Imprecise Label Learning: A Unified Framework for Learning with Various Imprecise Label Configurations

Hao Chen[1,*], Ankit Shah[1], Jindong Wang[2], Ran Tao[1], Yidong Wang[3], Xiang Li[1], Xing Xie[2], Masashi Sugiyama[4,5], Rita Singh[1], Bhiksha Raj[1,6]

[1]Carnegie Mellon University, [2]Microsoft Research,[3]Peking University, [4] RIKEN AIP, [5]The University of Tokyo, [6]MBZUAI

Abstract

Learning with reduced labeling standards, such as noisy label, partial label, and supplementary unlabeled data, which we generically refer to as *imprecise* label, is a commonplace challenge in machine learning tasks. Previous methods tend to propose specific designs for every emerging imprecise label configuration, which is usually unsustainable when multiple configurations of imprecision co-exist. In this paper, we introduce imprecise label learning (ILL), a framework for the unification of learning with various imprecise label configurations. ILL leverages expectation-maximization (EM) for modeling the imprecise label information, treating the precise labels as latent variables. Instead of approximating the correct labels for training, it considers the entire distribution of all possible labeling entailed by the imprecise information. We demonstrate that ILL can seamlessly adapt to partial label learning, semi-supervised learning, noisy label learning, and, more importantly, a mixture of these settings, with closed-form learning objectives derived from the unified EM modeling. Notably, ILL surpasses the existing specified techniques for handling imprecise labels, marking the first practical and unified framework with robust and effective performance across various challenging settings. We hope our work will inspire further research on this topic, unleashing the full potential of ILL in wider scenarios where precise labels are expensive and complicated to obtain. Code is available at: `https://github.com/Hhhhhhao/General-Framework-Weak-Supervision`.

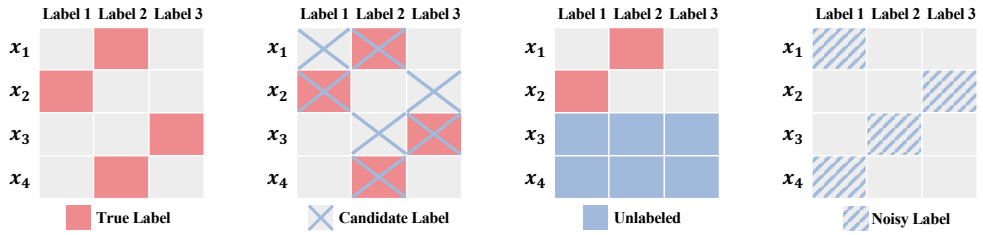
1 Introduction

One of the critical challenges in machine learning is the collection of annotated data for model training [1–6]. Ideally, every data instance would be fully annotated with precise labels. However, collecting such data can be expensive, time-consuming, and error-prone. Often, the labels can be intrinsically difficult to ascertain precisely. Factors such as a lack of annotator expertise and privacy concerns can also negatively affect the quality and completeness of the annotations.

In an attempt to circumvent this limitation, several methods have been proposed to permit model learning from the data annotated with reduced labeling standards, which are generally easier to obtain. We will refer to such labels as *imprecise*. Fig. 1 illustrates some typical mechanisms of label imprecision that are commonly addressed in the literature. Label imprecision requires a modification of the standard supervised training mechanism to build models for each specific case. For instance, *partial label learning* (PLL) [7–13] allows instances to have a set of candidate labels, instead of a single definitive one. *Semi-supervised Learning* (SSL) [14–23] seeks to enhance the generalization ability when only a small set of labeled data is available, supplemented by a larger unlabeled set.

*haoc3@andrew.cmu.edu

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) Full Label (b) Partial Label (c) Semi-Supervised (d) Noisy Label

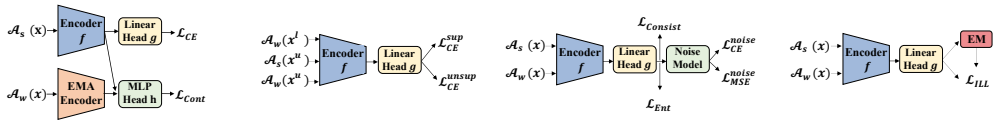
Figure 1: Illustration of the full label and imprecise label configurations. We use an example dataset of 4 training instances and 3 classes. (a) Full label, the annotation is a single true label; (b) Partial label, the annotation is a label candidate set containing true label; (c) Semi-supervised, only part of the dataset is labeled, and the others are unlabeled; (d) Noisy label, the annotation is mislabeled.

Noisy label learning (NLL) [24–37] deals with noisy scenarios where the labels are corrupted or incorrect. There is a greater variety of other forms of label imprecision, including crowd-sourcing [38, 39], programmable weak supervision [40, 41], and bag-level supervision [42–47], among others.

While prior arts have demonstrated success in handling individual configurations of label imprecision, their approaches often differ substantially. They are tailored to a *specific* form of imprecision, as depicted in Fig. 2. Such specificity not only imposes the necessity of devising a solution for emerging types of label imprecision scenarios, but also complicates the deployment in practical settings, where the annotations can be highly complex and may involve *multiple coexisting and interleaved* imprecision configurations. For instance, considering a scenario where both noisy labels and partial labels appear together, it might be challenging to adapt previous methods in NLL or PLL to this scenario since they either rely on the assumption of definite labels [39] or the existence of the correct label among label candidates [48], thus requiring additional algorithmic design. In fact, a few recent works have attempted to address the combinations of imprecise labels in this way, such as partial noisy label [49, 50] and semi-supervised partial label learning [51, 52]. However, simply utilizing a more sophisticated or ad-hoc design can hardly scale to other settings. In addition, most of these approaches attempt to infer the correct labels given the imprecise information (*e.g.* through consistency with adjacent data [14, 53, 54], iterative refinement [55, 56], average over given labels [57, 58], etc., to train the model, which inevitably accumulates error during training.

In this paper, we formulate the problem from a different perspective: rather than taking the imprecise label information provided as a potentially noisy or incomplete attempt at assigning labels to instances, we treat it generically as the information that imposes a deterministic or statistical restriction of the actual applicable true labels. We then train the model over the distribution of all possible labeling entailed by the given imprecise information. More specifically, for a dataset with samples X and imprecise label information I, we treat the inaccessible full and precise labels Y as a latent variable. The model is then trained to maximize the likelihood of the provided information I. Since the likelihood computed over the joint probability $P(X, I; \theta) = \sum_Y P(X, I, Y; \theta)$ must marginalize out Y, the actual information I provided could permit a potentially exponential number of labeling. To deal with the resulting challenge of maximizing the logarithm of an expectation, we use the common approach of *expectation-maximization* (EM) [59], where the E-step computes the expectation of $P(X, I, Y; \theta)$ given the posterior of current belief $P(Y|X, I; \theta^t)$ at time step t and the M-step maximizes the tight variational lower bound over $P(X, I; \theta)$. The overall framework is thus largely agnostic to the various nature of label imprecision, with the imprecise label only affecting the manner in which the posterior $P(Y|X, I; \theta^t)$ is computed. In fact, current approaches designed for various imprecise label scenarios can be treated as specific instances of our framework. Our approach can serve as a solution towards a *unified and generalized* view for learning with *various* imprecise labels.

While there exist earlier attempts on generalized or EM solutions for different (other) imprecise supervisions or fuzzy observations [60–64, 45, 65–67], they usually require additional assumptions and approximations on the imprecise information for learnablility [48, 68], thus presenting limited scalability on practical settings [62]. On the contrary, the unified framework we propose subsumes all of these and naturally extends to the more practical "mixed" style of data, where different types of imprecise labels coexist. Moreover, for noisy labels, our framework inherently enables the learning of a *noise model*, as we will show in Section 3.2. Through comprehensive experiments, we demonstrate that the proposed imprecise label learning (ILL) framework not only outperforms previous methods



(a) Partial Label (b) Semi-Supervised (c) Noisy Label (d) Imprecise Label

Figure 2: Baseline model pipelines for various imprecise label configurations. (a) PiCO [13] for partial label learning. (b) FixMatch [18] for semi-supervised learning. (c) SOP [69] for noisy label learning. (d) The proposed unified framework. It accommodates *any* imprecise label configurations and also mixed imprecise labels with an EM formulation.

for dealing with single imprecise labels of PLL, NLL, and SSL, but also presents robustness and effectiveness for mixed imprecise label learning (MILL) settings, leveraging the full potential to more challenging scenarios. Our contributions are summarized as follows:

- We propose an EM framework towards the unification of learning from *various* imprecise labels.
- We establish scalable and consistent state-of-the-art (SOTA) performance with the proposed method on partial label learning, semi-supervised learning, and noisy label learning, demonstrating our method's robustness in more diverse, complex label noise scenarios.
- To the best of our knowledge, our work is the first to show the robustness and effectiveness of a single unified method for handling the mixture of various imprecise labels.

2 Preliminary

In this section, we illustrate the notations and baselines from different imprecise label settings that adopt various solutions. We will show later how our proposed method generalize and subsumes these prior arts. Let $\mathcal{X}$ denote the input space, and $\mathcal{Y} = [C] := \{1, \ldots, C\}$ represent the label space with C distinct labels. A fully annotated training dataset of size N is represented as $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i \in [N]}$. Learning with imprecise labels involves approximating the mapping function $f \circ g : \mathcal{X} \to \mathcal{Y}$ from a training dataset where the true label y is not fully revealed from the annotation process. Here f is the backbone for feature extraction, g refers to the classifier built on top of the features, and the output from $f \circ g$ is the predicted probability $\mathbf{p}(y|\mathbf{x}; \theta)$, where θ is the learnable parameter for $f \circ g$. In this study, we primarily consider three imprecise label configurations (as illustrated in Fig. 1) and their corresponding representative learning paradigms (as shown in Fig. 2), namely partial label learning, semi-supervised learning, and noisy label learning.

Partial label learning (PLL). PLL aims to learn with a candidate label set $\mathbf{s} \subset \mathcal{Y}$, where the ground truth label $y \in \mathcal{Y}$ is concealed in $\mathbf{s}$. The training data for partial labels thus becomes $\mathcal{D}_{\mathrm{PLL}} = \{(\mathbf{x}_i, \mathbf{s}_i)\}_{i \in [N]}$. PiCO [13] is a recent contrastive method that employs class prototypes to enhance label disambiguation (as shown in Fig. 2(a)). It optimizes the cross-entropy (CE)[2] loss between the prediction of the augmented training sample $\mathcal{A}_{\mathrm{w}}(\mathbf{x})$ and the disambiguated labels $\hat{\mathbf{s}}$. PiCO learns a set of class prototypes from the features associated with the same pseudo-targets. A contrastive loss, based on MOCO [70], is employed to better learn the feature space, drawing the projected and normalized features $\mathbf{z}_{\mathrm{w}}$ and $\mathbf{z}_{\mathrm{s}}$ of the two augmented versions of data $\mathcal{A}_{\mathrm{w}}(\mathbf{x})$ and $\mathcal{A}_{\mathrm{s}}(\mathbf{x})$ [3] closer. The objective of PiCO is formulated as:

$$\mathcal{L}_{\mathrm{PiCO}} = \mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}(y|\mathcal{A}_{\mathrm{w}}(\mathbf{x}); \theta), \hat{\mathbf{s}}\right) + \mathcal{L}_{\mathrm{Cont}}\left(\mathbf{z}_{\mathrm{w}}, \mathbf{z}_{\mathrm{s}}, \mathcal{M}\right). \tag{1}$$

Semi-supervised learning (SSL). For SSL, we can define the labeled dataset as $\mathcal{D}^{\mathrm{L}}_{\mathrm{SSL}} = \{(\mathbf{x}^{\mathrm{l}}_i, y^{\mathrm{l}}_i)\}_{i \in [N^{\mathrm{L}}]}$, and the unlabeled dataset as $\mathcal{D}^{\mathrm{U}} = \{\mathbf{x}^{\mathrm{u}}_j\}_{j \in [N^{\mathrm{L}}+1, N^{\mathrm{L}}+N^{\mathrm{U}}]}$, with $N^{\mathrm{L}} \ll N^{\mathrm{U}}$. A general confidence-thresholding based self-training [53, 18] pipeline for SSL is shown in Fig. 2(b). Consider FixMatch [18] as an example; there are usually two loss components: the supervised CE loss on labeled data and the unsupervised CE loss on unlabeled data. For the unsupervised objective, the pseudo-labels $\hat{y}^{\mathrm{u}}$ from the network itself are used to train on the unlabeled data. A "strong-weak" augmentation [53] is commonly adopted. To ensure the quality of the pseudo-labels,

[2]For simplicity, we use $\mathcal{L}_{\mathbf{CE}}$ for labels of the formats of class indices, one-hot vectors, and class probabilities.

