# Learning with Partial-Label and Unlabeled Data: A Uniform Treatment for Supervision Redundancy and Insufficiency

**Yangfan Liu** [*1,2] **Jiaqi Lv** [*1,2] **Xin Geng** [1,2] **Ning Xu** [1,2]

## Abstract

One major challenge in weakly supervised learning is learning from inexact supervision, ranging from partial labels (PLs) with *redundant* information to the extreme of unlabeled data with *insufficient* information. While recent work has made significant strides in specific inexact supervision contexts, supervision forms typically *coexist* in complex combinations. This is exemplified in *semi-supervised partial label learning*, where PLs act as the exclusive supervision in a semi-supervised setting. Current strategies addressing combined inexact scenarios are usually composite, which can lead to incremental solutions that essentially replicate existing methods. In this paper, we propose a novel approach to *uniformly* tackle both label redundancy and insufficiency, derived from a mutual information-based perspective. We design a label channel that facilitates dynamic label exchange within the candidate label sets, which identifies potential true labels and filters out likely incorrect ones, thereby minimizing error accumulation. Experimental results demonstrate the superiority of our method over existing state-of-the-art PL and semi-supervised learning approaches by directly integrating them. Furthermore, our extended experiments on partial-complementary label learning underscore the flexibility of our uniform treatment in managing diverse supervision scenarios.

[*]Equal contribution [1]School of Computer Science and Engineering, Southeast University, Nanjing, China [2]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China. Correspondence to: Ning Xu <xning@seu.edu.cn>.

*Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. 

## 1. Introduction

Over the last decade and more, the remarkable progress in deep neural networks has been primarily driven by the availability of an enormous amount of manually labeled data. However, in scenarios with limited labeled data, even state-of-the-art supervised learning methods often face substantial challenges in performing well. On the other hand, the collection of massive data with high-quality annotations is laborious and costly. To mitigate this issue, a prevalent strategy is resorting to crowd-sourcing labels (Brabham, 2008) to trade off the cost and quality of annotations, which has highlighted the critical need for developing algorithms capable of learning from weakly supervised data (Zhou, 2017).

Inexact supervision (Zhou, 2017) is an important type of weak supervision, considering scenarios where supervision information is not as exact as desired. Inexact supervised classification is typically exemplified by *partial labels* (PLs) (Nguyen & Caruana, 2008; Cour et al., 2011; Zhang et al., 2017; Lv et al., 2020; Feng et al., 2020) and *complementary labels* (CLs) (Ishida et al., 2017; Yu et al., 2018; Ishida et al., 2019; Gao & Zhang, 2021) — a PL for an instance is a set of candidate labels, where a fixed but unknown candidate is the true label, and a CL specifies one class that an instance does not belong to. Thus, PLs offer redundant information that obscures the true label, and CLs are the extreme case of PLs by offering minimal guidance on the true label. In contrast, *unlabeled data* serve as the opposite extreme, generally necessitating integration with a small amount of supervised data for a slight supervision, i.e., semi-supervised learning (SSL) (Laine & Aila, 2017; Tarvainen & Valpola, 2017; Berthelot et al., 2019; Sohn et al., 2020). For specific forms of supervision, existing learning paradigms have been extensively developed, such as partial label learning (PLL), complementary label learning (CLL), and SSL, with corresponding techniques being elaborately tailored. However, in reality, forms of supervision *rarely exist in isolation*. Rather, they often co-occur in varied combinations, requiring flexible learning frameworks to effectively integrate and leverage multiple disparate sources of data to navigate the complexities of the real world.

*Semi-supervised partial label learning* (SSPLL) (Wang

et al., 2019; Wang & Zhang, 2020) is an emerging problem, which is proposed to exploit PL samples in conjunction with unlabeled data since one can easily access massive wild unlabeled samples. One intuitive way to address the combinations is to "divide and conquer", that is, to *compose approaches to individual problems*. Specifically, the typical SSPLL model incorporates two key components: a specific PLL method and a specific SSL process. A PLL method first disambiguates (Zhang & Yu, 2015) the redundant labels, i.e., decouples instances and their associated spurious labels, and applies SSL method to generate pseudo-labels for unlabeled data. The effectiveness of composite strategies can be anticipated to depend on the performance of the components and the heuristic means of combining them. Follow this pattern, real-world problems are reduced to a mere stack of fragmented tasks, and then corresponding incremental methods could be easily proposed, however, fundamentally, resulting in reinventing the wheel.

In this paper, we aim to address the realistic and challenging combined inexact scenarios. Instead of composite design, we propose a strategy to uniformly treat all data, regardless of whether their initial labeling information is characterized by redundancy or insufficiency, then allowing for the interaction among training samples across different forms of supervision. We design a label channel that enables labels to circulate through the candidate label set, which identifies potential true labels and filters out likely incorrect ones based on the mutual information, thereby minimizing error accumulation. We theoretically guarantee that, it is feasible to safely assign weak supervision signals to the training samples to initiate the training process under mild assumptions. With our strategy, one can easily instantiate a specific learning algorithm for various combinations of PLL, CLL, and SSL scenarios. We summarize our main contributions as follows: (1) Taking the SSPLL problem as a key example, we instantiate our strategy and introduce the SPMI method. This method offers a probabilistic formulation that effectively unifies challenges related to both label redundancy and insufficiency, drawing on principles of mutual information. (2) Experimental results demonstrate the effectiveness of our method compared with the direct combination of current state-of-the-art PLL and SSL methods. We additionally conduct experiments on the partial-complementary label learning problem, further confirming the flexibility of the uniform treatment for mixed scenarios.

## 2. Related Work

In this section, we briefly outline the progress in three aspects of weak supervision: partial label learning, complementary label learning, and semi-supervised learning. Furthermore, we discuss semi-supervised partial label learning, which represents a typical hybrid domain.

**Partial label learning.** Traditional PLL has two principal research directions: the identification-based strategy (IBS) and the average-based strategy (ABS). In the IBS, label disambiguation is conducted by selecting the most likely true label from the candidate label set for training (Jin & Ghahramani, 2002; Chen et al., 2014; Feng & An, 2019). In contrast, the ABS assumes equal probabilities for all labels in the candidate label set and utilizes all candidate labels for training (Hüllermeier & Beringer, 2006; Cour et al., 2011; Zhang et al., 2017). Leveraging the powerful capabilities of deep neural networks, substantial research progress has been achieved in deep PLL (Lv et al., 2020; Xu et al., 2021). Wang *et al.*(2022) introduced contrastive learning, utilizing prototypes to guide label disambiguation. Wu *et al.*(2022) performed consistency regularization through multiple augmented alignments. Xu *et al.*(2023a) utilized label enhancement (Xu et al., 2019; 2020; 2023b) to purify the candidate label sets and refine the classifier iteratively. However, PLL is a paradigm designed to handle label information redundancy, based on the assumption that each instance is associated with a candidate label set, making it challenging to address other types of information status issues.

**Complementary label learning.** CLL (Ishida et al., 2017; Yu et al., 2018; Ishida et al., 2019) assigns each training instance one complementary label, indicating the class it does not belong to. Ishida *et al.*(2017) proposed the CLL setting and provided conditions for obtaining unbiased estimates of the classification risk from complementary labeled data. Gao *et al.*(2021) derived a theoretically sound discriminative model and introduced weighted loss to maximize the predictive gap between potential true label and the complementary label. CLL can be considered an extreme case of PLL with maximum redundancy in label information, where each instance is associated with a candidate label set that excludes only one label.

