A Closer Look to Positive-Unlabeled Learning from Fine-grained Perspectives: An Empirical Study

Yuanchao Dai[1,2], Zhengzhang Hou[1,2], Changchun Li[1,2], Yuanbo Xu[1], En Wang[1], Ximing Li[1,2,3]*
[1]College of Computer Science and Technology, Jilin University, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University, China
[3]RIKEN Center for Advanced Intelligence Project
{liximing86, yuanchaodai, changchunli93}@gmail.com

Abstract

Positive-Unlabeled (PU) learning refers to a specific weakly-supervised learning paradigm that induces a binary classifier with a few positive labeled instances and massive unlabeled instances. To handle this task, the community has proposed dozens of PU learning methods with various techniques, demonstrating strong potential. In this paper, we conduct a comprehensive study to investigate the basic characteristics of current PU learning methods. We organize them into two fundamental families of PU learning, including *disambiguation-free empirical risks*, which approximate the expected risk of supervised learning, and *pseudo-labeling methods*, which estimate pseudo-labels for unlabeled instances. First, we make an empirical analysis on disambiguation-free empirical risks such as uPU, nnPU, and DistPU, and suggest a novel risk-consistent set-aware empirical risk from the perspective of aggregate supervision. Second, we make an empirical analysis of pseudo-labeling methods to evaluate the potential of pseudo-label estimation techniques and widely applied generic tricks in PU learning. Finally, based on those empirical findings, we propose a general framework of PU learning by integrating the set-aware empirical risk with pseudo-labeling. Compared with existing PU learning methods, the proposed framework can be a practical benchmark in PU learning.

1 Introduction

Positive-**U**nlabeled (**PU**) learning refers to a specific weakly-supervised learning paradigm [1, 2, 3] for binary classification, which trains a binary classifier with a few positive labeled instances and massive unlabeled instances [4]. It arises in various practical scenarios such as automatic face tagging, spam detection, and Inlier-based outlier detection [5]. Due to its wide applicability, PU learning has increasingly attracted more attention from the machine learning community.

During the past decades, many emerging practical PU learning methods have been proposed with various advanced techniques [6, 7]. Because in PU learning, negative labeled instances are unavailable, how to deal with unlabeled instances becomes its key challenge; and from this taxonomic perspective, we organize the existing PU learning methods into two fundamental families, namely *disambiguation-free empirical risks* [5, 8, 9, 10] and *pseudo-labeling methods* [11, 12, 13, 14, 15, 16].

The disambiguation-free empirical risks, as the name suggests, directly apply only positive labeled instances and unlabeled instances to approximate the expected risk of supervised learning. Under certain data generation assumptions, previous studies suggest unbiased empirical risk uPU [17, 5] and several practical variants such as nnPU with non-negativity constraint [8], abs-PU with absolute-

*Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

value constraint [9], and DistPU with positive-class prior constraint [10]. In parallel, the basic idea of pseudo-labeling methods is estimating pseudo-labels for unlabeled instances, and training the binary classifier with them in a self-training manner. Analogous to semi-supervised learning, these methods typically estimate pseudo-labels by iteratively updating the current predictions. For example, RP [18] iteratively identifies reliable negative examples from unlabeled data and assigns hard pseudo-labels to them for subsequent training. Another recent Self-PU [11] utilizes a soft pseudo-labeling strategy that continuously refines label assignments by incorporating the evolving confidence scores throughout the training process. Additionally, they apply several generic tricks such as mixup augmentation, exponential moving average, and knowledge distillation to further improve the classification performance [7, 13, 19].

The current PU learning methods have demonstrated strong potential, but we find that most of them, especially pseudo-labeling ones, are commonly complicated by integrating with specific tricks. Accordingly, some of their basic characteristics are still unclear, such as what kind of pseudo-labeling techniques and generic tricks are practical. In this paper, we conduct a comprehensive study to investigate the basic characteristics from a fine-grained perspective of PU learning. First, we make an empirical analysis on disambiguation-free empirical risks; suggest a novel risk-consistent set-aware empirical risk from the perspective of aggregate supervision, and empirically validate that it can be a practical candidate for disambiguation-free empirical risks. Second, we turn to pseudo-labeling methods, and make an empirical analysis on basic techniques to estimate pseudo-label such as hard pseudo-labeling technique, soft pseudo-labeling technique, and high-confident pseudo-label selection strategies; additionally, we empirically analyze several widely applied generic tricks in PU learning. Finally, based on those empirical findings, we propose a general framework of PU learning by integrating the set-aware empirical risk with pseudo-labeling, namely **GPU**. We further suggest specification principles within GPU. Compared with existing PU learning methods, the proposed GPU framework can be a practical benchmark in PU learning. In summary, the contributions of this paper is outlined below:

- We conduct a comprehensive empirical study to the current PU learning methods, and make extensive empirical observations on the effectiveness of basic techniques and tricks in PU Learning.
- We propose a novel risk-consistent set-aware empirical risk from the perspective of aggregate supervision, which can be a practical candidate for disambiguation-free empirical risks, and then formulate a novel general framework of PU learning by integrating it with pseudo-labeling.
- We suggest implementation principles of GPU. Compared with existing PU learning methods, the proposed GPU framework can be a practical benchmark in PU learning.

2 Preliminaries

In this section, we review the problem setting of PU learning and the two main families of PU learning methods.

Problem formulation and notations Formally, under the two-sample problem setting [20] and completely selected at random (SCAR) assumption [21], given a positive dataset $\mathcal{D}_p = \{(\mathbf{x}_i, +1)\}_{i=1}^{n_p} \overset{i.i.d.}{\sim} p_p(\mathbf{x}) = p(\mathbf{x}|y=+1)$ with n_p instances drawn from the positive-class conditional density $p_p(\mathbf{x})$ and an unlabeled dataset $\mathcal{D}_u = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_u} \overset{i.i.d.}{\sim} p(\mathbf{x})$ with n_u instances drawn from the marginal density $p(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^d$ and $y \in \{-1, +1\}$ denote the d-dimensional feature vector and the corresponding binary label, respectively. The objective of PU learning is to induce a classifier $g : \mathbb{R}^d \to \mathbb{R}$ over $\mathcal{D}_p \cup \mathcal{D}_u$, which can predict labels for unseen instances.

2.1 PU Learning with Disambiguation-free Empirical Risks

PU learning methods with disambiguation-free empirical risks directly utilize labeled positive data and unlabeled data to approximate the expected risk of supervised learning. In this work, we review several representative ones, including **uPU** [5], **nnPU** [8], **absPU** [9], and **DistPU** [10].

Table 1: A summary of basic techniques and widely applied generic tricks for PU learning methods with pseudo-labeling.

