and 80% probabilities are assumed to be given, based on the fine-grained transition matrix $P(y' = 2|y = 1)$. The details of the used fine grained transition matrix can be referred to appendix page Figure 8.

	$P(\vec{Y}\|y)$						$P(\vec{Y}\|y, y', x)P(y'\|y)$					
	$\vec{y}_1$	$\vec{y}_2$	$\vec{y}_3$	$\vec{y}_4$	$\vec{y}_5$	$\vec{y}_6$	$\vec{y}_1$	$\vec{y}_2$	$\vec{y}_3$	$\vec{y}_4$	$\vec{y}_5$	$\vec{y}_6$
$y = 1, y' = 1$	$\frac{1}{3}$	0	0	$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{0.2}{3}$	0	0	$\frac{0.2}{3}$	$\frac{0.2}{3}$	0
$y = 1, y' = 3$	-	-	-	-	-	-	0	0	0	0	0	0
$y = 2, y' = 2$	0	$\frac{1}{3}$	0	$\frac{1}{3}$	0	$\frac{1}{3}$	0	$\frac{0.7}{3}$	0	$\frac{0.7}{3}$	0	$\frac{0.7}{3}$
$y = 3, y' = 1$	-	-	-	-	-	-	0	0	0	0	0	0
$y = 3, y' = 3$	0	0	$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	0	0	$\frac{0.6}{3}$	0	$\frac{0.6}{3}$	$\frac{0.6}{3}$
$y = 1, y' = 2$	-	-	-	-	-	-	0	0	0	0	$\frac{0.8}{3}$	0
$y = 2,y' = 1$	-	-	-	-	-	-	0	0	0	$\frac{0.3}{3}$	0	0
$y = 2, y' = 3$	-	-	-	-	-	-	0	0	0	0	0	0
$y = 3,y' = 2$	-	-	-	-	-	-	0	0	0	0	0	$\frac{0.4}{3}$

TABLE I: Comparison of Probability Distributions: Uniform versus Partially Non-Uniform

I. Introduction

Partial Label Learning (PLL), as discussed in [1, 2], primarily focuses on scenarios where each instance is associated with a set of candidate labels. This set contains a true label along with either uniform or non-uniform false positive labels. In addition, [3] proposes instance-dependent partial label learning, which incorporates the feature aspect into the label generation process. To address the label ambiguity inherent in partial label learning, various frameworks have been proposed. Probabilistic graphical model-based methods [4, 5, 6, 7], as well as clustering-based or unsupervised approaches [8], leverage graph structures and prior information in the feature space for label disambiguation. Similarly, average-based perspective methods [1, 4] assume a uniform treatment of all candidates; however, they are susceptible to false positive labels, which can lead to misleading predictions. To mitigate this issue, an identification-based method [9] was introduced, treating the true label as a latent variable to better handle label disambiguation. Following this, representative approaches such as the maximum margin method [10, 11, 12, 13] have been used for further label clarification. More recently, self-training perspective methods [14, 15, 16] have emerged, demonstrating promising performance. In a related development, techniques such as those presented in [17, 18] use augmented inputs to learn features from unlabeled sample data. The learning objective here is to differentiate between similar and dissimilar parts of the input, thereby maximizing the acquisition of high-quality representations.
In addition, we believe that the fine-grained PLL is realistic since it considers human cognition behaviour [19, 20, 21] into the annotation process. For instance, in the fine-grained PLL problem, the quality of the annotation is significantly influenced by the human visual system [22, 23], which greatly affects how an annotator makes decisions, hindering annotator to accurately measuring length, discerning gray levels, and dealing with complex backgrounds [24].

Although the fine-grained Partial Label Learning (PLL) approach is realistic, it presents inherent challenges in terms of learnability. This is particularly evident when considering the variables of fine-grained features and subtle differences between categories, which result in a partially non-uniform structure. In the case of a highly similar fine-grained label learning task, where the number of fine-grained labels approaches the total size of the class, intractability issues arise. This implies that accurately predicting the true posterior probability becomes exceedingly challenging. Such a situation makes the accurate prediction of the true label nearly impossible,

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00116

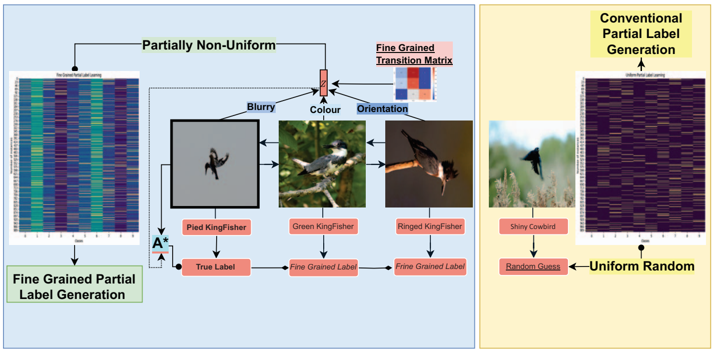


Fig. 1: The left heat-map shows the distribution of fine-grained labels in our problem scenario, while the right heat-map demonstrates the label distribution of conventional partial labels.

particularly if prior knowledge about the generation of fine-grained labels is not incorporated into the modelling process. To address this issue, previous studies [25, 26] have shown that patterns or statistical summaries of the pitfalls of the human visual system can actually be studied, meaning that the dataset annotation structure or fine-grained partial label hierarchy, which is accessible, can be formulated. In our case, the correlation among specific classes (or fine-grained labels) can be accessed or estimated based on the dataset annotations that include superclass information. For each fine-grained label, we can establish correlations by observing its superclass and then identifying connections to other classes that belong to the same superclass. The superclass information considers broader categories that group together fine-grained labels. Failing to consider such correlations during the model's encoding can lead to inconsistencies in the classifier and risk function. To address this, we incorporate the correlation of fine-grained labels, denoted as the fine-grained transition matrix, through specific class-hierarchical regularisation (see Eq. 9) and global label-hierarchical-wise embedding regularisation (see Eq. 10) to disambiguate labels. In addition, we introduce contrastive prototype regularisation, aiming to enhance the precision of the prototypes by leveraging the prototype vector margin. The **main contributions** of the work are summarised:

- We introduce a realistic fine-grained PLL, in which the true category has marginally distinctive features, while the other categories exhibit subtle differences in their features compared to the true category.
- We propose specific class-hierarchical regularisation for label disambiguation. Additionally, this method can be universally applied to other related fine-grained PLL learning methods. Thereafter, global label-hierarchical-wise embedding regularisation is proposed, using the positive samples from the consistent classifier to learn more distinct representation from fine-grained instances. Lastly, we propose contrastive prototype regularisation for updating pseudo labels.
- A new ambiguity condition (5) is proposed for fine-grained PLL. Theoretically, we have proven that the method is a Classifier-Consistent Risk Estimator.