[3]We use $\mathcal{A}_{\mathrm{w}}$ to indicate the weaker data augmentation and $\mathcal{A}_{\mathrm{s}}$ to indicate the stronger data augmentation.

only the pseudo-labels whose confidence scores $\hat{p}^{\mathrm{u}}$ are greater than a threshold τ are selected to participate in training:

$$\mathcal{L}_{\mathrm{Fix}} = \mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}(y|\mathcal{A}_{\mathrm{w}}(\mathbf{x}^{\mathrm{l}});\theta), y^{\mathrm{l}}\right) + \mathbb{1}\left(\hat{p}^{\mathrm{u}} \geq \tau\right)\mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}(y|\mathcal{A}_{\mathrm{s}}(\mathbf{x}^{\mathrm{u}});\theta), \hat{y}^{\mathrm{u}}\right). \tag{2}$$

Noisy label learning (NLL). NLL aims at learning with a dataset of corrupted labels, $\mathcal{D}_{\mathrm{NLL}} = \{(\mathbf{x}_i, \hat{y}_i)\}_{i\in[N]}$. We illustrate the NLL pipeline (in Fig. 2(c)) with the recent sparse over-parameterization (SOP) model [69], where a sparse *noise model* consisting of parameters $\mathbf{u}_i, \mathbf{v}_i \in [-1, 1]^C$ for each sample is adopted. The noise model transforms the network prediction from the true label distribution into the noisy label distribution. A CE loss and a mean-squared-error (MSE) loss optimize parameter $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ respectively:

$$\mathcal{L}_{\mathrm{SOP}} = \mathcal{L}_{\mathrm{CE}}\left(\phi\left(\mathbf{p}(y|\mathcal{A}_{\mathrm{w}}(\mathbf{x});\theta) + \mathbf{m}\right), \hat{y}\right) + \mathcal{L}_{\mathrm{MSE}}\left(\mathbf{p}(y|\mathcal{A}_{\mathrm{w}}(\mathbf{x});\theta) + \mathbf{m}, \hat{y}\right), \tag{3}$$

where ϕ denotes the L_∞ normalization and $\mathbf{m}_i = \mathbf{u}_i \odot \mathbf{u}_i \odot \hat{\mathbf{y}}_i^{\mathrm{oh}} - \mathbf{v}_i \odot \mathbf{v}_i \odot \left(1 - \hat{\mathbf{y}}_i^{\mathrm{oh}}\right)$, with $\hat{\mathbf{y}}_i^{\mathrm{oh}}$ referring to the one-hot version of y_i. Consistency regularization with strong-weak augmentation and entropy class-balance regularization are additionally utilized for better performance in SOP [69].

3 Imprecise Label Learning

Although current techniques demonstrate potential in addressing particular forms of imprecise labels, they frequently fall short in adaptability and transferability to more complicated and more realistic scenarios where multiple imprecise label types coexist and interleave. This section first defines the proposed expectation-maximization (EM) formulation for learning with various imprecise labels. Then, we demonstrate that our unified framework seamlessly extends to partial label learning, semi-supervised label learning, noisy label learning, and the more challenging setting of mixed imprecise label learning. Connections and generalization to previous pipelines can also be drawn clearly under the proposed EM framework.

3.1 A Unified Framework for Learning with Imprecise Labels

Exploiting information from imprecise labels. The challenge of learning with imprecise labels lies in learning effectively with inaccurate or incomplete annotation information. Per the analysis above, prior works catering to specific individual imprecise labels either explicitly or implicitly attempt to infer the precise labels from the imprecise label information. For example, partial label learning concentrates on the disambiguation of the ground truth label from the label candidates [13, 71, 50] or averaging equally over the label candidates [72]. In semi-supervised learning, after the model initially learns from the labeled data, the pseudo-labels are treated as correct labels and utilized to conduct self-training on the unlabeled data [73, 18]. Similarly, for noisy label learning, an integral part that helps mitigate overfitting to random noise is the implementation of an accurate noise model capable of identifying and rectifying the incorrect labels [33, 69], thereby ensuring the reliability of the learning process. However, inferring the correct labels from the imprecise labels or utilizing the imprecise labels directly can be very challenging and usually leads to errors accumulated during training [73, 74], which is also known as the confirmation bias. In this work, we take a different approach: we consider all possible labeling along with their likelihood that the imprecise labels fulfill to train the model, rather than using a single rectified label from the imprecise information. Such an approach also eliminates the requirements for designing different methods for various imprecise labels and provides a unified formulation instead, where closed-form solutions can be derived.

A unified framework for learning with imprecise labels (ILL). Let $\{\mathbf{x}_i\}_{i\in[N]}$ represent the features as realizations from X and $\{y_i\}_{i\in[N]}$ represent their precise labels as realizations from Y for the training data. Ideally, Y would be fully specified for X. In the imprecise label scenario, however, Y is not provided; instead we obtain imprecise label information I. We view I not as *labels*, but more abstractly as a variable representing the *information* about the labels. From this perspective, the actual labels Y would have a distribution $P(Y|I)$, and I can present in various forms. When the information I provided is the precise true label of the data, $P(Y|I)$ would be a delta distribution, taking a value 1 at the true label, and 0 elsewhere. If I represents partial labels, then $P(Y|I)$ would have non-zero value over the candidate labels, and be 0 elsewhere. When I represents a set of noisy labels, $P(Y|I)$ would represent the distribution of the true labels, given the noisy labels. When I does not contain any information, i.e., unlabeled data, Y can take any value.

By the maximum likelihood estimation (MLE) principle, we must estimate the model to maximize the likelihood of the data/information we have been provided, namely X and I. Let $P(X, I; \theta)$ represent a parametric form for the joint distribution of X and I[4] Explicitly considering the labels Y, we have $P(X, I; \theta) = \sum_Y P(X, Y, I; \theta)$. The maximum likelihood principle requires us to find:

$$\theta^* = \arg\max_\theta \ \log P(X, I; \theta) = \arg\max_\theta \ \log \sum_Y P(X, Y, I; \theta), \tag{4}$$

with θ^* denotes the optimal value of θ. Eq. (4) features the log of an expectation and cannot generally be solved in closed-form, and requires iterative hill-climbing solutions. Of these, arguably the most popular is the expectation-maximization (EM) algorithm [59], which iteratively maximizes a tight variational lower bound on the log-likelihood. In our case, applying it becomes:

$$\begin{aligned} \theta^{t+1} &= \arg\max_\theta \mathbb{E}_{Y|X,I;\theta^t} \left[\log P(X, Y, I; \theta)\right] \\ &= \arg\max_\theta \mathbb{E}_{Y|X,I;\theta^t} \left[\log P(Y|X; \theta) + \log P(I|X, Y; \theta)\right], \end{aligned} \tag{5}$$

where θ^t is the tth estimate of the optimal θ. Note that $P(X; \theta)$ is omitted from Eq. (5) since $P(X)$ does not rely on θ. The detailed derivation of the variational lower bound is shown in Appendix C.1. There are several implications from Eq. (5). (i) The expectation over the posterior $P(Y|X, I; \theta^t)$ equates to considering *all* labeling entailed by the imprecise label information I, rather than any single (possibly corrected) choice of label. For independent instances setting mostly studied in this paper, we can derive closed-form training objectives from this formulation as shown in Section 3.2. (ii) The property of the second term $\log P(I|X, Y; \theta)$ is dependent on the nature of imprecise label I. If I is derivable from true labels Y, such as the actual labels or the label candidates, it can be reduced to $P(I|Y)$, *i.e.*, the probability of I is no longer dependent on X or θ and thus can be ignored from Eq. (5). If I represents the noisy labels, $P(I|X, Y; \theta)$ instead includes a potentially learnable noise model. (iii) It is a general framework towards the unification of any label configuration, including full labels, partial labels, low-resource labels, noisy labels, etc. In this work, we specialize the proposed EM framework to PLL, SSL, NLL, and the mixture of them in the following.

3.2 Instantiating the Unified EM Formulation

We illustrate how to seamlessly expand the formulation from Eq. (5) to partial label learning, semi-supervised learning, noisy label learning, and mixture settings, with derived closed-form loss function[5] for each setting here. The actual imprecise labels only affect the manner in which the posterior $P(Y|X, I; \theta^t)$ is computed for each setting. We show that all learning objectives derived from Eq. (5) naturally include a consistency term with the posterior as the soft target. We also demonstrate that the proposed unified EM framework closely connects with the prior arts, which reveals the potential reason behind the success of these techniques. Note that while we only demonstrate the application of the proposed framework to four settings here, it can also be flexibly extended to other settings. More details of derivation below are shown in Appendix C.

Partial label learning (PLL). The imprecise label I for partial labels is defined as the label candidate sets S containing the true labels. These partial labels indicate that the posterior $P(Y|X, S; \theta^t)$ can only assign its masses on the candidate labels. Since S can be derived from true labels Y, $P(S|X, Y; \theta)$ reduces to $P(S|Y)$, and thus can be ignored. We also demonstrate with instance dependent partial labels that maintains $P(S|X, Y; \theta)$ in Appendix D.2.2. Defining the label candidates as $\{\mathbf{s}_i\}_{i\in[N]}$ and substituting it in Eq. (5), we have the loss function of PLL derived using ILL framework:

$$\mathcal{L}_{\text{ILL}}^{\text{PLL}} = -\sum_{Y\in[C]} P(Y|X, S; \theta^t) \log P(Y|X; \theta) \equiv \mathcal{L}_{\text{CE}}\left(\mathbf{p}(y|\mathcal{A}_{\text{s}}(\mathbf{x}); \theta), \mathbf{p}(y|\mathcal{A}_{\text{w}}(\mathbf{x}), \mathbf{s}; \theta^t)\right), \tag{6}$$

where $\mathbf{p}(y|\mathcal{A}_{\text{w}}(\mathbf{x}), \mathbf{s}; \theta^t)$ is the normalized probability that $\sum_{k\in C} p_k = 1$, and $p_k = 0, \forall k \in \mathbf{s}$. Eq. (6) corresponds exactly to consistency regularization [53], with the normalized predicted probability as the soft pseudo-targets. We use $\mathcal{A}_s$ and $\mathcal{A}_w$ to denote the strong and weak augmentation

[4]The actual parameters θ may apply only to some component such as $P(Y|X; \theta)$ of the overall distribution; we will nonetheless tag the entire distribution $P(X, I; \theta)$ with θ to indicate that it is dependent on θ overall.

[5]To formulate the loss function, we convert the problem to minimization of the negative log-likelihood.

as stated earlier. This realization on PLL shares similar insights as [12] which exploits a gradually induced loss weight for PLL on multiple augmentations of the data. However, our framework is much simpler and more concise as shown in Appendix D.2.2, which does not require additional techniques.

Semi-supervised learning (SSL) In SSL, the input X consists of the labeled data X^{L} and the unlabeled data X^{U}. The imprecise label for SSL is realized as the limited number of full labels Y^{L} for X^{L}. The labels Y^{U} for unlabeled X^{U} are unknown and become the latent variable. Interestingly, for the unlabeled data, there is no constraint on possible labels it can take. The posterior $P(Y^{\mathrm{U}}|X^{\mathrm{L}}, X^{\mathrm{U}}, Y^{\mathrm{L}}; \theta)$, which is the actual prediction from the network, can be directly utilized as soft targets for self-training. Since Y^{L} is conditionally independent with Y^{U} given X, the second term of Eq. (5): $P(Y^{\mathrm{L}}|X^{\mathrm{L}}, X^{\mathrm{U}}, Y^{\mathrm{U}}; \theta)$, is reduced to $P(Y^{\mathrm{L}}|X^{\mathrm{L}}; \theta)$, which corresponds to the supervised objective on labeled data. The loss function for SSL thus becomes:

$$\begin{aligned}\mathcal{L}_{\mathrm{ILL}}^{\mathrm{SSL}} &= -\sum_{Y\in[C]} P(Y^{\mathrm{U}}|X^{\mathrm{U}}, X^{\mathrm{L}}, Y^{\mathrm{L}}; \theta^t) \log P(Y^{\mathrm{U}}|X^{\mathrm{U}}, X^{\mathrm{L}}; \theta) - \log P(Y^{\mathrm{L}}|X^{\mathrm{L}}; \theta) \\ &\equiv \mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}(y|\mathcal{A}_{\mathrm{s}}(\mathbf{x}^{\mathrm{u}}); \theta), \mathbf{p}(y|\mathcal{A}_{\mathrm{w}}(\mathbf{x}^{\mathrm{u}}); \theta^t)\right) + \mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}(y|\mathcal{A}_{\mathrm{w}}(\mathbf{x}^{\mathrm{l}}); \theta), y^{\mathrm{l}}\right)\end{aligned} \tag{7}$$

The first term corresponds to the unsupervised consistency regularization usually employed in SSL, and the second term refers to the supervised CE loss only on labeled data. Eq. (7) has several advantages over the previous methods. It adopts the prediction as soft-targets of all possible labeling on unlabeled data, potentially circumventing the confirmation bias caused by pseudo-labeling and naturally utilizing all unlabeled data which resolves the quantity-quality trade-off commonly existing in SSL [18, 23]. It also indicates that previous pseudo-labeling with confidence threshold implicitly conducts the EM optimization, where the maximal probable prediction approximates the expectation, and the degree of the approximation is determined by the threshold τ, rationalizing the effectiveness of dynamic thresholding.