**Semi-supervised learning.** To alleviate underfitting caused by data scarcity, SSL is proposed, which involves learning from a training set consisting of a limited quantity of labeled data and a large amount of unlabeled data. Deep SSL (Yang et al., 2022) can be categorized into four technical directions: deep generative methods (Kingma et al., 2014; Li et al., 2019; Liu et al., 2020), consistency regularization methods (Sajjadi et al., 2016; Tarvainen & Valpola, 2017; Xie et al., 2020), graph-based methods (Wang et al., 2016; Kipf & Welling, 2017), and pseudo-labeling methods (Qiao et al., 2018; Chen et al., 2020). Recent popular SSL methods (Verma et al., 2019; Berthelot et al., 2019; Sohn et al., 2020) primarily incorporate a mixture of various techniques, such as consistency regularization, entropy minimization, and data augmentation. For instance, FixMatch (Sohn et al., 2020) employed a fixed confidence threshold to pseudo-label the weakly-augmented outputs,

which are subsequently aligned with the strongly-augmented outputs. Subsequent works have further extended it, placing emphasis on the flexible selection of the pseudo-labeling threshold (Zhang et al., 2021; Wang et al., 2023). Nevertheless, SSL relies on the assumption that labeled data is fully supervised, and if the supervision information is inexact, it can significantly impact the model's regularization.

**Semi-supervised partial label learning.** Due to the difficulty and diversity of data annotation, addressing the challenge of effectively leveraging mixed weak supervision annotations has become imperative. SSPLL has emerged as a scenario to meet this demand, combining partial label data and unlabeled data for learning. SSPL (Wang et al., 2019) employed label propagation to disambiguate the candidate label sets of partial label data and assigned valid labels to unlabeled data. PARM (Wang & Zhang, 2020) employed label propagation to instantiate the labeling confidence of partial label data and introduced confidence-rated margin maximization to jointly optimize the model and estimate latent labeling confidence for unlabeled data. The above methods independently handle partial label data and unlabeled data, falling into the strategy of concatenating two tasks. There is an urgent need for a uniform framework to simultaneously handle data with various types of annotation information.

## 3. Method

### 3.1. Preliminaries

Consider a $c$-class classification problem. Let $\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^d$ be the features and $y \in \mathcal{Y} = \{1, 2, \ldots, c\}$ be the labels. In ordinary multi-class classification, training samples are independently drawn from an unknown probability distribution with density $p(\boldsymbol{x}, y)$. The goal is to obtain a multi-class classifier $f : \mathcal{X} \to \mathbb{R}^c$ that minimizes the classification risk: $\mathcal{R} = \mathbb{E}_{p(\boldsymbol{x},y)}\mathcal{L}(f(\boldsymbol{x}), y)$, where $\mathbb{E}$ denotes the expectation and $\mathcal{L}$ is a loss function. In PLL, a dataset denotes by $\mathcal{D}_P = \{\boldsymbol{x}_i, S_i\}_{i=1}^L$, where $S_i$ is the candidate label set of $\boldsymbol{x}_i$. We require each candidate label set not to be empty nor the whole label set, i.e., $S_i \in \mathcal{S}, \mathcal{S} = \{2^{\mathcal{Y}} \backslash \mathcal{Y} \backslash \emptyset\}$, where $2^{\mathcal{Y}}$ denotes the power set. PLL training data are drawn from a corrupted distribution $p(\boldsymbol{x}, S)$ of $p(\boldsymbol{x}, y)$ with $p(\boldsymbol{x})$ unchanged.

In the semi-supervised partial label learning task, there is also an unlabeled dataset $\mathcal{D}_U = \{\boldsymbol{x}_{L+i}\}_{i=1}^U$ drawn from $p(\boldsymbol{x}, y)$, while the labels are inaccessible, and generally $L \ll U$. Compared with the PLL task, it is critical to leverage the unlabeled data in semi-supervised learning.

### 3.2. The Uniform Treatment from Data Perspective

An intuitive treatment to tackle semi-supervised partial label learning is to utilize existing PLL methods to learn from partial label data, followed by assigning pseudo labels to unlabeled data, and then jointly employing both subsets for training. This perspective separates partial label data and unlabeled data without establishing an effective connection between the disambiguation of candidate labels and the pseudo label assignment for unlabeled data. Furthermore, the existing SSPLL methods (Wang et al., 2019; Wang & Zhang, 2020) focus on utilizing unlabeled data to populate the feature representation space, aiding label disambiguation and establishing smooth decision boundaries. However, a critical foundational fact has been overlooked: the only distinction between partial label data and unlabeled data lies in the density of label information, and the objectives of label disambiguation and pseudo-labeling are essentially the same. Therefore, it is reasonable to handle both within the same framework simultaneously.

Our proposed framework adopts an egalitarian principle, treating all data uniformly by considering partial label data as redundant information and unlabeled data as insufficient information. Concretely, each instance is associated with a variable pseudo candidate label set, serving as a repository for the currently potential true labels, instead of enforcing the selection of a single label through a threshold. To facilitate the progressive process of candidate label generation and redundant label disambiguation, it is necessary to design a label channel for dynamically including and excluding candidate labels.

In our methodology, we consistently apply supervised loss to all instances with pseudo candidate label sets. Let $S_i^t$ denote the generated candidate label set of instance $\boldsymbol{x}_i$ in the $t$-th epoch and let $f_j(\boldsymbol{x}_i)$ represent the output of the model for the $j$-th class on the $i$-th instance. The loss function of our framework can be expressed as:

$$\mathcal{L} = \sum_{i=1}^{n} \sum_{j=1}^{c} w_{ij} \ell(f_j(\boldsymbol{x}_i), S_i^t), \tag{1}$$

where $\ell$ is the cross-entropy loss and the weight $w_{ij}$ is updated by the current model output weight corresponding to each candidate label (Lv et al., 2020):

$$w_{ij} = \begin{cases} \frac{f_j(\boldsymbol{x}_i)}{\sum_{k \in S_i^t} f_k(\boldsymbol{x}_i)} & \text{if } j \in S_i^t \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

This encourages the model to enhance outputs for labels in the generated candidate label set and motivates the learning of suitable feature representations, allocating higher weights to more probable labels, thereby gradually revealing potential true labels.

According to Eq.(1), our framework emphasizes the explicit manipulation of the candidate label set $S$ rather than optimizing the weight $w$. The objective is to identify candidate labels in each epoch and optimize the model using pseudo

candidate label sets. The pseudo candidate label set can be dynamically manipulated based on the relationship between the features of each instance and its label space, where relevant labels are added while irrelevant ones are removed, ultimately facilitating the identification of true labels. The framework consists of two components: *label expansion* and *label condensation*, which are respectively responsible for adding candidate labels and removing irrelevant labels. These components are detailed in the following subsections.

### 3.3. Label Expansion

*Label expansion* aims to assign pseudo candidate labels to unlabeled data and recover mistakenly removed labels for partial label data.

**Initialization.** Notice that for unlabeled data, the candidate labels are entirely absent. The purpose of initialization is to guarantee that, even in the early stages of training when the model's learning is insufficient, the true labels can be added to the pseudo candidate label sets with high credibility. The initialization process plays a crucial role in providing a substantial quantity of high-quality labels, effectively reducing the label space and expediting the training process.

For an instance $\boldsymbol{x}_i$, if $f_k(\boldsymbol{x}_i) > 1/c$, then:

$$\ell(f(\boldsymbol{x}_i), k) < \frac{1}{c}\sum_{j=1}^{c}\ell(f(\boldsymbol{x}_i), j), \tag{3}$$

where $\ell$ is a convex function. The derivation is provided in Appendix A.1. This indicates that if a label has a confidence greater than the random output (i.e., $1/c$), its loss value will be smaller than the average loss value for all labels. Under the mild assumption that the output for the true label is greater than the random output, we can infer an equivalent contrapositive statement: if the loss for a label is greater than the average loss value for all labels, then this label is not the true label. In other words, labels that satisfy this condition can be considered as complementary labels, thereby establishing the initial candidate label sets in reverse. Thanks to this mild assumption, it is feasible to assign initial candidate labels to unlabeled training samples in a reliable manner.

