Positive-Unlabeled Learning by Latent Group-Aware Meta Disambiguation

Lin Long[1*] Haobo Wang[1*] Zhijie Jiang[1] Lei Feng[2] Chang Yao[1†] Gang Chen[1] Junbo Zhao[1]
[1] Zhejiang University, China [2] Singapore University of Technology and Design, Singapore
{llong, wanghaobo, zjjjj882, changy, cg, j.zhao}@zju.edu.cn, lfengqaq@gmail.com

Abstract

*Positive-Unlabeled (PU) learning aims to train a binary classifier using minimal positive data supplemented by a substantially larger pool of unlabeled data, in the specific absence of explicitly annotated negatives. Despite its straightforward nature as a binary classification task, the currently best-performing PU algorithms still largely lag behind the supervised counterpart. In this work, we identify that the primary bottleneck lies in the difficulty of deriving discriminative representations under unreliable binary supervision with poor semantics, which subsequently hinders the common label disambiguation procedures. To cope with this problem, we propose a novel PU learning framework, namely **La**tent **G**roup-**A**ware **M**eta Disambiguation (**LaGAM**), which incorporates a hierarchical contrastive learning module to extract the underlying grouping semantics within PU data and produce compact representations. As a result, LaGAM enables a more aggressive label disambiguation strategy, where we enhance the robustness of training by iteratively distilling the true labels of unlabeled data directly through meta-learning. Extensive experiments show that LaGAM significantly outperforms the current state-of-the-art methods by an average of **6.8%** accuracy on common benchmarks, approaching the supervised baseline. We also provide comprehensive ablations as well as visualized analysis to verify the effectiveness of our LaGAM. The code is available at* https://github.com/llong-cs/LaGAM.

1. Introduction

The remarkable success of deep learning can be largely attributed to the availability of comprehensively annotated data which provides valid supervision. However, the necessity for high-quality labeled data is not always feasible in many scenarios [21, 39, 50]. A real-world case lies in the medical field [50], where the application of deep learning for diagnosing certain chronic diseases can be challenging due to the scarcity of diagnosed (positive) samples available for training. While data from a larger population of undiagnosed individuals are more accessible, they are all "unlabeled" for possibly being either diseased or healthy. Such label ambiguity will hamper effective model training.

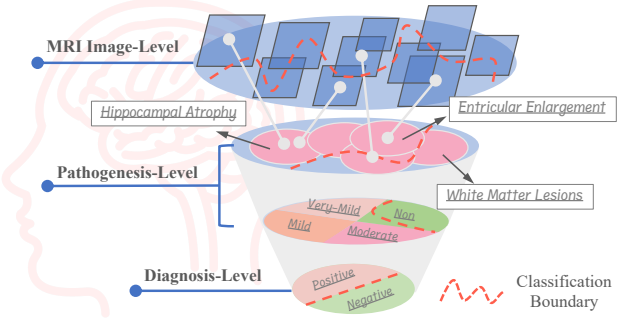


Figure 1. Illustration for hierarchical Alzheimer's Disease diagnosis with brain MRI. Given an MRI image (1st Level), firstly we need to identify any suspicious structural changes and match them with typical lesion characteristics of AD. (2nd Level). The final diagnosis (4th Level) will then be given based on a comprehensive assessment of the number and severity of symptoms (3rd Level).

Addressing this issue, **P**ositive-**U**nlabeled (PU) learning has been increasingly studied in recent years [2]. PU learning refers to a specific binary classification task where only a limited amount of positive samples are explicitly labeled but all other instances are unlabeled[30, 32]. A popular strategy for PU learning is to cast it as a cost-sensitive classification task through importance reweighting [12, 13, 26, 43]. For instance, Self-PU [10] pioneers meta-learning to diminish the influence of false negatives. Despite the promising results, samples being "filtered out" are actually not being effectively utilized for model training. Another set of works tries to select reliable negative (or positive) samples from the unlabeled set to construct a pseudo-binary labeled set [10, 29, 32, 43]. However, the selection results are not always precise, which may lead to confirmation bias or accumulated error [18, 49, 51].

While most previous studies target at straight label dis-

*Joint first authors.
†Corresponding author.

ambiguation, there always seems to be a dilemma where model generalizability and supervision intensity cannot be simultaneously ensured, making PU learning still largely lag behind the fully supervised counterpart nowadays. Carefully looking into this trade-off, we spot that the primary bottleneck actually lies in the difficulty of developing discriminative representations under the constraints of unreliable binary labels. In other words, *insufficient supervision is provided by the ambiguous yet semantically poor binary labels*. Meanwhile, we observe that in real-world PU scenarios, the binary classes generally originate from finer-grained categories. For example, as Figure 1 shows, the diagnosis of Alzheimer's Disease (AD) can be idealized as an inductive process, where it is the underlying pathogenesis that provides a more concrete representation of a patient.

Following this intuition, we propose a novel PU learning framework, namely **La**tent **G**roup-**A**ware **M**eta Disambiguation (**LaGAM**), to produce compact representations which encapsulate the intrinsic group semantics of PU data. To achieve this goal, we introduce a hierarchical contrastive learning module based on three guidelines: (i)-alignment with unsupervised prototypes to achieve latent group awareness; (ii)-refinement with dichotomized cut-off for enhanced binary prediction; (iii)-unification between local neighbors for further smoothness. Visual representations in Figure 5 validate that LaGAM does evolve the ability to distinguish valid latent categories. Moreover, based on the reliable representations, LaGAM integrates a meta-disambiguation objective, with which we iteratively refine the pseudo-labels of unlabeled samples, enabling the training of a robust binary classifier. Extensive experiments on four benchmark datasets show that LaGAM surpasses the current state-of-the-art by an average of $\mathbf{6.8}\%$ accuracy. Particularly for AD recognition 2, LaGAM has a significant advantage of $\mathbf{6.1}\%$, indicating its practicability.

Our main contributions can be summarized as follows:

- **(Insights)** This study represents a pioneering effort in the semantic analysis of PU data within real-world contexts, addressing the bottleneck of representation learning using unreliable binary labels with a lack of semantics.
- **(Methodology)** We propose LaGAM, leveraging a group-aware contrastive learning objective to explore the latent categories beyond binary labels and meta-learning to refine the labels of unlabeled data for label disambiguation.
- **(Experiments)** We conduct extensive experiment to show that our LaGAM outperforms the current state-of-the-art, approaching the supervised learning counterpart.