Noisy label learning (NLL). Things become more complicated here since the noisy labels $\hat{Y}$ do not directly reveal the true information about Y, thus $P(\hat{Y}|Y, X; \theta)$ inherently involves a noise model that needs to be learned. We define a simplified instance-independent[6] noise transition model $\mathcal{T}(\hat{Y}|Y; \omega)$ with parameters ω, and take a slightly different way to formulate the loss function for NLL from the ILL framework:

$$\begin{aligned}\mathcal{L}_{\mathrm{ILL}}^{\mathrm{NLL}} &= -\sum_{Y\in[C]} P(Y|X, \hat{Y}; \theta^t, \omega^t) \log P(Y|X, \hat{Y}; \theta, \omega^t) - \log P(\hat{Y}|X; \theta, \omega) \\ &\equiv \mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}(y|\mathcal{A}_{\mathrm{s}}(\mathbf{x}), \hat{y}; \theta, \omega^t), \mathbf{p}(y|\mathcal{A}_{\mathrm{w}}(\mathbf{x}), \hat{y}; \theta^t, \omega^t)\right) + \mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}(\hat{y}|\mathcal{A}_{\mathrm{w}}(\mathbf{x}); \theta, \omega), \hat{y}\right),\end{aligned} \tag{8}$$

where the parameters ω and θ are learned end-to-end. The first term corresponds to the consistency regularization of prediction conditioned on noisy labels and the second term corresponds to the supervised loss on noisy predictions that are converted from the ground truth predictions. Both quantities are computed using the noise transition model given the noisy label $\hat{y}$:

$$\mathbf{p}(y|\mathbf{x}, \hat{y}; \theta, \omega^t) \propto \mathbf{p}(y|\mathbf{x}; \theta)\mathcal{T}(\hat{y}|y; \omega^t), \text{and } \mathbf{p}(\hat{y}|\mathbf{x}; \theta, \omega) = \sum_{y\in[C]} \mathbf{p}(y|\mathbf{x}; \theta)\mathcal{T}(\hat{y}|y; \omega). \tag{9}$$

Mixture imprecise label learning (MILL). We additionally consider a more practical setting, mixture of imprecise label learning, with partial labels, noisy labels, and unlabeled data interleaved together. On the unlabeled data, the unsupervised objective is the same as the unsupervised consistency regularization of SSL as shown in Eq. (7). The labeled data here present partial and noisy labels $\hat{\mathbf{s}}$. Thus the noisy supervised objective in Eq. (9) becomes the supervised consistency regularization as in Eq. (6) of partial label setting to train the noise transition model, and the noisy unsupervised objective becomes the consistency regularization of the prediction conditioned on noisy partial labels. Thus we have the loss function for MILL derived as:

$$\begin{aligned}\mathcal{L}_{\mathrm{ILL}}^{\mathrm{MILL}} &= \mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}\left(y \mid \mathcal{A}_{\mathrm{s}}(\mathbf{x}^l), \hat{\mathbf{s}}^l; \theta, \omega^t\right), \mathbf{p}\left(y \mid \mathcal{A}_{\mathrm{w}}(\mathbf{x}^l), \hat{\mathbf{s}}^l; \theta^t, \omega^t\right)\right) \\ &\quad + \mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}\left(\hat{y} \mid \mathcal{A}_{\mathrm{w}}(\mathbf{x}^l); \theta, \omega\right), \hat{\mathbf{s}}^l\right) \\ &\quad + \mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}(y|\mathcal{A}_{\mathrm{s}}(\mathbf{x}^{\mathrm{u}}); \theta), \mathbf{p}(y|\mathcal{A}_{\mathrm{w}}(\mathbf{x}^{\mathrm{u}}); \theta^t)\right)\end{aligned} \tag{10}$$

[6] A more complicated instance-dependent noise model $\mathcal{T}(\hat{Y}|Y, X; \omega)$ can also be formulated under our unified framework, but not considered in this work. Also, since we use $\mathcal{T}$ both in forward fashion and backward fashion, it is unidentifiable in this work.

Table 1: Accuracy of different partial ratio q on CIFAR-10, CIFAR-100, and CUB-200 for **partial label learning**. The best and the second best results are indicated in **bold** and underline respectively.

Dataset	CIFAR-10			CIFAR-100			CUB-200
Partial Ratio q	0.1	0.3	0.5	0.01	0.05	0.1	0.05
Fully-Supervised		94.91±0.07			73.56±0.10		-
LWS [11]	90.30±0.60	88.99±1.43	86.16±0.85	65.78±0.02	59.56±0.33	53.53±0.08	39.74±0.47
PRODEN [55]	90.24±0.32	89.38±0.31	87.78±0.07	62.60±0.02	60.73±0.03	56.80±0.29	62.56±0.10
CC [9]	82.30±0.21	79.08±0.07	74.05±0.35	49.76±0.45	47.62±0.08	35.72±0.47	55.61±0.02
MSE [79]	79.97±0.45	75.65±0.28	67.09±0.66	49.17±0.05	46.02±1.82	43.81±0.49	22.07±2.36
EXP [79]	79.23±0.10	75.79±0.21	70.34±1.32	44.45±1.50	41.05±1.40	29.27±2.81	9.44±2.32
PiCO [13]	94.39±0.18	94.18±0.12	93.58±0.06	73.09±0.34	72.74±0.30	69.91±0.24	**72.17±0.72**
Ours	**96.37±0.08**	**96.26±0.03**	**95.91±0.05**	**75.31±0.19**	**74.58±0.03**	**74.00±0.02**	70.77±0.29

We can compute both quantity through the noise transition model:

$$\mathbf{p}(y|\mathbf{x}, \hat{\mathbf{s}}; \theta, \omega^t) \propto \mathbf{p}(y|\mathbf{x}; \theta) \prod_{\hat{y} \in \hat{\mathbf{s}}} \mathcal{T}(y|\hat{y}; \omega^t), \text{and } \mathbf{p}(\hat{y}|\mathbf{x}; \theta, \omega) = \sum_{y \in [C]} \mathbf{p}(y|\mathbf{x}; \theta) \mathcal{T}(\hat{y}|y; \omega). \quad (11)$$

4 Experiments

In this section, we conduct extensive experiments to evaluate ILL. Albeit simple, the ILL framework achieves comparable state-of-the-art performance regarding previous methods on partial label learning, semi-supervised learning, and noisy label learning. Moreover, our experiments show that ILL could be easily extended to a more practical setting with a mixture of various imprecise label configurations. For all settings, we additionally adopt an entropy loss for balancing learned cluster sizes [75, 76], similarly as [69, 22]. Experiments are conducted with three runs using NVIDIA V100 GPUs.

4.1 Partial Label Learning

Setup. Following [13], we evaluate our method on partial label learning setting using CIFAR-10 [77], CIFAR-100 [77], and CUB-200 [78]. We generate partially labeled datasets by flipping negative labels to false positive labels with a probability q, denoted as a partial ratio. The $C-1$ negative labels are then uniformly aggregated into the ground truth label to form a set of label candidates. We consider $q \in \{0.1, 0.3, 0.5\}$ for CIFAR-10, $q \in \{0.01, 0.05, 0.1\}$ for CIFAR-100, and $q = 0.05$ for CUB-200. We choose six baselines for PLL using ResNet-18 [1]: LWS [11], PRODEN [55], CC [9], MSE and EXP [79], and PiCO [13]. The detailed hyper-parameters, comparison with the more recent method R-CR [12] that utilizes a different training recipe and model [80], and comparison with instance-dependent partial labels [81] are shown in Appendix D.2.2.

Results. The results for PLL are shown in Table 1. Our method achieves the best performance compared to the baseline methods. Perhaps more surprisingly, on CIFAR-10 and CIFAR-100, our method even outperforms the fully-supervised reference, indicating the potential better generalization capability using the proposed framework, sharing similar insights as in Wu et al. [12]. While PiCO adopts a contrastive learning objective, our method still surpasses PiCO by an average of $\mathbf{2.13}\%$ on CIFAR-10 and $\mathbf{2.72}\%$ on CIFAR-100. Our approach can be further enhanced by incorporating contrastive learning objectives, potentially leading to more significant performance.

4.2 Semi-Supervised Learning

Setup. For experiments of SSL, we follow the training and evaluation protocols of USB [82] on image and text classification. To construct the labeled dataset for semi-supervised learning, we uniformly select l/C samples from each class and treat the remaining samples as the unlabeled dataset. We present the results on CIFAR-100 and STL-10 [77] for image classification, and IMDB [83] and Amazon Review [84] for text classification. We compare with the current methods with confidence thresholding, such as FixMatch [18], AdaMatch [85], FlexMatch [20], FreeMatch [22], and SoftMatch [23]. We also compare with methods with the contrastive loss, CoMatch [86] and SimMatch [87]. A full comparison of the USB datasets and hyper-parameters is shown in Appendix D.3.

Results. We present the results for SSL on Table 2. Although no individual SSL algorithm dominates the USB benchmark [82], our method still shows competitive performance. Notably, our method

Table 2: Error rate of different number of labels l on CIFAR-100, STL-10, IMDB, and Amazon Review datasets for **semi-supervised learning**.

Datasets	CIFAR-100		STL-10		IMDB		Amazon Review	
# Labels l	200	400	40	100	20	100	250	1000
AdaMatch [85]	22.32±1.73	16.66±0.62	13.64±2.49	7.62±1.90	8.09±0.99	7.11±0.20	45.40±0.96	**40.16±0.49**
FixMatch [18]	29.60±0.90	19.56±0.52	16.15±1.89	8.11±0.68	7.72±0.33	7.33±0.13	47.61±0.83	43.05±0.54
FlexMatch [20]	26.76±1.12	18.24±0.36	14.40±3.11	8.17±0.78	7.82±0.77	7.41±0.38	45.73±1.60	42.25±0.33
CoMatch [86]	35.08±0.69	25.35±0.50	15.12±1.88	9.56±1.35	7.44±0.30	7.72±1.14	48.76±0.90	43.36±0.21
SimMatch [87]	23.78±1.08	17.06±0.78	11.77±3.20	**7.55±1.86**	7.93±0.55	**7.08±0.33**	45.91±0.95	42.21±0.30
FreeMatch [22]	**21.40±0.30**	**15.65±0.26**	12.73±3.22	8.52±0.53	8.94±0.21	7.95±0.45	46.41±0.60	42.64±0.06
SoftMatch [23]	22.67±1.32	16.84±0.66	13.55±3.16	7.84±1.72	7.76±0.58	7.97±0.72	45.29±0.95	42.21±0.20
Ours	22.06±1.06	16.40±0.54	**11.09±0.71**	8.10±1.02	**7.32±0.12**	7.64±0.67	**43.96±0.32**	42.32±0.02

Table 3: Accuracy of synthetic noise on CIFAR-10 and CIFAR-100 and instance noise on Clothing1M and WebVision for **noisy label learning**. We use noise ratio of $\{0.2, 0.5, 0.8\}$ for synthetic symmetric noise and 0.4 for asymmetric label noise. The instance noise ratio is unknown.

Dataset	CIFAR-10				CIFAR-100				Clothing1M	WebVision
Noise Type	Sym.			Asym.	Sym.			Asym.	Ins.	Ins.
Noise Ratio η	0.2	0.5	0.8	0.4	0.2	0.5	0.8	0.4	-	-
CE	87.20	80.70	65.80	82.20	58.10	47.10	23.80	43.30	69.10	-
Mixup [90]	93.50	87.90	72.30	-	69.90	57.30	33.60	-	-	-
DivideMix [33]	96.10	94.60	93.20	93.40	77.10	74.60	60.20	72.10	**74.26**	77.32
ELR [32]	95.80	94.80	93.30	93.00	77.70	73.80	60.80	77.50	72.90	76.20
SOP [69]	96.30	95.50	94.00	93.80	**78.80**	**75.90**	63.30	**78.00**	73.50	76.60
Ours	**96.78±0.11**	**96.60±0.15**	**94.31±0.07**	**94.75±0.81**	77.49±0.28	75.51±0.52	**66.46±0.72**	75.82±1.89	74.02±0.12	**79.37±0.09**

performs best on STL-10 with 40 labels and Amazon Review with 250 labels, outperforming the previous best by $\mathbf{0.68}\%$ and $\mathbf{1.33}\%$. In the other settings, the performance of our method is also very close to the best-performing methods. More remarkably, our method does not employ any thresholding, re-weighting, or contrastive techniques to achieve current results, demonstrating a significant potential to be further explored.

4.3 Noisy Label Learning

Setup. We conduct the experiments of NLL following SOP [69] on both synthetic symmetric/asymmetric noise on CIFAR-10 and CIFAR-100, and more realistic and larger-scale instance noise on Clothing1M [88], and WebVision [89]. To introduce the synthetic symmetric noise to CIFAR-10 and CIFAR-100, we uniformly flip labels for a probability η into other classes. For asymmetric noise, we only randomly flip the labels for particular pairs of classes. The introduced noise is then treated as ground truth labels to train the model. We mainly select three previous best methods as baselines: DivideMix [33]; ELR [32]; and SOP [69]. We also include the normal cross-entropy (CE) training and mixup [90] as baselines. More comparisons of other methods [91, 28] and on CIFAR-10N [92] with training details and more baselines [93, 28] are shown in Appendix D.4.