**Label Generation.** After a period of training, the model incrementally learns proper patterns associated with each class, identifying potential true labels that are not present in the current candidate label set. During each epoch, this requires adding potential candidate labels for unlabeled data and reinstating possibly correct but previously removed labels for partial label data from the original candidate label set.

To achieve this objective, we introduce mutual information (MI) (Tishby et al., 2000; Tishby & Zaslavsky, 2015; Alemi et al., 2017) to dynamically generate candidate labels. MI is a measure of the correlation between random variables and can be used to estimate information gain under a given condition. Information Bottleneck (IB) (Tishby & Zaslavsky, 2015) is an application of MI in neural networks, and its optimization objective aligns with that of neural networks. Considering input $X$, extracted feature $Z$, and label $Y$ as random variables, the aim of the IB is to minimize:

$$\mathcal{L}_{IB} = I(X;Z) - \beta I(Z;Y), \tag{4}$$

where $\beta$ as a trade-off parameter that controls the compression ratio of the feature representation. Specifically, $I(X;Z)$ is considered as the encoder, with the objective of compressing the input $X$ as much as possible to derive an optimal representation $Z$. On the other hand, $I(Z;Y)$ acts as the decoder, responsible for maintaining consistency between the compressed representation $Z$ and the label $Y$.

As $I(X;Z)$ corresponds to the feature extraction encoder through a neural network and its optimization is independent of the operations on the candidate label set, we exclusively concentrate on the optimization to maximize $I(Z;Y)$ in the subsequent discussion. According to the definition of mutual information, we have

$$\begin{aligned} I(Z;Y) &= \int p(\boldsymbol{z}, y)\log\frac{p(\boldsymbol{z}, y)}{p(\boldsymbol{z})p(y)}\mathrm{d}\boldsymbol{z}\mathrm{d}y \\ &= \int p(\boldsymbol{z}, y)\log\frac{p(y|\boldsymbol{z})}{p(y)}\mathrm{d}\boldsymbol{z}\mathrm{d}y. \end{aligned} \tag{5}$$

Given the challenge of obtaining the actual distribution $p(y|\boldsymbol{z})$, inspired by (Alemi et al., 2017), we utilize the output $q(y|\boldsymbol{z})$ of neural network as an approximation. Since the Kullback-Leibler divergence is always non-negative, denoted as $D_{KL}[p(Y|Z)||q(Y|Z)] \geq 0$, we have

$$\int p(y|\boldsymbol{z})\log p(y|\boldsymbol{z})\mathrm{d}y \geq \int p(y|\boldsymbol{z})\log q(y|\boldsymbol{z})\mathrm{d}y. \tag{6}$$

Thus there is:

$$I(Z;Y) \geq \int p(\boldsymbol{z}, y)\log\frac{q(y|\boldsymbol{z})}{p(y)}\mathrm{d}\boldsymbol{z}\mathrm{d}y \doteq R. \tag{7}$$

Assuming that $z$ is determined solely by $x$ and is independent of $y$, there is $p(\boldsymbol{z}|\boldsymbol{x}, y) = p(\boldsymbol{z}|\boldsymbol{x})$. So we have $p(\boldsymbol{z}, y) = \int p(\boldsymbol{x}, \boldsymbol{z}, y)\mathrm{d}\boldsymbol{x} = \int p(\boldsymbol{x}, y)p(\boldsymbol{z}|\boldsymbol{x}, y)\mathrm{d}\boldsymbol{x} = \int p(\boldsymbol{x}, y)p(\boldsymbol{z}|\boldsymbol{x})\mathrm{d}\boldsymbol{x}$. Consequently, the right term of Eq.(7) can be expressed as:

$$R = \int p(\boldsymbol{x}, y)p(\boldsymbol{z}|\boldsymbol{x})\log\frac{q(y|\boldsymbol{z})}{p(y)}\mathrm{d}\boldsymbol{z}\mathrm{d}y\mathrm{d}\boldsymbol{x}. \tag{8}$$

Then it can be approximated as:

$$\begin{aligned} \hat{R} &= \frac{1}{n}\sum_{i=1}^{n}\int p(\boldsymbol{z}|\boldsymbol{x}_i)\log\frac{q(y_i|\boldsymbol{z})}{p(y_i)}\mathrm{d}\boldsymbol{z} \\ &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\boldsymbol{x}_i)}\log\frac{q(y_i|\boldsymbol{z})}{p(y_i)}. \end{aligned} \tag{9}$$

According to Eq.(7) and Eq.(9), we can obtain:

$$I(Z;Y) \geq \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\boldsymbol{x}_i)}\log\frac{q(y_i|\boldsymbol{z})}{p(y_i)}. \quad (10)$$

Guided by Eq.(10), an intuitive approach is to use a greedy strategy to determine whether the label $k \notin S_i$ should be added to the candidate label set $S$ by calculating $I(Z,k)$. According to the properties of mutual information, $I(Z,k) > 0$ indicates that $k$ is related to $Z$ and then $k$ can be considered as a potential true label. For each label $k$ corresponding to the features $\boldsymbol{z}_i$ of instance $\boldsymbol{x}_i$, if $k$ satisfies:

$$\begin{aligned}&\log\frac{q(k|\boldsymbol{z}_i)}{p(k)} > 0\\ &\Leftrightarrow q(k|\boldsymbol{z}_i) > p(k),\end{aligned} \quad (11)$$

where $p(k)$ is the $k$-th class prior probability, then $k$ is added to the candidate label set.

### 3.4. Label Condensation

Unlike *label expansion* which aims to select numerous potential true labels, *label condensation* involves removing the most reliably incorrect labels from the candidate label set to boost the information density of the remaining labels. Once the true label is erroneously removed, it misaligns the model's optimization objective, resulting in intolerable consequences. Hence, a rational approach is to remove the label with the least probability from the candidate label set. Guided by this principle, in our framework, multiple candidate labels are generated, but only one candidate label is removed in each epoch.

Similar to the calculations in Section 3.3, there is:

$$I(Z;Y) \geq \frac{1}{n}\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\boldsymbol{x}_i)}\log q(y_i|\boldsymbol{z}) + H(Y), \quad (12)$$

where $H(Y)$ is the entropy of $Y$. The derivation is provided in Appendix A.2. Since entropy is non-negative, we have

$$I(Z;Y) \geq \frac{1}{n}\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\boldsymbol{x}_i)}\log q(y_i|\boldsymbol{z}). \quad (13)$$

In our framework, we can only access the pseudo candidate label set and use it as a substitute for the label $Y$. Let $\boldsymbol{s}_i$ represent the uniform label distribution related to $S_i$ (e.g., if $c = 3$ and $S_i = \{1,3\}$, then $\boldsymbol{s}_i = [0.5, 0, 0.5]$). For each label $k \in S_i$, define $T_i^k = S_i \setminus k$, which represents the removal of $k$ from $S_i$. Let $\boldsymbol{t}_i^k$ represent the uniform label distribution of $T_i^k$. To measure the impact of removing a label, we quantify the change in mutual information. Intuitively, removing the least relevant labels has the least impact on the change in mutual information. Therefore, we introduce a method to align their mutual information by minimizing the difference between their model outputs. To achieve this, we can use the KL divergence to measure the gap between them:

$$D = D_{KL}[q(\boldsymbol{t}_i^k|\boldsymbol{z}_i)||q(\boldsymbol{s}_i|\boldsymbol{z}_i)]. \quad (14)$$

Define an information score function

$$G(\boldsymbol{x}_i, S_i, k) = D_{KL}[(f_{S_i\setminus k}(\boldsymbol{x}_i)||f_{S_i}(\boldsymbol{x}_i)], \quad (15)$$

where $f_{S_i\setminus k}(\boldsymbol{x}_i)$ and $f_{S_i}(\boldsymbol{x}_i)$ respectively denote the model output distributions on $S_i \setminus k$ and $S_i$. It quantifies the extent of information loss after condensing the candidate label set. The value of this function decreases when the mutual information between $k$ and $\boldsymbol{x}_i$ reduces, signifying a weakened relationship between them. According to the property of this function, the least probable label can be identified by the minimum value of $G(\boldsymbol{x}_i, S_i, k)$ of the instance $\boldsymbol{x}_i$. Our objective is to find $k$ from $S_i$ under the given $\boldsymbol{x}_i$ and $S_i$ such that it minimizes $G(\boldsymbol{x}_i, S_i, k)$.