2. Related Work

Positive-Unlabeled (PU) Learning. The existing PU learning algorithms primarily fall into two categories. The current mainstream adopts the framework of cost-sensitive learning, the key of which is decomposing the naive binary classification risk into individual risks over positive and unlabeled data respectively, to correct the bias derived from negativity estimation. uPU [12, 13] first proposes the unbiased risk estimator of PU learning, with follow-up works like nnPU [26], ImbPU [42] and Self-PU [10] trying to improve this technique from different aspects. Notably, most of these methods are established on the Selected Completely at Random (SCAR) assumption [15] and greatly rely on the knowledge of class prior, which rarely hold in practice. To mitigate such limitations, works like PUSB [24], bPU [3, 24, 41] and aPU [17] manage to relax the prerequisites to more general settings. Correspondingly, there also emerges some class prior estimation algorithms, including PE [14], Pen-L1 [14], KM1, KM2 [37] and TIcE [1].

The second type of PU learning algorithm is based on sample selection[32, 52]. As the name suggests, the basic idea of this type of method is to select reliable instances from the unlabeled set, followed by traditional (semi-)supervised algorithms. The early studies mainly focus on exploiting various heuristic strategies for negative sample selection, such as 1-DNF [35, 51, 52], Naive Bayes [32], Rocchio extraction [29], kNN [53] and k-means [7]. Recent studies like PUbN [22], GenPU [20], PULNS [33] and PAN [23] leverage auxiliary models for sample identification or generation. Meanwhile, VPU [23], MixPUL [47] and P^{3}MIX [28] adopt mixup technique to refine the imprecise supervision.

Contrastive Learning. As a promising paradigm of self-supervised learning, contrastive learning (CL) [19, 45] has been frequently used for learning discriminative representation with the absence of manually annotated data [9, 19, 45]. The essence of contrastive learning is to find a feature space wherein positive pairs are drawn closer together while negative pairs are pushed apart. To this end, most of the existing algorithms differ in the specific ways to construct this sort of data pairs. One commonly used pretext task treats each instance as a class and constructs positive pairs through different data augmentations [9, 19, 48], which is also known as Instance Discrimination (ID). There are also methods using the clustering results to guide the pair construction [5, 31], which proves to be especially effective for latent group recognition.

Meta-Learning. Meta-learning [16, 34], i.e. learning to learn better, has recently emerged as an effective framework for weakly-supervised tasks including noisy label disambiguation [10, 38, 40] and semi-supervised learning [36]. The existing methods generally contain two loops of learning: an inner loop, where virtual training on meta-model occurs, and an outer loop, which optimizes the evaluation loss w.r.t. meta-parameters. Most of these methods rely on a golden support set for evaluating the meta-model, and to

correct the biased training labels by adjusting their importance weights [38, 40].

3. Notations and Preliminaries

Positive-Unlabeled Learing. Let $\boldsymbol{x} \in \mathbb{R}^d$ and y be the input feature and binary label, respectively. Following [28], we specify the positive instances with $y=1$ and the negative ones with $y=0$. Under PU setting, the training dataset $\mathcal{D}_{\text{train}}$ is composed of a positive set $\mathcal{P}=\{(\boldsymbol{x}_i, y_i=1)\}_{i=1}^{n_p}$, and an unlabeled set $\mathcal{U}=\{(\boldsymbol{x}_i)\}_{i=n_p+1}^{n_p+n_u}$ with pseudo-labels $\tilde{\boldsymbol{y}}$, where n_p and n_u refer to the number of positive and unlabeled samples, respectively. The target of PU learning is to train a binary classifier $f(\boldsymbol{x}): \mathbb{R}^d \mapsto [0,1]$ parameterized by θ, with training set $\mathcal{D}_{\text{train}}=\mathcal{P}\cup\mathcal{U}$, where the loss can be denoted as $\mathcal{L}_{\text{cls}}(\mathcal{D},\theta)=\frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{x}_i,y_i)\in\mathcal{D}}\ell(f_\theta(\boldsymbol{x}_i),y_i)$. $\ell(\cdot)$ can be arbitrary loss like Binary Cross-Entropy (BCE):

$$\ell(f_\theta(\boldsymbol{x}_i),y_i)=-[y_i\log f_\theta(\boldsymbol{x}_i)+(1-y_i)\log(1-f_\theta(\boldsymbol{x}_i))]. \tag{1}$$

Contrastive Learning. In this paper, we resemble [19, 25] to construct a basic framework of contrastive learning, where the per-sample contrastive loss can be defined as:

$$\ell_{\text{cont}}(\boldsymbol{x},\mathcal{P})=-\frac{1}{|\mathcal{P}(\boldsymbol{x})|}\sum_{\boldsymbol{k}_+\in\mathcal{P}(\boldsymbol{x})}\log\frac{e^{\boldsymbol{q}\cdot\boldsymbol{k}_+/\tau}}{\sum_{\boldsymbol{k}_i\in A\setminus\{\boldsymbol{q}\}}e^{\boldsymbol{q}\cdot\boldsymbol{k}_i/\tau}}, \tag{2}$$

where $\boldsymbol{q}$ is the embedding of $\boldsymbol{x}$ and τ is the temperature. A refers to a set that contains all contrastive embeddings. The essence of contrastive learning is to align the representations of similar samples as defined by *positive set* $\mathcal{P}(\boldsymbol{x})$, and to separate the dissimilar ones. In addition to Instance Discrimination (ID) [9, 19, 48], where we let $P(\boldsymbol{x})$ consist of the embeddings from different augmentations of $\boldsymbol{x}$, there are many other ways to construct the positive set. Each of them offers a unique definition of similarity, allowing the model to learn different semantics via contrastive learning.

4. Method

To tackle the PU problem, LaGAM encapsulates (i)-a hierarchical contrastive learning module to improve the representation quality, and (ii)-a meta-disambiguation objective to refine the noisy labels. Next, we will describe these two components in detail.

4.1. Latent Group-Aware Representation Learning

As mentioned earlier, data representation plays a critical part in PU learning, the quality of which directly determines the effect of upper-level label disambiguation. Based on our observation, the binary class in PU learning generally encompasses finer-grained sub-categories, where there exists some mappings between the underlying groups and the highly abstracted binary labels. Motivated by this, we resort to a *hierarchical contrastive learning* objective, which actively explores the latent groups that reflect the essential semantics of PU data. Though contrastive learning has been extensively studied for representation learning in recent literature, the PU setting posits unique **challenges** for adapting this technique, which mainly lies in two aspects.