Results. We present the noisy label learning results in Table 3. The proposed method is comparable to the previous best methods. On synthetic noise of CIFAR-10, our method demonstrates the best performance on both symmetric noise and asymmetric noise. On CIFAR-100, our method generally produces similar results comparable to SOP. One may notice that our method shows inferior performance on asymmetric noise of CIFAR-100; we argue this is mainly due to the oversimplification of the noise transition model. Our method also achieves the best results on WebVision, outperforming the previous best by $\mathbf{2.05}\%$. On Clothing1M, our results are also very close to DivideMix, which trains for 80 epochs compared to 10 epochs in ours.

4.4 Mixed Imprecise Label Learning

Setup. We evaluate on CIFAR-10 and CIFAR-100 in a more challenging and realistic setting, the mixture of various imprecise label configurations, with unlabeled, partially labeled, and noisy labeled data existing simultaneously. We first sample the labeled dataset and treat other samples as the unlabeled. On the labeled dataset, we generate partial labels and randomly corrupt the true label of the partial labels. We set $l \in \{1000, 5000, 50000\}$ for CIFAR-10, and $l \in \{5000, 10000, 50000\}$ for

Table 4: Accuracy comparison of **mixture of different imprecise labels**. We report results of full labels, partial ratio q of 0.1 (0.01) and 0.3 (0.05) for CIFAR-10 (CIFAR-100), and noise ratio η of 0.1, 0.2, and 0.3 for CIFAR-10 and CIFAR-100.

Method	q	CIFAR-10, l=50000			q	CIFAR-100, l=50000		
		η=0.1	η=0.2	η=0.3		η=0.1	η=0.2	η=0.3
PiCO+ [94]	0.1	93.64	93.13	92.18	0.01	71.42	70.22	66.14
IRNet [71]		93.44	92.57	92.38		71.17	70.10	68.77
DALI [50]		94.15	94.04	93.77		72.26	71.98	71.04
PiCO+ Mixup [50]		94.58	94.74	94.43		75.04	74.31	71.79
DALI Mixup [50]		95.83	95.86	95.75		76.52	76.55	76.09
Ours		**96.47**$\pm$0.11	**96.09**$\pm$0.20	**95.83**$\pm$0.05		**77.53**$\pm$0.24	**76.96**$\pm$0.02	**76.43**$\pm$0.27
PiCO+ [94]	0.3	92.32	92.22	89.95	0.05	69.40	66.67	62.24
IRNet [71]		92.81	92.18	91.35		70.73	69.33	68.09
DALI [50]		93.44	93.25	92.42		72.28	71.35	70.05
PiCO+ Mixup [50]		94.02	94.03	92.94		73.06	71.37	67.56
DALI Mixup [50]		95.52	95.41	94.67		76.87	75.23	74.49
Ours		**96.2**$\pm$0.02	**95.87**$\pm$0.14	**95.22**$\pm$0.06		**77.07**$\pm$0.16	**76.34**$\pm$0.08	**75.13**$\pm$0.63

Table 5: Robust test accuracy results of our method on **more mixture of imprecise label configurations**. l, q and η are the number of labels, partial, and noise ratio.

l	q	CIFAR10				l	q	CIFAR100			
		η=0.0	η=0.1	η=0.2	η=0.3			η=0.0	η=0.1	η=0.2	η=0.3
5,000	0.1	95.29$\pm$0.18	93.90$\pm$0.11	92.02$\pm$0.22	89.02$\pm$0.63	10,000	0.01	69.90$\pm$0.23	68.74$\pm$0.15	66.87$\pm$0.34	65.34$\pm$0.02
	0.3	95.13$\pm$0.16	92.95$\pm$0.37	90.14$\pm$0.61	87.31$\pm$0.27		0.05	69.85$\pm$0.20	68.08$\pm$0.28	66.78$\pm$0.43	64.83$\pm$0.17
	0.5	95.04$\pm$0.10	92.18$\pm$0.52	88.39$\pm$0.62	83.09$\pm$0.56		0.10	68.92$\pm$0.45	67.15$\pm$0.63	64.44$\pm$1.29	60.26$\pm$1.96
1,000	0.1	94.48$\pm$0.09	91.68$\pm$0.17	87.17$\pm$0.51	81.04$\pm$1.13	5,000	0.01	65.66$\pm$0.27	63.13$\pm$0.27	60.93$\pm$0.17	58.36$\pm$0.56
	0.3	94.35$\pm$0.05	89.94$\pm$1.90	82.06$\pm$1.52	69.20$\pm$2.16		0.05	65.06$\pm$0.04	62.28$\pm$0.47	58.92$\pm$0.34	53.24$\pm$1.69
	0.5	93.92$\pm$0.29	86.34$\pm$2.37	70.86$\pm$2.78	38.19$\pm$6.55		0.10	63.32$\pm$0.55	58.73$\pm$1.33	53.27$\pm$1.57	46.19$\pm$1.04

CIFAR-100. For partial labels, we set $q \in \{0.1, 0.3, 0.5\}$ for CIFAR-10, and $q \in \{0.01, 0.05, 0.1\}$ for CIFAR-100. For noisy labels, we set $\eta \in \{0, 0.1, 0.2, 0.3\}$ for both datasets. Since there is no prior work that can handle all settings all at once, we compare on partial noisy label learning with PiCO+ [94], IRNet [71], and DALI [50]. Although there are also prior efforts on partial semi-supervised learning [51, 52], they do not scale on simple dataset even on CIFAR-10. Thus, we did not include them in comparison. We conduct additional validation of our method on more complex settings for partial noisy labels with unlabeled data to demonstrate its robustness to various imprecise labels.

Results. We report the comparison with partial noisy label learning methods in Table 4. Compared to previous methods, the proposed method achieves the best performance. Despite the simplicity, our method outperforms PiCO+ and DALI with mixup, showing the effectiveness of dealing with mixed imprecise labels. We also report the results of our methods on more mixed imprecise label configurations in Table 5. Our method demonstrates significant robustness against various settings of the size of labeled data, partial ratio, and noise ratio. Note that this is the first work that naturally deals with all three imprecise label configurations simultaneously, with superior performance than previous methods handling specific types or combinations of label configurations. This indicates the enormous potential of our work in realistic applications for handling more practical and complicated data annotations common in real world applications.

5 Conclusion

We present the imprecise label learning (ILL) framework, a unified and consolidated solution for learning from all types of imprecise labels. ILL effectively employs an expectation-maximization (EM) algorithm for maximum likelihood estimation (MLE) of the distribution over the latent ground truth labels Y, imprecise label information I, and data X. It naturally extends and encompasses previous formulations for various imprecise label settings, achieving promising results. Notably, in scenarios where mixed configurations of imprecise labels coexist, our method exhibits substantial robustness against diverse forms of label imprecision. The potential **broader impact** of the ILL framework is substantial. It stands poised to transform domains where obtaining precise labels poses a challenge, offering a simple, unified, and effective approach to such contexts. Beyond the three imprecise label configurations we have demonstrated in this study, the ILL framework shows promise for an extension to more intricate scenarios such as multi-instance learning [42] and multi-label crowd-sourcing learning [38]. However, it is also crucial to acknowledge the **limitations** of the ILL

framework. Although its effectiveness has been substantiated on relatively smaller-scale datasets, additional empirical validation is necessary to assess its scalability to larger datasets. Furthermore, our study only considers balanced datasets; thus, the performance of the ILL framework when dealing with imbalanced data and open-set data still remains an open area for future exploration. We hope that our study will constitute a significant stride towards a comprehensive solution for imprecise label learning and catalyze further research in this crucial field.

Acknowledge

Masashi Sugiyama was supported by the Institute for AI and Beyond, UTokyo.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021.

[6] OpenAI. Gpt-4 technical report. 2023.

[7] Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *The Journal of Machine Learning Research*, 12:1501–1536, 2011.

[8] Jie Luo and Francesco Orabona. Learning from candidate labeling sets. *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

[9] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. *ArXiv*, abs/2007.08929, 2020.

[10] Dengbao Wang, Min-Ling Zhang, and Li Li. Adaptive graph guided disambiguation for partial label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:8796–8811, 2019.

[11] Hongwei Wen, Jingyi Cui, Hanyuan Hang, Jiabin Liu, Yisen Wang, and Zhouchen Lin. Leveraged weighted loss for partial label learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 11091–11100. PMLR, 2021.

[12] Dong-Dong Wu, Deng-Bao Wang, and Min-Ling Zhang. Revisiting consistency regularization for deep partial label learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 24212–24225. PMLR, 17–23 Jul 2022.

[13] Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. PiCO: Contrastive label disambiguation for partial label learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[14] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, page 896, 2013.

[15] Laine Samuli and Aila Timo. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, page 6, 2017.

[16] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[17] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations (ICLR)*, 2019.

[18] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

[19] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10687–10698, 2020.

[20] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[21] Xudong Wang, Zhirong Wu, Long Lian, and Stella X Yu. Debiased learning from naturally imbalanced pseudo-labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 14647–14657, 2022.

[22] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, , Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Bernt Schiele, and Xing Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2023.

[23] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2023.

[24] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2691–2699, 2015.

[25] Alan Joseph Bekker and Jacob Goldberger. Training deep neural-networks based on unreliable labels. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2682–2686, 2016.

[26] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations (ICLR)*, 2016.

[27] Aritra Ghosh, Himanshu Kumar, and P. Shanti Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[28] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Wai-Hung Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[29] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[30] Junnan Li, Yongkang Wong, Qi Zhao, and M. Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5046–5054, 2018.

[31] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 322–330, 2019.

[32] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

[33] Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[34] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Monazam Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[35] Qi Wei, Haoliang Sun, Xiankai Lu, and Yilong Yin. Self-filtering: A noise-aware sample selection for label noise with confidence penalization. In *European Conference on Computer Vision*, pages 516–532. Springer, 2022.

[36] Yivan Zhang, Gang Niu, and Masashi Sugiyama. Learning noise transition matrix from only noisy labels via total variation regularization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[37] Qi Wei, Lei Feng, Haoliang Sun, Ren Wang, Chenhui Guo, and Yilong Yin. Fine-grained classification with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11651–11660, 2023.

[38] Shahana Ibrahim, Tri Nguyen, and Xiao Fu. Deep learning from crowdsourced labels: Coupled cross-entropy minimization, identifiability, and regularization. In *International Conference on Learning Representations (ICLR)*, 2023.

[39] Jiaheng Wei, Zhaowei Zhu, Tianyi Luo, Ehsan Amid, Abhishek Kumar, and Yang Liu. To aggregate or not? learning with separate noisy labels. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aug 2023.

[40] Jieyu Zhang, Cheng-Yu Hsieh, Yue Yu, Chao Zhang, and Alexander Ratner. A survey on programmatic weak supervision, 2022.

[41] Renzhi Wu, Shen-En Chen, Jieyu Zhang, and Xu Chu. Learning hyper label model for programmatic weak supervision. In *International Conference on Learning Representations (ICLR)*, 2023.

[42] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International Conference on Machine Learning (ICML)*, pages 2127–2136. PMLR, 2018.

[43] Nan Lu, Gang Niu, Aditya Krishna Menon, and Masashi Sugiyama. On the minimal supervision for training any binary classifier from only unlabeled data. *arXiv preprint arXiv:1808.10585*, 2018.

[44] Clayton Scott and Jianxin Zhang. Learning from label proportions: A mutual contamination framework. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:22256–22267, 2020.

[45] Y. Zhang, N. Charoenphakdee, Z. Wu, and M. Sugiyama. Learning from aggregate observations. pages 7993–8005, 2020.

[46] Saurabh Garg, Yifan Wu, Alex Smola, Sivaraman Balakrishnan, and Zachary C. Lipton. Mixture proportion estimation and pu learning: A modern approach, 2021.

[47] Lei Feng, Senlin Shu, Nan Lu, Bo Han, Miao Xu, Gang Niu, Bo An, and Masashi Sugiyama. Pointwise binary classification with pairwise confidence comparisons. In *International Conference on Machine Learning*, pages 3252–3262. PMLR, 2021.

[48] Andrea Campagner. Learnability in "learning from fuzzy labels". In *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6. IEEE, 2021.

[49] Zheng Lian, Mingyu Xu, Lan Chen, Licai Sun, Bin Liu, and Jianhua Tao. Arnet: Automatic refinement network for noisy partial label learning. *arXiv preprint arXiv:2211.04774*, 2022.

[50] Mingyu Xu, Zheng Lian, Lei Feng, Bin Liu, and Jianhua Tao. Dali: Dynamically adjusted label importance for noisy partial label learning, 2023.

[51] Qian-Wei Wang, Yu-Feng Li, and Zhi-Hua Zhou. Partial label learning with unlabeled data. In *IJCAI*, pages 3755–3761, 2019.

[52] Wei Wang and Min-Ling Zhang. Semi-supervised partial label learning via confidence-rated margin maximization. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6982–6993, 2020.

[53] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

[54] Yu Yao, Tongliang Liu, Mingming Gong, Bo Han, Gang Niu, and Kun Zhang. Instance-dependent label-noise learning under a structural causal model. *Advances in Neural Information Processing Systems*, 34:4409–4420, 2021.

[55] Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 6500–6510. PMLR, 2020.