**Algorithm 1** SPMI

**Input:** The partial label training set $\mathcal{D}_P$, the unlabeled training set $\mathcal{D}_U$, the threshold $\tau$, the class prior $\mu$, the number of epoch $T$, the warm-up epoch $T_w$;

**for** $t = 1, ..., T$ **do**
  **if** $t < T_w$ **then**
    Train the predictive model $f$ on $\mathcal{D}_P$ by Eq.(1);
  **else**
    **if** $t == T_w$ **then**
      Initialize pseudo candidate label sets on $\mathcal{D}_U$ by Eq.(3);
      $\mu^t = \mu$;
    **end if**
    **for** $i = 1, ..., n$ **do**
      $S_i^{t+1} = S_i^t$;
      **if** $\exists j \in S_i^t, G(\boldsymbol{x}_i, S_i^t, j) > \tau$ **then**
        $k = \arg\min_j G(\boldsymbol{x}_i, S_i^t, j)$;
        $S_i^{t+1} = S_i^t \setminus k$;
      **end if**
      **for** $j \notin S_i^t$ **do**
        **if** $f_j(\boldsymbol{x}_i) > \mu_j^t$ and ($\boldsymbol{x}_i \in \mathcal{D}_U$ or ($\boldsymbol{x}_i \in \mathcal{D}_P$ and $j \in S_i^0$)) **then**
          $S_i^{t+1} = S_i^{t+1} \cup j$;
        **end if**
      **end for**
    **end for**
    Train the predictive model $f$ with $S^{t+1}$ on $\mathcal{D}_P \bigcup \mathcal{D}_U$ by Eq.(1);
    Update the $\mu^{t+1}$ using Eq.(16);
  **end if**
**end for**

**Output:** The predictive model $f$.

*Table 1.* Test accuracy (mean±std) of each PLL approach on benchmark datasets under different numbers of labeled instances $l$ and partial rate $p$.

| Dataset | F-MNIST | | | CIFAR-10 | | CIFAR-100 | | SVHN |
|---|---|---|---|---|---|---|---|---|
| $l$ | 1000 | 4000 | | 1000 | 4000 | 5000 | 10000 | 1000 |
| $p$ | 0.3 | 0.3 | 0.7 | 0.3 | 0.3 | 0.05 | 0.05 | 0.3 |
| PRODEN | 86.18±0.69 | 89.08±0.11 | 86.05±0.15 | 88.85±0.78 | 92.19±0.13 | 52.06±0.77 | 70.90±0.30 | 96.51±0.19 |
| PLCR | 84.59±0.44 | 88.59±0.21 | 85.92±0.30 | 71.19±0.82 | 85.71±0.27 | 27.23±0.15 | 64.36±0.54 | 90.41±1.20 |
| POP | 84.21±0.87 | 88.18±0.13 | 85.35±0.41 | 68.63±1.28 | 85.75±0.23 | 43.04±0.34 | 65.21±0.53 | 66.33±1.31 |
| SPMI | **86.47**±0.30 | **89.82**±0.04 | **86.83**±0.08 | **90.41**±0.24 | **92.81**±0.06 | **66.43**±0.48 | **73.63**±0.37 | **96.59**±0.14 |

*Table 2.* Test accuracy (mean±std) of each SSL approach on benchmark datasets under different numbers of labeled instances $l$ and partial rate $p$.

| Dataset | F-MNIST | | | CIFAR-10 | | CIFAR-100 | | SVHN |
|---|---|---|---|---|---|---|---|---|
| $l$ | 1000 | 4000 | | 1000 | 4000 | 5000 | 10000 | 1000 |
| $p$ | 0.3 | 0.3 | 0.7 | 0.3 | 0.3 | 0.05 | 0.05 | 0.3 |
| MixMatch | 83.70±0.85 | 87.82±0.03 | 85.46±0.42 | 40.03±0.81 | 82.85±0.33 | 36.44±0.81 | 55.75±1.33 | 89.25±1.39 |
| Fixmatch | 86.18±0.69 | 89.08±0.11 | 86.05±0.15 | 88.85±0.78 | 92.19±0.13 | 52.06±0.77 | 70.90±0.30 | 96.51±0.19 |
| FlexMatch | 85.86±0.36 | 89.54±0.09 | 86.79±0.26 | 91.08±0.11 | 92.32±0.20 | 64.61±0.13 | 73.36±0.16 | 94.04±0.84 |
| FreeMatch | 86.12±0.40 | 89.76±0.13 | 86.76±0.23 | **91.20**±0.41 | 92.73±0.02 | **67.81**±0.44 | 73.15±0.17 | 91.80±0.70 |
| SPMI | **86.47**±0.30 | **89.82**±0.04 | **86.83**±0.08 | 90.41±0.24 | **92.81**±0.06 | 66.43±0.48 | **73.63**±0.37 | **96.59**±0.14 |

From another perspective, if the removed label is the true label, it would result in significant information loss, leading to a large value for $G$. Therefore, the maximum value of $G(\boldsymbol{x}_i, S_i, k)$ can serve as an indicator of the true label prominence and learning effectiveness of the instance $\boldsymbol{x}_i$. Unlike current prevalent SSL methods, our approach establishes a threshold based on the distribution of all possible labels, rather than relying on a single label with the highest confidence. This facilitates the comprehensive utilization of samples where multiple label outputs exhibit high confidence.

Based on the above analysis, if there exist $(\boldsymbol{x}_i, S_i)$ such that $G_{max}(\boldsymbol{x}_i, S_i, k) > \tau$ for $k \in S_i$, where $\tau$ is a hyper-parameter, then $j = \arg\min_k G(\boldsymbol{x}_i, S_i, k)$ and $j$ is removed from the candidate label set $S_i$.

### 3.5. Implementation Details

Our framework integrates label generation and disambiguation, employing *label expansion* to add possible labels to the candidate label set, and *label condensation* to remove the least probable label from the candidate label set. Specifically, we initially train only on $\mathcal{D}_P$ during the warm-up period, then utilize Eq.(3) to initialize the pseudo candidate label sets for $\mathcal{D}_U$. Subsequently, before each training epoch, we perform both label generation and label condensation on $\mathcal{D}_P \bigcup \mathcal{D}_U$, ensuring that the number of candidate labels is within the range of $[1, c-1]$. The training loss function is given by Eq.(1).