PUL methods	pseudo-labeling technique				generic trick	
	pseudo-label		high confidence selection		mixup	moving average
	hard	soft	with	w/o		
RP [18]	✓			✓		
AdaSampling [22]	✓			✓		
GenPU [23]	✓					
Self-PU [11]		✓		✓	✓	
VPU [12]		✓	✓		✓	
PULNS [24]	✓			✓		
P^{3}Mix [13]	✓		✓		✓	
RobustPU [14]	✓			✓		
HolisticPU [15]	✓		✓			
LaGAM [25]		✓	✓		✓	✓
PUL-CPBF [16]	✓		✓			
VQ-Encoder [26]		✓	✓			

Formally, let $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$ be any loss function, and $\pi = p(y = +1)$ be the positive-class prior. Since negative samples are not directly accessible in PU learning, the SCAR assumption fortunately provides a solution. Under this assumption, where the labeled positive examples are selected completely at random from all positive examples, given $R_p^+(g) = \mathbb{E}_{p_p(\mathbf{x})}\left[\ell\big(g(\mathbf{x}\big), +1)\right]$, $R_p^-(g) = \mathbb{E}_{p_p(\mathbf{x})}\left[\ell\big(g(\mathbf{x}\big), -1)\right]$, $R_n^-(g) = \mathbb{E}_{p_n(\mathbf{x})}\left[\ell\big(g(\mathbf{x}), -1\big)\right]$, and $R_u^-(g) = \mathbb{E}_{p(\mathbf{x})}\left[\ell\big(g(\mathbf{x}), -1\big)\right]$, we obtain that $p(\mathbf{x}) = \pi p_p(\mathbf{x}) + (1-\pi)p_n(\mathbf{x})$, such that $(1-\pi)R_n^-(g) = R_u^-(g) - \pi R_p^-(g)$. Then, suppose π is known, one can formulate [5], which is an unbiased risk of PU learning uPU for the expected risk of supervised learning, formulated as follows:

$$R_{\mathrm{uPU}}(g) = \pi R_p^+(g) + R_u^-(g) - \pi R_p^-(g), \tag{1}$$

and its empirical risk over $D_p \cup D_u$ is given below:

$$\widehat{R}_{\mathrm{uPU}}(g) = \pi \widehat{R}_p^+(g) + \widehat{R}_u^-(g) - \pi \widehat{R}_p^-(g) \tag{2}$$

However, uPU suffers from overfitting when using flexible models due to negative empirical risk $\widehat{R}_u^-(g) - \pi\widehat{R}_p^-(g)$. Some methods attempt to impose a non-negative constraint on $\widehat{R}_u^-(g) - \pi\widehat{R}_p^-(g)$ to prevent the empirical risk from becoming negative. To achieve this, nnPU [8] incorporates the max function:

$$\widehat{R}_{\mathrm{nnPU}}(g) = \pi \widehat{R}_p^+(g) + \max\{0, \widehat{R}_u^-(g) - \pi \widehat{R}_p^-(g)\}, \tag{3}$$

and absPU incorporates the absolute value function [9]:

$$\widehat{R}_{\mathrm{absPU}}(g) = \pi \widehat{R}_p^+(g) + \left|\widehat{R}_u^-(g) - \pi \widehat{R}_p^-(g)\right| \tag{4}$$

Additionally, under the case of symmetric losses where $\ell(z) + \ell(-z) = 1$, we have $\widehat{R}_p^-(g) = 1 - \widehat{R}_p^+(g)$ naturally holds. Leveraging this property, Dist-PU [10] reformulates uPU as follows:

$$\widehat{R}_{\mathrm{DistPU}}(g) = 2\pi \widehat{R}_p^+(g) + \left|\widehat{R}_u^-(g) - \pi\right|, \tag{5}$$

where the absolute value function is introduced to impose the non-negative constraint on $\widehat{R}_u^-(g) - \pi$.

2.2 PU Learning with Pseudo-labeling

PU learning methods with pseudo-labeling, as the name suggests, estimate pseudo-labels for unlabeled data, and train the classifier g in a self-training manner [18, 22, 23, 24, 13, 14, 15, 16]. Referring to semi-supervised learning, we can formulate the generic objective of pseudo-labeling below:

$$\mathcal{L}(g; D_p, D_u) = \widehat{R}_p^+(g) + \mathcal{L}_u(g, \hat{y}; D_u), \tag{6}$$

where $\mathcal{L}_u(g, \hat{y}; D_u)$ is the self-training objective with unlabeled data, and $\hat{y}$ denotes the pseudo-label.

Table 2: Positive and negative label groups of datasets and the statistics of those PU learning sets.

Dataset	π	Positive Class	Negative Class	Feature	Train	Backbone
F-MNIST-1	0.4	0, 2, 4, 6	1, 3, 5, 7, 8, 9	28×28	60,000	LeNet-5
F-MNIST-2	0.6	1, 3, 5, 7, 8, 9	0, 2, 4, 6	28×28	60,000	LeNet-5
CIFAR-10-1	0.4	0, 1, 8, 9	2, 3, 4, 5, 6, 7	$3 \times 32 \times 32$	50,000	7-Layer CNN
CIFAR-10-2	0.6	2, 3, 4, 5, 6, 7	0, 1, 8, 9	$3 \times 32 \times 32$	50,000	7-Layer CNN
STL-10-1	–	0, 2, 3, 8, 9	1, 4, 5, 6, 7	$3 \times 96 \times 96$	105,000	7-Layer CNN
STL-10-2	–	1, 4, 5, 6, 7	0, 2, 3, 8, 9	$3 \times 96 \times 96$	105,000	7-Layer CNN

We review existing pseudo-labeling methods and summarize the basic techniques and widely applied generic tricks in Table 1. Specifically, the basic problem of pseudo-labeling is the techniques to estimate pseudo-labels $\hat{y}$ for unlabeled data with current predictions, and they typically include **hard** and **soft** and pseudo-labeling techniques. Let $q = g(\mathbf{x})$ and $\phi(q) \in [0, 1]$ denote the prediction of the classifier and the confidence belonging to the positive class, respectively, where ϕ is a transformation function, and here we apply the sigmoid function. Then, the hard and soft pseudo-labels are estimated as $\hat{y} = \text{sign}\,(\phi(q) - 0.5) \in \{-1, +1\}$ and $\hat{y} = \phi(q)$, respectively. In addition, some studies suggest selecting high-confident pseudo-labels, rather than applying all of them [18, 22, 11, 24, 14], where the representatives include various thresholding strategies.

Generic tricks We briefly review two widely applied generic tricks in PU learning studies [11, 12, 13, 25], such as **mixup** and **moving average**. Mixup is an efficient data augmentation trick with the convex combination of instance pairs [7]. Given an instance pair $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$, it generates an augmented instance $(\tilde{\mathbf{x}}, \tilde{y})$ as follows:

$$\tilde{\mathbf{x}} = \lambda \mathbf{x}_i + (1 - \lambda)\mathbf{x}_j, \quad \tilde{y} = \lambda y_i + (1 - \lambda) y_j, \qquad \lambda \sim \text{Beta}(\alpha, \alpha) \tag{7}$$

Here, the moving average refers to updating pseudo-labels with historical predictions during the classifier training process [27]. Formally, its update equation for pseudo-labels is given below:

$$\phi(q) \leftarrow \epsilon \phi(q^h) + (1 - \epsilon)\phi(q), \tag{8}$$

where q^h denotes the historical prediction, and ϵ is a smoothing parameter.