II. Fine Grained Partial Label Problem Setting

Notations: Given a feature space $\mathcal{X} \subseteq \mathbb{R}^d$ and a fully supervised label space defined as $\mathcal{Y} = \{1, \ldots, c\}$, with the number of classes denoted as $|c| > 2$, the partially non-uniform fine-grained partial label set has a space of $\vec{\mathcal{Y}} := \{\vec{y} \mid \vec{y} \subseteq \mathcal{Y}\}$. This implies that there are a total of 2^c possible selections of subsets in $\mathcal{Y}$, which include the empty set and the full candidate set. Under the paradigm of partially non-uniform fine-grained partial labels, each instance $X \in \mathcal{X}$ has a candidate set of $\vec{Y} \in \vec{\mathcal{Y}}$. The distribution of the fine-grained partial label dataset, denoted as $\vec{D}$, includes elements $(X, \vec{Y})$ from the Cartesian product $\mathcal{X} \times \vec{\mathcal{Y}}$. The objective is to learn a classifier from the partially non-uniform fine-grained partial label sample of size m, defined as $\vec{\mathcal{D}} = \{(X_1, \vec{Y}1), \ldots, (Xm, \vec{Y}_m)\}$, which are independently and identically drawn from the distribution $\vec{D}$. The aim is for the classifier to accurately assign the true labels to the testing dataset.

III. Distinction between Conventional (Uniform and Non-Uniform) and Fine-Grained PLL using Causal Graph Model

In this section, we have compared conventional (uniform and non-uniform) and fine-grained partial labels using a causal graph model. It allows a clear visual representation of fine-grained label generation. Figure $2(a)$ depicts the generation process for a uniform partial label. The generation of $\vec{Y}$ depends only on the true category Y, and no additional variables are considered. This scenario is quite unrealistic where the labels in $\vec{Y}$ are generated with uniform probability. Thus, Figure $2(b)$ presents a non-random approach for the generation of $\vec{Y}$ by considering variables Y and Y'. It offers a more accurate depiction of how partial labels should realistically be generated. Here, Y' denotes categories with subtle differences from the true category Y, resulting in a non-uniform partial label. In the fine-grained partial label scenario, the fine-grained feature variable X is used to represent the nature of subtle differences in the features of fine-grained objects. For example, in a dataset of bird species, the true category Y could be a specific species

$$\begin{tikzcd} Y \arrow[r] & \vec{Y} \end{tikzcd}$$

(a) Uniform Partial Label

$$\begin{tikzcd} Y \arrow[r] & Y' \arrow[r] & \vec{Y} \end{tikzcd}$$

(b) Non-Uniform Partial Label

$$\begin{tikzcd} Y \arrow[r] & Y' \arrow[r] & \vec{Y} \\ & X \arrow[u] \arrow[ur] & \end{tikzcd}$$

(c) **Fine Grained Partial Label**

Fig. 2: In the case of uniform partial labeling, the true category Y possesses **distinct features** that make other equally likely categories distinguishable from it. In non-uniform partial labeling, the true category Y has **relatively distinctive features** compared to other subtle differences categories Y', which shares relatively similar features with the true category. In Fine-Grained Partial Labeling, the true category Y has **subtle nuances of features** like voice, face and fine grained animal species dataset, while the other categories Y' exhibit subtle differences in their features X compared to the true category. Fine-Grained Partial Label Generation is introduced in Figure 1 (c). Let $\vec{Y}$ denotes the observed candidate set of annotations provided by the annotator for an fine-grained feature X. Y' is denoted as a subtle difference category. The Y stands for true category to which the fine-grained feature instance X belongs.

of bird, while the subtly different category Y' might include other species that have subtle differences from the true species category of fine-grained bird X. Based on Figure 1(c), we can establish a joint probability distribution (as described in Eq. (3)).

$$P\left(Y,Y',\vec{Y},X\right) = P_\theta\left(\vec{Y}|Y,Y',X\right)P\left(Y'|Y,X\right)P\left(X\right)P\left(Y\right) \quad (1)$$

We have weakened the assumption from $P(Y' \mid Y, X)$ to $P(Y' \mid Y)$ to make variable Y' independent of X. This means that the generation of Y' depends solely on Y. Consequently, the equation becomes

$$P\left(Y,Y',\vec{Y},X\right) = P_\theta\left(\vec{Y} \mid Y',Y,X\right)P\left(Y' \mid Y\right)P(X)P(Y) \quad (2)$$

Here, $\mathrm{P}(\vec{Y} = \vec{y} \mid Y = y, Y' = y', X = x)$ represents the conditional probability that the model aims to learn and optimise. $P(X)$ and $P(Y)$ are the marginal probabilities of X and Y, respectively, which represent a standard normal distribution of the image and a uniform distribution of the label. Since X is given and is explicitly captured in the learning process, and $P(Y)$ is uniform, meaning it does not vary across instances, it is not necessary to include it in the learning process. This provides the causal graph model perspective of the fine-grained PLL. In this context, $P(Y'|Y)$ is named the fine-grained transition matrix. It is denoted as $Z(x)_{y,y'}$, which can be defined as the transition probability from a specific true class y of the fine-grained object x to a specific subclass y'. If classes y and y' are in the same superclass, this could be reflected in $\mathbf{Z}(x)y, y'$ having a higher value, since classes belonging to the same superclass tend to have a higher likelihood of sharing subtle nuances. Conversely, $\mathbf{Z}(x)_{y,y'} = 0$ indicates that classes y and y' do not belong to the same superclass, showing a weaker or negative correlation. Each superclass consists of many specific classes. It is reasonable to assume that considering the superclass information of the dataset, the fine-grained transition matrix is usually accessible. The entries of the fine-grained transition matrix, $Z(x)_{y,y'}$, are defined as $Z(x)_{y,y'} = P(Y' = y'|Y = y)$, with $y, y' \in \{1, \ldots, c\}$.

IV. Fine-Grained Partial Label Distribution

The conditional probability of fine-grained partial labels is derived as follow:

$$\begin{aligned}
&\sum_{y \in Y} \mathrm{P}(\vec{Y} = \vec{y}, Y = y \mid X = x) \\
&= \sum_{y \in Y}\sum_{y' \in Y'} \mathrm{P}(\vec{Y} = \vec{y}, Y = y, Y' = y' \mid X = x) \\
&= \sum_{y \in Y}\sum_{y' \in Y'} \underbrace{\mathrm{P}(\vec{Y} = \vec{y} \mid Y = y, Y' = y', X = x)P(Y' = y' \mid Y = y)}_{\textbf{Noised Induced Fine-Grained Partial Label Transition Matrix}}\mathrm{P}(Y = y \mid X = x),
\end{aligned} \quad (3)$$

where,

$$\begin{aligned}
&\mathrm{P}(\vec{Y} = \vec{y} \mid Y = y, Y' = y', X = x)P(Y' = y' \mid Y = y) \\
&= \begin{cases} \frac{1}{2^{c-1}-1}P(y' \mid y) & \text{if } y, y \in \vec{Y} \\ 0 & \text{if } y, y' \notin \vec{Y} \end{cases}
\end{aligned} \quad (4)$$

Here, $\mathrm{P}(\vec{Y} = \vec{y} \mid Y = y, Y' = y', X = x)$ is the conditional probability of $\vec{Y}$ given $Y, Y^{'}$, and X. This is also known as the Fine-Grained Partial Label Transition Matrix. For simplicity, we have assumed the $\vec{Y}$ is independent of $Y^{'}$, and X which is $\frac{1}{2^{c-1}-1}$ according to [16]. Unlike the previous work [16] which has studied under the uniform partial label transition matrix, we have studied partially non-uniform type by introducing the transition matrix Z. The full detailed explanation of conventional partial label distribution and fine-grained partial label distribution is illustrated in Table 1.

