PARTIAL AUC MAXIMIZATION FROM POSITIVE-UNLABELED DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

The partial area under the receiver operating characteristic curve (pAUC) is an important performance metric for binary classification that summarizes true positive rates within a specific range of false positive rates (FPRs). Classifiers that achieve high pAUC need to be obtained in many real-world applications such as cybersecurity, medical care, and advertising. Although many methods for maximizing the pAUC have been proposed, they typically require both labeled positive and negative data for training. However, in practice, labeled negative data are often difficult to collect due to privacy concerns or the need for high expertise to annotate them. In this paper, we propose a method for maximizing the pAUC from positive and unlabeled (PU) data without negative data. Within an empirical risk minimization framework, we show that the pAUC, including its FPR-dependent thresholds, can be represented using only the positive and marginal densities, and derive an empirical estimator from PU data. A classifier is then trained by maximizing the derived smoothed empirical pAUC estimator. We experimentally demonstrate the effectiveness of the proposed method with ten real-world datasets.

1 INTRODUCTION

The area under the receiver operating characteristic (ROC) curve (AUC) is a widely used evaluation metric for binary classification, especially in class-imbalanced scenarios where positive data are much smaller than negative data (Bradley, 1997; Cortes & Mohri, 2003). The ROC curve is obtained by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. By maximizing the AUC, we can obtain classifiers that achieve high average TPRs over all FPR values from zero to one (Yang & Ying, 2022).

In many real-world applications, we are interested in achieving high TPRs within a specific range of FPRs. For example, in cybersecurity, detection systems often need a low FPR since falsely detecting benign software or communications as malicious intensely hinders the work of end users and operators, but they still require high TPRs to reliably detect truly malicious activities (Nishiyama et al., 2024). In medical care, a low FPR needs to be maintained to avoid unnecessary biopsies, which are invasive and impose physical and psychological burdens on patients as well as financial costs (Baker & Pinsky, 2001; Iwata et al., 2020). In advertising, a low FPR is also important since irrelevant advertisements substantially degrade the user experience. In such applications, a partial AUC (pAUC), which is the partial area under the ROC curve within a specific FPR range, is a more appropriate metric than the AUC (Yang & Ying, 2022). Figure 1 illustrates the difference between the AUC and pAUC.

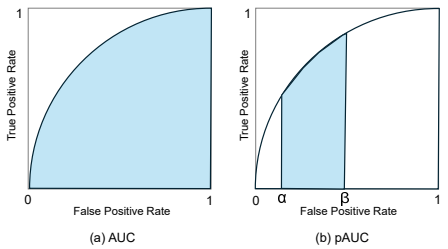


Figure 1: The comparison between the AUC and pAUC: (a) AUC. (b) pAUC within FPR range $[\alpha, \beta]$.

Although many methods for maximizing the pAUC have been proposed (Narasimhan & Agarwal, 2013a; Ueda & Fujino, 2018; Iwata et al., 2020; Yao et al., 2022; Zhu et al., 2022), they require both labeled positive and negative (PN) data for training. However, in practice, labeled negative data are often difficult to collect. For example, in cybersecurity, although some malicious data (positive

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

data) can be obtained from public blocklists, benign data (negative data) are often unavailable due to privacy reasons, and identifying benign data within unlabeled data requires a high level of expertise (Mirsky et al., 2018; Kumagai et al., 2024; 2025). In medical care and advertising, although positive data can be obtained from biopsy-confirmed cases or clicked ads, data that are not biopsy-confirmed cases or clicked ads cannot be regarded as negative because they may contain false negatives, such as early-stage patients incorrectly classified as healthy or advertisements that users are interested in but happened not to click (Bekker & Davis, 2020). Therefore, such data are better treated as unlabeled data. In this setting, most existing pAUC maximization methods cannot be applied.

The only exception is the work by Iwata et al. (2020), which proposed a semi-supervised learning method for maximizing the pAUC and presented an approximation of the pAUC from positive and unlabeled (PU) data. This method sorts the unlabeled instances by their current classifier scores and heuristically treats all instances within a specified rank range as negatives, assuming that the selected instances are indeed negative. However, the resulting objective generally does not coincide with the pAUC. Moreover, when this heuristic assumption is violated, its performance can degrade.

In this paper, we propose an approach for maximizing the pAUC from PU data by directly reformulating the pAUC without relying on this heuristic assumption. Our approach builds on the empirical risk minimization (ERM) framework of PU learning, which allows us to rewrite the risk (i.e., the objective for classifier training) using only the positive and marginal densities (Sugiyama et al., 2022; Du Plessis et al., 2015). However, unlike standard classification and AUC objectives used in prior PU learning work (Kiryo et al., 2017; Sakai et al., 2018; Xie & Li, 2018), simply applying this risk rewriting to the pAUC is insufficient because the pAUC additionally depends on FPR-dependent score thresholds, which are determined by the unknown negative density. We show that the FPR, and consequently the FPR-dependent thresholds required for the pAUC, can also be represented using only the positive and marginal densities, thereby enabling a direct reformulation of the pAUC. Based on this reformulation, we derive an empirical estimator of the pAUC from PU data. By maximizing the derived smoothed empirical pAUC estimator, we can train classifiers that achieve high pAUC even without negative data. The proposed method can use any differentiable classifiers such as linear classifiers, kernel-based classifiers, and neural network-based classifiers, which is preferable in practical use.

Our main contributions are summarized as follows: (1) We show that the pAUC, including its FPR-dependent thresholds, can be reformulated using only the positive and marginal densities, and derive an empirical estimator of the pAUC from PU data. (2) We propose a method for learning classifiers that achieve high pAUC by maximizing the derived smoothed empirical pAUC estimator with PU data. (3) We experimentally demonstrate the effectiveness of the proposed method against existing PU learning, AUC maximization, and pAUC maximization methods with ten real-world datasets.

2 Related Work

Many methods for maximizing the AUC have been proposed (Yang & Ying, 2022; Ying et al., 2016; Yuan et al., 2021b; Zhang et al., 2023; Yuan et al., 2021a). Although these methods perform well, particularly on class-imbalanced data (Yuan et al., 2021a;b), they require labeled PN data for training and are not designed for maximizing the pAUC. Since classifiers achieving high AUC are not necessarily optimal with respect to the pAUC (Narasimhan & Agarwal, 2013a;b), pAUC needs to be directly maximized.

PU learning methods have been proposed for training binary classifiers from only PU data (Bekker & Davis, 2020; Elkan & Noto, 2008). A representative approach is the ERM, which rewrites the risk using only PU data (Sugiyama et al., 2022; Du Plessis et al., 2015). Some methods use this approach to maximize the AUC with PU data (Sakai et al., 2018; Xie et al., 2024; Xie & Li, 2018). However, extending this approach to the pAUC is nontrivial because, unlike the AUC, the pAUC depends on FPR-dependent score thresholds determined by the negative density. Thus, rewriting the pairwise terms alone is insufficient. We show that these thresholds can also be represented using the positive and marginal densities, enabling the pAUC to be reformulated from PU data.

Many pAUC maximization methods have been proposed (Narasimhan & Agarwal, 2013a;b; Iwata et al., 2020; Yao et al., 2022; Zhu et al., 2022; Ueda & Fujino, 2018; Zhou et al., 2025). Although these methods are promising, they require labeled PN data for training. Iwata et al. (2020) presented

an approximation of the pAUC from PU data by heuristically treating all unlabeled instances whose ranks fall within a specified range as negative. However, the resulting objective generally does not coincide with the pAUC. In contrast, the proposed method reformulates the pAUC from PU data without relying on this heuristic. Although Xie et al. (2024) also use score-based instance selection, their method targets the full AUC rather than the pAUC.

3 PRELIMINARIES

In this section, we briefly explain AUC and pAUC. Let input instance $\mathbf{x} \in \mathcal{X}$ and its class label $y \in \{-1, +1\}$ be associated with probability density $p(\mathbf{x}, y)$, where $+1$ and -1 correspond to PN classes, respectively. The conditional probability densities for PN classes are denoted as $p^{\mathrm{p}}(\mathbf{x}) := p(\mathbf{x}|y = +1)$ and $p^{\mathrm{n}}(\mathbf{x}) := p(\mathbf{x}|y = -1)$. Let $s : \mathcal{X} \to \mathbb{R}$ be a score function that represents the relative tendency of an instance to be positive. The classifier is obtained by the score function with threshold t: $y = \mathrm{sign}(s(\mathbf{x}) - t)$, where sign denotes a sign function.

The TPR with threshold t is defined as the probability that the score of positive data is higher than threshold t:

$$\mathrm{TPR}(t) := \mathbb{E}_{\mathbf{x}\sim p^{\mathrm{p}}}\left[I(s(\mathbf{x}) > t)\right], \tag{1}$$

where $\mathbb{E}_{\mathbf{x}\sim p}[\cdot]$ is the expectation over density p, and $I(\cdot)$ is the indicator function; $I(Z) = 1$ if Z is true and $I(Z) = 0$ otherwise. Similarly, the FPR with threshold t is defined as the probability that the score of negative data is higher than threshold t:

$$\mathrm{FPR}(t) := \mathbb{E}_{\mathbf{x}\sim p^{\mathrm{n}}}\left[I(s(\mathbf{x}) > t)\right]. \tag{2}$$

The AUC is defined as the area under the ROC curve plotted by $\mathrm{TPR}(t)$ against $\mathrm{FPR}(t)$ with varying threshold t. Assuming there are no ties, it can be calculated by

$$\mathrm{AUC} := \int_0^1 \mathrm{TPR}(\mathrm{FPR}^{-1}(u))du = \mathbb{E}_{\mathbf{x}^{\mathrm{p}}\sim p^{\mathrm{p}}}\mathbb{E}_{\mathbf{x}^{\mathrm{n}}\sim p^{\mathrm{n}}}\left[I(s(\mathbf{x}^{\mathrm{p}}) > s(\mathbf{x}^{\mathrm{n}}))\right], \tag{3}$$

where

$$\mathrm{FPR}^{-1}(u) = \inf\{t \in \mathbb{R} \ : \ \mathrm{FPR}(t) \leq u\} \tag{4}$$

(Yang & Ying, 2022). $\mathrm{FPR}^{-1}(u)$ returns the lower threshold at which the FPR is no greater than u. The AUC can be regarded as the probability that the score of positive data is higher than that of negative data.

The pAUC between α and β, where $0 \leq \alpha < \beta \leq 1$, is defined as the normalized area under the ROC curve plotted by $\mathrm{TPR}(t)$ against $\mathrm{FPR}(t)$ where $\mathrm{FPR}(t)$ is between α and β as follows:

$$\begin{aligned}\mathrm{pAUC}(\alpha, \beta) &:= \frac{1}{\beta - \alpha}\int_\alpha^\beta \mathrm{TPR}(\mathrm{FPR}^{-1}(u))du \\ &= \frac{1}{\beta - \alpha}\mathbb{E}_{\mathbf{x}^{\mathrm{p}}\sim p^{\mathrm{p}}}\mathbb{E}_{\mathbf{x}^{\mathrm{n}}\sim p^{\mathrm{n}}}\left[I(s(\mathbf{x}^{\mathrm{p}}) > s(\mathbf{x}^{\mathrm{n}}))I(s(\mathbf{x}^{\mathrm{n}}) \in [t_\beta, t_\alpha])\right],\end{aligned} \tag{5}$$

where $t_\alpha := \mathrm{FPR}^{-1}(\alpha)$ and $t_\beta := \mathrm{FPR}^{-1}(\beta)$ (Yang & Ying, 2022)[1]. The pAUC between α and β is the probability that positive data have higher scores than negative data whose score is between $t_\beta = \mathrm{FPR}^{-1}(\beta)$ and $t_\alpha = \mathrm{FPR}^{-1}(\alpha)$. Due to the normalization factor $1/(\beta - \alpha)$, the pAUC takes values between zero and one.