3 Empirical Findings, Analysis, and Modifications

3.1 Settings of Empirical Study

We conduct empirical evaluations on 3 standard benchmark datasets, *i.e.* Fashion-MNIST (F-MNIST), CIFAR-10, and STL-10. Following [16], we transform them into a set of binary classification problems by partitioning their original 10 classes into positive and negative categories by varying the class prior $\pi \in \{0.4, 0.6\}$. For all datasets, the number of positive labeled instances is fixed as $n_p = 1,000$. The details of datasets are summarized in Table 2.

For each PU learning method, We employ dataset-appropriate backbones as follows: LeNet-5 for F-MNIST, 7-layer CNN for CIFAR-10 and STL-10; the MLP layer is used as the classification layer across all datasets. The mini-batch is fixed as 512 and the number of epochs is set to 100 for F-MNIST and 200 for others.

In addition, we employ the classification accuracy (**ACC**) as the evaluation metric. All experiments are conducted with five different random seeds on a server equipped with two Nvidia RTX4090 GPUs, and we report the mean and standard deviation of the results.

3.2 Disambiguation-free Empirical Risks

In this section, we suggest a novel set-aware empirical risk of PU learning and empirically evaluate it and existing disambiguation-free empirical risks with various surrogate loss functions.

Table 3: Properties of commonly used loss functions.

Loss	Formula	Convex	Differ.	Symm.	Lipsc.
hinge	$\max\{0, 1-z\}$	✓			✓
logistic	$\log(1+e^{-z})$	✓	✓		✓
sigmoid	$1/(1+e^{z})$		✓	✓	✓
squared	$(1-z)^2$	✓	✓		
ramp	$\min\{1,\ \max\{0,\ (1-z)/2\}\}$			✓	✓
double-hinge	$\max\{0,\ (1-z)/2,\ -z\}$			✓	✓

Set-aware empirical risk of PU learning In PU learning, we are given the positive labeled data and unlabeled data $\mathcal{D}_p \cup \mathcal{D}_u$, and the positive-class prior π. Inspired by previous weakly-supervised learning studies with aggregate supervision [28], we can arrange the training data as $\mathcal{D}_p \cup (\mathcal{D}_u, \pi)$, where we treat $(\mathcal{D}_u, \pi)$ as a set of instances with its approximate label proportion.[2] Accordingly, we can formulate the following set-aware empirical risk of PU learning (**SAPU**):

$$\widehat{R}_{\mathrm{SAPU}}(g) = \widehat{R}_p^+(g) + \ell_{CE}\left(\frac{1}{n_u}\sum_{\mathbf{x}_i \in \mathcal{D}_u} g(\mathbf{x}_i), \pi\right), \tag{9}$$

where ℓ_{CE} denotes the cross-entropy loss. Because the size of $\mathcal{D}_u$ can be too large, directly fitting π in the second term of $\widehat{R}_{\mathrm{SAPU}}(g)$ may result in smoothing instance-level predictions. To alleviate this potential issue, we can randomly divide $\mathcal{D}_u$ into many subsets $\{\mathcal{S}_i\}_{i=1}^{n_s}$, where $\mathcal{S}_i = \{\mathbf{x}_{ij}\}_{j=1}^{S}$, n_s is the number of subsets, and S is the number of instances in each subset; and if S is large enough, we can also approximate the label proportion of each subset as π. Upon these ideas, we can rearrange the training data as $\mathcal{D}_p \cup \{(\mathcal{S}_i, \pi)\}_{i=1}^{n_s}$, and then reformulate Eq.9 as follows:

$$\widehat{R}_{\mathrm{SAPU}}(g) = \widehat{R}_p^+(g) + \frac{1}{n_s}\sum_{i=1}^{n_s} \ell_{CE}\left(\frac{1}{S}\sum_{\mathbf{x}_{ij} \in \mathcal{S}_i} g(\mathbf{x}_{ij}), \pi\right) \tag{10}$$

We consider SAPU as a practical candidate disambiguation-free empirical risks of PU learning. We show the following theorem to indicate that it is risk-consistent for the expected risk of supervised learning. The proof is presented in the Appendix.

Lemma 3.1. *Let $\hat{\pi}_i = \frac{1}{S}\sum_{i=1}^{S} \mathbf{1}[y_{ij} = +1]$ be the true proportion of positive instance in set $\mathcal{S}_i$. When the set size satisfies $S \geq \frac{3\pi(1-\pi)\log(2/\delta)}{2\epsilon^2}$, with probability at least $1-\delta$, we have $|\hat{\pi}_j - \pi| \leq \epsilon$ for each set $\mathcal{S}_i$ and Var$(\hat{\pi}_j) \leq \frac{\pi(1-\pi)}{S}$.*

Theorem 3.2. *Let $g^* = \arg\min_{g\in\mathcal{G}} R(g)$ is the minimizer of the true classification risk and $\hat{g}_{\mathrm{SAPU}} = \arg\min_{g\in\mathcal{G}} \hat{R}_{\mathrm{SAPU}}(g)$ denotes the minimizer of the risk form in Eq.10. Suppose that the pseudo-dimensions of $\{\mathbf{x} \mapsto g(\mathbf{x}) | g \in \mathcal{G}\}$ and $\{\mathbf{x} \mapsto \ell_{CE}(g(\mathbf{x}), \pi) | g \in \mathcal{G}\}$ are finite, and there exist constants L_g, L_ℓ such that $|g(\mathbf{x})| \leq L_g$ and $|\ell_{CE}(g(\mathbf{x}), \pi)| \leq L_\ell$ for all $\mathbf{x} \in \mathcal{X}$ and all $g \in \mathcal{G}$. Then, for any $\delta > 0$, with probability at least $1-\delta$:*

$$R(\hat{g}_{\mathrm{SAPU}}) - R(g^*) \leq O\left(\sqrt{\frac{\log(1/\delta)}{n_p}}\right) + O\left(\sqrt{\frac{\log(1/\delta)}{n_s}}\right) + L_\ell \cdot O\left(\sqrt{\frac{\pi(1-\pi)\log(1/\delta)}{S}}\right) \tag{11}$$

Results and analysis We empirically investigate the proposed SAPU and 4 existing disambiguation-free empirical risks with different commonly used loss functions. Table 3 presents 6 loss functions, *i.e.*, hinge, logistic, sigmoid, squared, ramp, and double-hinge, along with their mathematical formulations and theoretical properties. These loss functions are selected to represent diverse characteristics across 4 key properties: convexity, differentiability, symmetry, and Lipschitz continuity. Because the positive prior π for the STL-10 dataset is unavailable, we conduct experiments on the CIFAR-10 and F-MNIST datasets.