A. Assertion Conditions in Fine-Grained Partial Label Generation Set

The learning conditions for fine-grained partial label are described below. According to [1], a certain degree of ambiguity is required for the learnable PLL. The **Fine Grained Partial Label ERM Learnability** condition, referred to as **Lemma 1**, is proposed as follows:

$$P_{y',\bar{y}} := \mathrm{P}(y', \bar{y} \in \vec{Y} \mid y', \bar{y} \neq y, x). \quad (5)$$

Here, y' represents the fine-grained label, while $\bar{y}$ denotes a random false positive label. Certain ambiguity conditions must be satisfied to ensure the learnability of the fine-grained (PLL) problem, where $y' \neq y$ and $\bar{y} \neq y$. These conditions, as proposed by [27], guarantee the Empirical Risk Minimization (ERM) learnability of the fine-grained PLL problem, given a certain degree of ambiguity. In our case, this condition is $P_{y',\bar{y}} < 1$. The term y represents the true label corresponding to each instance x. Here, P_y is defined as $\mathrm{P}(y \in \vec{Y} | Y = y)$, where $P_y = 1$ ensures that the ground-truth label is included in the partially non-uniform fine-grained partial label set for each instance. The term $P_{y'}$ is defined as nearly one in a relatively small-class dataset, while $P_{y'}$ is greater than $P_{\bar{y}}$ in a relatively large-class dataset.

V. Specific Classes-Hierarchical Regularisation

According to Equation (7), the fine-grained transition matrix Z, which represents correlations across specific classes, plays a crucial role in the process of fine-grained label generation. This generation process can be decomposed into two parts: Z and the noise fine-grained partial label transition matrix A. Therefore, by integrating the fine-grained transition matrix $\boldsymbol{Z}$ into our model, we can significantly reduce the ambiguity associated with the term A^*, the noise induced fine-grained partial label transition matrix. This matrix is defined as the probability of observing the candidate label set $\vec{Y}$ given a true label Y, an instance X, and a fine-grained label Y'. It is expressed as $A^*_{ij} = P(\vec{Y} = \vec{y} \mid Y = y, Y' = y', X = x)P(Y = y \mid Y' = y')$, where $\vec{y}$ belongs to the set $\mathcal{Y}$ and $\vec{y}$ is in the range of $[2^c - 1]$. To achieve this, we propose the specific classes hierarchical regularisation approach. This method incorporates the fine-grained transition matrix Z to explicitly encode the correlations across specific classes into the fine-grained PLL framework.

$$
\begin{aligned}
P(\vec{Y} \mid X = x) &= \boldsymbol{A}^* \quad \underbrace{P(Y \mid X = x),}_{\textbf{True posterior probability}} \\
\boldsymbol{A}^{*-1}\, P(\vec{Y} \mid X = x) &= \underbrace{P(Y \mid X = x),}_{\textbf{True posterior probability}}
\end{aligned} \tag{6}
$$

$$
Z^{T^{-1}} A^{-1} P(\vec{Y} \mid X = x) = \underbrace{P(Y \mid X = x).}_{\textbf{True posterior probability}} \tag{7}
$$

Nonetheless, we cannot yet claim to have obtained a classifier-consistent risk estimator since the term A remains unsolved. Furthermore, the complexity of A increases exponentially with the expansion of the label space. As the number of possible subsets, $2^c - 1$, grows, accurately estimating $P(\vec{Y} \mid Y, Y', X)$ becomes infeasible. Consequently, instead of attempting to estimate $P(\vec{Y} \mid Y, Y', X)$ precisely, we have relaxed the assumption of A by positing the independence of $\vec{Y}$ from X and Y'. Although the transition matrix $A = P(\vec{Y} \mid Y, Y', X)$ differs from $P(\vec{Y} \mid y)$, it is important to note that as long as the fine-grained transition matrix $P(Y' \mid Y)$ is captured, which is the primary factor impacting the outcome of label generation, the differences arising from transitioning from a uniform to a partially non-uniform label distribution can be neglected without significantly affecting the optimisation process. Moreover, as [16] states, fully recovering the term $P(\vec{Y} \mid y)$, which depends solely on y, is not necessary for optimising the loss function. For precise expression, we assume that $\vec{Y}$ is independent of y', y, and x, thus modifying the term A to the actual value $\frac{1}{(2^k-1)-1}$. Consequently, we have derived the specific classes-hierarchical regularisation as follows:

$$
\begin{aligned}
&\vec{\mathcal{L}}(f(x), \vec{y}) \\
&= -\frac{1}{N} \sum_{m=1}^{N} \left(\sum_{j=1}^{2^c-1} \mathbb{I}\left(\vec{Y}_m = \vec{y}_j\right) \log\left(\boldsymbol{A}^*[:, j]^\top g(\boldsymbol{x}_m)\right) \right) \\
&= -\frac{1}{N} \sum_{m=1}^{N} \sum_{i=1}^{c} 1(\vec{Y}_m = a_m) \log \left(\frac{1}{2^{c-1} - 1} \frac{\sum_{j=1}^{c} Z_{ji} \exp\left(f_j(x_m)\right)}{\sum_{k=1}^{c} \exp\left(f_k(x_m)\right)} \right)
\end{aligned} \tag{8}
$$

where $Z_{i,j} \in [0,1]^{c \times c}$, $Z_{i,i} = 1$, for $\forall_{i=j} \in [c]$, $I_{i,j} = 0$, for $\forall_{i \neq j} \in [c]$. The $\frac{1}{2^{c-1}-1}$ is derived according to [16] and denotes as A. The $\mathbb{I}$ is the indicator function.

$$
\mathcal{L}_{(f(X), \vec{Y})_{\textbf{SCH}}} = -\sum_{m=1}^{N} \sum_{i=1}^{c} 1(\vec{Y}_m = \bar{a}_m) \log \left(A \frac{\sum_{j=1}^{c} Z_{ji} \exp\left(f_j(x_m)\right)}{\sum_{k=1}^{c} \exp\left(f_k(x_m)\right)} \right), \tag{9}
$$

where N denotes the total number of examples. The classifier $f(x_m)$ maps m-th example x_m to the logit space. $f_j(x_m)$ is the logit for j-th class of the m -th input. Z_{ij} denotes the elements of the transition matrix, Z. The $\bar{a}_m$ is the predicted class label for the m-th example. Given the Z, the uncertainty of the intractable A^* is greatly reduced, as shown in equation (7). We have replaced the hard target label candidate set with pseudo labels $\bar{a}$, which is updated according to (12).