In Section 3.3, the class prior probability is required. If the dataset is balanced, the class prior is $1/c$. However, even if the number of true labels is balanced in PLs, label ambiguity can impact the bias of class priors. In practice, the $j$-th class prior $\mu_j$ can be approximated using the class posterior from the previous epoch:

$$\mu_j = \frac{1}{n} \sum_{i=1}^{n} f_j(\boldsymbol{x}_i). \tag{16}$$

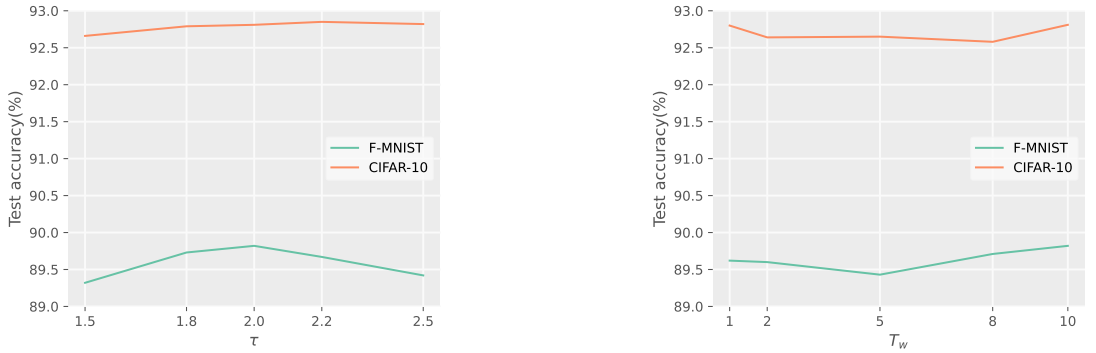
Algorithm 1 describes the algorithm process of SPMI.

## 4. Experiments

### 4.1. Datasets

To validate the effectiveness of our framework, we utilize four extensively employed benchmark datasets, including `Fashion-MNIST` (Xiao et al., 2017), `CIFAR-10`, `CIFAR-100` (Krizhevsky & Hinton, 2009), and `SVHN` (Netzer et al., 2011). Following previous research in PLL (Wang et al., 2022; Wu et al., 2022) and SSL (Sohn et al., 2020; Zhang et al., 2021; Wang et al., 2023), our experiments encompass diverse combinations involving varying partial rates $p$ and varying numbers of labeled instances $l$.

For the training set of each dataset, we initially partition the data into a labeled subset and an unlabeled subset, and then manually corrupt labeled subset into partially labeled




(a) Sensitivity analysis of $\tau$

(b) Sensitivity analysis of $T_w$

*Figure 1.* The sensitivity analysis on F-MNIST and CIFAR-10.

versions by uniform generation process (Lv et al., 2020). Specifically, we first extract a labeled subset from the training set by averaging sampling per class according to the given total number of labeled instances $l$, while the remaining samples constitute the unlabeled subset after removing labels. Subsequently, for the labeled subset, we employ a unified generation process to construct the candidate label set. In this process, for each sample, the true label is added to the candidate label set, and each incorrect label $\bar{y}$ is added with a flipping probability $p$ ($0 < p < 1$), where $p$ is the partial rate.

### 4.2. Baselines

Previous works (Wang et al., 2019; Wang & Zhang, 2020) in SSPLL are based on traditional techniques and are not inherently applicable to deep neural networks. Given the current lack of research on deep SSPLL, for a fair comparison, it is necessary to conduct comparisons by combining existing state-of-the-art PLL and SSL methods to adapt to the SSPLL scenario. Our comparative experiments involve three PLL methods and four SSL methods. Details of the baselines are described in Appendix A.3.

In the comparison experiment with PLL methods, considering the embedability of methods, we choose FixMatch from the SSL methods as the fundamental method. We select the following PLL approaches: (1) PRODEN (Lv et al., 2020); (2) PLCR (Wu et al., 2022); (3) POP (Xu et al., 2023a), combined with FixMatch.

Additionally, in the comparison experiment with SSL method, we choose PRODEN from the PLL methods as the basic method. We employ the following SSL approaches: (1) MixMatch (Berthelot et al., 2019); (2) FixMatch (Sohn et al., 2020); (3) FlexMatch (Zhang et al., 2021); (4) FreeMatch (Wang et al., 2023), combined with PRODEN.

We employ the same backbone, optimizer, epochs and data augmentation to train all methods on the same dataset. We use LeNet (LeCun et al., 1998) for F-MNIST, Wide-ResNet-28-2 (Zagoruyko & Komodakis, 2016) for CIFAR-10 and SVHN, and Wide-ResNet-28-8 (Zagoruyko & Komodakis, 2016) for CIFAR-100. We apply the same data augmentation strategy to all methods, including PRODEN, which originally did not have data augmentation. The initial value for the class prior $\mu$ is set to $1/c$. The threshold $\tau$ is configured to be 3 for partial label data and 2 for unlabeled data. We run three trials with different random seeds to record the mean and standard deviation. More details on the experimental settings are provided in Appendix A.4.

*Table 3.* Test accuracy (mean±std) of SPMI and its variants in ablation study. The w/o indicates removing this component.

| Method | F-MNIST | CIFAR-10 | CIFAR-100 | SVHN |
|---|---|---|---|---|
| SPMI | **89.82**±0.04 | **92.81**±0.06 | **73.63**±0.37 | **96.59**±0.14 |
| SPMI w/o *init* | 89.38±0.15 | 92.47±0.12 | 71.95±0.28 | 96.26±0.26 |
| SPMI w/o $LG$ | 87.51±0.36 | 74.64±0.81 | 40.12±1.17 | 82.99±0.94 |
| SPMI w/o $LC$ | 87.56±0.28 | 85.14±0.55 | 64.36±0.84 | 91.13±0.51 |

### 4.3. Experimental Results

Table 1 reports the comparison results with PLL approach combined with FixMatch on benchmark datasets. The best results are highlighted in bold. SPMI achieves the best performance against the variations of existing PLL approaches and exhibits significant performance gaps in many cases. The results indicate that when labeled data is extremely limited, methods tailored for PLL may mislead classifiers or even fail if they do not effectively leverage unlabeled data for label disambiguation.

Table 2 reports the comparison results with SSL approach combined with PRODEN on benchmark datasets. The results demonstrate that SPMI outperforms or achieves competitive results compared to other methods. Although SPMI shows slightly weaker performance when there is very little annotated data, such as CIFAR-10 with $l = 1000, p = 0.3$ and CIFAR-100 with $l = 5000, p = 0.05$, it should be noted

*Table 4.* Test accuracy (mean±std) of each CLL approach on benchmark datasets under different numbers of partial label instances $l_p$, numbers of complementary label instances $l_c$ and partial rate $p$.

| Dataset | F-MNIST | CIFAR-10 | CIFAR-100 | SVHN |
|---|---|---|---|---|
| $l_p$ | 30000 | 25000 | 25000 | 36000 |
| $l_c$ | 30000 | 25000 | 25000 | 37257 |
| $p$ | 0.3 | 0.3 | 0.05 | 0.3 |
| Forward | 92.65±0.06 | 93.45±0.24 | 75.02±0.42 | 96.86±0.01 |
| L-W | 92.49±0.03 | 93.44±0.15 | 75.19±0.39 | 96.77±0.05 |
| NLL | 92.59±0.04 | 93.55±0.13 | 75.35±0.28 | 96.87±0.04 |
| SPMI | **92.69**±0.07 | **94.55**±0.16 | **78.67**±0.06 | **97.04**±0.02 |

that FlexMatch and FreeMatch are SSL methods explicitly designed for scenarios with extremely limited labeled data.