[2]We declare that the label proportion approaches π, as n_u goes to ∞

Table 4: The ACC scores (mean±std) of disambiguation-free empirical risks with widely used loss functions on F-MNIST and CIFAR-10. The highest scores are indicated in **bold**.

Dataset	Method	S	hinge	logistic	sigmoid	squared	ramp	double-hinge
F-MNIST-1	uPU	-	68.0±0.5	68.5±0.5	69.8±0.9	**77.1±2.2**	70.8±1.8	68.4±0.6
	nnPU	-	93.8±0.4	93.0±0.5	93.9±0.7	93.2±1.7	93.8±1.0	**94.8±0.3**
	absPU	-	**94.2±0.4**	93.3±0.5	93.3±0.7	93.7±0.4	93.6±0.6	94.1±0.2
	Dist-PU	-	-	-	94.3±0.4	-	94.0±0.2	**94.7±0.2**
	SAPU	32	93.9±1.0	95.9±0.2	95.9±0.2	94.6±0.6	94.5±1.0	94.5±0.9
		64	92.8±0.3	96.0±0.2	96.0±0.1	93.0±0.0	93.8±0.8	94.0±0.9
		128	92.8±0.1	96.1±0.0	**96.2±0.1**	91.0±0.5	93.5±0.8	92.9±0.1
		256	93.0±0.5	96.0±0.2	**96.2±0.0**	90.8±0.1	92.9±0.5	92.8±0.3
		n_u	92.4±0.5	95.4±0.6	96.0±0.0	90.8±0.3	92.9±0.6	92.6±0.1
F-MNIST-2	uPU	-	47.8±0.6	47.7±0.3	49.1±0.9	**62.4±2.7**	50.8±1.4	48.7±0.7
	nnPU	-	92.4±0.4	91.7±1.3	91.0±0.4	92.7±0.5	93.1±0.7	**93.4±0.3**
	absPU	-	92.6±0.6	91.5±0.7	91.0±1.2	92.0±0.3	92.4±0.7	**93.4±0.4**
	Dist-PU	-	-	-	91.3±0.9	-	**93.3±0.6**	92.2±0.3
	SAPU	32	94.5±0.5	95.7±0.0	95.8±0.2	94.7±0.5	95.3±0.0	94.9±0.3
		64	94.6±0.1	95.8±0.0	95.8±0.1	92.8±0.5	94.0±0.4	94.5±0.5
		128	93.6±0.4	95.9±0.2	96.0±0.1	90.8±0.2	93.2±0.4	93.9±0.3
		256	93.7±0.2	95.8±0.2	**96.1±0.0**	88.2±1.8	93.4±0.5	93.2±0.0
		n_u	92.8±0.2	95.9±0.0	96.0±0.1	88.3±1.4	93.6±0.4	93.5±0.4
CIFAR-10-1	uPU	-	80.5±0.7	**81.7±0.9**	81.6±1.9	66.1±2.4	77.3±2.3	79.9±0.7
	nnPU	-	**86.4±0.4**	84.3±0.7	85.1±1.4	83.2±1.0	86.1±0.5	**86.4±0.1**
	absPU	-	85.6±0.4	82.9±0.7	85.7±1.5	81.9±1.2	**86.3±0.9**	85.7±0.6
	Dist-PU	-	-	-	86.0±0.9	-	86.2±0.6	**86.6±0.5**
	SAPU	32	85.3±0.6	86.5±0.5	86.6±0.2	76.6±4.7	86.2±0.7	84.5±0.6
		64	83.8±0.3	86.4±0.2	86.7±0.3	77.0±3.4	86.7±0.7	85.3±0.4
		128	84.7±0.5	85.6±0.4	**87.0±0.5**	79.5±1.5	85.0±0.0	83.6±0.2
		256	83.5±0.2	86.6±0.3	86.8±0.2	75.3±5.5	85.4±0.6	84.4±0.6
		n_u	84.1±0.2	85.3±0.4	86.8±0.3	78.7±2.5	85.5±1.2	84.1±0.7
CIFAR-10-2	uPU	-	76.1±0.9	**77.3±1.1**	76.9±2.4	55.7±2.0	67.9±2.4	75.5±1.0
	nnPU	-	**84.7±1.0**	80.7±1.4	83.7±1.3	81.0±1.7	84.3±1.0	83.8±1.4
	absPU	-	**84.4±0.9**	78.0±1.7	83.8±1.4	79.3±2.7	**84.4±0.7**	82.4±1.4
	Dist-PU	-	-	-	82.1±1.1	-	83.4±1.6	**85.6±0.7**
	SAPU	32	60.7±0.0	85.1±0.7	85.1±0.7	81.3±0.3	74.7±6.3	74.7±6.3
		64	62.4±1.7	85.2±0.7	84.9±0.7	81.3±0.1	74.9±6.1	75.7±5.3
		128	61.6±0.9	85.2±0.7	84.7±0.6	81.1±0.0	72.3±9.1	72.6±8.7
		256	60.7±0.0	**85.4±0.6**	84.5±0.5	81.3±0.0	73.8±7.4	74.0±7.1
		n_u	61.6±0.9	85.3±0.7	84.7±0.7	81.2±0.1	74.5±6.9	74.8±7.2

The experimental results in Table 4 demonstrate that the choice of loss function significantly influences classification accuracy across different datasets and methods. For F-MNIST dataset, the sigmoid loss consistently delivers superior performance, achieving a remarkable accuracy of 96% with our method. The double-hinge loss also performs exceptionally well, particularly with nnPU method and Dist-PU. On the more challenging CIFAR-10 dataset, the sigmoid loss still demonstrates robust performance (around 86% with SAPU), while the double-hinge loss excels in several configurations, notably with Dist-PU on CIFAR-10-2 (85.6%). Interestingly, the effectiveness of each loss function varies substantially across different empirical risk methods and datasets. For example, while SAPU achieves optimal results with sigmoid on CIFAR-10-1 (86.8%), its performance degrades considerably with the squared loss. Additionally, our empirical analysis confirms that smooth, differentiable losses (sigmoid, logistic) achieve better compatibility with SAPU's set-aware architecture, while non-smooth losses (double-hinge, ramp) align better with traditional point-wise optimization methods. This non-uniform behavior suggests a complex interaction between loss functions and model architectures

that cannot be reduced to simple heuristics, underscoring the importance of careful loss function selection based on specific application contexts.

Based on the above analysis, we can summarize the following guiding principles: (1) The smooth, differentiable losses (such as sigmoid, logistic loss) achieve better compatibility with the set-aware architecture of SAPU, while non-smooth losses (*e.g.*, double-hinge, ramp loss) align better with traditional point-wise optimization methods. (2) Convex losses generally provide better optimization guarantees. (3) Simple datasets (*e.g.*, F-MNIST) benefit from smooth losses, enabling fine-grained optimization, while complex datasets (*e.g.*, CIFAR-10) may require losses with stronger regularization properties.