VI. Global Label-Hierarchical-wise Embedding Regularisation

Given an more precised pseudo label of the positive sample set provided by function (9) the global label-hierarchical-wise embedding regularisation is applied to facilitate the model to learn more distinct representation of the fine grained object by pushing apart the dissimilar fine grained sample and grouping the similar fine grained sample. The norm embedding of u and v as the current anchor and key normalised embedding, respectively, derived from the feature extraction network f_Θ and the key neural network f'_Θ. The global label-hierarchical-wise embedding regularisation is defined as follows:

$$
\mathcal{L}_{(f(x), \tau, c)_{\textbf{GLHE}}} = -\frac{1}{N_+(x)} \sum_{v_+ \in N_+(x)} \log \frac{\exp(u^\top v_+ / \tau)}{\sum_{v' \in \bar{c}(x)} \exp(u^\top v' / \tau)}, \tag{10}
$$

1) Positive Sample Selection: The D_q and D_v are vectorial embeddings corresponding to the anchor and key views of the current mini-batch. Given an instance $\boldsymbol{x}$, the global label-hierarchical-wise embedding regularisation of each sample is denoted by contrasting its anchor embedding with the remaining samples of the total sample pool $\bar{c}$. The $\bar{S}(\boldsymbol{x})$ is the sample set excluding the anchor set q and is defined as $\bar{S}(\boldsymbol{x}) = \bar{c} \backslash \boldsymbol{q}$, where $\bar{c} = D_q \cup D_v \cup \text{queue}$. The positive sample set is defined as $N_+(\boldsymbol{x}) = \boldsymbol{v}' \mid \boldsymbol{v}' \in \bar{\mathcal{S}}\left(\boldsymbol{x}\right), \bar{y}' = (\hat{y} = c)$, which includes samples from the current mini-batch with the predicted label $\hat{y} = \text{argmax}_{y \in Y} f_y(\text{aug}_q(x))$ equal to the prediction $\bar{y}'$ of instances from $\bar{\mathcal{S}}(x)$. Finally, we have the final loss function expressed as:

$$
\mathcal{L}_{\textbf{SCH}} + \mathcal{L}_{\textbf{GLHE}} = \lambda \mathcal{L}(f(x), \tau, c) + \vec{\mathcal{L}}(f(X), \vec{Y}). \tag{11}
$$

Overall, the specific classes-hierarchical regularisation (9) is designed to incorporate fine grained correlation information through the use of Z. This process leads to the identification of more precise positive samples. These positive samples are subsequently exploited using the global label-hierarchical-wise embedding regularisation (12) to help model learn more distinctive representation of the fine-grained instances. Ultimately, this leads to improved classification accuracy. λ is the hyperparameter.

VII. Contrastive Prototype Regularisation

We present a contrastive prototype regularisation approach. This method updates the prototype to allow prototypes of different classes with subtle differences to have as large a distance or margin as possible. Initially, we need to determine the margin between the prototype vector $\boldsymbol{v}_i$ and the prototype vector $\boldsymbol{v}_j$, both in $\mathbb{S}^{d-1}$, which is defined as $m_{ij} = \exp(-\boldsymbol{v}_i^\top \boldsymbol{v}_j)$. The m_{ij} quantifies the margin between the prototype vectors of v_i and v_j on the unit sphere. A smaller value of m_{ij} indicates more similarity between v_i and v_j. For the prototype $\boldsymbol{v}_i$, we define the normalised margin between $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ as $\bar{m}_{ij} = \frac{\exp(-\boldsymbol{v}_i^\top \boldsymbol{v}_j)}{\sum_{j\neq i}\exp(-\boldsymbol{v}_i^\top \boldsymbol{v}_j)}$. For each $\boldsymbol{v}_i$, $i \in \{1, \cdots, K\}$, momentum updating is implemented, where the new prototype vector $\boldsymbol{v}_i^{t+1}$ is a combination of the normalised margin between $\boldsymbol{v}_j$ and $\boldsymbol{v}_i$ for all $j \neq i$ as a regularisation. The resulting new update rule is given as $\boldsymbol{v}_i^{t+1} = \sqrt{1-\alpha^2}\boldsymbol{v}_i^t + \alpha\frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2}$, where the gradient $\boldsymbol{g}$ is defined as $\boldsymbol{g} = \boldsymbol{u} - \beta\sum_{j\neq i}\bar{m}_{ij}^t \boldsymbol{v}_j^t$, and $\boldsymbol{u}$ is the anchor embedding whose prediction is class i. Here, $\bar{m}_{ij}^t$ is the normalised margin between prototype vectors at step t (i.e., $\boldsymbol{v}_j^t, j \neq i$). The $\boldsymbol{g}$ uses $\boldsymbol{u}$, the anchor embedding of x, to update the prototype vector v_i by considering the similarity of prototype vectors. If the similarity is high, resulting in a smaller value of $\bar{m}_{ij}$, more weight is given to the original $\boldsymbol{u}$ in the prototype updating, leading to fewer changes in $\boldsymbol{g}$. Consequently, the impact on $\boldsymbol{v}$ from less similar classes will be less, and vice versa. This ensures a larger separation of embeddings with subtle differences for different classes, endowing the model to learn more distinctiveness for dissimilar classes through greater adjustments to $\boldsymbol{v}_i$, and less modification for similar classes. The updating mechanism of pseudo labels $\bar{\boldsymbol{a}}$ is defined as follows:

$$\bar{\boldsymbol{a}} = \phi\bar{\boldsymbol{a}} + (1-\phi)\boldsymbol{r}_c,\ r_c \ = \begin{cases} 1 & \text{if } c = \arg\max_{y\in Y} \boldsymbol{u}^\top \boldsymbol{v}_{\boldsymbol{y}}, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Here, $\bar{\boldsymbol{a}}$ is the average weighted combination of the uniform probability $\frac{1}{|c|}1$, the initial format of $\bar{\boldsymbol{a}}$, and r_c. The r_c is constructed based on the dot product of the anchor embedding $\boldsymbol{u}$ and $\boldsymbol{v}_y$, ensuring that classes with more similar prototypes are selected. The $\boldsymbol{v}_y$ is the normalised prototype vector associated with the y-th class. The hyper-parameter ϕ controls the updating of $\bar{\boldsymbol{a}}$.

VIII. Classifier-Consistent Risk Estimator

1) Learning with True labels: Lets denote $f(X) = (g_1(x), \ldots, g_c(x))$ as the classifier, in which $g_c(x)$ is the classifier for label $c \in [c]$. The prediction of the classifier $f_c(x)$ is $P(Y = c \mid x)$. We want to obtain a classifier $f(X)$ =$\arg\max_{c\in[c]} g_c(x)$. The loss function is to measure the loss given classifier $f(X)$. To this end, the true risk can be denoted as:

$$R(f) = \mathbb{E}_{(X,Y)}[\mathcal{L}\left(f\left(X\right), Y\right)]. \tag{13}$$

The ultimate objective is to learn the optimal classifier $f^* = \arg\min_{f\in\mathcal{F}} R(f)$ for all loss functions, aiming for the convergence of empirical risk $\bar{R}_{fg}(f)$ to the true risk $R(h)$. To achieve the optimal classifier, we need to prove that the modified loss function is risk consistent as if it can converge to the true loss function.