Given N^{p} positive instances $\{\mathbf{x}_1^{\mathrm{p}}, \ldots, \mathbf{x}_{N^{\mathrm{p}}}^{\mathrm{p}}\}$ drawn from $p^{\mathrm{p}}(\mathbf{x})$ and N^{n} negative instances $\{\mathbf{x}_1^{\mathrm{n}}, \ldots, \mathbf{x}_{N^{\mathrm{n}}}^{\mathrm{n}}\}$ drawn from $p^{\mathrm{n}}(\mathbf{x})$, the empirical estimator of the pAUC is calculated as

$$\widehat{\mathrm{pAUC}}(\alpha, \beta) = \frac{1}{(\beta - \alpha)N^{\mathrm{p}}N^{\mathrm{n}}}\sum_{n=1}^{N^{\mathrm{p}}}\sum_{m=j_\alpha+1}^{j_\beta} I(s(\mathbf{x}_n^{\mathrm{p}}) > s(\mathbf{x}_{(m)}^{\mathrm{n}})), \tag{6}$$

where $j_\alpha := \alpha N^{\mathrm{n}}$, $j_\beta := \beta N^{\mathrm{n}}$, and $\mathbf{x}_{(m)}^{\mathrm{n}}$ is the negative instance ranked in the m-th position among negative data in a descending order of scores $s(\mathbf{x})$[2] (Dodd & Pepe, 2003). The empirical pAUC is the empirical probability that positive data have higher scores than negative data that are ranked between j_α and j_β.

[1] The second equality is exact when the negative-score distribution has no mass at the boundary thresholds.

[2] For simplicity, we assumed that j_α and j_β are both integers following the previous studies (Yao et al., 2022; Narasimhan & Agarwal, 2013b).

4 PROPOSED METHOD

We first explain our problem setting in Section 4.1. Then, we derive the empirical estimator of the pAUC with PU data in Section 4.2. Lastly, we describe the pAUC maximization method with the derived pAUC estimator in Section 4.3.

4.1 PROBLEM SETTING

Suppose that we are given a set of positive instances X^{p} and a set of unlabeled instances X:

$$X^{\mathrm{p}} := \{\mathbf{x}_n^{\mathrm{p}}\}_{n=1}^{N^{\mathrm{p}}} \overset{\text{i.i.d.}}{\sim} p^{\mathrm{p}}(\mathbf{x}), \;\; X := \{\mathbf{x}_n\}_{n=1}^{N} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x}) = \pi p^{\mathrm{p}}(\mathbf{x}) + (1-\pi)p^{\mathrm{n}}(\mathbf{x}), \tag{7}$$

where $p(\mathbf{x})$ is the marginal density, and $\pi := p(y=+1)$ is the positive class-prior. We assume that π is known as in the previous PU learning studies (Du Plessis et al., 2015; Kiryo et al., 2017; Iwata et al., 2020). When π is unknown, it can be estimated from PU data (Scott, 2015; Ramaswamy et al., 2016; Yao et al., 2021), as also illustrated in our experiments. Our goal is to learn score function s that can maximize the pAUC between α and β from given PU data $X^{\mathrm{p}} \cup X$.

4.2 PAUC ESTIMATOR WITH PU DATA

We derive the empirical estimator of the pAUC with PU data. First, from the definition of marginal density $p(\mathbf{x}) = \pi p^{\mathrm{p}}(\mathbf{x}) + (1-\pi)p^{\mathrm{n}}(\mathbf{x})$ in Eq. (7), the negative density can be represented as

$$p^{\mathrm{n}}(\mathbf{x}) = \frac{1}{1-\pi}\left[p(\mathbf{x}) - \pi p^{\mathrm{p}}(\mathbf{x})\right]. \tag{8}$$

By substituting Eq. (8) into Eq. (5), we obtain

$$\begin{aligned}\mathrm{pAUC}(\alpha,\beta) =& \frac{1}{(\beta-\alpha)(1-\pi)}\left[\mathbb{E}_{\mathbf{x}^{\mathrm{p}}\sim p^{\mathrm{p}}}\mathbb{E}_{\mathbf{x}\sim p}\left[I(s(\mathbf{x}^{\mathrm{p}}) > s(\mathbf{x}))I(s(\mathbf{x}) \in [t_\beta, t_\alpha])\right]\right. \\ & \left. -\pi\mathbb{E}_{\mathbf{x}^{\mathrm{p}}\sim p^{\mathrm{p}}}\mathbb{E}_{\bar{\mathbf{x}}^{\mathrm{p}}\sim p^{\mathrm{p}}}\left[I(s(\mathbf{x}^{\mathrm{p}}) > s(\bar{\mathbf{x}}^{\mathrm{p}}))I(s(\bar{\mathbf{x}}^{\mathrm{p}}) \in [t_\beta, t_\alpha])\right]\right]. \end{aligned} \tag{9}$$

In Eq. (9), each expectation depends on positive or marginal densities; they can be approximated with PU data $X^{\mathrm{p}} \cup X$. However, thresholds $t_\alpha = \mathrm{FPR}^{-1}(\alpha)$ and $t_\beta = \mathrm{FPR}^{-1}(\beta)$ still depend on negative density $p^{\mathrm{n}}(\mathbf{x})$. Therefore, we next represent thresholds $t_\alpha = \mathrm{FPR}^{-1}(\alpha)$ and $t_\beta = \mathrm{FPR}^{-1}(\beta)$ using only the positive and marginal densities.

From Eq. (8), the FPR in Eq. (2) can be also rewritten as

$$\mathrm{FPR}(t) = \mathbb{E}_{\mathbf{x}\sim p^{\mathrm{n}}}\left[I(s(\mathbf{x}) > t)\right] = \frac{1}{1-\pi}\left[\mathbb{E}_{\mathbf{x}\sim p}\left[I(s(\mathbf{x}) > t)\right] - \pi\mathbb{E}_{\mathbf{x}^{\mathrm{p}}\sim p^{\mathrm{p}}}\left[I(s(\mathbf{x}^{\mathrm{p}}) > t)\right]\right]. \tag{10}$$

Since $t_\alpha = \mathrm{FPR}^{-1}(\alpha)$ and $t_\beta = \mathrm{FPR}^{-1}(\beta)$, Eq. (10) implies that these population thresholds are determined only by the positive and marginal densities. Thus, Eqs. (9) and (10) show that the pAUC, including its thresholds, can be represented using only the positive and marginal densities. From Eq. (10), the empirical estimator of the FPR with threshold t is represented as

$$\widehat{\mathrm{FPR}}(t) = \frac{1}{1-\pi}\left[\frac{1}{N}\sum_{n=1}^{N} I(s(\mathbf{x}_n) > t) - \frac{\pi}{N^{\mathrm{p}}}\sum_{n=1}^{N^{\mathrm{p}}} I(s(\mathbf{x}_n^{\mathrm{p}}) > t)\right]. \tag{11}$$

Since $t_\alpha = \mathrm{FPR}^{-1}(\alpha) = \inf\{t \in \mathbb{R} \;:\; \mathrm{FPR}(t) \le \alpha\}$, we want to estimate the lower threshold at which Eq. (11) is no greater than α. Since the value of Eq. (11) changes discretely when threshold t crosses the score of training instance, we can determine threshold $\hat{t}_\alpha$ by simply examining the scores of training PU data $X^{\mathrm{p}} \cup X$, as shown in Figure 2. Specifically, threshold $\hat{t}_\alpha$ can be determined as

$$\hat{t}_\alpha = \min\{s(\mathbf{x}) \;:\; \widehat{\mathrm{FPR}}(s(\mathbf{x})) \le \alpha, \mathbf{x} \in X^{\mathrm{p}} \cup X\}. \tag{12}$$

Similarly, threshold $\hat{t}_\beta$ can be determined as

$$\hat{t}_\beta = \min\{s(\mathbf{x}) \;:\; \widehat{\mathrm{FPR}}(s(\mathbf{x})) \le \beta, \mathbf{x} \in X^{\mathrm{p}} \cup X\}. \tag{13}$$

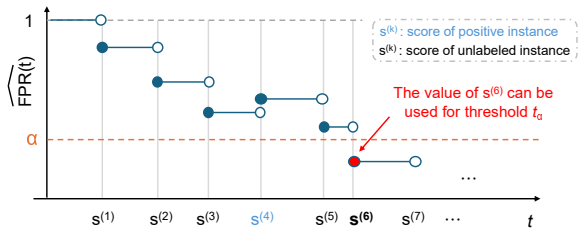


Figure 2: A visual explanation of determining threshold $\hat{t}_\alpha$: The vertical and horizontal axes represent $\widehat{\mathrm{FPR}}(t)$ and threshold t, respectively. $s^{(k)}$ denotes the k-th smallest score within the scores of all training data $\{s(\mathbf{x})\}$, where $\mathbf{x} \in X^{\mathrm{P}} \cup X$. From Eq. (11), $\widehat{\mathrm{FPR}}(t) = 1$ for any t smaller than $s^{(1)}$ and $\widehat{\mathrm{FPR}}(t) = 0$ for any t greater than or equal to $s^{(N^{\mathrm{P}}+N)}$. $\widehat{\mathrm{FPR}}(t)$ is a discrete function where its value changes only when t crosses one of the score $s^{(k)}$. When threshold t exceeds the score of the unlabeled (positive) instance, $\widehat{\mathrm{FPR}}(t)$ decreases (increases). By checking $s^{(1)}, s^{(2)}, \cdots$ in ascending order and identifying the first (minimum) score at which $\widehat{\mathrm{FPR}}(t) \leq \alpha$, we can take the score ($s^{(6)}$ in the figure) as the threshold $\hat{t}_\alpha$.

Note that when the value of Eq. (11) was less than zero or greater than one, we clipped the value to zero and one, respectively.

By substituting estimated thresholds $\hat{t}_\alpha$ and $\hat{t}_\beta$ into Eq. (9), the empirical estimator of the pAUC can be represented as

$$\widehat{\mathrm{pAUC}}(\alpha, \beta) = \frac{1}{(\beta - \alpha)(1 - \pi)} \left[\sum_{n=1}^{N^{\mathrm{P}}} \sum_{m=1}^{N} \left[\frac{I(s(\mathbf{x}_n^{\mathrm{P}}) > s(\mathbf{x}_m)) I(s(\mathbf{x}_m) \in [\hat{t}_\beta, \hat{t}_\alpha])}{N N^{\mathrm{P}}} \right] - \pi \sum_{n \neq m}^{N^{\mathrm{P}}, N^{\mathrm{P}}} \left[\frac{I(s(\mathbf{x}_n^{\mathrm{P}}) > s(\mathbf{x}_m^{\mathrm{P}})) I(s(\mathbf{x}_m^{\mathrm{P}}) \in [\hat{t}_\beta, \hat{t}_\alpha])}{N^{\mathrm{P}}(N^{\mathrm{P}} - 1)} \right] \right], \tag{14}$$

where we omitted the case of $n = m$ in the second term to avoid biases following the study (Sakai et al., 2018). The finite-sample properties of this estimator are analyzed in Section A.