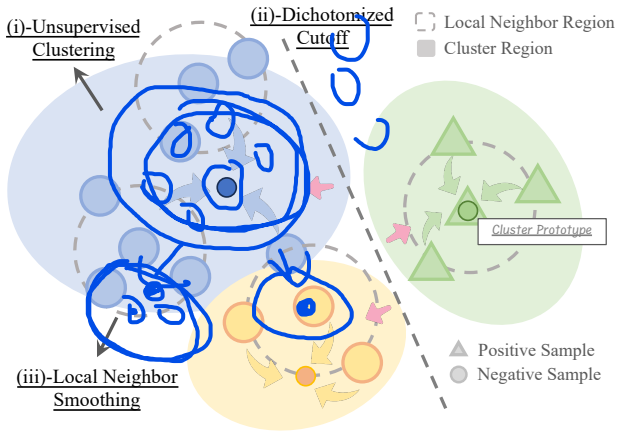


Figure 2. Overview of the hierarchical contrastive learning framework. The arrows indicate the samples' moving direction in the latent space guided by $\mathcal{L}_{\text{cont}}$, where samples are: (i)-converged towards the cluster centers; (ii)-separated from the binary boundary; and (iii)-aggregated within the local neighbor regions.

Challenge 1: ***How to find the latent groups?*** Without explicit labels, constructing a positive set that can guide the model to distinguish samples from different latent groups is not as straightforward. To this, we generate pseudo-positive pairs based on unsupervised clustering. Specifically, we apply k-means [7] clustering in the embedding space on $\mathcal{D}_{\text{train}}$ at the beginning of each epoch, yielding k cluster centers:

$$C(\boldsymbol{x}): \mathbb{R}^d \mapsto \{1,2,\ldots,k\}, \tag{3}$$

where $C(\boldsymbol{x})$ maps an input $\boldsymbol{x}$ to one of the k cluster centers. We use $\mathcal{S}_{\text{cluster}}(\boldsymbol{x})$ to represent a set that consists of the embeddings of all samples belonging to the same cluster center as $\boldsymbol{x}$. In this way, we artificially define a series of latent categories and thus can construct a positive set that encourages closely aligned representations of samples from the same latent group. In other words, we facilitate the spontaneous organization of data according to latent groups by manipulating features, which leads to better separability between those (potentially) semantically distinct data in the embedding space.

Challenge 2: ***How do the latent groups enhance the binary prediction?*** Considering that unsupervised clustering might yield some inaccurate results that will mislead the model's learning, regular alignment between the latent categories and the given binary labels is necessary. There-

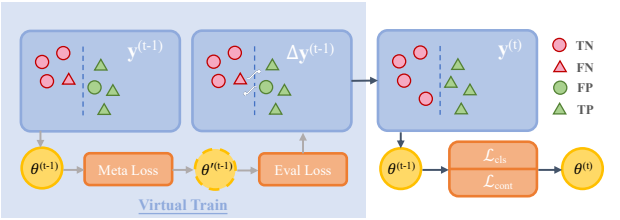


Figure 3. Illustration of the meta-disambiguation process.

fore, we cut off those mis-clustered pairs according to the dichotomized outputs of the classifier, restricting the latent categories to be semantically binarizable. Specifically, two samples from the same cluster are considered as a positive pair only if they carry the same predicted label. Similarly, we use $\mathcal{S}_{\text{binary}}(\boldsymbol{x})$ to represent a set that consists of the embeddings of all samples that have the same label as $\boldsymbol{x}$. With this extra constraint, we effectively mitigate the risk of confirmation bias caused by incorrect clustering.

Furthermore, we also adopt data-driven neighbor augmentation based on kNN[53], which helps to distribute the representations more smoothly within local neighborhoods. The Latent Group-Aware (LGA) positive set can thus be:

$$\mathcal{P}_{\text{LGA}}(\boldsymbol{x}) = (\mathcal{S}_{\text{cluster}}(\boldsymbol{x}) \cap \mathcal{S}_{\text{binary}}(\boldsymbol{x})) \cup \mathcal{S}_{\text{neighbor}}(\boldsymbol{x}), \quad (4)$$

where $\mathcal{S}_{\text{neighbor}}(\boldsymbol{x})$ refers to a set consists of the k nearest neighbors of $\boldsymbol{x}$. The overall contrastive objective that combines the losses of Instance Discrimination (ID) and Latent Group-Awareness can thus be formulated as:

$$\mathcal{L}_{\text{cont}}(\mathcal{D},\theta) = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x_i},y_i)\in\mathcal{D}} [\ell_{\text{cont}}(\boldsymbol{x_i},\mathcal{P}_{\text{ID}}) + \ell_{\text{cont}}(\boldsymbol{x_i},\mathcal{P}_{\text{LGA}})]. \quad (5)$$

4.2. Meta Label Disambiguation

Recalling the trade-off arising between model generalizability and supervision intensity, though meta-learning has already demonstrated powerful capabilities in sample-wise tuning [10] (with the cost of only a small golden support set consisting of a few labeled samples), it still relies on importance reweighting to achie label disambiguation, where the sample utilization rate is sacrificed. However, the enhanced representations attained through our proposed hierarchical contrastive learning enable the model to better comprehend the relationships between PU data and their binary labels, which may allow more aggressive label disambiguation with *direct label refinement*.

Meta Objective for Disambiguation. Before label disambiguation, we first set $\tilde{\boldsymbol{y}}$ to $\mathbf{0}$ by default, i.e., assuming all unlabeled data are negative. In contrast to Self-PU[10] which employs meta-learning for sample reweighting, we would like to directly distill the true labels of unlabeled data through meta-learning. Formally, we define the following objective function, using a bi-level optimization structure of meta-learning:

$$\tilde{\boldsymbol{y}}^* = \arg\min_{\tilde{\boldsymbol{y}}} \mathcal{L}_{\text{cls}}(\mathcal{D}_{\text{sup}}, \theta(\tilde{\boldsymbol{y}})) \quad (6)$$

$$\text{s.t. } \theta(\tilde{\boldsymbol{y}}) = \arg\min_{\theta} \mathcal{L}_{\text{cls}}(\mathcal{D}_{\text{train}}, \theta). \quad (7)$$

That is, in the inner loop (7), we first acquire model parameters $\theta(\tilde{\boldsymbol{y}})$ that minimize the supervised loss over training data pseudo-labeled with $\tilde{\boldsymbol{y}}$, which will then be evaluated on the support set $\mathcal{D}_{\text{sup}}$. To this, in the outer loop (6) we minimize the evaluation loss w.r.t. the pseudo-labels $\tilde{\boldsymbol{y}}$. Intuitively, the closer $\tilde{\boldsymbol{y}}$ aligns with the ground-truth, the less label ambiguity there is, and thus the model trained with it is supposed to exhibit better generalizability on the golden support set. In this way, label disambiguation can be transformed into an optimization problem of minimizing the evaluation loss to obtain $\tilde{\boldsymbol{y}}^*$. The corresponding classifier trained with $\tilde{\boldsymbol{y}}^*$ will be our target model.