2) Learning with Fine-Grained Partial Label: An input $X \in \mathcal{X}$ has a candidate set of $\vec{Y} \in \vec{\mathcal{Y}}$ but a only true label $Y \in \vec{\mathcal{Y}}$. Given the fine-grained partial label $\vec{Y} \in \vec{\mathcal{Y}}$ and instance $X \in \mathcal{X}$ that the objective of the loss function is denoted as: $\hat{R}(f) = \mathbb{E}_{(X,\vec{Y})}\vec{\mathcal{L}}\left(f\left(X\right), \vec{Y}\right)$. The optimal classifier $\hat{f}^* = \arg\min_{f\in\mathcal{F}} \hat{R}(f)$. Since the true fine-grained partial label distribution $\bar{\mathcal{D}}$ is unknown, our goal is approximate the optimal classifier with sample distribution $\bar{D}_{fg}$ by minimising the empirical risk function, namely

$$\hat{R}_{fg}(f) = \frac{1}{n}\sum_{i=1}^{n}\vec{\mathcal{L}}\left(f\left(\boldsymbol{x}_i\right), \vec{y}_i\right). \tag{14}$$

Assumption 1. According to [28] that the minimisation of the expected risk $R(f)$ given clean true population implies that the optimal classifier is able to do the mapping of $f_i^*(X) = P(Y = i \mid X), \forall i \in [c]$. Under Assumption 1, we can draw conclusion that $\hat{f}^* = f^*$ according to Theorem 2 in the appendix page.

Theorem 1. *Assume that the fine-grained transition matrix $Z_{y,y'}$ is fully ranked and the assumption 1 is met, the the minimizer of $\hat{f}^*$ of $\hat{R}(f)$ will converge to f^* of $R(f)$, meaning $\hat{f}^* = f^*$.*

Remark. If A and $Z_{y,y'}$ are accurately estimated, the empirical risk of the algorithm trained with partially non-uniform fine-grained partial labels will converge to the expected risk of the optimal classifier trained with true labels. As the number of samples approaches infinity, considering the fine-grained partial labels, $\hat{f}_n$ theoretically converges to $\hat{f}$. Consequently, $\hat{f}_n$ will converge to the optimal classifier f^*, as stated in Theorem 1.

Generalisation Error. Define $\hat{R}$ and $\hat{R}_{fg}$ as the true risk and the empirical risk, respectively, given the fine-grained partial label dataset. The empirical loss classifier is obtained as $\hat{f}_{fg} = \arg\min_{f\in\mathcal{F}} \hat{R}_{fg}(f)$. Suppose a set of real hypotheses $\mathcal{F}_{\vec{y}_k}$ exists, with $f_i(X) \in \mathcal{F}$ for all $i \in [c]$. Also, assume its loss function $\vec{\mathcal{L}}(\boldsymbol{f}(X), \vec{Y})$ is L-Lipschitz continuous with respect to $f(X)$ for all $\vec{y}_k \in \vec{\mathcal{Y}}$ and upper-bounded by M, i.e., $M = \sup_{x\in\mathcal{X}, f\in\mathcal{F}, y_k\in\vec{Y}} \vec{\mathcal{L}}\left(f(x), \vec{y}_k\right)$. The expected Rademacher complexity of $\mathcal{F}_{\vec{y}_k}$ is denoted as $\Re_n(\mathcal{F}_{\vec{y}_k})$[29]. As the number of samples approaches infinity ($n \to \infty$), $\Re_n(\mathcal{F}_{\vec{y}_k})$ tends to zero with a bounded norm. Subsequently, $\bar{R}(\hat{f})$ converges to $\bar{R}(\hat{f}^\star)$ as the number of training data becomes infinitely large.

Theorem 2. *For any $\delta > 0$, with probability at least $1 - \delta$,*

$$\hat{R}\left(\hat{f}_{fg}\right) - \hat{R}\left(\hat{f}^\star\right) \leq 4\sqrt{2}L\sum_{k=1}^{c}\Re_n\left(\mathcal{F}_{\vec{y}_k}\right) + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}}. \tag{15}$$

The proof is given in Appendix Theorem 2.

Dataset	Method	$q^* = 0.03 \pm 0.02$	$q^* = 0.05 \pm 0.02$	$q^* = 0.1 \pm 0.02$
CIFAR100	$\mathcal{L}_{\mathbf{SCH}} + \mathcal{L}_{\mathbf{GLHE}}$	**73.36** ±0.32	**72.76**±0.14	**54.09 ±1.88**
	PiCO*	72.87±0.26	72.53 ±0.37	48.03 ±3.32
	ML-PLL*	63.98 ±1.12	56.10±0.55	25.74 ±0.29
	LWS*	46.8 ±0.06	24.82 ±0.17	4.53 ±0.47
	PRODEN *	59.33 ±0.48	41.20 ±0.27	13.44±0.41
CUB200	$\mathcal{L}_{\mathbf{SCH}} + \mathcal{L}_{\mathbf{GLHE}}$	**72.04** ±0.73	**71.95**±0.38	**56.03**±0.69
	PiCO*	71.85±0.53	71.15±0.41	50.31±1.01
	ML-PLL*	5.13±0.68	2.39±0.38	0.84±0.18
	LWS*	9.6±0.62	4.02±0.03	1.44±0.06
	PRODEN*	18.71±0.45	17.63±0.89	17.99±0.62
Dataset	Method	$q^* = 0.1 \pm 0.02$	$q^* = 0.3 \pm 0.02$	$q^* = 0.5 \pm 0.02$
CIFAR10	$\mathcal{L}_{\mathbf{SCH}} + \mathcal{L}_{\mathbf{GLHE}}$	93.54±0.08	92.79±0.3	**89.81**±0.65
	PiCO*	**93.64**±0.24	**92.85**±0.43	81.45±0.57
	ML-PLL *	92.47 ±0.33	88.97 ±0.17	66.74±0.90
	LWS*	87.34±0.87	39.9±0.72	9.89±0.55
	PRODEN*	88.80±0.14	81.88±0.51	20.32±3.43

TABLE II: Accuracy comparison on three benchmark datasets. Superior results are indicated in bold. Our proposed methods have shown comparable results to fully supervised learning and outperform previous methods in a more challenging learning scenario, such as the partial rate at 0.5 (CIFAR10) and 0.1 (CIFAR100, CUB200). (The symbol $*$ indicates fine-grained partial label dataset).

IX. Experiments

Dataset: We evaluated our proposed method using three benchmark datasets: CIFAR10, CIFAR100 [30], and CUB200 [31]. CIFAR100 comprises 50,000 training images and 10,000 test images, distributed across 100 classes. In contrast, CIFAR10 consists of the same total number of images but is divided into just 10 classes. The CUB200 dataset includes 11,788 images of birds, categorized into 200 distinct classes. The data for CUB200 is split into 5,994 training images and 5,794 testing images.

Main Empirical Results for CIFAR10: We trained the model using a fine-grained partial label dataset at rates of $q = 0.1 \pm 0.02, 0.3 \pm 0.02, 0.5 \pm 0.02$. Classification accuracy for all experiments is presented in Table 1. We benchmarked our results against prior works on CIFAR-10, including PiCo [12], LWS [32], PRODEN [15], and ML-PLL [33], all of which employ class-hierarchical regularization. Our method consistently outperformed these previous approaches in scenarios involving fine-grained Partial Label Learning (PLL) with $q = 0.3 \pm 0.02, 0.5 \pm 0.02$. Notably, our proposed method demonstrated a significant improvement, achieving an 8.36% increase in classification accuracy at a 0.5 fine-grained partial rate compared to the best-performing previous work [12]. Additionally, our results were competitive at partial rates of 0.1 and 0.3.