4.3 pAUC Maximization with PU Data

We explain the pAUC maximization method using Eq. (14). Since the gradient of the indicator function $I(\cdot)$ is zero almost everywhere, the empirical pAUC in Eq. (14) cannot be maximized by using gradient descent-based methods. To make the empirical pAUC differentiable, following the previous studies (Iwata et al., 2020; Kumagai et al., 2024; 2025), we used the sigmoid function $\sigma(s(\mathbf{x}) - s(\mathbf{x}')) = 1/(1 + \exp(-s(\mathbf{x}) + s(\mathbf{x}')))$ instead of $I(s(\mathbf{x}) > s(\mathbf{x}'))$ in Eq. (14):

$$\widetilde{\mathrm{pAUC}}(\alpha, \beta) = \frac{1}{(\beta - \alpha)(1 - \pi)} \left[\sum_{n=1}^{N^{\mathrm{P}}} \sum_{m=1}^{N} \left[\frac{\sigma(s(\mathbf{x}_n^{\mathrm{P}}) - s(\mathbf{x}_m)) I(s(\mathbf{x}_m) \in [\hat{t}_\beta, \hat{t}_\alpha])}{N N^{\mathrm{P}}} \right] - \pi \sum_{n \neq m}^{N^{\mathrm{P}}, N^{\mathrm{P}}} \left[\frac{\sigma(s(\mathbf{x}_n^{\mathrm{P}}) - s(\mathbf{x}_m^{\mathrm{P}})) I(s(\mathbf{x}_m^{\mathrm{P}}) \in [\hat{t}_\beta, \hat{t}_\alpha])}{N^{\mathrm{P}}(N^{\mathrm{P}} - 1)} \right] \right]. \tag{15}$$

Here, we do not replace the indicator function term for instance selection $I(s(\mathbf{x}) \in [\hat{t}_\beta, \hat{t}_\alpha])$ with the sigmoid function for simplicity and stability. This treatment is standard in previous pAUC maximization studies (Ueda & Fujino, 2018; Iwata et al., 2020; Yao et al., 2022). Thus, gradients are only propagated through the sigmoid function terms during training. Note that we can use any surrogate functions such as logistic and ramp functions instead of the sigmoid function (Sakai et al., 2018). Maximizing $\widetilde{\mathrm{pAUC}}(\alpha, \beta)$ is equivalent to minimizing $1 - \widetilde{\mathrm{pAUC}}(\alpha, \beta)$. Therefore, our loss function for learning s is given as

$$\mathcal{L}(s; \alpha, \beta) = 1 - \widetilde{\mathrm{pAUC}}(\alpha, \beta). \tag{16}$$

Algorithm 1 Training procedure of the proposed method

Require: PU data $X^{\mathrm{P}} \cup X$, positive class-prior π, FPR range values α and β, positive mini-batch size P, and unlabeled mini-batch size U
Ensure: Model parameters of score function s
1: **repeat**
2: Sample unlabeled data with size U from X
3: Sample positive data with size P from X^{P}
4: Calculate thresholds $\hat{t}_\alpha$ and $\hat{t}_\beta$ in Eqs. (12) and (13) with the sampled data
5: Calculate the loss in Eq. (16) on the sampled data
6: Update parameters of s with the gradient of the loss
7: **until** End condition is satisfied;

Algorithm 1 shows the training procedure of the proposed method with stochastic gradient descent methods. We first sample PU data from $X^{\mathrm{P}} \cup X$ (Lines 2–3). Then, we calculate thresholds $\hat{t}_\alpha$ and $\hat{t}_\beta$ by using Eqs. (12) and (13), respectively (Line 4). By using the estimated thresholds, we calculate the loss in Eq. (16) for learning score function s (Line 5). Here, estimating the thresholds and evaluating the loss on the same mini-batch is natural, because the thresholds themselves are part of the empirical pAUC objective. Lastly, we update parameters of score function s by using the gradient of the loss (Line 6).

5 Experiments

5.1 Data

We used eight real-world datasets in the main paper: MNIST (LeCun et al., 1998), Fashion MNIST (FMNIST) (Xiao et al., 2017), SVHN (Netzer et al., 2011), CIFAR10 (Krizhevsky et al., 2009), SUSY (Baldi et al., 2014), CODRNA (Uzilov et al., 2006), IJCNN (Prokhorov, 2001), and EPSILON (Tsoumakas et al., 2008). These datasets have been commonly used in PU learning studies (Kumagai et al., 2024; 2025; Xie et al., 2024; Kiryo et al., 2017; Jiang et al., 2023; Sakai & Shimizu, 2019; Hammoudeh & Lowd, 2020). In Section C.6, we used Mammography (Vanschoren et al., 2014) and Online-shoppers (Sakar et al., 2019), which are medical- and advertising-related datasets.

MNIST comprises handwritten images of 10 digit classes, and FMNIST comprises images belonging to 10 fashion categories. Each image in MNIST and FMNIST is represented as a 28×28 grayscale image. SVHN consists of images of 10 printed digits cropped from photographs of house number plates, while CIFAR-10 consists of images of 10 animal and vehicle categories. Each image in SVHN and CIFAR-10 is represented as a 32×32 RGB image. SUSY, CODRNA, IJCNN, and EPSILON are binary tabular datasets taken from the site of LIBSVM (Chang & Lin, 2011). The numbers of features of SUSY, CODRNA, IJCNN, and EPSILON are 18, 8, 22, and 2,000, respectively. For image datasets, we created binary classification tasks from each dataset following the data creation procedure of previous studies (Kumagai et al., 2024; 2025; Xie et al., 2024). Specifically, for MNIST and SVHN, we used odd digits as positive and even digits as negative. For FMNIST, we used non-upper garment classes (1, 5, 7, 8, and 9) as positive and the others as negative. For CIFAR10, we used the vehicle classes (0, 1, 8, and 9) as positive and the animal classes as negative.

For training in each dataset, we used 5,000 unlabeled data and changed the numbers of positive data within $\{30, 50, 100\}$. Additionally, we used 50 positive and 1,000 unlabeled data for validation. For unlabeled data in training and validation, we changed class-prior π within $\{0.1, 0.2\}$. We used 2,500 positive and 2,500 negative data as test data for evaluation. Training, validation, and test data did not overlap. We conducted 10 different experiments for each positive class-prior and labeled positive data size changing the random seeds and evaluated mean test pAUC.

5.2 Comparison Methods

We compared the proposed method (Ours) with five methods: non-negative PU learning method (NNPU) (Kiryo et al., 2017), AUC maximization method with PU data (PUAUC) (Xie & Li, 2018; Xie et al., 2024), pAUC maximization method (PAUC) (Ueda & Fujino, 2018), pAUC maximization

Table 1: Test average pAUCs and their standard deviations with $\alpha = 0$, $\beta = 0.1$ over different labeled positive data sizes $N^{\mathrm{p}} = 30$, 50, and 100. Values in bold are not statistically different at the 5% level from the best performing method in each row according to a paired t-test. '# Best' row represents the number of results of each method that are the best or comparable to it.

Data	π	Ours	NNPU	PUAUC	PAUC	PUpAUCh	2step
MNIST	0.1	**0.683 ± 0.074**	0.522 ± 0.131	0.627 ± 0.056	0.597 ± 0.070	0.675 ± 0.073	0.633 ± 0.058
	0.2	**0.633 ± 0.071**	0.577 ± 0.068	0.592 ± 0.054	0.473 ± 0.058	0.624 ± 0.064	0.592 ± 0.054
FMNIST	0.1	**0.924 ± 0.025**	0.756 ± 0.022	0.895 ± 0.037	0.824 ± 0.064	**0.925 ± 0.028**	0.907 ± 0.037
	0.2	**0.914 ± 0.028**	0.770 ± 0.035	0.867 ± 0.047	0.649 ± 0.119	**0.912 ± 0.036**	0.873 ± 0.047
SVHN	0.1	**0.247 ± 0.081**	0.055 ± 0.004	0.208 ± 0.070	0.201 ± 0.053	0.233 ± 0.078	0.202 ± 0.075
	0.2	**0.191 ± 0.073**	0.054 ± 0.004	0.171 ± 0.053	0.161 ± 0.032	0.152 ± 0.065	0.157 ± 0.053
CIFAR10	0.1	**0.478 ± 0.078**	0.050 ± 0.039	0.459 ± 0.078	0.362 ± 0.075	0.466 ± 0.076	**0.469 ± 0.079**
	0.2	**0.433 ± 0.075**	0.070 ± 0.069	**0.420 ± 0.078**	0.256 ± 0.043	**0.426 ± 0.075**	**0.423 ± 0.079**
SUSY	0.1	**0.454 ± 0.034**	0.357 ± 0.078	0.414 ± 0.051	0.366 ± 0.055	0.437 ± 0.032	0.415 ± 0.052
	0.2	**0.424 ± 0.045**	0.370 ± 0.058	0.389 ± 0.054	0.245 ± 0.052	0.389 ± 0.047	0.385 ± 0.054
CODRNA	0.1	0.741 ± 0.047	0.317 ± 0.266	0.749 ± 0.057	0.630 ± 0.071	0.744 ± 0.050	**0.754 ± 0.054**
	0.2	**0.716 ± 0.056**	0.691 ± 0.064	0.719 ± 0.062	0.495 ± 0.085	**0.720 ± 0.060**	**0.726 ± 0.060**
IJCNN	0.1	**0.620 ± 0.094**	0.264 ± 0.118	0.563 ± 0.087	0.525 ± 0.071	0.600 ± 0.100	0.572 ± 0.090
	0.2	**0.565 ± 0.096**	0.449 ± 0.088	0.525 ± 0.083	0.393 ± 0.062	0.440 ± 0.194	0.524 ± 0.085
EPSILON	0.1	**0.193 ± 0.041**	0.078 ± 0.006	**0.191 ± 0.043**	0.175 ± 0.034	**0.193 ± 0.044**	**0.190 ± 0.046**
	0.2	**0.163 ± 0.045**	0.081 ± 0.010	**0.170 ± 0.039**	0.139 ± 0.026	**0.162 ± 0.039**	**0.172 ± 0.043**
# Best		15	0	3	0	6	6

method with PU data based on the heuristic scoring assumption (PUpAUCh) (Iwata et al., 2020), and two-step pAUC maximization method with PU data (2step). All methods including the proposed method used neural networks for modeling classifiers (score functions).

NNPU is a widely used PU learning method for balanced binary classification, which trains the classifier by minimizing the non-negative classification risk rewritten with PU data. PUAUC trains the classifier by maximizing the AUC rewritten with PU data. PAUC maximizes the pAUC by using PU data regarding all unlabeled data as negative. PUpAUCh maximizes the pAUC from PU data by heuristically treating all unlabeled instances within a specified rank range based on their current scores as negative during training. 2step first trains a classifier by using PUAUC, and then assigns pseudo PN labels to the unlabeled data by using the trained classifier. Specifically, it selects the top πN unlabeled instances according to the classifier scores as pseudo-positive, and assigns the remaining instances as pseudo-negative. Then, it applied PAUC for the positive and pseudo PN data to train the classifier. The proposed method, NNPU, PUpAUCh, and 2step used class-prior value π for training. The details of settings such as network architectures are described in section B.

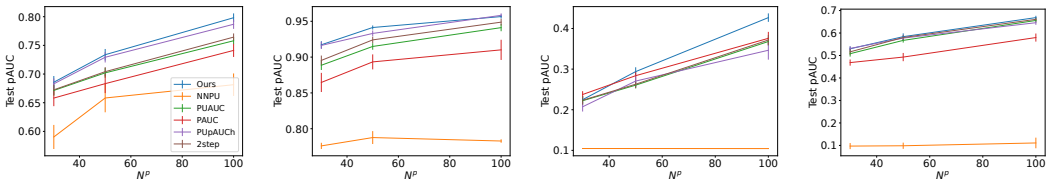
5.3 Results

Tables 1 and 2 show the average test pAUCs and their standard deviations for two settings $\alpha = 0$, $\beta = 0.1$, and $\alpha = 0$, $\beta = 0.2$, respectively. We used pAUC with $\alpha = 0$ since it is often used in previous studies and real-world applications (Narasimhan & Agarwal, 2013a; Iwata et al., 2020; Zhu et al., 2022). The results with $\alpha = 0.1$ and $\beta = 0.2$ are also reported in Section C.1. The proposed method performed the best or comparably to it in almost all cases (31 out of 32). NNPU did not work well since it is not designed for both class-imbalanced data and pAUC maximization. PUAUC outperformed NNPU since maximizing the AUC is appropriate for class-imbalanced data. However, it was often outperformed by the proposed method by a large margin, which indicates the importance of explicitly maximizing the pAUC. PAUC did not perform well either even though it aims to maximize the pAUC. This is because it merely treats all unlabeled data as negative, which leads to very biased results. PUpAUCh outperformed these methods (NNPU, PUAUC, and PAUC) since it is explicitly designed for maximizing the pAUC with PU data. However, it is often outperformed by the proposed method. Section C.8 shows that its heuristic assumption is frequently violated during training. Although 2step performed comparably to PUpAUCh, its performance was limited by the label noise inherent in the pseudo PN data. Overall, these results show the effectiveness of our method for maximizing the pAUC from PU data.