Online Disambiguation. However, from Eq. (6)(7) we can observe that, the calculation of $\tilde{\boldsymbol{y}}^*$ requires two nested loops of optimization, each of which can be very expensive. Therefore, inspired by [10, 38, 40], we iteratively adapt an online $\tilde{\boldsymbol{y}}$ and alternate minimization between the inner and outer loop, gradually converging towards the ground-truth labels. Specifically, the online optimization comprises two main steps: **Label-Update** and **Actual-Training**.

At the stage of **Label-Update**, we try to align $\tilde{\boldsymbol{y}}$ with the label distribution of $\mathcal{D}_{\text{sup}}$, which is considered to be clean and unbiased. Specifically, consider the t^{th} iteration, we first perform a one-step "virtual" update on the model parameters with the given training mini-batch $\mathcal{B}_{\text{train}}$:

$$\theta'^{(t)}(\tilde{\boldsymbol{y}}) = \theta^{(t)} - \lambda \nabla_\theta \mathcal{L}_{\text{cls}}(\mathcal{B}_{\text{train}}, \theta)|_{\theta=\theta^{(t)}}, \quad (8)$$

where λ is the learning rate. $\theta'^{(t)}(\tilde{\boldsymbol{y}})$ can be considered as a reasonable approximation for the optimal model parameter obtained with $\tilde{\boldsymbol{y}}$. Being "virtually" trained with $\tilde{\boldsymbol{y}}$, the generalizability of the meta-model $f_{\theta'^{(t)}}$ can also reflect the correctness of the current $\tilde{\boldsymbol{y}}$ to some extent.

Therefore, we then calculate the gradients of the evaluation loss of $f_{\theta'^{(t)}}$ on the support set $\mathcal{D}_{\text{sup}}$, w.r.t. $\tilde{\boldsymbol{y}}$, which indicates the direction of label update towards higher model generalizability and thus better $\tilde{\boldsymbol{y}}$:

$$\delta^{(t)} = -\nabla_{\tilde{\boldsymbol{y}}} \mathcal{L}_{\text{cls}}(\mathcal{D}_{\text{sup}}, \theta'^{(t)}(\tilde{\boldsymbol{y}}))\Big|_{\tilde{\boldsymbol{y}}=\tilde{\boldsymbol{y}}^{(t)}}, \quad (9)$$

where $\tilde{\boldsymbol{y}}^{(t)}$ refers to the values of pseudo-labels at time step t. However, we empirically found that direct gradient descent yields poor performance possibly due to the bias from

negativity estimation [43]. To mitigate this issue, we instead use the projected gradients together with the Exponential Moving Average (EMA) for label updating:

$$s_i^{(t)} = \begin{cases} 1, & \delta_i^{(t)} \geq 0, \\ 0, & \delta_i^{(t)} < 0, \end{cases} \tag{10}$$

$$\tilde{\boldsymbol{y}}^{(t+1)} = \epsilon \tilde{\boldsymbol{y}}^{(t)} + (1-\epsilon) s^{(t)}, \tag{11}$$

where $\epsilon \in [0, 1)$ is the EMA parameter for controlling the evolving speed of pseudo-labels $\tilde{\boldsymbol{y}}$, to ensure label consistency among different time steps. Simply speaking, we first map each element in $\delta^{(t)}$ to $\{0, 1\}$, through which we drop the magnitude information of the gradient, retaining only the direction information. This together with the introduction of the EMA updating strategy significantly increases the model's tolerance for bias and thus enhances its robustness. Particularly, when ϵ is set to 0, a replacement strategy is adopted rather than momentum stepping, which provides stronger yet less consistent supervision signals. We further study the effect of different updating strategies in Section 5.3. With each update, $\tilde{\boldsymbol{y}}$ is getting more and more accurately annotated, i.e., towards the direction that leads to better generalizability, and thus can be better utilized for subsequent training of a robust binary classifier. For better understanding, we elaborate our theoretical insights of this debiased learning procedure from an influence function perspective in Appendix B.

Next, for the **Actual-Training** of the classifier, we optimize θ to fit the pseudo-labels $\tilde{\boldsymbol{y}}^{(t+1)}$ using a binary classification loss $\mathcal{L}_{\text{cls}}$ coupled with the contrastive loss $\mathcal{L}_{\text{cont}}$:

$$\theta^{(t+1)} = \theta^{(t)} - \lambda \nabla_\theta [\mathcal{L}_{\text{cls}}(\mathcal{B}_{\text{train}}, \theta) + \beta \mathcal{L}_{\text{cont}}(\mathcal{B}_{\text{train}}, \theta)]|_{\theta=\theta^{(t)}}, \tag{12}$$

where β is a weighting parameter.

Alternatively updating $\tilde{\boldsymbol{y}}$ and θ, we iteratively approach the optimal solution together with the distillation of $\tilde{\boldsymbol{y}}$. The detailed procedure can be referred to in the Appendix.

5. Experiment

5.1. Experimental Settings

Datasets. In the experiments, we employ two prevalent benchmark datasets, CIFAR-10 [27] and STL-10 [11]. Additionally, we extend the setup in [8, 26] to CIFAR-100 [27], which contains more sub-categories, for evaluating the model's performance on data with more complex distributions. For each dataset, we divide its category labels (which range from 0 to 9 for CIFAR-10 and STL-10, and 0 to 19 for the superclasses of CIFAR-100) into two disjoint subsets of comparable sizes, and generate two synthetic PN datasets by specifying one of them as positive set. Particularly for CIFAR-100, we instead use two different partitions: one is balanced and the other is imbalanced, to examine the performance of LaGAM under different class distributions. Following [28], for each dataset, we uniformly select 1000 positive instances from the training set to be labeled, and split 500 instances from the validation set to form the support set. To show that LaGAM not only works on the contrived datasets, we also evaluate its performance with the real-world Alzheimer dataset [1] for Alzheimer's Disease diagnosis [56], which doesn't contain explicitly given latent categories. Detailed statistics are given in Appendix C.