### 4.4. Further Analysis

**Ablation study.** To assess the effectiveness of each component of our framework, an ablation study is conducted to measure their contributions. Our framework consists of three core components: initialization (*init*), label generation (*LG*), and label condensation (*LC*). Considering that ablating on the partial label data would significantly impact performance, we conduct ablation experiments only on the unlabeled data operations while keeping the operations on partial label data unchanged to evaluate in a reasonable manner. The ablation study is conducted on F-MNIST and CIFAR-10 with $l = 4000, p = 0.3$, CIFAR-100 with $l = 10000, p = 0.05$, and SVHN with $l = 1000, p = 0.3$. As shown in Table 3, each component contributes to the overall performance, with the effects of *LG* and *LC* being more pronounced. It is evident that *LG* effectively recovers unrecognized labels into the candidate labels, and *LC* successfully eliminates redundant labels, while the contribution of initialization is mainly to expedite the convergence speed of training by assigning a substantial number of initial labels.

**Sensitivity analysis.** Figures 1(a) and 1(b) illustrate the sensitivity analysis of the threshold $\tau$ and the warm-up epoch $T_w$ on F-MNIST and CIFAR-10 with $l = 4000, p = 0.3$ under different parameters. The hyper-parameters $\tau$ and $T_w$ are varied within the ranges of $\{1.5, 1.8, 2.0, 2.2, 2.5\}$ and $\{1, 2, 5, 8, 10\}$, respectively. The experimental results demonstrate that, for different parameters, the variation range of accuracy is approximately within $0.5\%$, indicating that our framework is robust to the choice of these two hyper-parameters under mild settings.

**The effectiveness of the pseudo-labeling mechanism.** To investigate the effectiveness of the joint label generation and label disambiguation mechanism, we record the error rate (i.e., the true label is not present in the pseudo candidate label set) and the average number of pseudo candidate labels for unlabeled data during the training process. We conduct experiments on F-MNIST with $l = 4000, p = 0.3$, and the results are shown in Figures 2 and 3. After the warm-up period, a large number of pseudo candidate labels are initialized for unlabeled instances, resulting in only an approximately 5% noise rate in the constructed pseudo candidate label sets, thus demonstrating the efficiency of the initialization process. As training progresses, the average number of labels in the candidate label set significantly decreased, while the error rate remained stable. This indicates that our framework is capable of progressively purifying true labels while maintaining a low error rate.

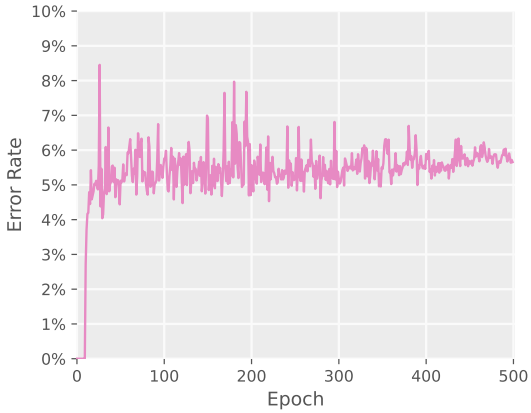



*Figure 2.* The error rate of the pseudo candidate label sets for unlabeled data on F-MNIST.

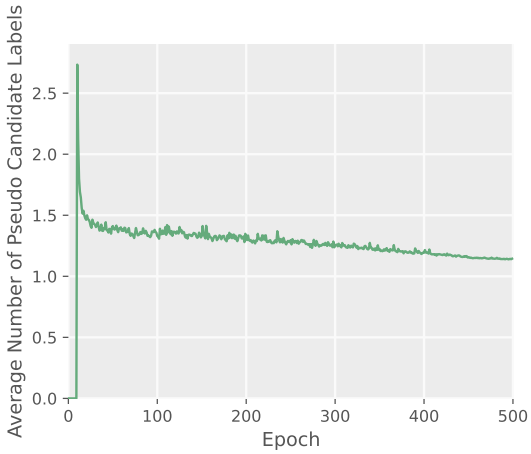



*Figure 3.* The average number of pseudo candidate labels for unlabeled data on F-MNIST.

**The compatibility on PCLL tasks.** To verify the compatibility of our framework with other weakly supervised learning tasks, we further extend experiments to datasets containing a mixture of partial labels and complementary labels, a scenario known as partial-complementary label learning

(PCLL). Experiment details for PCLL are provided in Appendix A.5. Table 4 reports the comparison results with CLL approach combined with PRODEN on benchmark datasets. The results demonstrate that our framework consistently outperforms all compared methods, underscoring its effectiveness in handling the mixture task of partial labels and complementary labels. The supplementary experiments in Appendix A.6 further validate its compatibility across other weakly supervised learning scenarios.

## 5. Conclusion

This paper explores the SSPLL problem which combines different forms of inexact supervision and proposes a novel approach named SPMI to uniformly treat label redundancy and insufficiency. The design of SPMI is rooted in mutual information, establishing a channel for labels to circulate through the candidate label sets for all data, independent of their initial state. Extensive experiments on benchmark datasets have validated that SPMI surpasses the direct combination of PLL and SSL methods, and experiments on the PCLL problem demonstrate the compatibility of our method with scenarios involving mixed inaccurate supervision information.

## Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## Acknowledgements

This research was supported by the National Science Foundation of China (62206050, 62125602, and 62076063), China Postdoctoral Science Foundation (2021M700023), Jiangsu Province Science Foundation for Youths (BK20210220), Young Elite Scientists Sponsorship Program of Jiangsu Association for Science and Technology (TJ-2022-078), the Fundamental Research Funds for the Central Universities (2242024k30035), and the Big Data Computing Center of Southeast University.

## References

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*, 2017.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems 32 (NeurIPS'19)*, pp. 5050–5060, 2019.

Brabham, C. D. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1): 75–90, 2008.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, volume 119, pp. 1597–1607. PMLR, 2020.

Chen, Y.-C., Patel, V. M., Chellappa, R., and Phillips, P. J. Ambiguously labeled learning using dictionaries. *IEEE Transactions on Information Forensics and Security*, 9 (12):2076–2088, 2014.

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

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR'19)*, pp. 113–123, 2019.

DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Feng, L. and An, B. Partial label learning with self-guided retraining. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI'19)*, pp. 3542–3549, 2019.

Feng, L., Lv, J., Han, B., Xu, M., Niu, G., Geng, X., An, B., and Sugiyama, M. Provably consistent partial-label learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, pp. 10948–10960, 2020.

Gao, Y. and Zhang, M. Discriminative complementary-label learning with weighted loss. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, volume 139, pp. 3587–3597. PMLR, 2021.

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

Ishida, T., Niu, G., Hu, W., and Sugiyama, M. Learning from complementary labels. In *Advances in Neural Information Processing Systems 30 (NeurIPS'17)*, pp. 5639–5649, 2017.

Ishida, T., Niu, G., Menon, A., and Sugiyama, M. Complementary-label learning for arbitrary losses and

models. In *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, volume 97, pp. 2971–2980. PMLR, 2019.

Jin, R. and Ghahramani, Z. Learning with multiple labels. In *Advances in Neural Information Processing Systems 15 (NeurIPS'02)*, pp. 897–904, 2002.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems 27 (NeurIPS'14)*, pp. 3581–3589, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*, 2017.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. 2009.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. In *Proceedings of 5th International Conference on Learning Representations (ICLR'17)*, 2017.

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

Li, Y., Pan, Q., Wang, S., Peng, H.and Yang, T., and Cambria, E. Disentangled variational auto-encoder for semi-supervised learning. *Information Sciences*, 482:73–85, 2019.

Liu, Y., Deng, G., Zeng, X., Wu, S., Yu, Z., and Wong, H. Regularizing discriminative capability of cgans for semi-supervised generative learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR'20)*, pp. 5720–5729, 2020.

Lv, J., Xu, M., Feng, L., Niu, G., Geng, X., and Sugiyama, M. Progressive identification of true labels for partial-label learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, volume 119, pp. 6500–6510. PMLR, 2020.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Nguyen, N. and Caruana, R. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (SIGKDD'08)*, pp. 551–559, 2008.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32 (NeurIPS'19)*, pp. 8024–8035, 2019.