As for the results of each class-prior π in Tables 1 and 2, all methods except NNPU tended to perform worse as π increased. This is because these (p)AUC maximization-based methods include a term in their objective functions that encourages the scores of positive data to be higher than those of unlabeled data (e.g., the first term in Eq. (15) for the proposed method). When π is small, the

Table 2: Test average pAUCs and their standard deviations with $\alpha = 0$, $\beta = 0.2$ over different labeled positive data sizes $N^{\mathrm{P}} = 30$, 50, and 100. Values in bold are not statistically different at the 5% level from the best performing method in each row according to a paired t-test. '# Best' row represents the number of results of each method that are the best or comparable to it.

Data	π	Ours	NNPU	PUAUC	PAUC	PUpAUCh	2step
MNIST	0.1	**0.755 ± 0.056**	0.610 ± 0.127	0.725 ± 0.048	0.736 ± 0.055	0.749 ± 0.055	0.730 ± 0.050
	0.2	**0.723 ± 0.065**	0.677 ± 0.062	0.696 ± 0.049	0.652 ± 0.062	**0.717 ± 0.056**	0.697 ± 0.049
FMNIST	0.1	**0.943 ± 0.022**	0.775 ± 0.016	0.925 ± 0.028	0.926 ± 0.032	**0.940 ± 0.023**	0.934 ± 0.027
	0.2	**0.934 ± 0.023**	0.789 ± 0.033	0.904 ± 0.034	0.853 ± 0.059	**0.931 ± 0.026**	0.911 ± 0.033
SVHN	0.1	**0.342 ± 0.092**	0.105 ± 0.005	0.305 ± 0.081	**0.334 ± 0.086**	0.329 ± 0.089	0.313 ± 0.084
	0.2	**0.289 ± 0.090**	0.104 ± 0.005	0.262 ± 0.063	0.264 ± 0.052	0.221 ± 0.067	0.259 ± 0.064
CIFAR10	0.1	**0.610 ± 0.074**	0.089 ± 0.058	0.595 ± 0.072	0.569 ± 0.070	**0.600 ± 0.062**	**0.608 ± 0.070**
	0.2	**0.578 ± 0.075**	0.116 ± 0.094	0.559 ± 0.078	0.459 ± 0.070	**0.569 ± 0.066**	**0.564 ± 0.077**
SUSY	0.1	**0.553 ± 0.032**	0.457 ± 0.075	0.522 ± 0.041	0.528 ± 0.046	0.540 ± 0.028	0.523 ± 0.042
	0.2	**0.528 ± 0.044**	0.474 ± 0.051	0.499 ± 0.047	0.434 ± 0.056	0.508 ± 0.039	0.500 ± 0.045
CODRNA	0.1	**0.853 ± 0.034**	0.386 ± 0.292	0.850 ± 0.041	0.830 ± 0.047	**0.855 ± 0.035**	**0.856 ± 0.038**
	0.2	**0.838 ± 0.039**	0.791 ± 0.059	0.827 ± 0.048	0.705 ± 0.072	**0.839 ± 0.039**	**0.833 ± 0.047**
IJCNN	0.1	**0.711 ± 0.079**	0.367 ± 0.111	0.670 ± 0.084	0.687 ± 0.073	0.687 ± 0.101	0.683 ± 0.084
	0.2	**0.659 ± 0.102**	0.543 ± 0.092	0.637 ± 0.087	0.573 ± 0.068	0.599 ± 0.132	0.641 ± 0.089
EPSILON	0.1	**0.305 ± 0.058**	0.128 ± 0.007	**0.298 ± 0.055**	0.290 ± 0.043	**0.304 ± 0.053**	**0.299 ± 0.060**
	0.2	**0.269 ± 0.046**	0.132 ± 0.012	**0.260 ± 0.050**	0.248 ± 0.037	**0.261 ± 0.053**	**0.270 ± 0.055**
# Best		16	0	2	1	9	6



(a) MNIST (b) FMNIST (c) SVHN (d) CIFAR10

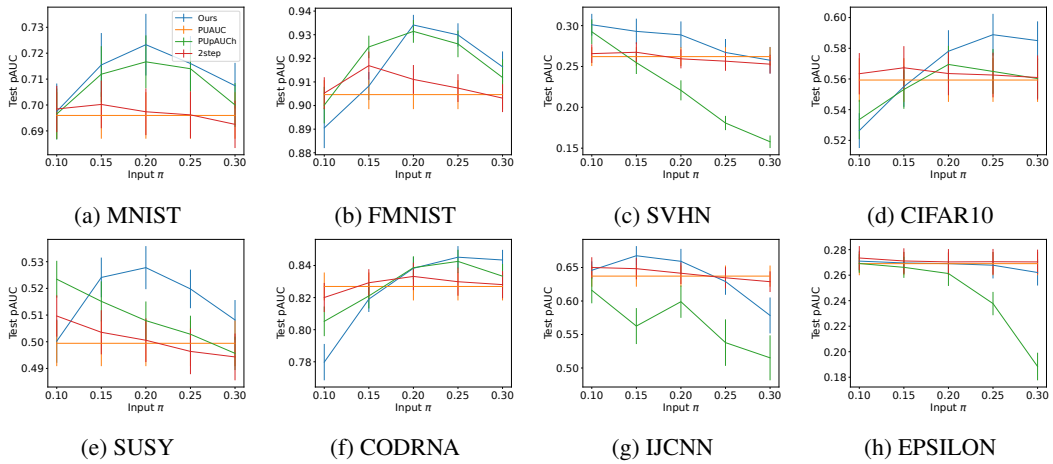
(e) SUSY (f) CODRNA (g) IJCNN (h) EPSILON

Figure 3: The average test pAUCs and their standard errors with $\alpha = 0$, $\beta = 0.2$ over different class-priors $\pi \in \{0.1, 0.2\}$ when changing the number of labeled positive data N^{P}.

unlabeled data can be regarded as mostly negative and thus the corresponding objective function term becomes more suitable for learning accurate score functions that can rank positive data higher than negative ones. As a result, the performance tends to improve (degrade) as π is decreased (increased). On the other hand, since increasing π reduces the severity of class-imbalance, NNPU tended to perform better. Nevertheless, since our focus is on the class-imbalanced problem, these results support the effectiveness of the proposed method. In Section C.2, we further confirmed the effectiveness of the proposed method under smaller and larger class-prior values.

Figure 3 shows the average test pAUCs with their standard errors when changing the number of labeled positive data N^{P}. The proposed method consistently performed well across different numbers of labeled positive data. Since labeled positive data are informative for training, the proposed method tended to perform better as N^{P} increased. In Section C.4, we also evaluated the proposed method with different unlabeled training data sizes N and found that it performed well across different N.

pAUC maximization methods with PU data (the proposed method, PUpAUCh, and 2step) need to know true class-prior π for training. Although we evaluated these methods with true class-priors in our experiments following the previous PU learning studies (Iwata et al., 2020; Kiryo et al., 2017; Du Plessis et al., 2015), there might be some noise in practice. Therefore, we investigated how the performance of the proposed method changes when using noisy class-priors for training. Figure 4



(a) MNIST (b) FMNIST (c) SVHN (d) CIFAR10

(e) SUSY (f) CODRNA (g) IJCNN (h) EPSILON

Figure 4: The average test pAUCs and their standard errors with $\alpha = 0$, $\beta = 0.2$ when true class-prior $\pi = 0.2$. 'Input π' means the class-prior value used for training.

Table 3: The results using estimated class-prior when the true class-prior is $\pi = 0.2$: test average pAUCs and their standard deviations with $\alpha = 0$, $\beta = 0.2$ over different labeled positive data sizes $N^{\mathrm{P}} = 30$, 50, and 100. Values in bold are not statistically different at the 5% level from the best performing method in each row according to a paired t-test. '# Best' row represents the number of results of each method that are the best or comparable to it.

Data	π	Ours	NNPU	PUAUC	PAUC	PUpAUCh	2step
MNIST	0.2	**0.721 ± 0.061**	0.676 ± 0.065	0.696 ± 0.049	0.652 ± 0.062	**0.716 ± 0.053**	0.697 ± 0.049
FMNIST	0.2	**0.933 ± 0.024**	0.808 ± 0.055	0.904 ± 0.034	0.853 ± 0.059	**0.932 ± 0.026**	0.910 ± 0.033
SVHN	0.2	**0.301 ± 0.073**	0.104 ± 0.005	0.262 ± 0.063	0.264 ± 0.052	**0.292 ± 0.083**	0.266 ± 0.063
CIFAR10	0.2	**0.571 ± 0.064**	0.494 ± 0.078	**0.559 ± 0.078**	0.459 ± 0.070	0.539 ± 0.081	**0.556 ± 0.078**
SUSY	0.2	**0.523 ± 0.044**	0.488 ± 0.051	0.499 ± 0.047	0.434 ± 0.056	0.505 ± 0.039	0.498 ± 0.046
CODRNA	0.2	0.821 ± 0.043	0.639 ± 0.167	0.827 ± 0.048	0.705 ± 0.072	**0.824 ± 0.043**	**0.829 ± 0.047**
IJCNN	0.2	**0.646 ± 0.043**	0.198 ± 0.043	0.637 ± 0.087	0.573 ± 0.068	0.616 ± 0.106	**0.650 ± 0.084**
EPSILON	0.2	**0.274 ± 0.049**	0.128 ± 0.008	**0.269 ± 0.050**	0.248 ± 0.037	**0.270 ± 0.046**	**0.270 ± 0.053**
# Best		7	0	2	0	5	4

shows the results. Here, PUAUC is a strong baseline that does not depend on π for training. The proposed method performed well in many cases, even when the class-prior used for training (Input π) differed from true class-prior π. These results show that the proposed method robustly works well as long as noise on the class-prior is not so large.

Furthermore, we examined the performance of the proposed method when using the class-priors estimated from PU data. For this, we used the kernel embedding-based class-prior estimation method (Ramaswamy et al., 2016). The estimated class-priors of MNIST, FMNIST, SVHN, CIFAR10, SUSY, CODRNA, IJCNN, and EPSILON were 0.214, 0.208, 0.096, 0.382, 0.226, 0.159, 0.113, and 0.159, respectively. Table 3 shows the average test pAUCs and their standard deviations when using the estimated class-priors. The proposed method still performed well. These results suggest that the proposed method remains effective and practically applicable even when the class-priors are estimated from data.

6 Conclusion

We proposed a method for maximizing the pAUC from PU data by representing the pAUC, including its FPR-dependent thresholds, using only the positive and marginal densities. Based on this reformulation, we derived an empirical estimator whose smoothed version enables classifier training without negative data. Experiments on ten real-world datasets demonstrated its effectiveness compared with existing PU learning, AUC maximization, and pAUC maximization methods.

AI USE STATEMENT

In this work, we used generative AI tools in the theoretical analysis section to assist in writing proofs and to provide ingredients for proving mathematical claims. We also used generative AI tools to edit the paper to improve readability and clarity. We carefully verified and, where necessary, corrected the proofs. The problem formulation, proposed risk reformulation, and methodological ideas were developed by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work proposes a general machine learning method and does not involve the collection of new human-subject data. All experiments were conducted using existing publicly available benchmark datasets. We are not aware of any specific ethical concerns associated with this work.

REPRODUCIBILITY STATEMENT

The details of the datasets used in our experiments are described in Sections 5.1 and C.6, and the experimental settings such as the neural network architectures are described in Section B. The pseudocode for our training procedure is provided in Algorithm 1. In addition, the assumptions underlying the proposed method are stated in Sections 3, 4, and A, and the theoretical analysis and proofs are provided in Section A.

REFERENCES

Stuart G Baker and Paul F Pinsky. A proposed design and analysis for comparing digital and analog mammography: special receiver operating characteristic methods for cancer screening. *Journal of the American Statistical Association*, 96(454):421–428, 2001.

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1):4308, 2014.

Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: a survey. *Machine Learning*, 109(4):719–760, 2020.

Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Corinna Cortes and Mehryar Mohri. Auc optimization vs. error rate minimization. *NeurIPS*, 2003.

Lori E Dodd and Margaret S Pepe. Partial auc estimation and regression. *Biometrics*, 59(3):614–623, 2003.

Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *ICML*, 2015.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *SIGKDD*, 2008.

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

Tomoharu Iwata, Akinori Fujino, and Naonori Ueda. Semi-supervised learning for maximizing the partial auc. In *AAAI*, 2020.

Yangbangyan Jiang, Qianqian Xu, Yunrui Zhao, Zhiyong Yang, Peisong Wen, Xiaochun Cao, and Qingming Huang. Positive-unlabeled learning with label distribution alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Diederik P Kingma and Jimmy Ba. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. *NeurIPS*, 2017.

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

Atsutoshi Kumagai, Tomoharu Iwata, Hiroshi Takahashi, Taishi Nishiyama, and Yasuhiro Fujiwara. Auc maximization under positive distribution shift. In *NeurIPS*, 2024.

Atsutoshi Kumagai, Tomoharu Iwata, Hiroshi Takahashi, Taishi Nishiyama, Kazuki Adachi, and Yasuhiro Fujiwara. Positive-unlabeled auc maximization under covariate shift. In *ICML*, 2025.

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

Pascal Massart. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *The annals of Probability*, pp. 1269–1283, 1990.

Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: an ensemble of autoencoders for online network intrusion detection. In *NDSS*, 2018.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*, volume 2. MIT press Cambridge, 2012.

Harikrishna Narasimhan and Shivani Agarwal. A structural svm based approach for optimizing partial auc. In *ICML*, 2013a.

Harikrishna Narasimhan and Shivani Agarwal. Svmpauctight: a new support vector method for optimizing partial auc based on a tight convex upper bound. In *KDD*, 2013b.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 7. Granada, Spain, 2011.

Taishi Nishiyama, Atsutoshi Kumagai, Akinori Fujino, and Kazunori Kamiya. Partial auc maximization for security log analysis robust to overfitting and noisy labels. In *GLOBECOM*, 2024.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Danil Prokhorov. Ijcnn 2001 neural network competition. *Slide presentation in IJCNN*, 1(97):38, 2001.

Harish Ramaswamy, Clayton Scott, and Ambuj Tewari. Mixture proportion estimation via kernel embeddings of distributions. In *ICML*, 2016.

Tomoya Sakai and Nobuyuki Shimizu. Covariate shift adaptation on learning from positive and unlabeled data. In *AAAI*, 2019.

Tomoya Sakai, Gang Niu, and Masashi Sugiyama. Semi-supervised auc optimization based on positive-unlabeled learning. *Machine Learning*, 107:767–794, 2018.

C Okan Sakar, S Olcay Polat, Mete Katircioglu, and Yomi Kastro. Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks. *Neural Computing and Applications*, 31(10):6893–6908, 2019.

Clayton Scott. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *AISTATS*, 2015.

Masashi Sugiyama, Han Bao, Takashi Ishida, Nan Lu, and Tomoya Sakai. *Machine learning from weak supervision: An empirical risk minimization approach*. MIT Press, 2022.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 workshop on mining multidimensional data*, 2008.

Naonori Ueda and Akinori Fujino. Partial auc maximization via nonlinear scoring functions. *arXiv preprint arXiv:1806.04838*, 2018.

Andrew V Uzilov, Joshua M Keegan, and David H Mathews. Detection of non-coding rnas on the basis of predicted secondary structure formation free energy change. *BMC bioinformatics*, 7(1): 173, 2006.

Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

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

Zheng Xie and Ming Li. Semi-supervised auc optimization without guessing labels of unlabeled data. In *AAAI*, 2018.

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

Tianbao Yang and Yiming Ying. Auc maximization in the era of big data and ai: a survey. *ACM computing surveys*, 55(8):1–37, 2022.

Yao Yao, Qihang Lin, and Tianbao Yang. Large-scale optimization of partial auc in a range of false positive rates. In *NeurIPS*, 2022.

Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Gang Niu, Masashi Sugiyama, and Dacheng Tao. Rethinking class-prior estimation for positive-unlabeled learning. In *ICLR*, 2021.

Yiming Ying, Longyin Wen, and Siwei Lyu. Stochastic online auc maximization. *NeurIPS*, 2016.

Zhuoning Yuan, Zhishuai Guo, Nitesh Chawla, and Tianbao Yang. Compositional training for end-to-end deep auc maximization. In *ICLR*, 2021a.

Zhuoning Yuan, Yan Yan, Milan Sonka, and Tianbao Yang. Large-scale robust deep auc maximization: a new surrogate loss and empirical studies on medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3040–3049, 2021b.

Xinwen Zhang, Yihan Zhang, Tianbao Yang, Richard Souvenir, and Hongchang Gao. Federated compositional deep auc maximization. *NeurIPS*, 2023.

Linli Zhou, Bokun Wang, My T. Thai, and Tianbao Yang. Stochastic primal-dual double block-coordinate for two-way partial auc maximization. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Dixian Zhu, Gang Li, Bokun Wang, Xiaodong Wu, and Tianbao Yang. When auc meets dro: optimizing partial auc for deep learning with non-convex convergence guarantee. In *ICML*, 2022.

A Theoretical Analysis

In this section, we analyze the finite-sample properties of the proposed pAUC estimator in Eq. (14). We first analyze the estimation error of the FPR estimator in Eq. (11) and the FPRs at the estimated thresholds $\hat{t}_\alpha$ and $\hat{t}_\beta$. Then, we derive a finite-sample error bound for the empirical pAUC estimator in Eq. (14). Finally, we analyze the effect of class-prior estimation error. Throughout this section, we assume $0 \leq \alpha < \beta < 1$, which covers all settings of interest.

For the analysis, we explicitly write the dependence on score function s when necessary. Let $\mathcal{S}$ be a class of score functions. For $s \in \mathcal{S}$, define

$$\overline{F}_s(t) := \mathbb{E}_{\mathbf{x}\sim p}\left[I(s(\mathbf{x}) > t)\right], \tag{17}$$

$$\overline{F}^{\mathrm{p}}_s(t) := \mathbb{E}_{\mathbf{x}^{\mathrm{p}}\sim p^{\mathrm{p}}}\left[I(s(\mathbf{x}^{\mathrm{p}}) > t)\right], \tag{18}$$

and their empirical versions as

$$\widehat{\overline{F}}_s(t) := \frac{1}{N}\sum_{n=1}^{N} I(s(\mathbf{x}_n) > t), \tag{19}$$

$$\widehat{\overline{F}}^{\mathrm{p}}_s(t) := \frac{1}{N^{\mathrm{p}}}\sum_{n=1}^{N^{\mathrm{p}}} I(s(\mathbf{x}^{\mathrm{p}}_n) > t). \tag{20}$$

Then, Eqs. (10) and (11) can be written as

$$\mathrm{FPR}_s(t) = \frac{\overline{F}_s(t) - \pi\overline{F}^{\mathrm{p}}_s(t)}{1-\pi}, \tag{21}$$

$$\widehat{\mathrm{FPR}}_s(t) = \frac{\widehat{\overline{F}}_s(t) - \pi\widehat{\overline{F}}^{\mathrm{p}}_s(t)}{1-\pi}. \tag{22}$$

We use the following assumption.

Assumption A.1. *For every $s \in \mathcal{S}$ considered in the analysis, the distribution of $s(\mathbf{x}^{\mathrm{n}})$ under $p^{\mathrm{n}}(\mathbf{x})$ has no point masses.*

Assumption A.1 serves as a convenient sufficient condition for handling the data-dependent estimated thresholds in the finite-sample analysis, as it ensures that $\mathrm{FPR}_s(t)$ is continuous in t. We use this continuity below when relating an estimated threshold to the FPR at that threshold. When $\alpha = 0$, it is possible that no finite threshold achieves $\mathrm{FPR}_s(t) = 0$. In this case, we set $t_{\alpha,s} = +\infty$, define $\mathrm{FPR}_s(+\infty) = 0$, and interpret $[t_{\beta,s}, t_{\alpha,s}]$ as $[t_{\beta,s}, \infty)$.

A.1 Uniform Convergence of the Score Survival Probabilities

Define the uniform estimation errors of the marginal and positive score survival probabilities as

$$\Delta_N := \sup_{s\in\mathcal{S},\, t\in\mathbb{R}} \left|\widehat{\overline{F}}_s(t) - \overline{F}_s(t)\right|, \tag{23}$$

$$\Delta_{N^{\mathrm{p}}} := \sup_{s\in\mathcal{S},\, t\in\mathbb{R}} \left|\widehat{\overline{F}}^{\mathrm{p}}_s(t) - \overline{F}^{\mathrm{p}}_s(t)\right|. \tag{24}$$

Thus, Δ_N and $\Delta_{N^{\mathrm{p}}}$ are the largest estimation errors of the marginal and positive score survival probabilities, respectively, over all $s \in \mathcal{S}$ and $t \in \mathbb{R}$.

We first consider a finite score-function class $\mathcal{S}$ with $|\mathcal{S}| = M$. For any fixed $s \in \mathcal{S}$, the Dvoretzky–Kiefer–Wolfowitz (DKW) inequality (Massart, 1990) gives

$$\mathbb{P}\left(\sup_{t\in\mathbb{R}} \left|\widehat{\overline{F}}_s(t) - \overline{F}_s(t)\right| > \epsilon\right) \le 2\exp(-2N\epsilon^2). \tag{25}$$

Although the DKW inequality is usually stated for cumulative distribution functions, it can also be directly applied to the survival probabilities since $\overline{F}_s(t) = 1 - F_s(t)$ and $\widehat{\overline{F}}_s(t) = 1 - \widehat{F}_s(t)$, where F_s and $\widehat{F}_s$ are the population and empirical cumulative distribution functions of $s(\mathbf{x})$, respectively.

Applying the union bound over all M score functions gives

$$\mathbb{P}\left(\Delta_N > \epsilon\right) \le 2M\exp(-2N\epsilon^2). \tag{26}$$

Setting the right-hand side to $\delta/2$ gives

$$\epsilon_N = \sqrt{\frac{\log(4M/\delta)}{2N}}, \tag{27}$$

and therefore

$$\mathbb{P}\left(\Delta_N \leq \epsilon_N\right) \geq 1 - \frac{\delta}{2}. \tag{28}$$

Applying the same argument to the N^{p} positive instances gives

$$\epsilon_{N^{\mathrm{p}}} = \sqrt{\frac{\log(4M/\delta)}{2N^{\mathrm{p}}}}, \tag{29}$$

such that $\mathbb{P}(\Delta_{N^{\mathrm{p}}} \leq \epsilon_{N^{\mathrm{p}}}) \geq 1 - \delta/2$. Thus, by the union bound, with probability at least $1 - \delta$,

$$\Delta_N \leq \epsilon_N, \qquad \Delta_{N^{\mathrm{p}}} \leq \epsilon_{N^{\mathrm{p}}} \tag{30}$$

hold simultaneously. In particular, for a finite score-function class, the empirical marginal and positive score survival probabilities converge uniformly to their population values, with errors of orders

$$O\left(\sqrt{\frac{\log(M/\delta)}{N}}\right) \quad \text{and} \quad O\left(\sqrt{\frac{\log(M/\delta)}{N^{\mathrm{p}}}}\right), \tag{31}$$

respectively.