Baselines. To verify the effectiveness of LaGAM, we utilize 8 PU learning baselines, including uPU [12, 13], nnPU [26], Self-PU [10], PAN [23], VPU [23], Dist-PU [56], and two versions of P^{3}MIX [28], together with the supervised counterpart for comparison. For all baselines, we unify the backbone used for each dataset. Specifically, we adopt 7-layer CNN for CIFAR-10 and STL-10, and 13-layer CNN and ResNet-50 for more complex CIFAR-100 and Alzheimer, respectively [56]. Furthermore, to show the superiority of LaGAM in terms of learning capacity brought by the representation learning module, we also conduct experiments replacing the backbone with deeper ResNet-18 and comparing our LaGAM with the current SOTA Dist-PU and P^{3}MIX-C under the same setup. Note that the cost-sensitive based methods including uPU[13], nnPU[26] and Self-PU[10], require prior knowledge of class proportion, which is however unknown for STL-10 wherein many training samples are naturally unlabeled. Addressing this problem, following the work of P^{3}MIX [28], we use the current SOTA KM2 [37] to estimate the class proportion of STL-10.

Implementation Details. Considering that the effect of meta-disambiguation largely relies on the quality of representations, the early introduction of meta-learning when the representation space is not yet in shape may lead to poor performance. Therefore, we disable meta-disambiguation and warm up the model with Eq. (12) for the first 20 epochs. Empirically, we find that warm-up significantly improves the model's performance and the convergence rate. Moreover, to accelerate the calculation of second-order gradients involved in Eq. (6)(7), all layers of the classifier except for the classification head are frozen during meta-learning. Surprisingly, we found that not only does it make the training process significantly faster, but it also leads to higher converged accuracy, possibly due to the mitigation of the negative interference of meta-disambiguation on representation learning. Besides, we linearly ramp down the EMA parameter ϵ from 0.95 to 0.8, to allow the label updating to be more conservative at the beginning, and switch to a larger step size as the training trajectory gradually stabilizes. We

[1] https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images

Backbone	Method	CIFAR-10-1	CIFAR-10-2	CIFAR-100-1	CIFAR-100-2	STL-10-1	STL-10-2
CNN	uPU	76.5±2.5	71.6±1.4	90.2±0.2	64.2±1.7	76.7±3.8	78.2±4.1
	nnPU	84.7±2.4	83.7±0.6	63.2±1.3	68.1±2.1	77.1±4.5	80.4±2.7
	Self-PU	85.1±0.8	83.9±2.6	87.1±2.9	68.4±1.4	78.5±1.1	80.8±2.1
	PAN	87.0±0.3	82.8±1.0	75.6±1.8	66.6±1.4	77.7±2.5	79.8±1.4
	VPU	86.8±1.2	82.5±1.1	90.1±0.1	50.0±0.1	78.4±1.1	82.9±0.7
	Dist-PU	86.8±0.7	87.2±0.9	69.2±2.4	72.9±0.6	79.8±0.6	82.9±0.4
	P^{3}MIX-E	88.2±0.4	84.7±0.5	87.3±1.3	54.9±1.6	80.2±0.9	83.7±0.7
	P^{3}MIX-C	88.7±0.4	87.9±0.5	88.1±0.9	52.9±1.2	80.7±0.7	**84.1±0.3**
	LaGAM (ours)	**89.6±0.4**	**90.6±0.8**	**92.6±0.7**	**85.9±0.1**	**87.5±0.3**	81.9±1.5
	Supervised	91.3±0.3	91.3±0.3	93.5±0.5	91.9±0.4	-	-
ResNet	Dist-PU	88.8±0.8	88.9±0.7	65.8±2.3	69.1±0.7	81.7±1.6	83.4±1.5
	P^{3}MIX-C	76.1±0.6	74.9±0.6	88.5±1.3	51.4±1.1	71.1±0.9	72.3±1.1
	LaGAM (ours)	**96.2±0.5**	**96.1±0.3**	**92.1±0.4**	**86.6±0.3**	**88.5±0.5**	**88.1±0.7**
	Supervised	98.7±0.5	98.7±0.5	94.2±0.4	92.8±0.7	-	-

Table 1. Results of classification accuracy (mean±std). CIFAR-100-1 and CIFAR-100-2 refer to imbalanced and balanced partitions, respectively. **Best** performance on each setup is highlighted. Supervised learning is unavailable for STL-10, the training data of which are mostly unlabeled.

Method	Accuracy	F1 Score	AUC
uPU	68.5±2.2	67.6±2.8	73.8±2.9
nnPU	68.3±2.1	68.6±3.2	72.9±2.8
Self-PU	70.9±0.7	72.1±1.1	75.9±1.8
VPU	67.4±0.7	70.2±1.1	73.1±0.9
Dist-PU	71.7±0.6	73.7±1.6	77.1±0.7
LaGAM (ours)	**77.8±2.8**	**84.5±2.6**	**83.2±1.7**

Table 2. Comparative results on Alzheimer (mean±std).

also apply the mixup technique [4, 8, 47, 54] to improve the robustness of the classification loss $\mathcal{L}_{\text{cls}}$ used for model training (12) [6, 44, 55].

5.2. Main Results

For each dataset, we independently run each method 5 times with random seeds and report the average classification accuracy together with the standard deviation as shown in Table 1, noting that part of the results are adopted from [28, 46]. For the Alzheimer dataset, we additionally report the metrics of F1 score and AUC as shown in Table 2, which are more practically significant.

Overall Performance. With a shallow CNN backbone, our proposed LaGAM outperforms most of the baselines across 7 setups, indicating its superiority. Compared with cost-sensitive methods like uPU [13], nnPU [26], Self-PU [10] and Dist-PU [56], LaGAM holds a significant advantage of average $\mathbf{9.5}\%$ accuracy without requiring class prior nor SCAR assumption, demonstrating its universality in real-world practice. Benefiting from effective representation learning, a dominant advantage of up to $\mathbf{13.0}\%$ can be observed from LaGAM in more challenging tasks of CIFAR-100, where the latent categories are more finely subdivided. For CIFAR-100-1, where the ratio between positive and negative categories is $2:18$, the classifier only needs to recognize the features of those specific two categories or can even get a 90% accuracy with the trivial solution of assigning all inputs to the same class. However, in CIFAR-100-2, with the ratio being $10:10$, the classifier needs to identify features of at least ten categories to achieve good performance, where common methods are generally challenged while LaGAM maintains stably outstanding performance. On the Alzheimer dataset, LaGAM still holds an absolute advantage over all three metrics, which confirms its practicability.