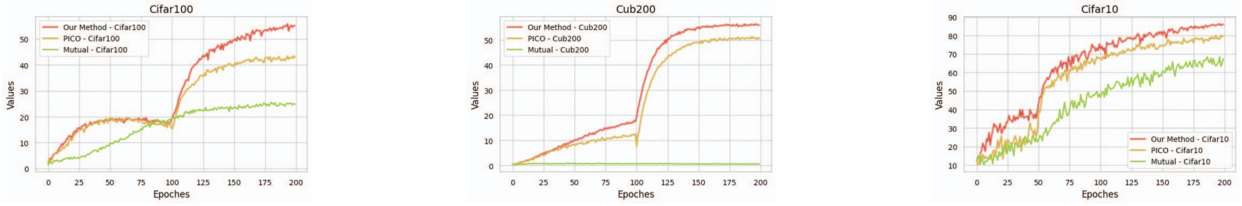
Main Empirical Results for CUB200 and CIFAR100: Our proposed method excelled in challenging tasks involving fine-grained partial labels, particularly notable at a 0.1 partial rate across both the CUB200 and CIFAR100 datasets. Specifically, for the CUB200 dataset, our method achieved significant improvements, showing a notable increase of **5.72%** at the 0.1 fine-grained partial rate, along with gains of 1.281% and 0.37% at the 0.05 and 0.03 rates, respectively. Similarly, on the CIFAR100 dataset, our approach demonstrated a substantial classification advantage, with margins of **6.06%** at the 0.1 rate, and 0.4181% and 0.5414% at the 0.05 and 0.03 rates, respectively. To ensure the reliability of these results, each experiment was conducted five times using different random seeds.

A. Ablation Study

In this section, we present a comparison of the classification performance for the Fine-Grained Partial Label task on two datasets, CUB200 and CIFAR100, of the state-of-the-art method *PLCR* [34] and our method, which integrates the identity transition matrix loss with PLCR. As shown in Table 5 and Table 6, our method has achieved superior results for partially non-uniform fine-grained learning tasks. Even under less challenging conditions where $q^* = 0.03$, our loss function demonstrates improved performance. Most importantly, we've observed a significant deterioration in the performance of the PLCR method under more challenging conditions of $q^* = \{0.05, 0.1\}$. This suggests that without properly accounting for the exogeneity of the dataset, the designed algorithm may not handle more realistic tasks efficiently. The table above presents a comparison of the PLCR method on its own and the combination of PLCR with our $\mathcal{L}_{\mathbf{SCH}}$. Overall, the table clearly shows that, regardless of the difficulty of learning tasks (q^*), the accuracy of PLCR significantly improves when integrated with our $\mathcal{L}_{\mathbf{SCH}}$. For $q^* = 0.03$, the $\mathcal{L}_{\mathbf{SCH}}$ has improved the accuracy of the PLCR method by 26.87%. Similarly, for $q^* = 0.05$, we have observed an accuracy improvement of 35.74%. Lastly, for $q^* = 0.1$, the accuracy improvement is 23.85%. These improvements prove the effectiveness of the $\mathcal{L}_{\mathbf{SCH}}$, especially in fine-grained PLL tasks.

Data	Method	$q^* = 0.03$	$q^* = 0.05$	$q^* = 0.1$
CUB200	PLCR*	34.08±0.18%	15.37±0.4%	4.05±0.23%
CUB200	PLCR*+ $\mathcal{L}_{\mathbf{SCH}}$	**60.95**±0.39%	**51.11**±0.64%	**27.90**±0.98%

TABLE III: Accuracy Comparison on Benchmark dataset (CUB200)

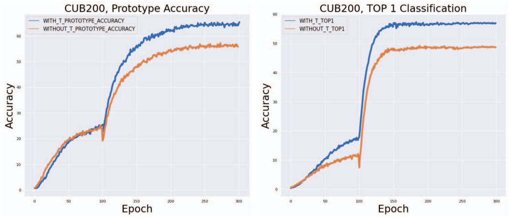


(a) (b) (c)

Fig. 3: Classification Comparison on datasets CIFAR10, CIFAR100, CUB200

Data	Method	q^*=0.1
CIFAR100	PiCO[12]	20.941(-24.015)%
Data	Method	q^*=0.1
CIFAR100	$\mathcal{L}_{\textbf{SCH}} + \mathcal{L}_{\textbf{GLHE}}$	**54.156(0.066)**%
Data	Method	q^*=0.1
CUB200	PiCO[12]	21.22(-25.155)%
Data	Method	q^*=0.1
CUB200	$\mathcal{L}_{\textbf{SCH}} + \mathcal{L}_{\textbf{GLHE}}$	**48.62(-7.64)**%

Fig. 4: Accuracy comparison for benchmark dataset (CUB200).



(a) (b)

Fig. 5: Prototype and Classification Comparison

Data	Method	q^*=0.1	$q^* = 0.2$	$q^* = 0.3$	$q^* = 0.4$
CIFAR100	PLCR*	73.67%±0.13	68.99% ±0.18	52.45%±0.99	36.50%±1.68
CIFAR100	PLCR*+ $\mathcal{L}_{\textbf{SCH}}$	**73.55**%±0.30	**70.66**%±0.08	**61.57**%±0.59	**47.87**%±0.49

TABLE IV: Accuracy Comparison on Benchmark datasets (CIFAR100)

Figure 5 examines the impact of updating entries in the original fine-grained transition matrix, denoted as $Z_{\textbf{Original}}$(with entries 0.2), to form a new matrix, $Z_{\textbf{New}}$(with entries 0.3), on the classification performance: Our results demonstrate that a fine-grained transition matrix is crucial for training a more robust model. Compared to the approach described in [12], our method exhibits greater robustness. Figure 6 illustrates how the proposed Equation (9) significantly reduces the uncertainty associated with the A^* transition matrix. This reduction in uncertainty subsequently improves both classification accuracy and prototype performance.

X. Conclusion

In this paper we have addressed the fine-grained PLL problem by capturing the dependency of the fine-grained label on the true label. However, accounting for this dependency increases the complexity of the transition matrix, potentially leading to an inconsistent classifier. To address this issue, we propose the specific classes-hierarchical regularisation. This approach not only offers a provably consistent classifier but also achieves superior performance. Thereafter, the global label hierarchical wise embedding regularisation is proposed, exploiting the positive samples from the consistent classifier to learn more distinct representation from fine grained instances, leading to better classification performance.

References

[1] T. Cour, B. Sapp, and B. Taskar, “Learning from partial labels,” *The Journal of Machine Learning Research*, vol. 12, pp. 1501–1536, 2011.

[2] E. Hüllermeier and J. Beringer, “Learning from ambiguously labeled examples,” *Intelligent Data Analysis*, vol. 10, no. 5, pp. 419–439, 2006.

[3] N. Xu, C. Qiao, X. Geng, and M.-L. Zhang, “Instance-dependent partial label learning,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[4] M.-L. Zhang, B.-B. Zhou, and X.-Y. Liu, “Partial label learning via feature-aware disambiguation,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1335–1344.

[5] T. Wang and P. Isola, “Understanding contrastive representation learning through alignment and uniformity on the hypersphere,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 9929–9939.

[6] N. Xu, J. Lv, and X. Geng, “Partial label learning via label enhancement,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5557–5564.

[7] G. Lyu, S. Feng, T. Wang, C. Lang, and Y. Li, “Gm-pll: graph matching based partial label learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 2, pp. 521–535, 2019.

[8] L. Liu and T. Dietterich, “A conditional multinomial mixture model for superset label learning,” *Advances in neural information processing systems*, vol. 25, 2012.