The following analysis only requires the uniform bounds in Eq. (30). Therefore, the results are not restricted to finite score-function classes. For a score function s and a threshold t, define

$$C_{s,t} := \{\mathbf{x} \in \mathcal{X} : s(\mathbf{x}) > t\}. \tag{32}$$

Consider the collection of these sets over all score functions and thresholds:

$$\mathcal{C} := \{C_{s,t} : s \in \mathcal{S},\ t \in \mathbb{R}\}. \tag{33}$$

Since

$$\overline{F}_s(t) = \mathbb{P}_{\mathbf{x}\sim p}\left(\mathbf{x} \in C_{s,t}\right), \qquad \widehat{\overline{F}}_s(t) = \frac{1}{N}\sum_{n=1}^{N} I(\mathbf{x}_n \in C_{s,t}), \tag{34}$$

we have

$$\Delta_N = \sup_{C\in\mathcal{C}} \left| \frac{1}{N}\sum_{n=1}^{N} I(\mathbf{x}_n \in C) - \mathbb{P}_{\mathbf{x}\sim p}\left(\mathbf{x} \in C\right) \right|. \tag{35}$$

Therefore, if $\mathcal{C}$ has finite VC dimension d, a standard VC uniform-convergence bound (Mohri et al., 2012) gives

$$\Delta_N = O\left(\sqrt{\frac{d\log(2N) + \log(2/\delta)}{N}}\right) \tag{36}$$

with probability at least $1 - \delta/2$. Applying the same argument to the positive distribution using the same set class $\mathcal{C}$ and the N^{p} positive instances gives

$$\Delta_{N^{\mathrm{p}}} = O\left(\sqrt{\frac{d\log(2N^{\mathrm{p}}) + \log(2/\delta)}{N^{\mathrm{p}}}}\right) \tag{37}$$

with probability at least $1 - \delta/2$. Thus, by the union bound, both bounds hold simultaneously with probability at least $1-\delta$. Therefore, the following results also apply to infinite score-function classes when the threshold-set class $\mathcal{C}$ has finite VC dimension.

A.2 Accuracy of the PU-based FPR Estimator and Estimated Thresholds

We first analyze the error of the FPR estimator in Eq. (11).

Theorem A.2. *Suppose that*

$$\Delta_N \le \epsilon_N, \qquad \Delta_{N^{\mathrm{P}}} \le \epsilon_{N^{\mathrm{P}}}. \tag{38}$$

Then,

$$\sup_{s\in\mathcal{S},\, t\in\mathbb{R}} \left|\widehat{\mathrm{FPR}}_s(t) - \mathrm{FPR}_s(t)\right| \le \epsilon_{\mathrm{FPR}}, \tag{39}$$

where

$$\epsilon_{\mathrm{FPR}} := \frac{\epsilon_N + \pi\epsilon_{N^{\mathrm{P}}}}{1-\pi}. \tag{40}$$

The same bound holds when $\widehat{\mathrm{FPR}}_s(t)$ is clipped to $[0,1]$ as in the proposed method.

Proof. Using Eqs. (21) and (22), we have

$$\begin{aligned}
&\left|\widehat{\mathrm{FPR}}_s(t) - \mathrm{FPR}_s(t)\right| \\
&= \frac{1}{1-\pi}\left|\left(\widehat{\overline{F}}_s(t) - \overline{F}_s(t)\right) - \pi\left(\widehat{\overline{F}}^{\mathrm{P}}_s(t) - \overline{F}^{\mathrm{P}}_s(t)\right)\right| \\
&\le \frac{1}{1-\pi}\left[\left|\widehat{\overline{F}}_s(t) - \overline{F}_s(t)\right| + \pi\left|\widehat{\overline{F}}^{\mathrm{P}}_s(t) - \overline{F}^{\mathrm{P}}_s(t)\right|\right] \\
&\le \frac{\Delta_N + \pi\Delta_{N^{\mathrm{P}}}}{1-\pi} \\
&\le \frac{\epsilon_N + \pi\epsilon_{N^{\mathrm{P}}}}{1-\pi} = \epsilon_{\mathrm{FPR}}.
\end{aligned} \tag{41}$$

Since this holds for every $s \in \mathcal{S}$ and $t \in \mathbb{R}$, by taking the supremum, we obtain Eq. (39). Since $\mathrm{FPR}_s(t) \in [0,1]$, clipping $\widehat{\mathrm{FPR}}_s(t)$ to $[0,1]$ cannot increase the estimation error. Thus, the same bound also holds for the clipped estimator. □

For $\gamma \in \{\alpha, \beta\}$, let $\hat{t}_{\gamma,s}$ denote the estimated threshold obtained by Eq. (12) or Eq. (13), respectively, for score function s. Since $\alpha < \beta$,

$$\left\{t : \widehat{\mathrm{FPR}}_s(t) \le \alpha\right\} \subseteq \left\{t : \widehat{\mathrm{FPR}}_s(t) \le \beta\right\}. \tag{42}$$

Therefore, $\hat{t}_{\beta,s} \le \hat{t}_{\alpha,s}$ even though $\widehat{\mathrm{FPR}}_s(t)$ is not necessarily monotonic in t.

Theorem A.3. *Suppose that Assumption A.1 holds, $0 \le \alpha < \beta < 1$, and*

$$\Delta_N \le \epsilon_N, \qquad \Delta_{N^{\mathrm{P}}} \le \epsilon_{N^{\mathrm{P}}}. \tag{43}$$

Then, for every $s \in \mathcal{S}$ and $\gamma \in \{\alpha, \beta\}$,

$$\left|\mathrm{FPR}_s(\hat{t}_{\gamma,s}) - \gamma\right| \le \epsilon_{\mathrm{FPR}}. \tag{44}$$

Proof. We fix s and omit its subscript for simplicity. Since $0 \le \gamma < 1$, clipping $\widehat{\mathrm{FPR}}_s(t)$ to $[0,1]$ does not change the set of thresholds satisfying $\widehat{\mathrm{FPR}}_s(t) \le \gamma$. From the definition of $\hat{t}_\gamma$ in Eqs. (12) and (13),

$$\widehat{\mathrm{FPR}}(\hat{t}_\gamma) \le \gamma. \tag{45}$$

Therefore, Theorem A.2 gives

$$\mathrm{FPR}(\hat{t}_\gamma) \le \gamma + \epsilon_{\mathrm{FPR}}. \tag{46}$$

For the lower bound, $\hat{t}_\gamma$ is the smallest score among the PU instances at which the empirical FPR is no greater than γ. Suppose first that $\hat{t}_\gamma$ is not the smallest score among the PU instances, and let t_γ^- be the largest distinct score smaller than $\hat{t}_\gamma$. By the definition of $\hat{t}_\gamma$,

$$\widehat{\mathrm{FPR}}(t_\gamma^-) > \gamma. \tag{47}$$

Since there is no PU instance whose score lies in $(t_\gamma^-, \hat{t}_\gamma)$, the empirical FPR is constant on this interval. Therefore,

$$\widehat{\mathrm{FPR}}(t) > \gamma, \qquad t \in (t_\gamma^-, \hat{t}_\gamma). \tag{48}$$

If $\hat{t}_\gamma$ is the smallest score among the PU instances, then all PU instances have scores greater than t for any $t < \hat{t}_\gamma$. Hence,

$$\widehat{\mathrm{FPR}}(t) = \frac{1-\pi}{1-\pi} = 1 > \gamma. \tag{49}$$

Thus, in either case, there is a left neighborhood of $\hat{t}_\gamma$ in which $\widehat{\mathrm{FPR}}(t) > \gamma$. This argument does not require $\widehat{\mathrm{FPR}}(t)$ to be monotonic. For t approaching $\hat{t}_\gamma$ from below, Theorem A.2 gives

$$\mathrm{FPR}(t) > \gamma - \epsilon_{\mathrm{FPR}}. \tag{50}$$

Assumption A.1 ensures that $\mathrm{FPR}(t)$ is continuous. Therefore, taking $t \uparrow \hat{t}_\gamma$ gives

$$\mathrm{FPR}(\hat{t}_\gamma) \geq \gamma - \epsilon_{\mathrm{FPR}}. \tag{51}$$

Combining Eqs. (46) and (51) gives

$$\left|\mathrm{FPR}(\hat{t}_\gamma) - \gamma\right| \leq \epsilon_{\mathrm{FPR}}. \tag{52}$$

Since $s \in \mathcal{S}$ and $\gamma \in \{\alpha, \beta\}$ were arbitrary, the result holds for every $s \in \mathcal{S}$ and $\gamma \in \{\alpha, \beta\}$. □

Theorem A.3 directly bounds the quantity needed for pAUC estimation. The estimated threshold values themselves do not need to be close to their population values as long as the FPRs at the estimated thresholds are close to the target values α and β.

A.3 Finite-sample Error of the Empirical pAUC Estimator

We next analyze the empirical pAUC estimator in Eq. (14). For $b \in \mathbb{R}$ and $a \in \mathbb{R} \cup \{+\infty\}$ with $b \leq a$, define

$$A_s(a,b) := \mathbb{E}_{\mathbf{x}^{\mathrm{p}} \sim p^{\mathrm{p}}} \mathbb{E}_{\mathbf{x} \sim p} \left[I(s(\mathbf{x}^{\mathrm{p}}) > s(\mathbf{x})) I(s(\mathbf{x}) \in [b,a]) \right], \tag{53}$$

$$B_s(a,b) := \mathbb{E}_{\mathbf{x}^{\mathrm{p}} \sim p^{\mathrm{p}}} \mathbb{E}_{\bar{\mathbf{x}}^{\mathrm{p}} \sim p^{\mathrm{p}}} \left[I(s(\mathbf{x}^{\mathrm{p}}) > s(\bar{\mathbf{x}}^{\mathrm{p}})) I(s(\bar{\mathbf{x}}^{\mathrm{p}}) \in [b,a]) \right]. \tag{54}$$

Their empirical versions are

$$\widehat{A}_s(a,b) := \frac{1}{N N^{\mathrm{p}}} \sum_{n=1}^{N^{\mathrm{p}}} \sum_{m=1}^{N} I(s(\mathbf{x}_n^{\mathrm{p}}) > s(\mathbf{x}_m)) I(s(\mathbf{x}_m) \in [b,a]), \tag{55}$$

$$\widehat{B}_s(a,b) := \frac{1}{N^{\mathrm{p}}(N^{\mathrm{p}}-1)} \sum_{\substack{n,m=1 \\ n \neq m}}^{N^{\mathrm{p}}} I(s(\mathbf{x}_n^{\mathrm{p}}) > s(\mathbf{x}_m^{\mathrm{p}})) I(s(\mathbf{x}_m^{\mathrm{p}}) \in [b,a]). \tag{56}$$

Define

$$Q_s(a,b) := \frac{A_s(a,b) - \pi B_s(a,b)}{(\beta - \alpha)(1-\pi)}. \tag{57}$$

Using Eq. (8), we have

$$Q_s(a,b) = \frac{1}{\beta - \alpha} \mathbb{E}_{\mathbf{x}^{\mathrm{p}} \sim p^{\mathrm{p}}} \mathbb{E}_{\mathbf{x}^{\mathrm{n}} \sim p^{\mathrm{n}}} \left[I(s(\mathbf{x}^{\mathrm{p}}) > s(\mathbf{x}^{\mathrm{n}})) I(s(\mathbf{x}^{\mathrm{n}}) \in [b,a]) \right]. \tag{58}$$

Therefore,

$$Q_s(t_{\alpha,s}, t_{\beta,s}) = \mathrm{pAUC}_s(\alpha, \beta). \tag{59}$$

We use the following lemma.

Lemma A.4. *Let $Z_1, \dots, Z_n$ be i.i.d. copies of a random variable Z and suppose that*

$$\sup_{t\in\mathbb{R}} \left| \frac{1}{n}\sum_{i=1}^{n} I(Z_i > t) - \mathbb{P}(Z > t) \right| \le \epsilon. \tag{60}$$

Then, for any non-increasing function $g : \mathbb{R} \to [0, 1]$,

$$\sup_{b\le a} \left| \frac{1}{n}\sum_{i=1}^{n} g(Z_i) I(Z_i \in [b, a]) - \mathbb{E}\left[g(Z) I(Z \in [b, a])\right] \right| \le 2\epsilon, \tag{61}$$

where the supremum is taken over $a, b \in \mathbb{R}$.