[56] Chidubem Arachie and Bert Huang. Constrained labeling for weakly supervised learning. In *Uncertainty in Artificial Intelligence*, pages 236–246. PMLR, 2021.

[57] Eyke Hüllermeier and Weiwei Cheng. Superset learning based on generalized loss minimization. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II 15*, pages 260–275. Springer, 2015.

[58] Jiaqi Lv, Biao Liu, Lei Feng, Ning Xu, Miao Xu, Bo An, Gang Niu, Xin Geng, and Masashi Sugiyama. On the robustness of average losses for partial-label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–15, 2023.

[59] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

[60] Thierry Denœux. Maximum likelihood estimation from fuzzy data using the em algorithm. *Fuzzy sets and systems*, 183(1):72–91, 2011.

[61] Eyke Hüllermeier. Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization. *International Journal of Approximate Reasoning*, 55(7):1519–1534, 2014.

[62] Benjamin Quost and Thierry Denoeux. Clustering and classification of fuzzy data using the fuzzy em algorithm. *Fuzzy Sets and Systems*, 286:134–156, 2016.

[63] Brendan Van Rooyen and Robert C Williamson. A theory of learning with corrupted labels. *J. Mach. Learn. Res.*, 18(1):8501–8550, 2017.

[64] Chen Gong, Jian Yang, Jane You, and Masashi Sugiyama. Centroid estimation with guaranteed efficiency: A general framework for weakly supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2841–2855, 2020.

[65] Chao-Kai Chiang and Masashi Sugiyama. Unified risk analysis for weakly supervised learning. *arXiv preprint arXiv:2309.08216*, 2023.

[66] Zixi Wei, Lei Feng, Bo Han, Tongliang Liu, Gang Niu, Xiaofeng Zhu, and Heng Tao Shen. A universal unbiased method for classification from aggregate observations. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 36804–36820. PMLR, 23–29 Jul 2023.

[67] Zheng Xie, Yu Liu, Hao-Yuan He, Ming Li, and Zhi-Hua Zhou. Weakly supervised auc optimization: A unified partial auc approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[68] Andrea Campagner. Learning from fuzzy labels: Theoretical issues and algorithmic solutions. *International Journal of Approximate Reasoning*, page 108969, 2023.

[69] Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. Robust training under label noise by over-parameterization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 14153–14172. PMLR, 17–23 Jul 2022.

[70] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020.

[71] Zheng Lian, Mingyu Xu, Lan Chen, Licai Sun, Bin Liu, and Jianhua Tao. Irnet: Iterative refinement network for noisy partial label learning, 2022.

[72] Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.

[73] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[74] Baixu Chen, Junguang Jiang, Ximei Wang, Jianmin Wang, and Mingsheng Long. Debiased pseudo labeling in self-training. *arXiv preprint arXiv:2202.07136*, 2022.

[75] David MacKay John Bridle, Anthony Heading. Unsupervised classifiers, mutual information and 'phantom targets. *Advances in Neural Information Processing Systems (NeurIPS)*, 1991.

[76] Armand Joulin and Francis Bach. A convex relaxation for weakly supervised classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2012.

[77] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[78] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

[79] Lei Feng, Takuo Kaneko, Bo Han, Gang Niu, Bo An, and Masashi Sugiyama. Learning with multiple complementary labels. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3072–3081. PMLR, 2020.

[80] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference (BMVC)*. British Machine Vision Association, 2016.

[81] Ning Xu, Congyu Qiao, Xin Geng, and Min-Ling Zhang. Instance-dependent partial label learning. *Advances in Neural Information Processing Systems*, 34:27119–27130, 2021.

[82] Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, Heli Qi, Zhen Wu, Yu-Feng Li, Satoshi Nakamura, Wei Ye, Marios Savvides, Bhiksha Raj, Takahiro Shinozaki, Bernt Schiele, Jindong Wang, Xing Xie, and Yue Zhang. Usb: A unified semi-supervised learning benchmark. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[83] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

[84] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172, 2013.

[85] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alex Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. *International Conference on Learning Representations (ICLR)*, 2021.

[86] Junnan Li, Caiming Xiong, and Steven CH Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 9475–9484, 2021.

[87] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching. *arXiv preprint arXiv:2203.06915*, 2022.

[88] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015.

[89] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data, 2017.

[90] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[91] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2233–2241, 2016.

[92] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088*, 2021.

[93] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pages 2304–2313. PMLR, 2018.

[94] Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. Pico+: Contrastive label disambiguation for robust partial label learning, 2022.

[95] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, pages 702–703, 2020.

[96] Min-Ling Zhang and Fei Yu. Solving the partial label learning problem: An instance-based approach. In *IJCAI*, pages 4048–4054, 2015.

[97] Chen Gong, Tongliang Liu, Yuanyan Tang, Jian Yang, Jie Yang, and Dacheng Tao. A regularization approach for instance-based superset label learning. *IEEE transactions on cybernetics*, 48(3):967–978, 2017.

[98] Ning Xu, Jiaqi Lv, and Xin Geng. Partial label learning via label enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5557–5564, 2019.

[99] Zhenguo Wu, Jiaqi Lv, and Masashi Sugiyama. Learning with proper partial labels. *Neural Computation*, 35(1):58–81, Jan 2023.

[100] Nam Nguyen and Rich Caruana. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 551–559, 2008.

[101] Min-Ling Zhang, Bin-Bin Zhou, and Xu-Ying Liu. Partial label learning via feature-aware disambiguation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1335–1344, 2016.

[102] Liping Liu and Thomas Dietterich. A conditional multinomial mixture model for superset label learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 25, 2012.

[103] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[104] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2018.

[105] Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 11525–11536. PMLR, 2021.

[106] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, 2019.

[107] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11557–11568, 2021.

[108] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2016.

[109] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[110] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, page 1–19, 2022.

[111] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[112] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction, 2020.

[113] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. In *IEEE Transactions on pattern analysis and machine intelligence*, pages 447–461, 2016.

[114] Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, Apr 2021.

[115] Yivan Zhang, Gang Niu, and Masashi Sugiyama. Learning noise transition matrix from only noisy labels via total variation regularization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 12501–12512. PMLR, 2021.

[116] Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu. Understanding and improving early stopping for learning with noisy labels, 2021.

[117] Junnan Li, Caiming Xiong, and Steven CH Hoi. Learning from noisy data with robust representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 9485–9494, 2021.

[118] Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. Selective-supervised contrastive learning with noisy labels. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2022.

[119] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2018.

[120] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[121] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. SELFIE: Refurbishing unclean samples for robust deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 5907–5915. PMLR, 09–15 Jun 2019.

[122] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction, 2019.

[123] Vinay Shukla, Zhe Zeng, Kareem Ahmed, and Guy Van den Broeck. A unified approach to count-based weakly-supervised learning. In *ICML 2023 Workshop on Differentiable Almost Everything: Differentiable Relaxations, Algorithms, Operators, and Simulators*, jul 2023.

[124] Massih-Reza Amini and Patrick Gallinari. Semi-supervised logistic regression. In *ECAI*, volume 2, page 11, 2002.

[125] Ning Xu, Jiaqi Lv, Biao Liu, Congyu Qiao, and Xin Geng. Progressive purification for instance-dependent partial label learning, 2022.

[126] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[127] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

[128] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[129] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021.

[130] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification. *arXiv preprint arXiv:2110.14795*, 2021.

[131] Jong-Chyi Su and Subhransu Maji. The semi-supervised inaturalist-aves challenge at fgvc7 workshop. *arXiv preprint arXiv:2103.06937*, 2021.

[132] Yelp dataset: http://www.yelp.com/dataset_challenge.

[133] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems (NeurIPS)*, 28:649–657, 2015.

[134] Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. Importance of semantic representation: Dataless classification. In *AAAI*, volume 2, pages 830–835, 2008.

[135] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. *arXiv preprint arXiv:2010.02347*, 2020.

Appendix

A Notation

We present the notation table for each symbol used in this paper in Table 6.

Table 6: Notation Table

Notation	Definition
$\mathbf{x}$	A training instance
y	A class index label
$\{\mathbf{x}_i\}_{i\in[N]}$	A set of data instances $\mathbf{x}$ of size N
$\{y_i\}_{i\in[N]}$	A set of precise label indices y of size N
$[\iota]$	An imprecise label, which might contain multiple class indices
$\{[\iota]_i\}_{i\in[N]}$	A set of imprecise labels $[\iota]$ of size N
X	Random variable of training instance
$\mathcal{X}$	Input space where $\mathbf{x}$ is drawn from
Y	Random variable of ground-truth labels
$\mathcal{Y}$	Label space where y is drawn from
I	Random variable of imprecise labels
f	Model backbone
g	Model classifier
h	Model multi-layer perceptron
$f \circ g$	Model mapping $\mathcal{X} \to \mathcal{Y}$
θ	Learnable parameters of $f \circ g$
$\mathbf{p}(y\|\mathbf{x};\theta)$	Output probability from model $f \circ g$
$f \circ h$	Model mapping $\mathcal{X} \to \mathcal{Z}$, where Z is a projected feature space
$\mathcal{D}$	Dataset
$\mathcal{L}$	Loss function
$\mathcal{A}_{\mathrm{w}}$	Weak data augmentation, usually is HorizontalFlip
$\mathcal{A}_{\mathrm{s}}$	Strong data augmentation, usually is RandAugment [95]
$\mathbf{z}_{\mathrm{w}}$	Projected features from $f \circ h$ on weakly-augmented data
$\mathbf{z}_{\mathrm{s}}$	Projected features from $f \circ h$ on strongly-augmented data
$\mathcal{M}$	Memory queue in MoCo [70]
$\mathbf{s}$	A partial label, with ground-truth label contained
$\{\mathbf{s}_i\}_{i\in[N]}$	A set partial labels, with ground-truth label contained of size N
S	Random variable of partial label
$\mathbf{x}^{\mathrm{l}}$	A labeled training example
y^{l}	A labeled class index
$\mathbf{x}^{\mathrm{u}}$	A unlabeled training example
y^{u}	A unknown class index for unlabeled data
X^{L}	A set of labeled data instances
Y^{L}	A set of labels for labeled data instances
X^{U}	A set of unlabeled data instances
Y^{U}	A set of unknown labels for unlabeled data instances
$\hat{p}^{\mathbf{u}}$	The maximum predicted probability on unlabeled data $\max(\mathbf{p}(y\|\mathbf{x}^{\mathrm{u}};\theta))$
$\hat{y}^{\mathrm{u}}$	The pseudo-label from the predicted probability on unlabeled data $\arg\max(\mathbf{p}(y\|\mathbf{x}^{\mathrm{u}};\theta))$
τ	The threshold for confidence thresholding
$\hat{y}$	A corrupted/noisy label
$\hat{y}^{\mathrm{oh}}$	An one-hot version of the corrupted/noisy label
$\hat{Y}$	Random variable of noisy labels
$\mathbf{u}, \mathbf{v}, \mathbf{m}$	Noise model related parameters in SOP [69]
$\mathcal{T}(\hat{y}\|y;\omega)$	The simplified noise transition model in ILL
ω	The parameters in the simplified noise model

B Related Work

Many previous methods have been proposed for dealing with the specific types and some combinations of imprecise label configurations. We revisit the relevant work in this section, especially the state-of-the-art popular baselines for learning with individual and mixture imprecise label configurations.

Partial label learning (PLL). The prior arts can be roughly divided into identification-based for label disambiguation [96–99] or average-based for utilizing all candidate labels [72, 7, 58]. The traditional average-based methods usually treat all candidate labels equally, which may involve the misleading false positive labels into training. To overcome these limitations, researchers have

explored identification-based methods, viewing the ground-truth label as a latent variable. They seek to maximize its estimated probability using either the maximum margin criterion [100, 101] or the maximum likelihood criterion [102]. Deep learning techniques have recently been incorporated into identification-based methods, yielding promising results across multiple datasets. For example, PRODEN [55] proposed a self-training strategy that disambiguates candidate labels using model outputs. CC [9] introduced classifier-consistent and risk-consistent algorithms, assuming uniform candidate label generation. LWS [11] relaxed this assumption and proposed a family of loss functions for label disambiguation. More recently, Wangt et al. [13] incorporated contrastive learning into PLL, enabling the model to learn discriminative representations and show promising results under various levels of ambiguity. RCR involves consistency regularization into PLL recently [12].

Semi-supervised learning (SSL). SSL is a paradigm for learning with a limited labeled dataset supplemented by a much larger unlabeled dataset. Consistency regularization and self-training, inspired by clusterness and smoothness assumptions, have been proposed to encourage the network to generate similar predictions for inputs under varying perturbations [103, 15, 104]. Self-training [14, 73, 18] is a widely-used approach for leveraging unlabeled data. Pseudo Label [14, 73], a well-known self-training technique, iteratively creates pseudo labels that are then used within the same model. Recent studies focus largely on generating high-quality pseudo-labels. MixMatch [16], for instance, generates pseudo labels by averaging predictions from multiple augmentations. Other methods like ReMixMatch [17], UDA [53], and FixMatch [18] adopt confidence thresholds to generate pseudo labels for weakly augmented samples, which are then used to annotate strongly augmented samples. Methods such as Dash [105], FlexMatch [20], and FreeMatch [22] dynamically adjust these thresholds following a curriculum learning approach. SoftMatch [23] introduces a novel utilization of pseudo-labels through Gaussian re-weighting. SSL has also seen improvements through the incorporation of label propagation, contrastive loss, and meta learning [106, 107, 86, 87, 82].