Furthermore, as the core of SAPU lies in dividing unlabeled data into multiple subsets for set-aware supervision, we further conduct experiments to verify how subset size affects the performance of the model. We systematically tested different subset sizes $S = \{32, 64, 128, 256, n_u\}$ across all datasets and recorded classification accuracy with various loss functions. The experimental results in Table 4 demonstrate that $S = 256$ yields optimal performance on most datasets, particularly when combined with the sigmoid loss function. For example, the model achieved peak accuracies of 96.2% and 96.1% respectively on F-MNIST-1 and F-MNIST-2 when $S = 256$ with sigmoid loss function. For simpler datasets like F-MNIST, this phenomenon can be explained that when subsets are too small, individual subsets struggle to accurately reflect the overall label distribution; conversely, when subsets become excessively large (approaching n_u), instance-level predictions become overly smoothed, reducing the model's discriminative power. Moreover, for more complex datasets like CIFAR-10, our experiments indicate that larger subset sizes tend to be more effective. Based on our comprehensive analysis across different datasets, we recommend setting medium-sized subsets (e.g., $S = 256$) as a generally effective configuration for our SAPU method.

3.3 Pseudo-labeling Methods

In this section, we investigate the pseudo-labeling techniques and thresholding techniques for selecting high-confident pseudo-labels. For comprehensive evaluations, we first suggest several base methods and then discuss the empirical results.

Base methods of pseudo-labeling We specify the generic objective of Eq.6 with specific pseudo-labeling techniques and thresholding strategies, leading to a set of base methods. First, we estimate pseudo-labels $\hat{y}$ by **hard** and **soft** pseudo-labeling techniques; and for clarity, we review them as $\hat{y} = \text{sign}\,(\phi(q) - 0.5) \in \{-1, +1\}$ and $\hat{y} = \phi(q)$, respectively. Second, the thresholding strategy refers to computing a threshold value τ to define the lower bound of high-confident pseudo-labels. Inspired by [29, 30], we specify 3 thresholding strategies to compute τ, described below:

- **Fixed thresholding** treats the threshold value τ as a hyper-parameter, and empirically sets it as a constant value. Here, we fix τ to 0.95.
- **Adaptive thresholding** gradually updates the threshold value τ during classifier training. Following the idea that the predictions can be more accurate as the classifier continues to be trained [31], we gradually increase τ as follows:

  $$\tau \leftarrow \tau_{max} \times \min(1, t/T),$$

  where t is the current epoch, τ_{max} is the maximum threshold value, and T is the ramp-up period.
- **Class-specific adaptive thresholding** gradually updates the threshold values τ_p and τ_n for positive- and negative-classes, respectively. Following [30], we gradually update τ_p and τ_n as follows:

  $$\tau_p \leftarrow \tau_p \times \mathcal{C}_p^{(t)}, \qquad \tau_n \leftarrow \tau_n \times \mathcal{C}_n^{(t)},$$

where $\mathcal{C}_p^{(t)}$ and $\mathcal{C}_n^{(t)}$ are the ratios between the pseudo-label accuracies of positive- and negative-classes and their higher accuracy at epoch t.

Based on these specific techniques, we can specify **6 base methods of pseudo-labeling**.

Results and analysis To comprehensively evaluate the effectiveness of different pseudo-label strategies and generic tricks under PU learning, we compare six base pseudo-labeling methods

Table 5: The ACC scores (mean±std) of 6 base methods of pseudo-labeling on F-MNIST and CIFAR-10. The highest scores are indicated in **bold**.

Label	Threshold	M.A.	Mixup	F-MNIST-1	F-MNIST-2	CIFAR-10-1	CIFAR-10-2
Hard	Fixed			90.0±1.7	92.7±0.1	85.1±0.6	83.1±3.8
		✓		89.9±1.6	89.3±2.3	84.2±1.8	82.0±3.5
			✓	90.5±0.8	92.8±0.2	85.1±0.3	85.0±1.0
		✓	✓	89.6±1.9	89.0±1.7	84.0±2.1	82.1±3.7
	Adaptive			91.4±0.7	92.8±0.2	84.3±0.5	83.7±0.8
		✓		71.7±4.2	65.5±5.8	80.5±0.2	82.8±3.0
			✓	90.6±0.6	92.9±0.1	84.5±0.2	83.8±0.2
		✓	✓	72.1±4.3	67.5±3.2	80.7±0.0	83.1±3.3
	Class Adaptive			91.5±0.4	89.0±0.0	84.3±0.4	83.5±0.5
		✓		71.7±4.2	65.5±5.8	82.0±1.0	82.0±1.5
			✓	91.0±1.0	89.6±0.2	84.5±0.5	84.0±0.5
		✓	✓	72.1±4.3	65.5±5.7	80.5±1.0	82.5±1.5
Soft	Fixed			95.4±0.4	93.7±0.4	84.3±0.5	83.8±0.2
		✓		95.2±0.3	71.1±1.7	83.5±1.7	83.0±1.5
			✓	95.4±0.4	93.9±0.4	84.3±0.5	83.8±0.2
		✓	✓	95.2±0.3	70.7±2.2	83.8±1.5	83.3±1.0
	Adaptive			95.4±0.4	94.1±0.3	**85.9±0.5**	82.9±3.4
		✓		74.0±4.0	69.9±3.0	83.5±1.5	83.0±1.3
			✓	**95.5±0.4**	94.2±0.4	**85.9±0.5**	84.8±0.0
		✓	✓	82.9±2.4	70.7±2.2	83.9±1.3	83.4±1.1
	Class Adaptive			95.0±0.1	94.1±0.4	84.3±0.5	83.8±0.2
		✓		77.2±0.1	93.7±1.5	80.8±1.5	83.0±1.5
			✓	**95.5±0.4**	**94.6±0.4**	85.6±0.5	**85.8±0.2**
		✓	✓	82.9±2.4	93.8±1.1	81.0±1.3	83.4±1.1

(including two pseudo-labeling techniques (hard vs. soft labeling) and three thresholding strategies (fixed, adaptive, and class-specific adaptive)) and two widely used enhancement techniques (mixup and moving average) on CIFAR-10 and F-MNIST datasets.

The results demonstrate that soft labeling consistently outperforms hard labeling, particularly when combined with class-adaptive thresholding on F-MNIST datasets (achieving up to 95.5%). Mixup proves to be the most consistent generic trick for ACC improvement across all experimental configurations, while moving average often leads to performance degradation when combined with other techniques. Mixup proves consistently beneficial because it addresses the fundamental challenge of decision boundary uncertainty in PU learning. By creating synthetic samples through convex combinations, mixup naturally smooths the decision boundaries in regions. This is particularly crucial in PU learning where the model must distinguish between true negatives and mislabeled positives within the unlabeled set. In contrast, the counterintuitive phenomenon of performance degradation with moving average techniques primarily stems from the unstable nature of pseudo-labels in PU learning. Unlike traditional semi-supervised learning where unlabeled data contains truly unlabeled instances, PU learning involves mislabeled negative samples, making historical predictions unreliable. The self-training process generates systematic biases, and moving average perpetuates rather than corrects these biases. On the other hand, moving average techniques may suppress the model's ability to rapidly adapt within the feature space to distinguish between positive and negative samples. Furthermore, the momentum parameter requires careful tuning, which significantly increases the experimental cost for hyperparameter optimization.