[9] R. Jin and Z. Ghahramani, “Learning with multiple labels,” *Advances in neural information processing systems*, vol. 15, 2002.

[10] N. Nguyen and R. Caruana, “Classification with partial labels,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 551–559.

[11] H. Wang, Y. Qiang, C. Chen, W. Liu, T. Hu, Z. Li, and G. Chen, "Online partial label learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2020, pp. 455–470.

[12] H. Wang, R. Xiao, Y. Li, L. Feng, G. Niu, G. Chen, and J. Zhao, "Pico: Contrastive label disambiguation for partial label learning," *ICLR*, 2022.

[13] F. Hong, J. Yao, Z. Zhou, Y. Zhang, and Y. Wang, "Long-tailed partial label learning via dynamic rebalancing," in *The Eleventh International Conference on Learning Representations*. OpenReview.net, 2023.

[14] L. Feng and B. An, "Partial label learning with self-guided retraining," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3542–3549.

[15] H. Wen, J. Cui, H. Hang, J. Liu, Y. Wang, and Z. Lin, "Leveraged weighted loss for partial label learning," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 11 091–11 100.

[16] L. Feng, J. Lv, B. Han, M. Xu, G. Niu, X. Geng, B. An, and M. Sugiyama, "Provably consistent partial-label learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10 948–10 960, 2020.

[17] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[18] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[19] M. Schrimpf, J. Kubilius, H. Hong, N. J. Majaj, R. Rajalingham, E. B. Issa, K. Kar, P. Bashivan, J. Prescott-Roy, F. Geiger, K. Schmidt, D. L. K. Yamins, and J. J. DiCarlo, "Brain-score: Which artificial neural network for object recognition is most brain-like?" *bioRxiv preprint*, 2018. [Online]. Available: https://www.biorxiv.org/content/10.1101/407007v2

[20] M. Schrimpf, J. Kubilius, M. J. Lee, N. A. R. Murty, R. Ajemian, and J. J. DiCarlo, "Integrative benchmarking to advance neurally mechanistic models of human intelligence," *Neuron*, 2020. [Online]. Available: https://www.cell.com/neuron/fulltext/S0896-6273(20)30605-X

[21] C. Zhuang, S. Yan, A. Nayebi, M. Schrimpf, M. C. Frank, J. J. DiCarlo, and D. L. Yamins, "Unsupervised neural network models of the ventral visual stream," *Proceedings of the National Academy of Sciences*, vol. 118, no. 3, p. e2014196118, 2021.

[22] W. J. Scheirer, S. E. Anthony, K. Nakayama, and D. D. Cox, "Perceptual annotation: Measuring human vision to improve computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1679–1686, 2014.

[23] G. Jacob, R. Pramod, H. Katti, and S. Arun, "Qualitative similarities and differences in visual object representations between brains and deep networks," *Nature communications*, vol. 12, no. 1, p. 1872, 2021.

[24] A. Dorais and D. Sagi, "Contrast masking effects change with practice," *Vision Research*, vol. 37, no. 13, pp. 1725–1733, 1997.

[25] C. Conwell, J. S. Prince, G. A. Alvarez, and T. Konkle, "What can 5.17 billion regression fits tell us about artificial models of the human visual system?" in *SVRHM 2021 Workshop@ NeurIPS*, 2021.

[26] B. Han, J. Yao, N. Gang, M. Zhou, I. Tsang, Y. Zhang, and M. Sugiyama, "Masking: A new perspective of noisy supervision," in *NeurIPS*, 2018, pp. 5839–5849.

[27] L. Liu and T. Dietterich, "Learnability of the superset label learning problem," in *International Conference on Machine Learning*. PMLR, 2014, pp. 1629–1637.

[28] X. Yu, T. Liu, M. Gong, and D. Tao, "Learning with biased complementary labels," in *ECCV*, 2018, pp. 68–83.

[29] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 463–482, 2002.

[30] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[31] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[32] J. Lv, M. Xu, L. Feng, G. Niu, X. Geng, and M. Sugiyama, "Progressive identification of true labels for partial-label learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 6500–6510.

[33] Y. Yan and Y. Guo, "Mutual partial label learning with competitive label noise," in *The Eleventh International Conference on Learning Representations*.

[34] D.-D. Wu, D.-B. Wang, and M.-L. Zhang, "Revisiting consistency regularization for deep partial label learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 24 212–24 225.

[35] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

XI. Appendix

A. Implementation Details

The contrastive network was equipped with a projection head featuring a 128-dimensional, 2-layer MLP embedding and used augmentation techniques as described in [12]. The network operated with a momentum of 0.999. Queue sizes were configured to 8192 for CIFAR-10 and CIFAR-100, and 4192 for CUB200.

For model training, the following parameters were used: the model trained for 299 epochs using an SGD optimizer with a momentum of 0.9, a batch size of 256, and a cosine learning rate schedule. The temperature τ was set at 0.07, and the loss weighting λ was 0.5. The hyperparameters α and β, controlling prototype updates, were set at 0.1 and 0.01 respectively.

Regarding the partial label rates, for CIFAR-10, the rates were $q \in \{0.1, 0.3, 0.5\}$, and for CIFAR-100/CUB200, the rates were $q \in \{0.03, 0.05, 0.1\}$. The fine-grained rates were defined as $q^* \in \{0.1 \pm 0.02, 0.3 \pm 0.02, 0.5 \pm 0.02\}$ for CIFAR-10 and $q^* \in \{0.03 \pm 0.02, 0.05 \pm 0.02, 0.1 \pm 0.02\}$ for CIFAR-100/CUB200.

Training durations varied by dataset and label specificity: CIFAR-10 required 1 epoch for all partial rates and 50 epochs for fine-grained labels. For CIFAR-100 and CUB200, the epochs were set to {20, 20, 100} for clean partial labels and {20, 100, 100} for fine-grained partial rates. Notably, CIFAR-10 training did not involve global label-hierarchical-wise embedding regularisation.

B. Implementation Details for Ablation Study

1) Implementation Details for CUB200: In the CUB200 dataset, for tasks with $q^* = \{0.03, 0.05\}$, we have assigned a total of 80 epochs, considering these to be less challenging. Conversely, for the more demanding task where $q^* = 0.1$, we have increased the total epochs to 100. Across all fine-grained Partial Label Learning (PLL) tasks on CUB200, the learning rate is uniformly set at 0.01. For all other methods, we have implemented the number of epochs that yields the best results.

2) Implementation Details for CIFAR100 and CIFAR10: For the CIFAR100 and CIFAR10 datasets, within our $\mathcal{L}_{\mathbf{SCH}}$ framework, the tasks with $q^* = 0.03$ are considered relatively easy and thus set to 150 epochs. Tasks with $q^* = \{0.05, 0.1\}$ also have a duration of 150 epochs, reflecting a standardized approach to managing these difficulty levels. For all other methods, we have implemented the epoch count that produces the most effective outcome. The learning rate for all fine-grained PLL tasks on CIFAR100 is set at 0.1.