Noisy label learning (NLL). Overfitting to the noisy labels could result in poor generalization performance, even if the training error is optimized towards zero [108, 109]. Several strategies to address the noisy labels have been proposed [110]. Designing loss functions that are robust to noise is a well-explored strategy for tackling the label noise problem [29, 31, 111, 112]. Additionally, methods that re-weight loss [113] have also been explored for learning with noisy labels. Another common strategy to handle label noise involves assuming that the noisy label originates from a probability distribution that depends on the actual label. Early works [26] incorporated these transition probabilities into a noise adaptation layer that is stacked over a classification network and trained in an end-to-end fashion. More recent work, such as Forward [91], prefers to estimate these transition probabilities using separate procedures. However, the success of this method is contingent upon the availability of clean validation data [114] or additional assumptions about the data [115]. Noise correction has shown promising results in noisy label learning recently [116–118, 69]. During the early learning phase, the model can accurately predict a subset of the mislabeled examples [32]. This observation suggests a potential strategy of correcting the corresponding labels. This could be accomplished by generating new labels equivalent to soft or hard pseudo-labels estimated by the model [119, 120]. Co-Teaching uses multiple differently trained networks for correcting noisy labels [28]. SELFIE [121] corrects a subset of labels by replacing them based on past model outputs. Another study in [122] uses a two-component mixture model for sample selection, and then corrects labels using a convex combination. Similarly, DivideMix [33] employs two networks for sample selection using a mixture model and Mixup [90].

Mixture imprecise label settings. Various previous works have explored dealing with distinct types of imprecise labels. However, they have yet to tackle a combination of partial labels, limited labels, and noisy labels, which is a highly realistic scenario. For instance, recent attention has been paid to the issue of partial noisy label learning. PiCO+ [94], an extended version of PiCO [13], is tailored specifically for partial noisy labels. IRNet [71] uses two modules: noisy sample detection and label correction, transforming the scenario of noisy PLL into a more traditional PLL. DALI [50] is another framework designed to reduce the negative impact of detection errors by creating a balance between the initial candidate set and model outputs, with theoretical assurances of its effectiveness. Additionally, some work has focused on semi-supervised partial label learning [51, 52]. No existing research can effectively address the challenge of handling a combination of partial, limited, and noisy labels simultaneously, which underscores the novelty and significance of our work.

Previous attempts towards unification of learning from imprecise labels. There are earlier attempts for the generalized solutions of different kinds of imprecise labels/observations. Denœux

[60] proposed an EM algorithm for the likelihood estimation of fuzzy data and verified the algorithm on linear regression and uni-variate normal mixture estimation. Van Rooyen et al. [63] developed an abstract framework that generically tackles label corruption via the Markov transition. Quost et al. [62] further extended the EM algorithm of fuzzy data on the finite mixture of Gaussians. Gong et al. [64] proposed a general framework with centroid estimation for imprecise supervision. A unified partial AUC optimization approach was also proposed earlier [67]. Zhang et al. [45] and Wei. et al. [66] proposed generalized solutions for aggregate observations. A unified solution based on dynamic programming for count-based weak supervision was also proposed [123] While relating to these works on the surface, ILL does not require any assumption on the imprecise information and generalizes well to more practical settings with noisy labels. Some other works for individual settings also related EM framework, but usually involved the approximation on the EM [124, 25, 13].

C Methods

C.1 Derivation of Variational Lower Bound

Evidence lower bound (ELBO), or equivalently variational lower bound [59], is the core quantity in EM. From Eq. (5), to model $\log P(X, I; \theta)$, we have:

$$\begin{aligned}
\log P(X, I; \theta) &= \int Q(Y) \log P(X, I; \theta) dY \\
&= \int Q(Y) \log P(X, I; \theta) \frac{P(Y|X, I; \theta)}{P(Y|X, I; \theta)} dY \\
&= \int Q(Y) \log \frac{P(X, I, Y; \theta) Q(Y)}{P(Y|X, I; \theta) Q(Y)} dY \\
&= \int Q(Y) \log \frac{P(X, I, Y; \theta)}{Q(Y)} dY - \int Q(Y) \log \frac{P(Y|X, I; \theta)}{Q(Y)} dY
\end{aligned} \tag{12}$$

where the first term is the ELBO and the second term is the KL divergence $\mathcal{D}_{KL}(Q(Y)||P(Y|X, I; \theta))$. Replacing $Q(Y)$ with $P(Y|X, I; \theta^t)$ at each iteration will obtain Eq. (5).

C.2 Instantiations to Partial Label Learning

The imprecise label I for partial labels is defined as the label candidate sets S with $\{\mathbf{s}_i\}_{i \in [N]}$ containing the true labels. Now we can derive Eq. (6) by replacing I with S in Eq. (5):

$$\begin{aligned}
&\mathbb{E}_{Y|X, I; \theta^t} \left[\log P(Y|X; \theta) + \log P(I|X, Y; \theta) \right] \\
&= \mathbb{E}_{Y|X, S; \theta^t} \left[\log P(Y|X; \theta) + \log P(I|X, Y; \theta) \right] \\
&= \sum_Y P(Y|X, S; \theta^t) \left[\log P(Y|X; \theta) + \log P(I|X, Y; \theta) \right] \\
&= \sum_Y P(Y|X, S; \theta^t) \left[\log P(Y|X; \theta) \right] + \log P(I|X, Y; \theta)
\end{aligned} \tag{13}$$

Note that $P(I|Y, X; \theta)$ can be moved out of the expectation because it is a fixed quantity to any Y. Now we replace Y, X, and S to y, $\mathbf{x}$, and $\mathbf{s}$ for each instance, and converting the maximization problem to negative log-likelihood minimization problem to drive the loss function:

$$\mathcal{L}_{\text{ILL}}^{\text{PLL}} = -\frac{1}{N} \sum_i^N \mathbf{p}(y_i|\mathbf{x}_i, \mathbf{s}_i; \theta^t) \log \mathbf{p}(y_i|\mathbf{x}_i; \theta) - \frac{1}{N} \sum_i^N \log \mathbf{p}(\mathbf{s}_i|\mathbf{x}_i, y_i; \theta). \tag{14}$$

The first term is the Cross-Entropy loss we derived in Eq. (6). If S is not instance-dependent, then knowing Y also knows S, the second term thus can be ignored in Eq. (6). If S becomes instance-dependent, the second term can be maintained as a supervised term as in [12] to optimize θ.

C.3 Instantiations to Semi-Supervised Learning

In SSL, the input X consists of the labeled data X^{L} and the unlabeled data X^{U}. The imprecise label for SSL is realized as the limited number of full labels Y^{L} for X^{L}. The labels Y^{U} for unlabeled X^{U}

are unknown and become the latent variable. Thus we can write:

$$\begin{aligned}
&\mathbb{E}_{Y|X,I;\theta^t}\left[\log P(Y|X;\theta)+\log P(I|X,Y;\theta)\right]\\
&=\mathbb{E}_{Y^{\mathrm{U}}|X^{\mathrm{U}},X^{\mathrm{L}},Y^{\mathrm{L}};\theta^t}\left[\log P(Y^{\mathrm{U}}|X^{\mathrm{U}},X^{\mathrm{L}};\theta)+\log P(Y^{\mathrm{L}}|X^{\mathrm{L}},X^{\mathrm{U}},Y^{\mathrm{U}};\theta)\right]\\
&=\sum_{Y^{\mathrm{U}}}P(Y^{\mathrm{U}}|X^{\mathrm{U}};\theta^t)\left[\log P(Y^{\mathrm{U}}|X^{\mathrm{U}};\theta)\right]+\log P(Y^{\mathrm{L}}|X^{\mathrm{L}};\theta).
\end{aligned} \tag{15}$$

The negative log-likelihood loss function for $\{\mathbf{x}_i^l, y_i^l\}_{i\in[N^L]}$ and $\{\mathbf{x}^u\}_{i\in[N^U]}$ thus becomes:

$$\mathcal{L}_{\mathrm{ILL}}^{\mathrm{SSL}}=\mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}(y|\mathbf{x}^{\mathrm{u}};\theta),\mathbf{p}(y|\mathbf{x}^{\mathrm{u}};\theta^t)\right)+\mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}(y|\mathbf{x}^{\mathrm{L}};\theta),y^{\mathrm{L}}\right) \tag{16}$$

C.4 Instantiations to Noisy Label Learning

We denote the given noisy labels as $\hat{Y}$. For noisy label learning, our method naturally supports a noise transition model $\mathcal{T}(\hat{Y}|Y;\omega)$ with learnable parameter ω, as we will show in the following:

$$\begin{aligned}
&\mathbb{E}_{Y|X,I;\theta^t}\left[\log P(Y|X;\theta)+\log P(I|X,Y;\theta)\right]\\
&=\mathbb{E}_{Y|X,\hat{Y};\theta^t}\left[\log P(Y,\hat{Y}|X;\theta)\right]\\
&=\mathbb{E}_{Y|X,\hat{Y};\theta^t}\left[\log P(Y|\hat{Y},X;\theta)+\log P(\hat{Y}|X;\theta)\right]\\
&=\sum_{Y}P(Y|\hat{Y},X;\theta^t)\log P(Y|\hat{Y},X;\theta)+\log P(\hat{Y}|X;\theta).
\end{aligned} \tag{17}$$

The loss function is:

$$\mathcal{L}_{\mathrm{ILL}}^{\mathrm{NLL}}=\mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}(y|\mathbf{x},\hat{y};\theta,\omega^t),\mathbf{p}(y|\mathbf{x},\hat{y};\theta^t,\omega^t)\right)+\mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}(\hat{y}|\mathbf{x};\theta,\omega),\hat{y}\right) \tag{18}$$

Note that both term is computed from the noise transition matrix as mentioned in Eq. (9).

C.5 Instantiations to Mixed Imprecise Label Learning

In this setting, we have both labeled data and unlabeled data, where the labels for the labeled data are both partial and noisy. On the unlabeled data, the unsupervised objective is the same as the unsupervised consistency regularization of semi-supervised learning shown in Eq. (7). On the labeled data, it mainly follows the Eq. (9) of noisy label learning, with the noisy single label becoming the noisy partial labels $\hat{\mathbf{s}}$. For noisy partial labels, the noisy supervised objective in Eq. 8 becomes the supervised consistency regularization as in Eq. 6 of partial label setting to train the noise transition model, and the noisy unsupervised objective becomes the consistency regularization of the prediction conditioned on noisy partial labels:

$$\mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}\left(y\mid\mathcal{A}_{\mathrm{s}}(\mathbf{x}),\hat{\mathbf{s}};\theta,\omega^t\right),\mathbf{p}\left(y\mid\mathcal{A}_{\mathrm{w}}(\mathbf{x}),\hat{y};\theta^t,\omega^t\right)\right)+\mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}\left(\hat{y}\mid\mathcal{A}_{\mathrm{w}}(\mathbf{x});\theta,\omega\right),\hat{\mathbf{s}}\right) \tag{19}$$

We can compute both quantity through the noise transition model:

$$\mathbf{p}(y|\mathbf{x},\hat{\mathbf{s}};\theta,\omega^t)\propto\mathbf{p}(y|\mathbf{x};\theta)\prod_{\hat{y}\in\hat{\mathbf{s}}}\mathcal{T}(y|\hat{y};\omega^t),\text{and }\mathbf{p}(\hat{y}|\mathbf{x};\theta,\omega)=\sum_{y\in[C]}\mathbf{p}(y|\mathbf{x};\theta)\mathcal{T}(\hat{y}|y;\omega). \tag{20}$$

D Experiments

D.1 Additional Training Details

We adopt two additional training strategies for the ILL framework. The first is the "strong-weak" augmentation strategy [53]. Since there is a consistency regularization term in each imprecise label formulation of ILL, we use the soft pseudo-targets of the weakly-augmented data to train the strongly-augmented data. The second is the entropy loss [75] for class balancing, which is also adopted in SOP [69] and FreeMatch [22]. We set the loss weight for the entropy loss uniformly for all experiments as 0.1.

D.2 Partial Label Learning

D.2.1 Setup

Following previous work [125, 11, 13], we evaluate our method on partial label learning setting using CIFAR-10, CIFAR-100, and CUB-200 [78]. We generate partially labeled datasets by flipping negative labels to false positive labels with a probability q, which is also denoted as a partial ratio. Specifically, the $C-1$ negative labels are uniformly aggregated into the ground truth label to form a set of label candidates. We consider $q \in \{0.1, 0.3, 0.5\}$ for CIFAR-10, $q \in \{0.01, 0.05, 0.1\}$ for CIFAR-100, and $q = 0.05$ for CUB-200. For CIFAR-10 and CIFAR-100, we use ResNet-18 [1] as backbone. We use SGD as an optimizer with a learning rate of 0.01, a momentum of 0.9, and a weight decay of $1\mathrm{e}{-3}$. For CUB-200, we initialize the ResNet-18 [1] with ImageNet-1K [126] pre-trained weights. We train 800 epochs for CIFAR-10 and CIFAR-100 [77], and 300 epochs for CUB-200, with a cosine learning rate scheduler. For CIFAR-10 and CIFAR-100, we use an input image size of 32. For CUB-200, we use an input image size of 224. A batch size of 256 is used for all datasets. The choice of these parameters mainly follows PiCO [13]. We present the full hyper-parameters systematically in Table 7.