Overall, the combination of soft labeling with class-adaptive thresholds and Mixup yields the best performance across nearly all datasets. The only exception occurs in the CIFAR-10-1 dataset, likely due to its complex visual diversity as a natural image dataset, making the combination of fixed thresholding with moving average and mixup more suitable for handling its complex decision boundaries. These findings suggest that the combination of soft labeling, class-adaptive thresholding, and mixup generally constitutes the most promising universal method.

Table 6: The ACC scores (mean±std) of existing PU learning methods and GPU.

Method	F-MNIST-1	F-MNIST-2	CIFAR-10-1	CIFAR-10-2	STL-10-1	STL-10-2	Rank
uPU	77.1±2.2	62.4±2.7	81.7±0.9	77.3±1.1	76.7±0.8	71.5±4.8	15.3
nnPU	94.8±0.3	93.4±0.3	86.4±0.1	84.7±1.0	77.1±4.5	81.9±1.0	8.7
absPU	94.2±0.4	93.4±0.3	86.3±0.9	84.4±0.9	75.3±2.2	82.0±0.7	9.8
Dist-PU	94.7±0.2	93.3±0.6	86.7±0.5	85.6±0.7	78.3±0.8	81.5±1.1	8.7
RP	94.4±0.6	93.3±0.5	78.0±1.9	84.2±1.1	71.3±0.8	75.5±2.6	12.2
AdaSampling	93.6±0.3	93.5±0.2	79.6±0.5	79.1±1.0	74.3±2.2	82.6±0.8	11.3
GenPU	78.1±0.4	86.2±1.4	71.2±1.9	68.3±2.5	68.5±1.4	57.3±1.5	16.5
Self-PU	90.8±0.4	89.1±0.7	85.1±0.8	83.9±2.6	78.5±1.1	80.8±2.1	12.2
VPU	92.6±1.2	90.5±0.8	86.8±1.2	82.5±1.1	78.4±1.1	82.9±0.7	10.3
PULNS	91.0±0.5	89.1±0.8	87.2±0.6	83.7±2.9	80.2±0.8	83.6±0.7	9.8
P^{3}Mix-E	92.6±0.4	91.8±0.2	88.2±0.4	84.7±0.5	80.2±0.9	83.7±0.7	7.8
P^{3}Mix-C	92.8±0.6	90.4±0.1	88.7±0.4	87.9±0.5	80.7±0.7	84.1±0.3	6.5
Robust-PU	90.0±0.5	85.5±0.7	80.0±0.6	85.2±1.1	79.6±0.9	80.4±0.8	12.3
HolisticPU	96.2±0.1	96.0±0.3	91.0±0.3	90.4±0.5	82.5±0.5	84.0±1.2	3.2
LaGAM	94.9±0.2	94.1±0.3	89.9±0.3	88.0±1.4	85.3±0.3	85.0±0.3	2.8
PUL-CPBF	96.7±0.3	96.5±0.2	91.4±0.2	91.0±0.3	83.4±0.7	85.4±1.2	1.2
GPU	96.4±0.1	96.1±0.5	88.4±0.1	87.9±0.4	82.7±0.7	84.9±0.4	3.2
PN learning	97.7±0.1	97.7±0.1	91.9±0.1	91.9±0.1	86.0±0.6	86.0±0.6	-

3.4 Proposed GPU Framework

By integrating SAPU with pseudo-labeling, we suggest an efficient PU learning framework **GPU**. Its generic objective is given as follows:

$$\mathcal{L}_{\mathrm{GPU}}(g) = \widehat{R}_p^+(g) + \mathcal{L}_u(g, \hat{y}; D_u) + \frac{\alpha}{n_s} \sum_{j=1}^{n_s} \ell_{CE}\left(\frac{1}{S} \sum_{\mathbf{x}_{ij} \in \mathcal{S}_i} g(\mathbf{x}_{ij}), \pi\right), \tag{12}$$

where α is a coefficient parameter.

We can interpret GPU as a regularized pseudo-labeling method of PU learning, where the set-aware term is treated as a regularization term. Based on the previous evaluations, we find that pseudo-labeling methods depend on high quality of pseudo-labels in the early training stage because they are in a self-training manner. Accordingly, we suggest a **warm-up stage** by minimizing the objective of SAPU. In addition, we can specify the pseudo-labeling techniques and thresholding strategies according to our empirical observations.

Results and analysis To evaluate the efficacy of our proposed GPU framework compared to existing PU learning methods, we conduct experiments on F-MNIST, CIFAR-10, and STL-10 datasets to assess its general performance across varying scenarios. For comprehensive comparison, we also include PN learning (*i.e.*, supervised learning) as an upper bound baseline. Our GPU implementation uses a subset size $S = 256$ with the sigmoid loss function and employs soft pseudo-labeling with class-adaptive thresholding based on our empirical observations. For the warm-up stage, we train using only SAPU for 20 epochs before introducing the pseudo-labeling component.

As evident from Table 6, our proposed GPU framework demonstrates competitive performance across all benchmark datasets, ranking 3.2 overall, tied with HolisticPU and slightly behind LaGAM (2.8), while remaining competitive with the leading PUL-CPBF. For example, GPU achieves accuracy scores of 96.4% and 96.1% on F-MNIST-1 and F-MNIST-2 respectively, which are comparable to the best-performing PUL-CPBF (96.7% and 96.5%). For CIFAR-10, GPU obtains 88.4% and 87.6%, positioning it among the top-tier methods but slightly below HolisticPU and PUL-CPBF. Similar competitive performance is also demonstrated on the STL-10 dataset. The performance gap between GPU and the best-performing methods reflects our focus on exploring fundamental techniques and integrating them with our novel set-aware empirical risk SAPU, rather than employing sophisticated techniques like ensemble methods in PUL-CPBF, trend detection in HolisticPU, or meta-learning in

LaGAM. GPU provides a general framework that can integrate future advances, while specialized methods may not generalize well. Notably, GPU significantly outperforms traditional PU learning methods across all datasets, demonstrating that combining set-aware empirical risk estimation with pseudo-labeling strategies effectively enhances the discriminative capability of the model.