C. The Proof for Theorem 1

The objective is to design a new loss function, enabling the hypothesis with fine-grained partial labels to converge towards the optimal classifier trained with true labels. We define $\vec{\mathcal{L}}$ as the newly proposed loss function for fine-grained PLL. The true and empirical loss functions regarding fine-grained partial labels are stated as $\hat{R}(f) = \mathbb{E}_{(X,\vec{Y}) \sim P_{(X,\vec{Y})}}[\vec{\mathcal{L}}(f(X), \vec{Y})]$ and $\hat{R}_{fg}(f) = \frac{1}{n}\sum_{i=1}^{n} \vec{\mathcal{L}}(f(x_i), \vec{y}_i)$, respectively. Furthermore, we define $\{(\mathbf{x}_i, \vec{y}_i)\}_{1 \le i \le n}$ as the fine-grained partial label sample space. The functions $\hat{f}^*$ and $\hat{f}_{fg}$ represent the optimal classifiers with minimum expected risk function $\hat{R}(f)$ and empirical risk function $\hat{R}_{fg}(f)$, respectively. Specifically, the model is formalised as $\hat{f}^* = \arg\min_{f \in \mathcal{F}} \hat{R}(f)$ and $\hat{f}_{fg} = \arg\min_{f \in \mathcal{F}} \hat{R}_{fg}(f)$. The goal of the proposed loss function $\vec{\mathcal{L}}$ is to ensure that the classifier, trained with a sample of fine-grained partial labels, converges to the optimal classifier trained with a population dataset of true labels. Formally, this convergence is represented as $\hat{f}_{fg} \xrightarrow{n} f^\star$. The true expected risk function for a classifier trained with true labels from the population is defined as $R(f) = \mathbb{E}_{(X,Y) \sim P(X,Y)}[\vec{\mathcal{L}}(f(X), Y)]$. The optimal classifier for the true expected risk function is defined as $f^\star = \arg\min_{f \in \mathcal{F}} R(f)$.

D. The Proof for Theorem 2

Definition 1: Let's denote $\vec{y}_k$ as the k_{th} element of the vector $\vec{y}$, being 1 while the others are 0, if $\vec{y}_k \in \vec{y}$. Here, $\vec{y}$ represents a candidate set of the fine-grained partial label of an instance. Based on Lemma 1 and Theorem 1, the estimation error bound has been proven through the following equation:

$$\begin{aligned}
&\hat{R}\left(\hat{f}_{fg}\right) - \min_{f \in F} \hat{R}(f) = \hat{R}\left(\hat{f}_{fg}\right) - \hat{R}\left(\hat{f}^\star\right) \\
&= \hat{R}\left(\hat{f}_{fg}\right) - \hat{R}_{fg}(\hat{f}) + \hat{R}_{fg}(\hat{f}) - \hat{R}_{fg}\left(\hat{f}^\star\right) + \hat{R}_{fg}\left(\hat{f}^\star\right) - \hat{R}\left(\hat{f}^\star\right) \\
&\le \hat{R}\left(\hat{f}_{fg}\right) - \hat{R}_{fg}(\hat{f}) + \hat{R}_{fg}\left(\hat{f}^\star\right) - \hat{R}\left(\hat{f}^\star\right) \\
&\le 2 \sup_{f \in \mathcal{F}} \left|\hat{R}(f) - \hat{R}_{fg}(f)\right| \\
&\le 4\Re\left(\mathcal{F}_v\right) + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}} \\
&\le 4\sqrt{2}L\sum_{k=1}^{c} \Re_n\left(\mathcal{F}_{\vec{y}_k}\right) + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}}.
\end{aligned} \tag{16}$$

The first inequality, $\hat{R}_{fg}(\hat{f}) - \hat{R}_{fg}(f^\star) \le 0$, establishes the initial step in the proof. The derivation and validation of the first three equations have been thoroughly demonstrated in [35]. The overarching framework and subsequent elements of this proof draw extensively on the methodologies and results discussed in [29].

The definition 2: Suppose a space D and a sample distribution D_S are given, in which $S = \{s_1, \ldots, s_n\}$ is a set of examples drawn independently and identically from the distribution D_S. Additionally, let $\mathcal{F}$ be defined as a class of functions $f : S \to \mathbb{R}$. The empirical Rademacher complexity of $\mathcal{F}$ is defined as:

$$\hat{\Re}_n(\mathcal{F}) = \mathbb{E}_\sigma\left[\sup_{f \in \mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(x_i)\right)\right]. \tag{17}$$

The expected Rademacher complexity of the function space $\mathcal{F}$ is denoted as:

$$\Re = \mathbb{E}_{D_S}\mathbb{E}_\sigma\left[\sup_{f \in \mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(x_i)\right)\right]. \tag{18}$$

The independent random variables $\sigma_1, \ldots, \sigma_n$ are uniformly selected from $\{-1, 1\}$ and are defined as Rademacher variables. Let M be the upper bound of the loss function. Subsequently, for any $\delta > 0$, with probability at least $1 - \delta$, we have:

$$\sup_{f \in \mathcal{F}}\left|\hat{R}(f) - \hat{R}_{fg}(f)\right| \le 2\Re(\vec{\mathcal{L}} \circ \mathcal{F}) + M\sqrt{\frac{\log(1/\delta)}{2n}}, \tag{19}$$

where $\Re(\vec{\mathcal{L}} \circ \mathcal{F})$ is the expected Rademacher complexity of the function space with the modified loss function $\vec{\mathcal{L}}$, defined as:

$$\vec{\mathcal{L}}(f(X), \vec{Y}) = -\sum_{i=1}^{c}(\bar{a}_i)\log\left((\mathbf{Z})^\top f(X)_i\right), \tag{20}$$

and $\mathcal{F}_V$ is:

$$\mathcal{F}_V = \left\{(X, \vec{Y}) \mapsto \sum_{i=1}^{c}(\bar{a}_i)\log\left((\mathbf{Z})^\top f(X)_i\right) \mid f \in \mathcal{F}\right\}. \tag{21}$$

Assuming the loss function $\vec{\mathcal{L}}(f(\mathbf{X}), \vec{Y})$ satisfies the L-Lipschitz property with respect to $f(\mathbf{X})$ for all $\vec{y}_k \in \vec{\mathcal{Y}}$, and applying the Rademacher vector contraction inequality, we obtain:

$$\Re(\mathcal{F}_V) \leq \sqrt{2}L \sum_{k=1}^{c} \Re_n(\mathcal{F}_{\vec{y}_k}). \tag{22}$$

The proof is thus completed.

E. The Transition Matrix of the Table 1

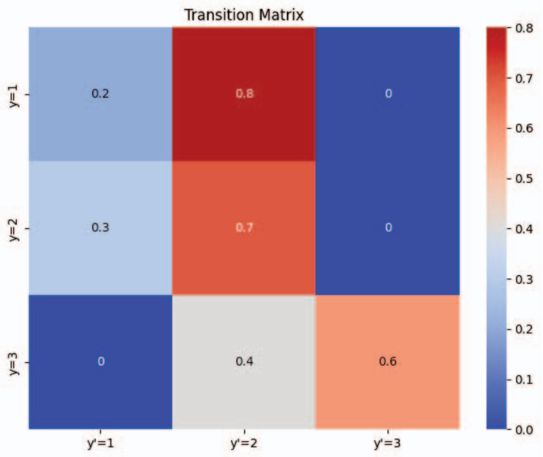


Fig. 6: The Transition Matrix