Table 7: Hyper-parameters for **partial label learning** used in experiments.

Hyper-parameter	CIFAR-10	CIFAR-100	CUB-200
Image Size	32	32	224
Model	ResNet-18	ResNet-18	ResNet-18 (ImageNet-1K Pretrained)
Batch Size	256	256	256
Learning Rate	0.01	0.01	0.01
Weight Decay	1e-3	1e-3	1e-5
LR Scheduler	Cosine	Cosine	Cosine
Training Epochs	800	800	300
Classes	10	100	200

D.2.2 Discussion

We additionally compare our method with R-CR [12], which uses a different architecture as the results in Table 1. R-CR uses Wide-ResNet34x10 as backbone, and adopts multiple strong data augmentations. It also adjusts the loss weight along training. For fair comparison, we use the same architecture without multiple augmentation and the curriculum adjust on loss. The results are shown in Table 8, where our method outperforms R-CR on CIFAR-10 and is comparable on CIFAR-100.

Table 8: Comparison with R-CR in partial label learning

Method	CIFAR-10		CIFAR-100	
	0.3	0.5	0.05	0.10
R-CR	97.28±0.02	97.05±0.05	82.77±0.10	82.24±0.07
Ours	97.55±0.07	97.17±0.11	82.46±0.08	82.22±0.05

We also provide the comparison of our method on instance-dependent partial label learning as proposed by Xu et al. [81, 125]. Due to the nature of instance-dependence, we maintain the term $P(S|Y, X; \theta)$ from Eq. (5) as a supervised term for optimization. We compare our method with VALEN [81], RCR [12], PiCO [13], and POP [125] on MNIST, Kuzushiji-MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100, with synthetic instance-dependent partial labels generated according to Xu et al. [125]. From the results in Table 9, we proposed method demonstrate the best performance across different datasets evaluated.

A recent work on PLL discussed and analyzed the robustness performance of different loss functions, especially the average-based methods [58]. We perform a similar analysis here for the derived loss function in ILL. Following the notation in [58], let $\mathbf{s}$ denote the candidate label set, $\mathbf{x}$ as the training instance, g as the probability score from the model, and f as the classifier $f(\boldsymbol{x}) = \arg\max_{i \in \mathcal{Y}} g_i(\boldsymbol{x})$, the average-based PLL can be formulated as:

$$\mathcal{L}_{avg-PLL}(f(\boldsymbol{x}), \mathbf{s}) = \frac{1}{|\mathbf{s}|} \sum_{i \in \mathbf{s}} \ell(f(\boldsymbol{x}), i) \tag{21}$$

Table 9: Comparison on instance-dependent partial label learning

	MNIST	Kuzushiji-MNIST	Fashion-MNIST	CIFAR-10	CIFAR-100
VALEN [81]	99.03	90.15	96.31	92.01	71.48
RCR [12]	98.81	90.62	96.64	86.11	71.07
PiCO [13]	98.76	88.87	94.83	89.35	66.30
POP [125]	**99.28**	91.09	96.93	93.00	71.82
Ours	99.19	**91.35**	**97.01**	**93.86**	**72.43**

Lv et al. [58] compared different loss functions ℓ on both noise-free and noisy PLL settings, where they find both theoretically and empirically that average-based PLL with *bounded* loss are robust under mild assumptions. Empirical study in [58] suggests that both *Mean Absolute Error* and *Generalized Cross-Entropy* loss [29] that proposed for noisy label learning achieves the best performance and robustness for average-based PLL.

Our solution for PLL can be viewed as an instantiation of the average-based PLL as in [58] with:

$$\ell(f(\mathbf{x}), i) = -\bar{g}_i(\mathbf{x}) \log g_i(\mathbf{x}) \tag{22}$$

where $\bar{g}$ is normalized probability over $\mathbf{s}$ with detached gradient. We can further show that the above loss function is bounded for $0 < \ell \leq \frac{1}{e}$ and thus bounded for summation of all classes, which demonstrates robustness, as we show in Table 4.

D.3 Semi-Supervised Learning

D.3.1 Setup

For experiments of SSL, we follow the training and evaluation protocols of USB [82] on image and text classification. To construct the labeled dataset for semi-supervised learning, we uniformly select l/C samples from each class and treat the remaining samples as the unlabeled dataset. For image classification tasks, ImageNet-1K [126] Vision Transformers [4] are used, including CIFAR-100 [77], EuroSAT [127], STL-10 [128], TissueMNIST [129, 130], Semi-Aves [131]. For text classification tasks, we adopt BERT [3] as backbone, including IMDB [83], Amazon Review [84], Yelp Review [132], AG News [133] , Yahoo Answer [134]. The hyper-parameters strictly follow USB, and are shown in Table 10 and Table 11.

Table 10: Hyper-parameters of **semi-supervised learning** used in vision experiments of USB.

Hyper-parameter	CIFAR-100	STL-10	Euro-SAT	TissueMNIST	Semi-Aves
Image Size	32	96	32	32	224
Model	ViT-S-P4-32	ViT-B-P16-96	ViT-S-P4-32	ViT-T-P4-32	ViT-S-P16-224
Labeled Batch size			16		
Unlabeled Batch size			16		
Learning Rate	5e-4	1e-4	5e-5	5e-5	1e-3
Weight Decay			5e-4		
Layer Decay Rate	0.5	0.95	1.0	0.95	0.65
LR Scheduler			$\eta = \eta_0 \cos(\frac{7\pi k}{16K})$		
Training epochs			20		
Classes	100	10	10	10	200
Model EMA Momentum			0.0		
Prediction EMA Momentum			0.999		
Weak Augmentation			Random Crop, Random Horizontal Flip		
Strong Augmentation			RandAugment [95]		

D.3.2 Results

In the main paper, we only provide the comparison on CIFAR-100, STL-10, IMDB, and Amazon Review. Here we provide the full comparison in Table 12 and Table 13. From the full results, similar conclusion can be drawn as in the main paper. Our ILL framework demonstrates comparable performance as previous methods.

Table 11: Hyper-parameters of **semi-supervised learning** NLP experiments in USB.

Hyper-parameter	AG News	Yahoo! Answer	IMDB	Amazom-5	Yelp-5
Max Length	512				
Model	Bert-Base				
Labeled Batch size	4				
Unlabeled Batch size	4				
Learning Rate	5e-5	1e-4	5e-5	1e-5	5e-5
Weight Decay	1e-4				
Layer Decay Rate	0.65	0.65	0.75	0.75	0.75
LR Scheduler	$\eta = \eta_0 \cos(\frac{7\pi k}{16K})$				
Training epochs	10				
Classes	4	10	2	5	5
Model EMA Momentum	0.0				
Prediction EMA Momentum	0.999				
Weak Augmentation	None				
Strong Augmentation	Back-Translation [53]				

Table 12: Error rate comparison of different number of labels on CIFAR-100, STL-10, EuroSAT, TissueMNIST, and SemiAves for **semi-supervised learning**. We use USB [82] image classification task results. The best results are indicated in bold. Our results are averaged over 3 independent runs.

Datasets	CIFAR-100		STL-10		EuroSat		TissueMNIST		SemiAves
# Labels	200	400	40	100	20	40	80	400	3959
Pseudo-Label [14]	33.99±0.95	25.32±0.29	19.14±1.33	10.77±0.60	25.46±1.36	15.70±2.12	56.92±4.54	50.86±1.79	40.35±0.30
Mean-Teacher [103]	35.47±0.40	26.03±0.30	18.67±1.69	24.19±10.15	26.83±1.46	15.85±1.66	62.06±3.43	55.12±2.53	38.55±0.21
VAT [104]	31.49±1.33	21.34±0.50	18.45±1.47	10.69±0.51	26.16±0.96	10.09±0.94	57.49±5.47	51.30±1.73	38.82±0.04
MixMatch [16]	38.22±0.71	26.72±0.72	58.77±1.98	36.74±1.24	24.85±4.85	17.28±2.67	55.53±1.51	49.64±2.28	37.25±0.08
ReMixMatch [17]	22.21±2.21	16.86±0.57	13.08±3.34	**7.21±0.39**	**5.05±1.05**	5.07±0.56	58.77±4.43	49.82±1.18	**30.20±0.03**
AdaMatch [85]	22.32±1.73	16.66±0.62	13.64±2.49	7.62±1.90	7.02±0.79	**4.75±1.10**	58.35±4.87	52.40±2.08	31.75±0.13
FixMatch [18]	29.60±0.90	19.56±0.52	16.15±1.89	8.11±0.68	13.44±3.53	5.91±2.02	**55.37±4.50**	51.24±1.56	31.90±0.06
FlexMatch [20]	26.76±1.12	18.24±0.36	14.40±3.11	8.17±0.78	5.17±0.57	5.58±0.81	58.36±3.80	51.89±3.21	32.48±0.15
Dash [105]	30.61±0.98	19.38±0.10	16.22±5.95	7.85±0.74	11.19±0.90	6.96±0.87	56.98±2.93	51.97±1.55	32.38±0.16
CoMatch [86]	35.08±0.69	25.35±0.50	15.12±1.88	9.56±1.35	5.75±0.43	4.81±1.05	59.04±4.90	52.92±1.04	38.65±0.18
SimMatch [87]	23.78±1.08	17.06±0.78	11.77±3.20	7.55±1.86	7.66±0.60	5.27±0.89	60.88±4.31	52.93±1.56	33.85±0.08
FreeMatch [22]	**21.40±0.30**	**15.65±0.26**	12.73±3.22	8.52±0.53	6.50±0.78	5.78±0.51	58.24±3.08	52.19±1.35	32.85±0.31
SoftMatch [23]	22.67±1.32	16.84±0.66	13.55±3.16	7.84±1.72	5.75±0.62	5.90±1.42	57.98±3.66	51.73±2.84	31.80±0.22
Ours	22.06±1.06	17.40±1.04	**11.09±0.71**	8.10±1.02	5.86±1.06	5.74±1.13	57.99±2.16	**50.95±2.03**	33.08±0.26

Table 13: Error rate comparison of different number of labels on IMDB, AG News, Amazon Review, Yahoo Answers, and Yelp Review for **semi-supervised learning**. We use USB [82] text classification task results. Best results are indicated in bold. Our results are averaged over 3 independent runs.

Datasets	IMDB		AG News		Amazon Review		Yahoo Answers		Yelp Review	
# Labels	20	100	40	200	250	1000	500	2000	250	1000
Pseudo-Label [14]	45.45±4.43	19.67±1.01	19.49±3.07	14.69±1.88	53.45±1.9	47.00±0.79	37.70±0.65	32.72±0.31	54.51±0.82	47.33±0.20
Mean-Teacher [103]	20.06±2.51	13.97±1.49	15.17±1.21	13.93±0.65	52.14±0.52	47.66±0.84	37.09±0.18	33.43±0.28	50.60±0.62	47.21±0.31
VAT [104]	25.93±2.58	11.61±1.79	14.70±1.19	11.71±0.84	49.83±0.46	46.54±0.31	34.87±0.41	31.50±0.35	52.97±1.41	45.30±0.32
MixMatch [16]	26.12±6.13	15.47±0.65	13.50±1.51	11.75±0.60	59.54±0.67	61.69±3.32	35.75±0.71	33.62±0.14	53.98±0.59	51.70±0.68
AdaMatch [85]	8.09±0.99	7.11±0.20	**11.73±0.17**	**11.22±0.95**	46.72±0.72	42.27±0.25	32.75±0.35	30.44±0.31	45.40±0.96	40.16±0.49
FixMatch [18]	7.72±0.33	7.33±0.13	30.17±1.87	11.71±1.95	47.61±0.83	43.05±0.54	**33.03±0.49**	30.51±0.53	46.52±0.94	40.65±0.46
FlexMatch [20]	7.82±0.77	7.41±0.38	16.38±3.94	12.08±0.73	45.73±1.60	42.25±0.33	35.61±1.08	31.13±0.18	**43.35±0.69**	40.51±0.34
Dash [105]	8.34±0.86	7.55±0.35	17.67±3.19	13.76±1.67	47.10±0.74	43.09±0.60	35.26±0.33	31.19±0.29	45.24±2.02	40.14±0.79
CoMatch [86]	7.44±0.30	7.72±1.14	11.95±0.76	10.75±0.35	48.76±0.90	43.36±0.21	33.48±0.51	30.25±0.35	45.40±1.12	40.27±0.51
SimMatch [87]	7.93±0.55	**7.08±0.33**	14.26±1.51	12.45±1.37	45.91±0.95	42.21±0.30	33.06±0.20	30.16±0.21	46.12±0.48	40.26±0.62
FreeMatch [22]	8.94±0.21	7.95±0.45	12.98±0.58	11.73±0.63	46.41±0.60	42.64±0.06	32.77±0.26	30.32±0.18	47.95±1.45	40.37±1.00
SoftMatch [23]	7.76±0.58	7.97±0.72	11.90±0.27	11.72±1.58	45.29±0.95	**42.21±0.20**	33.07±0.31	30.44±0.62	44.09±0.50	39.76±0.13
Ours	**7.32±0.12**	7.64±0.67	14.77±1.59	12.21±0.82	**43.96±0.32**	42.32±0.02	33.80±0.25	30.86±0.17	44.82±0.17	**39.67±0.71**

D.4 Noisy Label Learning

D.4.1 Setup

We conduct experiments of noisy label learning following SOP [69]. We evaluate the proposed method on both synthetic symmetric/asymmetric noise on CIFAR-10 and CIFAR-100, and more realistic and larger-scale instance noise on Clothing1M and WebVision. To introduce the synthetic symmetric noise to CIFAR-10 and CIFAR-100, we uniformly flip labels for a probability η into other classes. For asymmetric noise, we only randomly flip the labels for particular pairs of classes. For CIFAR-10 and CIFAR-100, we train PreAct-ResNet-18 with SGD using a learning rate of 0.02, a

weight decay of $1e-3$, and a momentum of 0.9. We train for 300 epochs with a cosine learning rate schedule and a batch size of 128. For WebVision, we use InceptionResNet-v2 as the backbone and set the batch size to 32. Other settings are similar to CIFAR-10. For Clothing1M, we use ImageNet-1K pre trained ResNet-50 as the backbone. We train it using SGD with an initial learning rate of $2e$-3 for a total of 10 epochs, where the learning rate is reduced by 10 after 5 epochs. In addition, we also conduct experiments on CIFAR-10N and CIFAR-100N. We present the detailed hyper-parameters in Table 14.