Proof. Fix a realization of $Z_1, \dots, Z_n$ satisfying Eq. (60). Then,

$$\left| \frac{1}{n}\sum_{i=1}^{n} I(Z_i > t) - \mathbb{P}(Z > t) \right| \le \epsilon \qquad \text{for all } t \in \mathbb{R}. \tag{62}$$

The probability of any interval can be expressed using differences and one-sided limits of the tail probabilities in Eq. (60). Therefore,

$$\left| \frac{1}{n}\sum_{i=1}^{n} I(Z_i \in J) - \mathbb{P}(Z \in J) \right| \le 2\epsilon \tag{63}$$

for any interval J. For example, for a closed left endpoint, we use

$$\mathbb{P}(Z \in [b, \infty)) = \lim_{t\uparrow b} \mathbb{P}(Z > t), \qquad \frac{1}{n}\sum_{i=1}^{n} I(Z_i \in [b, \infty)) = \lim_{t\uparrow b} \frac{1}{n}\sum_{i=1}^{n} I(Z_i > t). \tag{64}$$

Next, since $0 \le g(z) \le 1$, we use

$$g(z) = \int_0^1 I(g(z) > u)\, du. \tag{65}$$

Multiplying both sides by $I(z \in [b, a])$ gives

$$\begin{aligned} g(z) I(z \in [b, a]) &= \int_0^1 I(z \in [b, a]) I(g(z) > u)\, du \\ &= \int_0^1 I(z \in J_u)\, du, \end{aligned} \tag{66}$$

where

$$J_u = [b, a] \cap \{z : g(z) > u\}. \tag{67}$$

Since g is non-increasing, $\{z : g(z) > u\}$ is a half-line, i.e., a set of the form $(-\infty, c)$ or $(-\infty, c]$ (possibly empty or all of $\mathbb{R}$). Therefore, J_u is an interval, possibly empty, for every $u \in [0, 1]$. By Eq. (63),

$$\left| \frac{1}{n}\sum_{i=1}^{n} I(Z_i \in J_u) - \mathbb{P}(Z \in J_u) \right| \le 2\epsilon. \tag{68}$$

Applying the above identity to each Z_i and to Z, we have

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{n} g(Z_i) I(Z_i \in [b, a]) - \mathbb{E}\left[g(Z) I(Z \in [b, a])\right] \\ &= \int_0^1 \left\{ \frac{1}{n}\sum_{i=1}^{n} I(Z_i \in J_u) - \mathbb{P}(Z \in J_u) \right\} du. \end{aligned} \tag{69}$$

Therefore,

$$\begin{aligned}&\left|\frac{1}{n}\sum_{i=1}^{n} g(Z_i)I(Z_i \in [b,a]) - \mathbb{E}\left[g(Z)I(Z \in [b,a])\right]\right| \\ &\leq \int_0^1 \left|\frac{1}{n}\sum_{i=1}^{n} I(Z_i \in J_u) - \mathbb{P}(Z \in J_u)\right| du \\ &\leq \int_0^1 2\epsilon\, du = 2\epsilon. \end{aligned} \tag{70}$$

Since this holds for every $b \leq a$, taking the supremum proves Eq. (61). □

Lemma A.5. *Suppose that $\Delta_N \leq \epsilon_N$ and $\Delta_{N^{\mathrm{p}}} \leq \epsilon_{N^{\mathrm{p}}}$. Then, for every $s \in \mathcal{S}$ and $a, b \in \mathbb{R}$ with $b \leq a$,*

$$\left|\widehat{A}_s(a,b) - A_s(a,b)\right| \leq 2\epsilon_N + \epsilon_{N^{\mathrm{p}}}, \tag{71}$$

$$\left|\widehat{B}_s(a,b) - B_s(a,b)\right| \leq 3\frac{N^{\mathrm{p}}}{N^{\mathrm{p}}-1}\epsilon_{N^{\mathrm{p}}} + \frac{1}{N^{\mathrm{p}}-1}. \tag{72}$$

Proof. Using Eqs. (18) and (53), we have

$$A_s(a,b) = \mathbb{E}_{\mathbf{x}\sim p}\left[\overline{F}^{\mathrm{p}}_s(s(\mathbf{x}))I(s(\mathbf{x}) \in [b,a])\right]. \tag{73}$$

Similarly, Eq. (55) can be written as

$$\widehat{A}_s(a,b) = \frac{1}{N}\sum_{m=1}^{N} \widehat{\overline{F}}^{\mathrm{p}}_s(s(\mathbf{x}_m))I(s(\mathbf{x}_m) \in [b,a]). \tag{74}$$

Therefore,

$$\begin{aligned}&\left|\widehat{A}_s(a,b) - A_s(a,b)\right| \\ &\leq \left|\frac{1}{N}\sum_{m=1}^{N}\left[\widehat{\overline{F}}^{\mathrm{p}}_s(s(\mathbf{x}_m)) - \overline{F}^{\mathrm{p}}_s(s(\mathbf{x}_m))\right] I(s(\mathbf{x}_m) \in [b,a])\right| \\ &\quad + \left|\frac{1}{N}\sum_{m=1}^{N} \overline{F}^{\mathrm{p}}_s(s(\mathbf{x}_m))I(s(\mathbf{x}_m) \in [b,a]) - A_s(a,b)\right|. \end{aligned} \tag{75}$$

By the definition of $\Delta_{N^{\mathrm{p}}}$, the first term is at most $\epsilon_{N^{\mathrm{p}}}$. Since $\overline{F}^{\mathrm{p}}_s(z)$ is non-increasing in z, Lemma A.4, applied to the marginal scores $Z_m = s(\mathbf{x}_m)$ together with $\Delta_N \leq \epsilon_N$, bounds the second term by $2\epsilon_N$. This proves Eq. (71).

For $\widehat{B}_s(a,b)$, we first introduce the corresponding V-statistic so that the pairwise sum can be written using the empirical positive score survival probability:

$$\widehat{B}^{\mathrm{V}}_s(a,b) := \frac{1}{(N^{\mathrm{p}})^2}\sum_{n=1}^{N^{\mathrm{p}}}\sum_{m=1}^{N^{\mathrm{p}}} I(s(\mathbf{x}^{\mathrm{p}}_n) > s(\mathbf{x}^{\mathrm{p}}_m))I(s(\mathbf{x}^{\mathrm{p}}_m) \in [b,a]). \tag{76}$$

Using Eq. (20), this can be written as

$$\widehat{B}^{\mathrm{V}}_s(a,b) = \frac{1}{N^{\mathrm{p}}}\sum_{m=1}^{N^{\mathrm{p}}} \widehat{\overline{F}}^{\mathrm{p}}_s(s(\mathbf{x}^{\mathrm{p}}_m))I(s(\mathbf{x}^{\mathrm{p}}_m) \in [b,a]). \tag{77}$$

Also,

$$B_s(a,b) = \mathbb{E}_{\mathbf{x}^{\mathrm{p}}\sim p^{\mathrm{p}}}\left[\overline{F}^{\mathrm{p}}_s(s(\mathbf{x}^{\mathrm{p}}))I(s(\mathbf{x}^{\mathrm{p}}) \in [b,a])\right]. \tag{78}$$

As in the case of $\widehat{A}_s(a,b)$, replacing $\widehat{\overline{F}}_s^{\mathrm{p}}$ in Eq. (77) with $\overline{F}_s^{\mathrm{p}}$ gives an error of at most $\epsilon_{N^{\mathrm{p}}}$. Applying Lemma A.4 to the positive scores gives an additional error of at most $2\epsilon_{N^{\mathrm{p}}}$. Therefore,

$$\left|\widehat{B}_s^{\mathrm{V}}(a,b) - B_s(a,b)\right| \le 3\epsilon_{N^{\mathrm{p}}}. \tag{79}$$

The diagonal terms in $\widehat{B}_s^{\mathrm{V}}(a,b)$ are zero since

$$I(s(\mathbf{x}_n^{\mathrm{p}}) > s(\mathbf{x}_n^{\mathrm{p}})) = 0. \tag{80}$$

Therefore,

$$\widehat{B}_s(a,b) = \frac{N^{\mathrm{p}}}{N^{\mathrm{p}}-1}\widehat{B}_s^{\mathrm{V}}(a,b). \tag{81}$$

Using Eq. (79), we have

$$\begin{aligned}\left|\widehat{B}_s(a,b) - B_s(a,b)\right| &\le \frac{N^{\mathrm{p}}}{N^{\mathrm{p}}-1}\left|\widehat{B}_s^{\mathrm{V}}(a,b) - B_s(a,b)\right| + \frac{B_s(a,b)}{N^{\mathrm{p}}-1} \\ &\le 3\frac{N^{\mathrm{p}}}{N^{\mathrm{p}}-1}\epsilon_{N^{\mathrm{p}}} + \frac{1}{N^{\mathrm{p}}-1},\end{aligned} \tag{82}$$

where we used $0 \le B_s(a,b) \le 1$. This proves Eq. (72). □

Theorem A.6. *Suppose that Assumption A.1 holds,* $0 \le \alpha < \beta < 1$, $N^{\mathrm{p}} \ge 2$, *and*

$$\Delta_N \le \epsilon_N, \qquad \Delta_{N^{\mathrm{p}}} \le \epsilon_{N^{\mathrm{p}}}. \tag{83}$$

Then, simultaneously for every $s \in \mathcal{S}$,

$$\left|\widehat{\mathrm{pAUC}}_s(\alpha,\beta) - \mathrm{pAUC}_s(\alpha,\beta)\right| \le B_{N,N^{\mathrm{p}}}, \tag{84}$$

where

$$B_{N,N^{\mathrm{p}}} := \frac{1}{(\beta-\alpha)(1-\pi)}\left[4\epsilon_N + \left(1 + 2\pi + 3\pi\frac{N^{\mathrm{p}}}{N^{\mathrm{p}}-1}\right)\epsilon_{N^{\mathrm{p}}} + \frac{\pi}{N^{\mathrm{p}}-1}\right]. \tag{85}$$

Proof. We decompose the estimation error as

$$\begin{aligned}&\left|\widehat{\mathrm{pAUC}}_s(\alpha,\beta) - \mathrm{pAUC}_s(\alpha,\beta)\right| \\ &\le \left|\widehat{\mathrm{pAUC}}_s(\alpha,\beta) - Q_s(\hat{t}_{\alpha,s},\hat{t}_{\beta,s})\right| \\ &\quad + \left|Q_s(\hat{t}_{\alpha,s},\hat{t}_{\beta,s}) - Q_s(t_{\alpha,s},t_{\beta,s})\right|.\end{aligned} \tag{86}$$

The estimated thresholds $\hat{t}_{\alpha,s}$ and $\hat{t}_{\beta,s}$ are selected from the PU instance scores and are therefore finite. Thus, Lemma A.5 can be applied at these data-dependent thresholds. Using Eqs. (71) and (72), the first term in Eq. (86) is bounded by

$$\begin{aligned}&\left|\widehat{\mathrm{pAUC}}_s(\alpha,\beta) - Q_s(\hat{t}_{\alpha,s},\hat{t}_{\beta,s})\right| \\ &\le \frac{1}{(\beta-\alpha)(1-\pi)}\left[2\epsilon_N + \epsilon_{N^{\mathrm{p}}} + 3\pi\frac{N^{\mathrm{p}}}{N^{\mathrm{p}}-1}\epsilon_{N^{\mathrm{p}}} + \frac{\pi}{N^{\mathrm{p}}-1}\right].\end{aligned} \tag{87}$$

The bound in Lemma A.5 is uniform over the finite endpoints a and b. Thus, it also holds when the estimated thresholds and the empirical pAUC are calculated from the same instances.