Learning Capacity. Furthermore, to further investigate the difference in learning capacity brought by the well-designed representation learning framework, we replace the backbone model with deeper ResNet-18 and compare the performance of current SOTA Dist-PU [56] and P^{3}MIX-C [28] with LaGAM. The results show that the classification accuracy of LaGAM gets significantly improved by $\mathbf{0.8}\%$ to $\mathbf{6.2}\%$ after applying a deeper network. Dist-PU also shows a slight improvement while severe model degradation occurs in P^{3}MIX-C probably due to the mismatch between the learning capacity and model complex-

Ablation	Meta	$\mathcal{L}_{\text{cont}}$	CIFAR-10	CIFAR-100
LaGAM	✓	✓	**96.6**	**89.8**
nnPU w/ $\mathcal{L}_{\text{cont}}$	✗	✓	91.2	87.8
Only Meta	✓	✗	87.6	73.5
Naive BCE	✗	✗	60.0	50.0

Table 3. Ablation study on key components with ✓ indicating the enabling of meta-disambiguation or $\mathcal{L}_{\text{cont}}$.

ity given rough data. It indicates that group-aware representation learning together with stable supervision signals obtained from meta-disambiguation enables PU learning to better adapt to deeper network architectures in general cases, thereby fully leveraging the learning capacity of deep neural networks. Detailed investigation can be found in Appendix C.

5.3. Ablation Studies

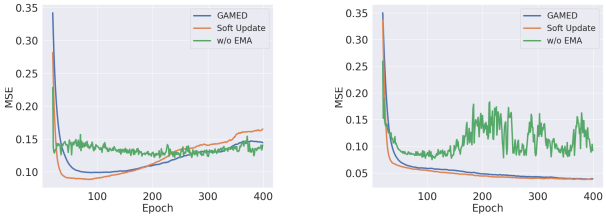
In this section, we present the ablation results to show the effectiveness as well as reasonability of our LaGAM.

Effect of $\mathcal{L}_{\text{cont}}$ and Meta Disambiguation. We ablate the contributions of the two key components of LaGAM: group-aware contrastive learning and meta label disambiguation. In particular, we compare LaGAM with three types of different variants: 1) *nnPU w/ $\mathcal{L}_{cont}$*, where $\mathcal{L}_{\text{cont}}$ is jointly used with the non-negative unbiased risk estimator [26] but without meta disambiguation; 2) *LaGAM w/o $\mathcal{L}_{cont}$*, where the target classifier is completely trained on a BCE loss with pseudo-labels obtained through meta disambiguation; and 3) *Naive BCE*, where the classifier is trained on a BCE loss assuming that all the unlabeled data are negative. From Tabel 3, we can observe that $\mathcal{L}_{\text{cont}}$ significantly boost the performance of nnPU compared with the results in Table 1, indicating that group-aware contrastive learning possesses a certain universality, being able to integrate well with some other methods by enhancing their performance in representation learning. On the other hand, the gap between LaGAM and variant 1 also suggests that due to the involvement of dichotomized cut-off, $\mathcal{L}_{\text{cont}}$ is able to achieve the best results if the classifier provides valid guidance that can effectively restrict its clustering direction. Besides, the significant difference between the performance of variant 2 and variant 3 demonstrates that meta-disambiguation can indeed correct the misestimated labels.

Effect of Different Label Updating Strategies. Since there are various ways to perform label updating in meta-learning, we also conduct ablation experiments with two variants of LaGAM on CIFAR-10, adopting different updating strategies: 1) *soft update* using normalized gradients and 2) pseudo-label update *w/o EMA*. Interestingly, from Table 4 we can see that though EMA update with projected gradients which is used in LaGAM has the worst performance under the circumstance without $\mathcal{L}_{\text{cont}}$, while the comparative results completely reverse once $\mathcal{L}_{\text{cont}}$ is involved. We hypothesize that variant 2 provides the strongest binary classification supervision signals which can mitigate overfitting on estimation bias without $\mathcal{L}_{\text{cont}}$, as shown in Figure 4. However, when paired with $\mathcal{L}_{\text{cont}}$, the intensity of the supervision signal may not be the priority, instead, stability and consistency take precedence, while variant 2 exhibits significant oscillation during the label updating process.

Ablation	Gradient	EMA	w/o CL	w/ CL
LaGAM	Projected	✓	87.6	**96.6**
Soft Update	Normalized	✓	88.4	96.1
w/o EMA	Projected	✗	**91.2**	95.3

Table 4. Ablation study on label updating strategies.



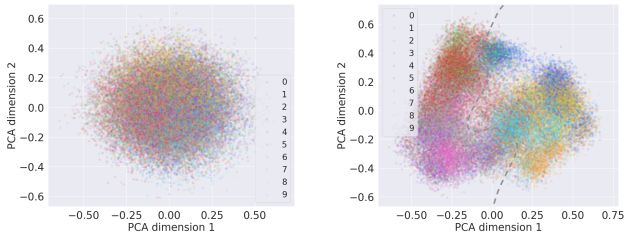
(a) w/o CL (b) w/ CL

Figure 4. The variation in mean square error (MSE) between pseudo-labels and ground-truths.

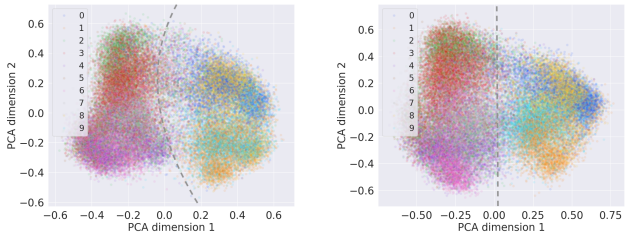
Ablation	ID	UC	DC	NS	Acc.
LaGAM	✓	✓	✓	✓	**96.6**
w/o Neighor Smoothing	✓	✓	✓	✗	96.3
w/o NS + Dichotomized Cutoff	✓	✓	✗	✗	96.1
Only Instance Discrimination	✓	✗	✗	✗	91.8

Table 5. Ablation study on positive set constructions in $\mathcal{L}_{\text{cont}}$.