Table 14: Hyper-parameters for **noisy label learning** used in experiments.

Hyper-parameter	CIFAR-10 (CIFAR-10N)	CIFAR-100 (CIFAR-100N)	Clothing1M	WebVision
Image Size	32	32	224	299
Model	PreAct-ResNet-18 (ResNet-34)	PreAct-ResNet-18 (ResNet-34)	ResNet-50 (ImageNet-1K Pretrained)	Inception-ResNet-v2
Batch Size	128	128	64	32
Learning Rate	0.02	0.02	0.002	0.02
Weight Decay	1e-3	1e-3	1e-3	5e-4
LR Scheduler	Cosine	Cosine	MultiStep	MultiStep
Training Epochs	300	300	10	100
Classes	10	100	14	50
Noisy Matrix Scale	1.0	2.0	0.5	2.5

D.4.2 Results

In addition to the results regarding noisy label learning provided in the main paper, we also present comparison results on CIFAR-10N and CIFAR-100N [92] in Table 15. We include a full comparison on Clothing1M and WebVision, incorporating methods like Co-Teaching, Forward, and CORES, in Table 16. As shown in Table 15, the proposed ILL framework achieves performance comparable to the previous best method, SOP [69]. On CIFAR-10N, our method yields results very close to SOP in the Random and Aggregate case noise scenarios and surpasses SOP in the Worst case noise scenario. However, on CIFAR-100N, our method slightly underperforms previous methods, possibly due to the oversimplified noise model utilized in ILL. We believe that a more realistic noise transition model and further tuning of our method could lead to improved performance.

Table 15: Test accuracy comparison of instance independent label noise on CIFAR-10N and CIFAR-100N for **noisy label learning**. The best results are indicated in **bold**, and the second best results are indicated in underline. Our results are averaged over three independent runs with ResNet34 as the backbone.

Dataset	CIFAR-10N						CIFAR-100N	
Noisy Type	Clean	Random 1	Random 2	Random 3	Aggregate	Worst	Clean	Noisy
CE	92.92±0.11	85.02±0.65	86.46±1.79	85.16±0.61	87.77±0.38	77.69±1.55	76.70±0.74	55.50±0.66
Forward [91]	93.02±0.12	86.88±0.50	86.14±0.24	87.04±0.35	88.24±0.22	79.79±0.46	76.18±0.37	57.01±1.03
Co-teaching [28]	93.35±0.14	90.33±0.13	90.30±0.17	90.15±0.18	91.20±0.13	83.83±0.13	73.46±0.09	60.37±0.27
DivideMix [33]	-	95.16±0.19	95.23±0.07	95.21±0.14	95.01±0.71	92.56±0.42	-	71.13±0.48
ELR [32]	95.39±0.05	94.43±0.41	94.20±0.24	94.34±0.22	94.83±0.10	91.09±1.60	78.57±0.12	66.72±0.07
CORES [135]	94.16±0.11	94.45±0.14	94.88±0.31	94.74±0.03	95.25±0.09	91.66±0.09	73.87±0.16	55.72±0.42
SOP [69]	**96.38±0.31**	95.28±0.13	95.31±0.10	95.39±0.11	95.61±0.13	93.24±0.21	**78.91±0.43**	67.81±0.23
Ours	96.21±0.29	**96.06±0.07**	**95.98±0.12**	**96.10±0.05**	**96.40±0.03**	**93.55±0.14**	78.53±0.21	**68.07±0.33**

D.5 Mixed Imprecise Label Learning

D.5.1 Setup

To create a mixture of various imprecise label configurations, we select CIFAR-10 and CIFAR-100 as base datasets. We first uniformly sample l/C labeled samples from each class to form the labeled dataset and treat the remaining samples as the unlabeled dataset. Based on the labeled dataset, we generate partially labeled datasets by flipping negative labels to false positive labels with the partial ratio q. After obtaining the partial labels, we randomly select η percentage of samples from each class, and recreate the partial labels for them by flipping the ground truth label uniformly to another class. In this setting, unlabeled data, partially labeled data, and noisy labeled data exist simultaneously, which is very challenging and more closely resembles realistic situations. For CIFAR-10, we set $l \in \{1000, 5000, 50000\}$, and for CIFAR-100, we set $l \in \{5000, 10000, 50000\}$. Similarly in

Table 16: Test accuracy comparison of realistic noisy labels on Clothing1M and WebVision for **noisy label learning**. The best results are indicated in **bold** and the second best results are indicated in underline. Our results are averaged over 3 independent runs. For Clothing1M, we use ImageNet-1K pre trained ResNet50 as the backbone. For WebVision, InceptionResNetv2 is used as the backbone.

Dataset	Clothing1M	WebVision
CE	69.10	-
Forward [91]	69.80	61.10
MentorNet [93]	66.17	63.00
Co-Teaching [28]	69.20	63.60
DivideMix [33]	**74.76**	77.32
ELR [32]	72.90	76.20
CORES [135]	73.20	-
SOP [69]	73.50	76.60
Ours	$74.02_{\pm 0.12}$	$\mathbf{79.37}_{\pm 0.09}$

the partial label setting, we set $q \in \{0.1, 0.3, 0.5\}$ for CIFAR-10, and $q \in \{0.01, 0.05, 0.1\}$ for CIFAR-100. For noisy labels, we set $\eta \in \{0.1, 0.2, 0.3\}$ for both datasets.

D.5.2 Results

We provide a more complete version of Table 4 in Table 17. On partial noisy labels of CIFAR-10 with partial ratio 0.5 and of CIFAR-100 with partial ratio 0.1, most baseline methods are more robust or even fail to perform. However, our ILL still shows very robust performance with minor performance degradation as increase of noise ratios.

D.6 Ablation on Strong-Augmentation and Entropy Loss

We provide the ablation study on the strong-augmentation and entropy loss components here, which are common techniques in each setting [18, 13, 69]. For example, in SSL, strong-weak augmentation is an important strategy for SSL algorithms widely used in existing works such as FixMatch [18] and FlexMatch [20]. Thus, it is important to adopt strong-weak augmentation to achieve better performance in SSL [22, 23, 82]. This is similar in PLL settings [13, 12]. PiCO [13, 12] also used strong augmentation). Strong-weak augmentation and entropy loss are also adopted in SOP [69] of NLL. However, we found these techniques are less important for our formulation of NLL. We provide an ablation study on the entropy loss of SSL, and both techniques for NLL and PLL here to demonstrate our discussions.

D.7 Runtime Analysis

We provide the runtime analysis on CIFAR-100 of our method on different settings, compared with the SOTA baselines. We compute the average runtime from all training iterations on CIFAR-100. The results are shown in Table 21. Our method in general present faster runtime without complex design such as contrastive loss.

Table 17: Test accuracy comparison of **mixture of different imprecise labels**. We report results of full labels, partial ratio q of $\{0.1, 0.3, 0.5\}$ for CIFAR-10 and $\{0.01, 0.05, 0.1\}$ for CIFAR-100, and noise ratio η of $\{0.1, 0.2, 0.3\}$ for CIFAR-10 and CIFAR-100.

Dataset	# Labels	Partial Ratio q	Noise Ratio η	0	0.1	0.2	0.3
CIFAR-10	50,000	0.1	PiCO+ [94]	95.99±0.03	93.64	93.13	92.18
			IRNet [71]	-	93.44	92.57	92.38
			DALI [50]	-	94.15	94.04	93.77
			PiCO+ w/ Mixup [50]	-	94.58	94.74	94.43
			DALI w/ Mixup [50]	-	95.83	95.86	95.75
			Ours	**96.55±0.08**	**96.47±0.11**	**96.09±0.20**	**95.83±0.05**
		0.3	PiCO+ [94]	95.73±0.10	92.32	92.22	89.95
			IRNet [71]	-	92.81	92.18	91.35
			DALI [50]	-	93.44	93.25	92.42
			PiCO+ w/ Mixup [50]	-	94.02	94.03	92.94
			DALI w/ Mixup [50]	-	95.52	95.41	94.67
			Ours	**96.52±0.12**	**96.2±0.02**	**95.87±0.14**	**95.22±0.06**
		0.5	PiCO+ [94]	95.33±0.06	91.07	89.68	84.08
			IRNet [71]	-	91.51	90.76	86.19
			DALI [50]	-	92.67	91.83	89.8
			PiCO+ w/ Mixup [50]	-	93.56	92.65	88.21
			DALI w/ Mixup [50]	-	95.19	93.89	92.26
			Ours	**96.28±0.13**	**95.82±0.07**	**95.28±0.08**	**94.35±0.08**
CIFAR-100	50,000	0.01	PiCO+ [94]	76.29±0.42	71.42	70.22	66.14
			IRNet [71]	-	71.17	70.10	68.77
			DALI [50]	-	72.26	71.98	71.04
			PiCO+ w/ Mixup [50]	-	75.04	74.31	71.79
			DALI w/ Mixup [50]	-	76.52	76.55	76.09
			Ours	**78.08±0.26**	**77.53±0.24**	**76.96±0.02**	**76.43±0.27**
		0.05	PiCO+ [94]	76.17±0.18	69.40	66.67	62.24
			IRNet [71]	-	70.73	69.33	68.09
			DALI [50]	-	72.28	71.35	70.05
			PiCO+ w/ Mixup [50]	-	73.06	71.37	67.56
			DALI w/ Mixup [50]	-	76.87	75.23	74.49
			Ours	**76.95±0.46**	**77.07±0.16**	**76.34±0.08**	**75.13±0.63**
		0.1	PiCO+ [94]	75.55±0.21	-	-	-
			IRNet [71]	-	-	-	-
			DALI [50]	-	-	-	-
			PiCO+ w/ Mixup [50]	-	-	-	-
			DALI w/ Mixup [50]	-	-	-	-
			Ours	**76.41±1.02**	**75.50±0.54**	**74.67±0.30**	**73.88±0.60**

Table 18: SSL ablation

	CIFAR100 l=200	STL10 l=40
Ours	22.06	11.09
Ours w/o ent.	22.41	11.23

Table 19: PLL ablation

	CIFAR10 $q = 0.5$	CIFAR100 $q = 0.1$
PiCO	93.58	69.91
Ours	95.91	74.00
PiCO w/o s. a.	91.78	66.43
Ours w/o s. a.	94.53	72.69
Ours w/o ent.	95.87	73.75

Table 20: NLL ablation

	CIFAR10 $\eta = 0.5$	CIFAR100 $\eta = 0.1$
SOP	94.00	63.30
Ours	94.31	66.46
SOP w/o s. a.	66.85	36.60
Ours w/o s. a.	93.56	65.89
SOP w/o ent.	93.04	62.85
Ours w/o ent.	94.16	66.12

Table 21: Runtime Analysis on CIFAR-100

Setting	Algorithm	CIFAR-100 Avg. Runtime (s/iter)
SSL	FreeMatch	0.2157
SSL	Ours	0.1146
PLL	PiCO	0.3249
PLL	Ours	0.2919
NLL	SOP	0.1176
NLL	Ours	0.1021

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We discussed our contribution in introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed our limitation in conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All are stated in Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We present all details in both main paper and Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We use publically available data and code will be released.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All results are obtained with 3 runs using different seeds and error bars are reported.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the standard deviation in the results, which are averaged over 3 independent runs across different experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We showed in experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All behaviors follows NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discussed in conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer:[Yes]

    Justification: We discussed in experiments.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.