4 Discussion and Future Works

In this paper, we comprehensively review the current families of PU learning and investigate their basic characteristics. We review the existing disambiguation-free empirical risks and suggest a novel set-aware empirical risk SAPU from the perspective of aggregate supervision, which is risk-consistent for the expected risk of supervised learning. We empirically evaluate them with various commonly applied loss functions. In addition, we review the basic techniques and widely applied generic tricks, *i.e.* mixup and moving average, in the existing pseudo-labeling methods. To empirically evaluate them, we formulate a set of base methods specified by hard and soft pseudo-labeling techniques with thresholding strategies for selecting high-confident pseudo-labels such as fixed, adaptive, and class-specific adaptive thresholding strategies. Finally, we propose an efficient PU learning framework GPU by integrating SAPU with pseudo-labeling. GPU involves a warm-up stage by minimizing SAPU and specify the framework according to our empirical observations. We compare GPU with the existing PU learning methods, and the empirical results demonstrate that GPU can be a practical benchmark in PU learning, and is scalable for future pseudo-labeling techniques.

In the future, there are two potential problems that require more attention. One basic problem is how to estimate more accurate pseudo-labels [32, 33, 34] since we only investigate the straightforward pseudo-labeling techniques. Some advanced techniques such as ensemble learning [35] demonstrate strong potential. Another problem is whether the existing PU learning can be effective for the scenarios with scarce positive labeled instances and how to deal with such scenarios, which can appear in many real-world applications.

Acknowledgements

We would like to acknowledge support for this project from the National Science and Technology Major Project (No.2021ZD0112500), and the National Natural Science Foundation of China (No.62276113).

References

[1] Kou, Z., J. Wang, Y. Jia, et al. Progressive label enhancement. *Pattern Recognition*, 160:111172, 2025.

[2] —. Inaccurate label distribution learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(10):10237–10249, 2024.

[3] —. Instance-dependent inaccurate label distribution learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):1425–1437, 2025.

[4] Bekker, J., J. Davis. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4):719–760, 2020.

[5] du Plessis, M. C., G. Niu, M. Sugiyama. Convex formulation for learning from positive and unlabeled data. In *International Conference on Machine Learning*, pages 1386–1394. 2015.

[6] Goodfellow, I., J. Pouget-Abadie, M. Mirza, et al. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[7] Zhang, H., M. Cissé, Y. N. Dauphin, et al. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*. 2018.

[8] Kiryo, R., G. Niu, M. C. du Plessis, et al. Positive-unlabeled learning with non-negative risk estimator. In *Neural Information Processing Systems*, pages 1675–1685. 2017.

[9] Hammoudeh, Z., D. Lowd. Learning from positive and unlabeled data with arbitrary positive shift. In *Neural Information Processing Systems*. 2020.

[10] Zhao, Y., Q. Xu, Y. Jiang, et al. Dist-pu: Positive-unlabeled learning from a label distribution perspective. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14441–14450. 2022.

[11] Chen, X., W. Chen, T. Chen, et al. Self-pu: Self boosted and calibrated positive-unlabeled training. In *International Conference on Machine Learning*, pages 1510–1519. 2020.

[12] Chen, H., F. Liu, Y. Wang, et al. A variational approach for learning from positive and unlabeled data. In *Neural Information Processing Systems*. 2020.

[13] Li, C., X. Li, L. Feng, et al. Who is your right mixup partner in positive and unlabeled learning. In *International Conference on Learning Representations*. 2022.

[14] Zhu, Z., L. Wang, P. Zhao, et al. Robust positive-unlabeled learning via noise negative sample self-correction. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3663–3673. 2023.

[15] Wang, X., W. Wan, C. Geng, et al. Beyond myopia: Learning from positive and unlabeled data through holistic predictive trends. In *Neural Information Processing Systems*. 2023.

[16] Li, C., Y. Dai, L. Feng, et al. Positive and unlabeled learning with controlled probability boundary fence. In *International Conference on Machine Learning*. 2024.

[17] du Plessis, M. C., G. Niu, M. Sugiyama. Analysis of learning from positive and unlabeled data. In *Neural Information Processing Systems*, pages 703–711. 2014.

[18] Northcutt, C. G., T. Wu, I. L. Chuang. Learning with confident examples: Rank pruning for robust classification with noisy labels. In *Conference on Uncertainty in Artificial Intelligence*. 2017.

[19] Hinton, G., O. Vinyals, J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[20] Niu, G., M. C. du Plessis, T. Sakai, et al. Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In *Neural Information Processing Systems*, pages 1199–1207. 2016.

[21] Elkan, C., K. Noto. Learning classifiers from only positive and unlabeled data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 213–220. 2008.

[22] Yang, P., W. Liu, J. Y. H. Yang. Positive unlabeled learning via wrapper-based adaptive sampling. In *International Joint Conference on Artificial Intelligence*, pages 3273–3279. 2017.

[23] Hou, M., B. Chaib-Draa, C. Li, et al. Generative adversarial positive-unlabeled learning. In *International Joint Conference on Artificial Intelligence*, pages 2255–2261. 2018.

[24] Luo, C., P. Zhao, C. Chen, et al. PULNS: positive-unlabeled learning with effective negative sample selector. In *AAAI Conference on Artificial Intelligence*, pages 8784–8792. 2021.

[25] Long, L., H. Wang, Z. Jiang, et al. Positive-unlabeled learning by latent group-aware meta disambiguation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23138–23147. 2024.

[26] Zamzam, O., H. Akrami, M. Soltanolkotabi, et al. Learning a disentangling representation for PU learning. *arXiv preprint arXiv:2310.03833*, 2024.

[27] Wang, H., R. Xiao, Y. Li, et al. Pico+: Contrastive label disambiguation for robust partial label learning. *arXiv preprint arXiv:2201.08984*, 2022.

[28] Busa-Fekete, R., H. Choi, T. Dick, et al. Easy learning from label proportions. In *Neural Information Processing Systems*, pages 14957 – 14968. 2023.

[29] Sohn, K., D. Berthelot, N. Carlini, et al. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Neural Information Processing Systems*, 33:596–608, 2020.

[30] Zhang, B., Y. Wang, W. Hou, et al. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Neural Information Processing Systems*, 34:18408–18419, 2021.

[31] Berthelot, D., R. Roelofs, K. Sohn, et al. Adamatch: A unified approach to semi-supervised learning and domain adaptation. *arXiv preprint arXiv:2106.04732*, 2021.

[32] Kou, Z., S. Qin, H. Wang, et al. Label distribution learning with biased annotations by learning multi-label representation, 2025.

[33] Kou, Z., J. Wang, J. Tang, et al. Exploiting multi-label correlation in label distribution learning. In *International Joint Conference on Artificial Intelligence*, pages 4326–4334. 2024.

[34] Kou, Z., H. Xuan, J. Zhu, et al. Tail-aware reconstruction of incomplete label distributions with low-rank and sparse modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2025.

[35] Li, C., Y. Dai, L. Feng, et al. Positive and unlabeled learning with controlled probability boundary fence. In *International Conference on Machine Learning*. 2024.

[36] Hoeffding, W. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pages 409–426, 1994.

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 have succinctly outlined the contributions of this paper in both the abstract and introduction sections, and the results presented in the experiments section robustly substantiate the effectiveness of our proposed method.

   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 have created a separate "Limitations" section in our paper, the details can be found in Appendix C.1.