Effect of Different Constructions of Positive Set. To better understand the impact of each criterion used for the construction of positive set in the hierarchical contrastive learning, we compare the model accuracy on CIFAR-10 after removing each criterion in Table 5 and visualize the corresponding representation distributions respectively. In addition to the increase in classification accuracy, we can also observe from Figure 5 that with each added criterion, the distinguishability of the sample representations distribution

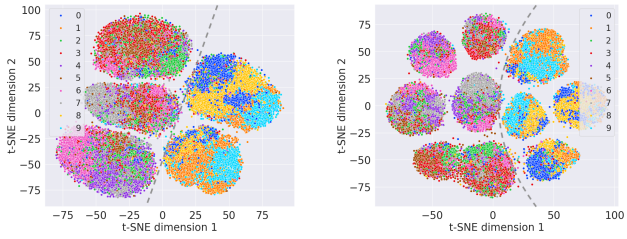


(a) Only Instance Discrimination (b) w/o NS + Dichotomized Cutoff

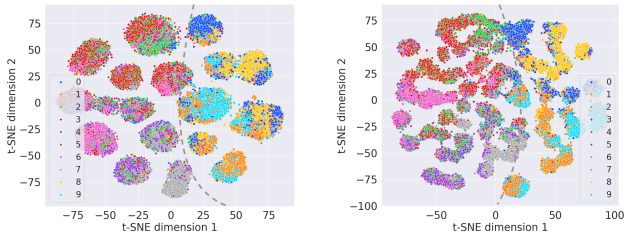


(c) w/o Neighor Smoothing (d) LaGAM

Figure 5. Representation layout obtained on CIFAR-10 using different contrastive learning frameworks. We use PCA since it produces a clearer decision boundary.



(a) # Cluster=5 (b) # Cluster=10



(c) # Cluster=20 (d) # Cluster=100

Figure 6. Representation layout obtained on CIFAR-10 using different cluster numbers. We use t-SNE since it produces clearer data clusters.

Cluster Number	CIFAR-10	CIFAR-100
5	96.4	88.4
10	**97.0**	89.8
20	96.6	**90.2**
100	96.6	89.8

Table 6. Sensitivity analysis on cluster number (default 100).

is significantly enhanced. Specifically, with **U**nsupervised **C**lustering (**UC**), the samples begin to distribute in clusters as the colors of different categories start to show up. With **D**ichotomized **C**utoff (**DC**), we can clearly see that the clusters begin to separate towards opposite sides with a gap emerging between them. With local **N**eighbor **S**moothing (**NS**), the clusters are more distinctly separated as the colors of different regions are more uniform.

5.4. Sensitivity Analysis

To verify the feasibility of LaGAM, we also conduct an experiment on the effect of specifying different cluster numbers in the k-means algorithm (3), to show that LaGAM is competitive without requiring any prior knowledge about the number of latent categories. Intuitively, as shown in Table 6, the classifier has the best performance when the number of true latent categories (which is 10 and 20 for CIFAR-10 and CIFAR-100, respectively) equals the one manually set. But more importantly, as the cluster number continues to increase, the decrease in classification accuracy is relatively minor, which implies that we can choose a necessarily large cluster number rather than the exact number of latent categories, without losing utility. To understand the underlying reasons, from Figure 6 we can observe that when the cluster number has far exceeded the number of true latent categories, different clusters belonging to the same latent category still tend to cluster together, and thus will not interfere with the model's classification boundary.

6. Conclusion

In this paper, we study the challenge of representation learning caused by the lack of semantics under PU setting, and manage to tackle it with a group-aware contrastive learning objective that extracts the underlying features consistent with the natural distribution of PU data. Based on this key idea, we propose a novel PU learning framework, namely LaGAM, wherein we also make an aggressive attempt at label disambiguation strategy, directly distilling the labels of unlabeled data through meta-learning. Empirically, we conduct extensive experiments and show that LaGAM establishes state-of-the-art performance and even approaches the supervised counterpart. Visualized results also prove that LaGAM can indeed learn effective representations that are well aligned with the real semantics. We hope our work will open new avenues for the community to explore PU learning from the perspective of representation learning.

Acknowledgements

This work is supported by the Pioneer R&D Program of Zhejiang (No. 2024C01035). Chang Yao is supported by the Key Research and Development Program of Zhejiang Province (No. 2023C03192).

References

[1] Jessa Bekker and Jesse Davis. Estimating the class prior in positive and unlabeled data through decision tree induction. In *AAAI*, pages 2712–2719. AAAI Press, 2018. 2

[2] Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: a survey. *Mach. Learn.*, 109(4):719–760, 2020. 1

[3] Jessa Bekker, Pieter Robberechts, and Jesse Davis. Beyond the selected completely at random assumption for learning from positive and unlabeled data. In *ECML*, pages 71–85. Springer, 2019. 2

[4] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, pages 5050–5060, 2019. 6

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2

[6] Luigi Carratino, Moustapha Cissé, Rodolphe Jenatton, and Jean-Philippe Vert. On mixup regularization. *J. Mach. Learn. Res.*, 23:325:1–325:31, 2022. 6

[7] Sneha Chaudhari and Shirish K. Shevade. Learning from positive and unlabelled examples using maximum margin clustering. In *ICONIP*, pages 465–473. Springer, 2012. 2, 3

[8] Hui Chen, Fangqing Liu, Yin Wang, Liyue Zhao, and Hao Wu. A variational approach for learning from positive and unlabeled data. In *NeurIPS*, 2020. 5, 6

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 2, 3

[10] Xuxi Chen, Wuyang Chen, Tianlong Chen, Ye Yuan, Chen Gong, Kewei Chen, and Zhangyang Wang. Self-pu: Self boosted and calibrated positive-unlabeled training. In *ICML*, pages 1510–1519, 2020. 1, 2, 4, 5, 6

[11] Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, pages 215–223. JMLR.org, 2011. 5

[12] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *ICML*, pages 1386–1394, 2015. 1, 2, 5

[13] Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *NeurIPS*, pages 703–711, 2014. 1, 2, 5, 6

[14] Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Class-prior estimation for learning from positive and unlabeled data. In *ACML*, pages 221–236. JMLR.org, 2015. 2

[15] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *SIGKDD*, pages 213–220, 2008. 2

[16] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. PMLR, 2017. 2

[17] Zayd Hammoudeh and Daniel Lowd. Learning from positive and unlabeled data with arbitrary positive shift. In *NeurIPS*, 2020. 2