Qiao, C., Xu, N., Lv, J., Ren, Y., and Geng, X. Fredis: A fusion framework of refinement and disambiguation for unreliable partial label learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, pp. 28321–28336, 2023.

Qiao, S., Shen, W., Zhang, Z., Wang, B., and Yuille, A. Deep co-training for semi-supervised image recognition. In *Proceedings of the 15th European Conference on Computer Vision (ECCV'18)*, volume 11219, pp. 135–152, 2018.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems 29 (NeurIPS'16)*, pp. 1163–1171, 2016.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, pp. 596–608, 2020.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30 (NeurIPS'17)*, pp. 1195–1204, 2017.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop*, pp. 1–5. IEEE, 2015.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Verma, V., Lamb, A., Kannala, J., Bengio, Y., and Lopez-Paz, D. Interpolation consistency training for semi-supervised learning. In *Proceedings of the 28nd International Joint Conference on Artificial Intelligence (IJCAI'19)*, pp. 3635–3641, 2019.

Wang, D., Cui, P., and Zhu, W. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (SIGKDD'16)*, pp. 1225–1234, 2016.

Wang, H., Xiao, R., Li, Y., Feng, L., Niu, G., Chen, G., and Zhao, J. Pico: Contrastive label disambiguation for partial label learning. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'22)*, 2022.

Wang, Q., Li, Y., Zhou, Z., et al. Partial label learning with unlabeled data. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*, pp. 3755–3761, 2019.

Wang, W. and Zhang, M. Semi-supervised partial label learning via confidence-rated margin maximization. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, pp. 6982–6993, 2020.

Wang, Y., Chen, H., Heng, Q., Hou, W., Fan, Y., Wu, Z., Wang, J., Savvides, M., Shinozaki, T., Raj, B., et al. Freematch: Self-adaptive thresholding for semi-supervised learning. In *Proceedings of the 11th International Conference on Learning Representations (ICLR'23)*, 2023.

Wu, D., Wang, D., and Zhang, M. Revisiting consistency regularization for deep partial label learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, volume 162, pp. 24212–24225. PMLR, 2022.

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

Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, pp. 6256–6268, 2020.

Xu, N., Liu, Y.-P., and Geng, X. Label enhancement for label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1632–1643, 2019.

Xu, N., Shu, J., Liu, Y.-P., and Geng, X. Variational label enhancement. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, pp. 10597–10606, 2020.

Xu, N., Qiao, C., Geng, X., and Zhang, M. Instance-dependent partial label learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS'21)*, pp. 27119–27130, 2021.

Xu, N., Liu, B., Lv, J., and Qiao, C.and Geng, X. Progressive purification for instance-dependent partial label learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, volume 202, pp. 38551–38565. PMLR, 2023a.

Xu, N., Shu, J., Zheng, R., Geng, X., Meng, D., and Zhang, M.-L. Variational label enhancement. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):6537–6551, 2023b.

Yang, X., Song, Z., King, I., and Xu, Z. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954, 2022.

Yu, X., Liu, T., Gong, M., and Tao, D. Learning with biased complementary labels. In *Proceedings of the 15th European Conference on Computer Vision (ECCV'18)*, volume 11205, pp. 68–83, 2018.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *Proceedings of the British Machine Vision Conference 2016*, 2016.

Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., and Shinozaki, T. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems 34 (NeurIPS'21)*, pp. 18408–18419, 2021.

Zhang, M. and Yu, F. Solving the partial label learning problem: An instance-based approach. In *Proceedings of the 24nd International Joint Conference on Artificial Intelligence (IJCAI'15)*, pp. 4048–4054, 2015.

Zhang, M., Yu, F., and Tang, C. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2155–2167, 2017.

Zhou, Z. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2017.

# A. Appendix

## A.1. The derivation of Eq.(3)

Due to the convexity of $\ell$, according to Jensen's inequality, it follows that $\ell(\frac{1}{c}\sum_{j=1}^{c} f_j(\boldsymbol{x}_i)) \leq \frac{1}{c}\sum_{j=1}^{c}\ell(f_j(\boldsymbol{x}_i))$. In practice, $-\ln(\boldsymbol{x}_i)$ is used as the loss function $\ell$.

For an instance $\boldsymbol{x}_i$, there is:

$$\begin{aligned} f_k(\boldsymbol{x}_i) > \frac{1}{c} &\Leftrightarrow -\ln(f_k(\boldsymbol{x}_i)) < -\ln(\frac{1}{c}) \\ &\Leftrightarrow -\ln(f_k(\boldsymbol{x}_i)) < -\ln(\frac{\sum_{j=1}^{c} f_j(\boldsymbol{x}_i)}{c}) \\ &\Leftrightarrow \ell(f_k(\boldsymbol{x}_i)) < \ell(\frac{\sum_{j=1}^{c} f_j(\boldsymbol{x}_i)}{c}) \\ &\Leftrightarrow \ell(f_k(\boldsymbol{x}_i)) < \frac{1}{c}\sum_{j=1}^{c}\ell(f_j(\boldsymbol{x}_i)) \\ &\Leftrightarrow \ell(f(\boldsymbol{x}_i), k) < \frac{1}{c}\sum_{j=1}^{c}\ell(f(\boldsymbol{x}_i), j). \end{aligned} \tag{17}$$

## A.2. The derivation of Eq.(12)

$$\begin{aligned} I(Z;Y) &= \int p(\boldsymbol{z}, y)\log\frac{p(y|\boldsymbol{z})}{p(y)}\mathrm{d}\boldsymbol{z}\mathrm{d}y \\ &\geq \int p(\boldsymbol{z}, y)\log\frac{q(y|\boldsymbol{z})}{p(y)}\mathrm{d}\boldsymbol{z}\mathrm{d}y \\ &= \int p(\boldsymbol{z}, y)\log q(y|\boldsymbol{z})\mathrm{d}\boldsymbol{z}\mathrm{d}y - \int p(y)\log p(y)\mathrm{d}y \\ &= \int p(\boldsymbol{x}, y)p(\boldsymbol{z}|\boldsymbol{x})\log q(y|\boldsymbol{z})\mathrm{d}\boldsymbol{z}\mathrm{d}y\mathrm{d}\boldsymbol{x} + H(Y). \end{aligned} \tag{18}$$

Then it can be approximated as:

$$\begin{aligned} I(Z;Y) &\geq \frac{1}{n}\sum_{i=1}^{N}\int p(\boldsymbol{z}|\boldsymbol{x}_i)\log q(y_i|\boldsymbol{z})\mathrm{d}\boldsymbol{z} + H(Y) \\ &= \frac{1}{n}\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\boldsymbol{x}_i)}\log q(y_i|\boldsymbol{z}) + H(Y). \end{aligned} \tag{19}$$

## A.3. Details of the baselines

The employed PLL methods include: (1) PRODEN (Lv et al., 2020) introduces a consistent classification risk estimator to update the model and performs label disambiguation through a progressive identification algorithm; (2) PLCR (Wu et al., 2022) utilizes consistency regularization of candidate labels to constrain the model, achieved by matching the multiple augmented outputs of an instance to a conformal label distribution; (3) POP (Xu et al., 2023a) updates the model and progressively purifies each candidate label set in every epoch with theoretical guarantees. In addition, the selected SSL methods consist of: (1) MixMatch (Berthelot et al., 2019) guesses low-entropy labels for data-augmented unlabeled data and conducts training with a mixture of labeled and unlabeled data using MixUp; (2) FixMatch (Sohn et al., 2020) uses high-confidence weakly-augmented prediction to generate pseudo-label, aligning it with the strongly-augmented output of the same image; (3) FlexMatch (Zhang et al., 2021) proposes a curriculum pseudo-labeling approach, flexibly adjusting the pseudo-labeling threshold for different classes at each time step; (4) FreeMatch (Wang et al., 2023) self-adaptively adjusts the confidence threshold based on the learning state of the model and designs a self-adaptive class fairness regularization penalty.