   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: We analyze the estimation error of SAPU in Theorem 3.2. All theoretical results are clearly supported by rigorous mathematical derivations in Appendix A and B.

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 have elaborated on the implementation principles and details of the method to facilitate the reproduction of the main experimental results presented in our paper. Additionally, we have submitted our code and datasets in the Supplementary Material.

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 have submitted our code and datasets in the Supplementary Material.

   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: We provide detailed descriptions of all necessary training and testing procedures in Section 3.1. Furthermore, all experimental setup details can be readily found in the submitted code.

   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 results regarding the standard errors of the mean in our experiments and ensure that our paper contains the calculation method for standard errors along with other essential information related to them.

   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 have provided sufficient information on the computer resources needed to reproduce our experiments in Section 3.1.

   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: We ensure that our research adheres to the NeurIPS Code of Ethics in all aspects.

   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 have created a separate "Broader Impacts" section in our paper, the details can be found in Appendix C.2.

    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: The paper pose no such risks.

    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 have provided proper citations for all models, code, and datasets utilized in our paper.

    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: [Yes]

    Justification: We have submitted the proposed new assets in the Supplementary Material. And the submitted files include structured explanatory documents regarding these new assets.

    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: Our paper does not involve crowdsourcing nor research with human subjects.

    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: Our paper does not involve crowdsourcing nor research with human subjects.

    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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Proof of Lemma 3.1

The boundary of the bag deviation Under the SCAR assumption [21], each sample in the unlabeled dataset has an independent probability π of being positive. Given a bag j containing s samples, since the variance of a Bernoulli random variable is $\pi(1-\pi)$, we can obtain a tighter bound using Bernstein's inequality [36] for any $\epsilon > 0$:

$$P(|\hat{\pi}_j - \pi| \geq \epsilon) \leq 2\exp(-\frac{S\epsilon^2}{2\pi(1-\pi) + 2\epsilon/3}) \leq \delta \tag{13}$$

Then,

$$S \geq \frac{3\pi(1-\pi)}{2\epsilon^2}\log(2/\delta) \tag{14}$$

The variance of $\hat{\pi}_j$ The variance of $\hat{\pi}_j$ can be given by the mean of S independent Bernoulli random variables:

$$\mathrm{Var}(\hat{\pi}_j) = \frac{\pi(1-\pi)}{S} \tag{15}$$

B Proof of Theorem 3.2

We decompose the excess risk as:

$$R(\hat{g}_{\mathrm{SAPU}}) - R(g^*) \leq \underbrace{|R(\hat{g}_{\mathrm{SAPU}}) - R_{\mathrm{SAPU}}(\hat{g}_{\mathrm{SAPU}})|}_{\text{Term 1}} + \underbrace{|R_{\mathrm{SAPU}}(\hat{g}_{\mathrm{SAPU}}) - R_{\mathrm{SAPU}}(g^*)|}_{\leq 0} + \underbrace{|R_{\mathrm{SAPU}}(g^*) - R(g^*)|}_{\text{Term 2}} \tag{16}$$

For Term 1, using the uniform convergence theory and the fact that the deviation in bag proportions is bounded by ϵ, we have:

$$|R(\hat{g}_{\mathrm{SAPU}}) - R_{\mathrm{SAPU}}(\hat{g}_{\mathrm{bag}})| \leq C_1\sqrt{\frac{d\log(n_p) + \log(1/\delta)}{n_p}} + C_2\sqrt{\frac{d\log(n_s) + \log(1/\delta)}{n_s}} + \epsilon L_\ell \tag{17}$$

where L_ℓ is the Lipschitz constant of the cross-entropy loss; C_1 and C_2 are universal constants; d is the pseudo-dimension of the function class.

According to Hoeffding's inequality, for any $\delta > 0$, $|\hat{\pi}_j - \pi| \leq \sqrt{\frac{\log(2/\delta)}{2S}}$ holds with probability at least $1-\delta$. Then, we have:

$$|R(\hat{g}_{\mathrm{SAPU}}) - R_{\mathrm{SAPU}}(\hat{g}_{\mathrm{bag}})| \leq C_1\sqrt{\frac{d\log(n_p) + \log(1/\delta)}{n_p}} + C_2\sqrt{\frac{d\log(n_s) + \log(1/\delta)}{n_s}} + L_\ell\sqrt{\frac{\log(2/\delta)}{2S}} \tag{18}$$

For Term 2, the deviation comes from the difference between the true positive class prior π and the bag proportions $\hat{\pi}_j$. Using the variance bound from Lemma 3.1 and applying Jensen's inequality:

$$|R_{\mathrm{SAPU}}(g^*) - R(g^*)| \leq L_\ell\sqrt{\mathbb{E}[(\hat{\pi}_j - \pi)^2]} = L_\ell\sqrt{\mathrm{Var}(\hat{\pi}_j)} = L_\ell\sqrt{\frac{\pi(1-\pi)}{S}} \tag{19}$$

Then,

$$
\begin{aligned}
R(\hat{g}_{\text{SAPU}}) - R(g^*) &\leq C_1\sqrt{\frac{d\log(n_p)+\log(1/\delta)}{n_p}} + C_2\sqrt{\frac{d\log(n_s)+\log(1/\delta)}{n_s}} \\
&\quad + L_\ell\sqrt{\frac{\log(2/\delta)}{2S}} + L_\ell\sqrt{\frac{\pi(1-\pi)}{S}} \\
&= C_1\sqrt{\frac{d\log(n_p)+\log(1/\delta)}{n_p}} + C_2\sqrt{\frac{d\log(n_s)+\log(1/\delta)}{n_s}} \\
&\quad + L_\ell\left(\sqrt{\frac{\log(2/\delta)}{2S}} + \sqrt{\frac{\pi(1-\pi)}{S}}\right) \\
&= O\left(\sqrt{\frac{\log(1/\delta)}{n_p}}\right) + O\left(\sqrt{\frac{\log(1/\delta)}{n_s}}\right) + L_\ell \cdot O\left(\sqrt{\frac{\pi(1-\pi)\log(1/\delta)}{S}}\right)
\end{aligned} \tag{20}
$$

C Limitations and Broader Impacts

C.1 Limitations

Despite our comprehensive empirical study, accurately estimating pseudo-labels remains challenging, especially with limited positive samples. Our methods could be further improved by incorporating advanced techniques such as ensemble learning to generate more reliable pseudo-labels. Additionally, the set-aware empirical risk method may face challenges with extremely imbalanced datasets where the positive class prior becomes difficult to estimate accurately.

C.2 Broader Impacts

Our GPU framework introduces a novel perspective by integrating empirical risk with pseudo-labeling methods, enhancing PU learning applicability in real-world scenarios such as medical diagnoses and fraud detection. The proposed set-aware empirical risk extends the theoretical foundation of PU learning through aggregate supervision, which could inspire further weakly-supervised learning research. By making PU learning more reliable with limited labeled data, our work contributes to reduced annotation costs and broader accessibility of machine learning in resource-constrained environments.