[18] Fengxiang He, Tongliang Liu, Geoffrey I Webb, and Dacheng Tao. Instance-dependent pu learning by bayesian optimal relabeling. *arXiv preprint arXiv:1808.02180*, 2018. 1

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735. Computer Vision Foundation / IEEE, 2020. 2, 3

[20] Ming Hou, Brahim Chaib-Draa, Chao Li, and Qibin Zhao. Generative adversarial positive-unlabelled learning. *arXiv preprint arXiv:1711.08054*, 2017. 2

[21] Cho-Jui Hsieh, Nagarajan Natarajan, and Inderjit S. Dhillon. PU learning for matrix completion. In *ICML*, pages 2445–2453. JMLR.org, 2015. 1

[22] Yu-Guan Hsieh, Gang Niu, and Masashi Sugiyama. Classification from positive, unlabeled and biased negative data. In *ICML*, pages 2820–2829. PMLR, 2019. 2

[23] Wenpeng Hu, Ran Le, Bing Liu, Feng Ji, Jinwen Ma, Dongyan Zhao, and Rui Yan. Predictive adversarial learning from positive and unlabeled data. In *AAAI*, pages 7806–7814, 2021. 2, 5

[24] Masahiro Kato, Takeshi Teshima, and Junya Honda. Learning from positive and unlabeled data with a selection bias. In *ICLR*, 2018. 2

[25] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 3

[26] Ryuichi Kiryo, Gang Niu, Marthinus Christoffel du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *NeurIPS*, pages 1675–1685, 2017. 1, 2, 5, 6, 7

[27] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[28] Changchun Li, Ximing Li, Lei Feng, and Jihong Ouyang. Who is your right mixup partner in positive and unlabeled learning. In *ICLR*. OpenReview.net, 2022. 2, 3, 5, 6

[29] Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *IJCAI*, pages 587–592, 2003. 1, 2

[30] Xiaoli Li and Bing Liu. Learning from positive and unlabeled examples with different data distributions. In *ECML*, pages 218–229. Springer, 2005. 1

[31] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *AAAI*, pages 8547–8555, 2021. 2

[32] Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. Partially supervised classification of text documents. In *ICML*, pages 387–394. Morgan Kaufmann, 2002. 1, 2

[33] Chuan Luo, Pu Zhao, Chen Chen, Bo Qiao, Chao Du, Hongyu Zhang, Wei Wu, Shaowei Cai, Bing He, Saravanakumar Rajmohan, and Qingwei Lin. PULNS: positive-unlabeled learning with effective negative sample selector. In *AAAI*, pages 8784–8792. AAAI Press, 2021. 2

[34] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. 2

[35] Tao Peng, Wanli Zuo, and Fengling He. SVM based adaptive learning method for text classification from positive and unlabeled documents. *Knowl. Inf. Syst.*, 16(3):281–301, 2008. 2

[36] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. Meta pseudo labels. In *CVPR*, pages 11557–11568. Computer Vision Foundation / IEEE, 2021. 2

[37] Harish G. Ramaswamy, Clayton Scott, and Ambuj Tewari. Mixture proportion estimation via kernel embeddings of distributions. In *ICML*, pages 2052–2060. JMLR.org, 2016. 2, 5

[38] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pages 4331–4340. PMLR, 2018. 2, 3, 4

[39] Yafeng Ren, Donghong Ji, and Hongbin Zhang. Positive unlabeled learning for deceptive reviews detection. In *EMNLP*, pages 488–498. ACL, 2014. 1

[40] Zhongzheng Ren, Raymond A. Yeh, and Alexander G. Schwing. Not all unlabeled data are equal: Learning to weight data in semi-supervised learning. In *NeurIPS*, 2020. 2, 3, 4

[41] Tomoya Sakai and Nobuyuki Shimizu. Covariate shift adaptation on learning from positive and unlabeled data. In *AAAI*, pages 4838–4845, 2019. 2

[42] Guangxin Su, Weitong Chen, and Miao Xu. Positive-unlabeled learning from imbalanced data. In *IJCAI*, pages 2995–3001. ijcai.org, 2021. 2

[43] Daiki Tanaka, Daiki Ikami, and Kiyoharu Aizawa. A novel perspective for positive-unlabeled learning via noisy labels. *CoRR*, abs/2103.04685, 2021. 1, 5

[44] Sunil Thulasidasan, Gopinath Chennupati, Jeff A. Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *NeurIPS*, pages 13888–13899, 2019. 6

[45] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. 2

[46] Xinrui Wang, Wenhai Wan, Chuanxin Geng, Shaoyuan LI, and Songcan Chen. Beyond myopia: Learning from positive and unlabeled data through holistic predictive trends. *arXiv preprint arXiv:2310.04078*, 2023. 6

[47] Tong Wei, Feng Shi, Hai Wang, Wei-Wei Tu, and Yu-Feng Li. Mixpul: Consistency-based augmentation for positive and unlabeled learning. *CoRR*, abs/2004.09388, 2020. 2, 6

[48] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *CoRR*, abs/1805.01978, 2018. 2, 3

[49] Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Jun Yu, Gang Niu, and Masashi Sugiyama. Sample selection with uncertainty of losses for learning with noisy labels. In *ICLR*. OpenReview.net, 2022. 1

[50] Peng Yang, Xiaoli Li, Jian-Ping Mei, Chee Keong Kwoh, and See-Kiong Ng. Positive-unlabeled learning for disease gene identification. *Bioinform.*, 28(20):2640–2647, 2012. 1

[51] Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. PEBL: positive example based learning for web page classification using SVM. In *SIGKDD*, pages 239–248. ACM, 2002. 1, 2

[52] Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. PEBL: web page classification without negative examples. *IEEE Trans. Knowl. Data Eng.*, 16(1):70–81, 2004. 2

[53] Bangzuo Zhang and Wanli Zuo. Reliable negative extracting based on knn for learning from positive and unlabeled examples. *J. Comput.*, 4(1):94–101, 2009. 2, 4

[54] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*. OpenReview.net, 2018. 6

[55] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization? In *ICLR*. OpenReview.net, 2021. 6

[56] Yunrui Zhao, Qianqian Xu, Yangbangyan Jiang, Peisong Wen, and Qingming Huang. Dist-pu: Positive-unlabeled learning from a label distribution perspective. In *CVPR*, pages 14461–14470, 2022. 5, 6