### A.4. Experimental Settings

The implementation is based on PyTorch (Paszke et al., 2019), and all experiments are conducted using NVIDIA RTX 3090 GPUs. We adopt the "weak" and "strong" augmentation strategy, where "weak" augmentation includes random horizontal flipping and random cropping, and "strong" augmentation builds upon the "weak" strategy and further incorporates AutoAugment (Cubuk et al., 2019) and Cutout (DeVries & Taylor, 2017). Due to the limitations of data augmentation techniques, we do not use AutoAugment on F-MNIST. The model is trained for 500 epochs and optimized by stochastic gradient descent (SGD) with a momentum of 0.9. The initial learning rate is selected from $\{0.05, 0.03, 0.01\}$ with cosine learning rate scheduling and the batch size is set to 256. The warm-up epoch $T_w$ is set from $\{5, 10, 20\}$. To maintain the stability of supervised information and reduce noise introduction, we employ the exponential moving average (EMA) to update the candidate label set in partial label data. Besides, we also use EMA to update the pseudo candidate label set for unlabeled data on CIFAR-100.

Considering the combined impact of two forms of weak supervision, we adopt a more lenient setting compared to each form individually. We set $\{l \in \{1000, 4000\}, p = 0.3\}$ and $\{l = 4000, p = 0.7\}$ for F-MNIST, $\{l \in \{1000, 4000\}, p = 0.3\}$ for CIFAR-10 and $\{l = 1000, p = 0.3\}$ for SVHN. Additionally, we set $\{l \in \{5000, 10000\}, p = 0.05\}$ for CIFAR-100.

Moreover, PiCO (Wang et al., 2022) is not included in the comparison experiments due to the contrastive learning classifier's reliance on a large number of negative samples and the extremely limited labeled data in the semi-supervised setting, resulting in the failure of model learning.

### A.5. Experimental Details of partial-complementary label learning

To make fair comparisons, we combine existing state-of-the-art CLL methods with the PLL method PRODEN. The compared CLL methods include: (1) Forward (Yu et al., 2018) extends standard deep neural network classifiers to learn with biased complementary labels and theoretically ensures that the classifier learned with complementary labels converges to the optimal one learned with true labels; (2) L-W (Gao & Zhang, 2021) derives a theoretically sound discriminative model and introduces weighted loss to maximize the predictive gap between potential ground-truth label and complementary label; (3) NLL (Wu et al., 2022) refers to the modified negative log likelihood loss, which is introduced by PLCR (Wu et al., 2022) to reduce the outputs of non-candidate labels.

The dataset generation process is similar to that in Section 4.1. First, the training data is divided into a partial label subset $\mathcal{D}_p$ and a complementary label subset $\mathcal{D}_c$. Then, a unified generation process (Lv et al., 2020) is applied to manually corrupt $\mathcal{D}_p$ into a partially labeled version based on the partial rate $p$. Meanwhile, $\mathcal{D}_c$ is assigned complementary labels based on the generation process following Ishida *et al.*(2017). Depending on the specific dataset, we configure different numbers of partial label instances $l_p$, numbers of complementary label instances $l_c$, and partial rate $p$. We set $\{l_p = 30000, l_c = 30000\}$ for F-MNIST, $\{l_p = 25000, l_c = 25000\}$ for CIFAR-10 and CIFAR-100, and $\{l_p = 36000, l_c = 37257\}$ for SVHN. Additionally, we set $p = 0.3$ for F-MNIST, CIFAR-10 and SVHN, and $p = 0.05$ for CIFAR-100. The other experimental settings remain consistent with those previously described.

### A.6. Supplementary experiments

To comprehensively assess the generality and efficacy of our framework, we conduct experiments on the combination of SSL and CLL here. We integrate existing state-of-the-art CLL methods with the SSL method FixMatch. The experimental setup involves 4000 samples with complementary labels, with the remaining samples being unlabeled for F-MNIST and CIFAR-10 datasets.

*Table 5.* Test accuracy of each CLL approach combined with FixMatch.

| Method | F-MNIST | CIFAR-10 |
|---|---|---|
| SPMI | **80.42** | **62.78** |
| Forward | 76.03 | 35.41 |
| L-W | 74.14 | 21.30 |
| NLL | 75.31 | 25.38 |

We also conduct experiments on mixed tasks involving PLL, SSL, and CLL. The baseline method for comparison consist of a combination of PRODEN (Lv et al., 2020), FixMatch (Sohn et al., 2020), and Forward (Yu et al., 2018). We set the number of partial labels $l_p$ to 2000 and the number of complementary labels $l_c$ to 2000 for both F-MNIST and CIFAR-10 datasets, with the remaining data being unlabeled. The partial rate $p$ is set to 0.3.

*Table 6.* Test accuracy of mixed approach involving PLL, SSL, and CLL.

| Method | F-MNIST | CIFAR-10 |
|---|---|---|
| SPMI | **88.52** | **91.61** |
| PRODEN + FixMatch + Forward | 81.27 | 48.07 |

Additionally, to further explore the applicability of our method, experiments are conducted on the independent tasks of SSL and PLL.

For the SSL experiments, we set the number of supervised labels to 4000 for F-MNIST and CIFAR-10 datasets.

*Table 7.* Test accuracy of each SSL approach.

| Method | F-MNIST | CIFAR-10 |
|---|---|---|
| SPMI | **90.75** | 93.18 |
| FixMatch | 90.30 | 93.05 |
| FlexMatch | 90.36 | 93.45 |
| FreeMatch | 90.42 | **93.65** |

For the PLL experiments, the partial rate is set to 0.3 for F-MNIST, CIFAR-10, and SVHN datasets, while it is set to 0.05 for CIFAR-100 dataset.

*Table 8.* Test accuracy of each PLL approach.

| Method | F-MNIST | CIFAR-10 | CIFAR-100 | SVHN |
|---|---|---|---|---|
| SPMI | 93.57 | **95.90** | 82.32 | **97.41** |
| PRODEN | 93.00 | 94.78 | 78.37 | 96.72 |
| PiCO | 93.47 | 94.37 | 78.24 | 96.50 |
| PLCR | 93.83 | 95.55 | 81.56 | 97.23 |
| POP | **93.91** | 95.66 | **82.48** | 97.39 |

From the experimental results presented above, it is evident that our framework also achieves competitive results in independent SSL or PLL tasks, thereby further demonstrating the generality of SPMI.

We conduct experiments with noisy label to further validate the compatibility of our framework. Considering that our framework is designed for the scenarios of inexact supervision, we choose to conduct experiments in a well-researched scenario containing noisy and redundant supervised information, namely unreliable partial label learning (UPLL) (Qiao et al., 2023), where the true label of each sample may not exist in the candidate label set. The experimental setup includes a partial rate $p = 0.3$ and a noise rate of $\eta = \{0.1, 0.2\}$ for F-MNIST and CIFAR-10 datasets.

*Table 9.* Test accuracy of each UPLL approach.

| Method | F-MNIST($\eta$=0.1) | F-MNIST($\eta$=0.2) | CIFAR-10($\eta$=0.1) | CIFAR-10($\eta$=0.2) |
|---|---|---|---|---|
| SPMI | **92.31** | **91.23** | **90.59** | **84.25** |
| FREDIS | 91.05 | 89.60 | 82.75 | 79.65 |