For the second term, changing either endpoint of the score interval affects Eq. (58) only through negative instances whose scores are between the original and changed endpoints. Since the pairwise indicator is bounded by one,

$$\begin{aligned}&\left|Q_s(\hat{t}_{\alpha,s},\hat{t}_{\beta,s}) - Q_s(t_{\alpha,s},t_{\beta,s})\right| \\ &\le \frac{1}{\beta-\alpha}\mathbb{E}_{\mathbf{x}^{\mathrm{n}}\sim p^{\mathrm{n}}}\left[\left|I\big(s(\mathbf{x}^{\mathrm{n}}) \in [\hat{t}_{\beta,s},\hat{t}_{\alpha,s}]\big) - I(s(\mathbf{x}^{\mathrm{n}}) \in [t_{\beta,s},t_{\alpha,s}])\right|\right] \\ &\le \frac{1}{\beta-\alpha}\mathbb{E}_{\mathbf{x}^{\mathrm{n}}\sim p^{\mathrm{n}}}\Big[\left|I(s(\mathbf{x}^{\mathrm{n}}) \ge \hat{t}_{\beta,s}) - I(s(\mathbf{x}^{\mathrm{n}}) \ge t_{\beta,s})\right| \\ &\qquad\qquad + \left|I(s(\mathbf{x}^{\mathrm{n}}) > \hat{t}_{\alpha,s}) - I(s(\mathbf{x}^{\mathrm{n}}) > t_{\alpha,s})\right|\Big] \\ &\le \frac{1}{\beta-\alpha}\left[\left|\mathrm{FPR}_s(\hat{t}_{\alpha,s}) - \mathrm{FPR}_s(t_{\alpha,s})\right| + \left|\mathrm{FPR}_s(\hat{t}_{\beta,s}) - \mathrm{FPR}_s(t_{\beta,s})\right|\right].\end{aligned} \tag{88}$$

Under Assumption A.1, $\mathrm{FPR}_s(t)$ is continuous. Thus, for finite population thresholds,

$$\mathrm{FPR}_s(t_{\gamma,s}) = \gamma. \tag{89}$$

Together with the convention $\mathrm{FPR}_s(+\infty) = 0$ when $t_{\alpha,s} = +\infty$, we have

$$\mathrm{FPR}_s(t_{\alpha,s}) = \alpha, \qquad \mathrm{FPR}_s(t_{\beta,s}) = \beta. \tag{90}$$

Then, Theorem A.3 gives

$$\begin{aligned} &\left|Q_s(\hat{t}_{\alpha,s}, \hat{t}_{\beta,s}) - Q_s(t_{\alpha,s}, t_{\beta,s})\right| \\ &\leq \frac{1}{\beta - \alpha}\left[\left|\mathrm{FPR}_s(\hat{t}_{\alpha,s}) - \alpha\right| + \left|\mathrm{FPR}_s(\hat{t}_{\beta,s}) - \beta\right|\right] \\ &\leq \frac{2\epsilon_{\mathrm{FPR}}}{\beta - \alpha}. \end{aligned} \tag{91}$$

Combining Eqs. (87) and (91), and substituting Eq. (40), proves Eq. (84). □

For a finite score-function class with $|\mathcal{S}| = M$, the DKW bounds in Section A.1 and a union bound show that the conditions of Theorem A.6 hold simultaneously with probability at least $1 - \delta$. Therefore,

$$\begin{aligned} &\sup_{s \in \mathcal{S}} \left|\widehat{\mathrm{pAUC}}_s(\alpha, \beta) - \mathrm{pAUC}_s(\alpha, \beta)\right| \\ &= O\left(\frac{1}{(\beta - \alpha)(1 - \pi)}\left[\sqrt{\frac{\log(M/\delta)}{N}} + \sqrt{\frac{\log(M/\delta)}{N^{\mathrm{P}}}} + \frac{1}{N^{\mathrm{P}}}\right]\right) \end{aligned} \tag{92}$$

with probability at least $1 - \delta$. Thus, the empirical pAUC estimator is uniformly consistent as $N, N^{\mathrm{P}} \to \infty$. The same conclusion applies to infinite score-function classes when the corresponding uniform errors in Eq. (30) converge to zero. Since the bound is uniform over $s \in \mathcal{S}$, it also applies to a score function selected from $\mathcal{S}$ using the training data.

A.4 Effect of Class-prior Estimation Error

We finally analyze the effect of using an inaccurate class-prior at the population level. Let $\widetilde{\pi} \in [0, 1)$ be the class-prior value used in the proposed method, while π is the true class-prior. Replacing π with $\widetilde{\pi}$ in Eq. (10) gives

$$\mathrm{FPR}_{\widetilde{\pi},s}(t) := \frac{\overline{F}_s(t) - \widetilde{\pi}\overline{F}^{\mathrm{P}}_s(t)}{1 - \widetilde{\pi}}. \tag{93}$$

Proposition A.7. *For every $s \in \mathcal{S}$ and $t \in \mathbb{R}$,*

$$\begin{aligned} &\mathrm{FPR}_{\widetilde{\pi},s}(t) - \mathrm{FPR}_s(t) \\ &= \frac{\pi - \widetilde{\pi}}{1 - \widetilde{\pi}}\left[\mathrm{TPR}_s(t) - \mathrm{FPR}_s(t)\right]. \end{aligned} \tag{94}$$

Consequently,

$$\sup_{s \in \mathcal{S},\, t \in \mathbb{R}} \left|\mathrm{FPR}_{\widetilde{\pi},s}(t) - \mathrm{FPR}_s(t)\right| \leq \epsilon_\pi, \qquad \epsilon_\pi := \frac{|\widetilde{\pi} - \pi|}{1 - \widetilde{\pi}}. \tag{95}$$

Proof. From Eq. (7),

$$\overline{F}_s(t) = \pi \mathrm{TPR}_s(t) + (1 - \pi)\mathrm{FPR}_s(t). \tag{96}$$

Substituting this into Eq. (93) gives

$$\mathrm{FPR}_{\widetilde{\pi},s}(t) = \frac{(\pi - \widetilde{\pi})\mathrm{TPR}_s(t) + (1 - \pi)\mathrm{FPR}_s(t)}{1 - \widetilde{\pi}}. \tag{97}$$

Subtracting $\mathrm{FPR}_s(t)$ gives Eq. (94). Since $\mathrm{TPR}_s(t), \mathrm{FPR}_s(t) \in [0, 1]$, Eq. (95) follows. □

For $b \in \mathbb{R}$ and $a \in \mathbb{R} \cup \{+\infty\}$ with $b \le a$, define

$$Q_{\widetilde{\pi},s}(a,b) := \frac{A_s(a,b) - \widetilde{\pi} B_s(a,b)}{(\beta-\alpha)(1-\widetilde{\pi})}. \tag{98}$$

Let

$$\widetilde{t}_{\gamma,s} := \inf\{t \in \mathbb{R} : \mathrm{FPR}_{\widetilde{\pi},s}(t) \le \gamma\}, \qquad \gamma \in \{\alpha, \beta\}, \tag{99}$$

where we set $\inf \emptyset = +\infty$. Since $\alpha < \beta$, every threshold satisfying $\mathrm{FPR}_{\widetilde{\pi},s}(t) \le \alpha$ also satisfies $\mathrm{FPR}_{\widetilde{\pi},s}(t) \le \beta$. Therefore,

$$\widetilde{t}_{\beta,s} \le \widetilde{t}_{\alpha,s}. \tag{100}$$

Theorem A.8. *Suppose that Assumption A.1 holds and $0 \le \alpha < \beta < 1$. Then, for every $s \in \mathcal{S}$,*

$$\begin{aligned} &\left|Q_{\widetilde{\pi},s}(\widetilde{t}_{\alpha,s}, \widetilde{t}_{\beta,s}) - \mathrm{pAUC}_s(\alpha,\beta)\right| \\ &\le \frac{3|\widetilde{\pi} - \pi|}{(\beta-\alpha)(1-\widetilde{\pi})}. \end{aligned} \tag{101}$$

Proof. We fix s. From Proposition A.7,

$$\sup_{t \in \mathbb{R}} |\mathrm{FPR}_{\widetilde{\pi},s}(t) - \mathrm{FPR}_s(t)| \le \epsilon_\pi. \tag{102}$$

Since $\mathrm{FPR}_{\widetilde{\pi},s}(t) \to 1$ as $t \to -\infty$ and $\gamma < 1$, there exists a finite T such that $\mathrm{FPR}_{\widetilde{\pi},s}(t) > \gamma$ for all $t < T$. Therefore, $\widetilde{t}_{\gamma,s} = -\infty$ cannot occur. Suppose first that $\widetilde{t}_{\gamma,s}$ is finite. Since the survival probabilities $\overline{F}_s(t)$ and $\overline{F}^{\mathrm{p}}_s(t)$ are right-continuous in t, $\mathrm{FPR}_{\widetilde{\pi},s}(t)$ is also right-continuous. Therefore, from the definition of $\widetilde{t}_{\gamma,s}$,

$$\mathrm{FPR}_{\widetilde{\pi},s}(\widetilde{t}_{\gamma,s}) \le \gamma. \tag{103}$$

Also, for every $t < \widetilde{t}_{\gamma,s}$,

$$\mathrm{FPR}_{\widetilde{\pi},s}(t) > \gamma. \tag{104}$$

Together with Eq. (102), these inequalities give

$$\mathrm{FPR}_s(\widetilde{t}_{\gamma,s}) \le \gamma + \epsilon_\pi, \tag{105}$$

$$\mathrm{FPR}_s(t) > \gamma - \epsilon_\pi \qquad \text{for } t < \widetilde{t}_{\gamma,s}. \tag{106}$$

Assumption A.1 ensures that $\mathrm{FPR}_s(t)$ is continuous. Therefore, taking $t \uparrow \widetilde{t}_{\gamma,s}$ in the second inequality gives

$$\mathrm{FPR}_s(\widetilde{t}_{\gamma,s}) \ge \gamma - \epsilon_\pi. \tag{107}$$

Thus,

$$\left|\mathrm{FPR}_s(\widetilde{t}_{\gamma,s}) - \gamma\right| \le \epsilon_\pi. \tag{108}$$

If $\widetilde{t}_{\gamma,s} = +\infty$, then necessarily $\gamma = 0$. With the convention $\mathrm{FPR}_s(+\infty) = 0$, Eq. (108) also holds. Therefore, Eq. (108) holds for $\gamma \in \{\alpha, \beta\}$.

As in the proof of Theorem A.6, Eq. (58) and the fact that the pairwise indicator is bounded by one give

$$\begin{aligned} &\left|Q_s(\widetilde{t}_{\alpha,s}, \widetilde{t}_{\beta,s}) - \mathrm{pAUC}_s(\alpha,\beta)\right| \\ &\le \frac{1}{\beta-\alpha}\left[\left|\mathrm{FPR}_s(\widetilde{t}_{\alpha,s}) - \alpha\right| + \left|\mathrm{FPR}_s(\widetilde{t}_{\beta,s}) - \beta\right|\right] \\ &\le \frac{2\epsilon_\pi}{\beta-\alpha}. \end{aligned} \tag{109}$$

Next, from Eq. (7),

$$A_s(a,b) = \pi B_s(a,b) + (1-\pi) C_s(a,b), \tag{110}$$

where

$$C_s(a,b) := \mathbb{E}_{\mathbf{x}^{\mathrm{P}}\sim p^{\mathrm{P}}}\mathbb{E}_{\mathbf{x}^{\mathrm{N}}\sim p^{\mathrm{N}}}\left[I(s(\mathbf{x}^{\mathrm{P}}) > s(\mathbf{x}^{\mathrm{N}}))I(s(\mathbf{x}^{\mathrm{N}}) \in [b,a])\right]. \tag{111}$$

Substituting $A_s(a,b) = \pi B_s(a,b) + (1-\pi)C_s(a,b)$ into $Q_{\widetilde{\pi},s}$ and Q_s gives

$$Q_{\widetilde{\pi},s}(a,b) = \frac{(\pi-\widetilde{\pi})B_s(a,b) + (1-\pi)C_s(a,b)}{(\beta-\alpha)(1-\widetilde{\pi})}, \tag{112}$$

$$Q_s(a,b) = \frac{C_s(a,b)}{\beta-\alpha}. \tag{113}$$

Therefore,

$$\begin{aligned} &Q_{\widetilde{\pi},s}(a,b) - Q_s(a,b) \\ &= \frac{\pi-\widetilde{\pi}}{(\beta-\alpha)(1-\widetilde{\pi})}\left[B_s(a,b) - C_s(a,b)\right]. \end{aligned} \tag{114}$$

Since $B_s(a,b), C_s(a,b) \in [0,1]$,

$$|Q_{\widetilde{\pi},s}(a,b) - Q_s(a,b)| \leq \frac{\epsilon_\pi}{\beta-\alpha}. \tag{115}$$

Applying Eq. (115) at

$$(a,b) = (\widetilde{t}_{\alpha,s}, \widetilde{t}_{\beta,s}) \tag{116}$$

and combining it with Eq. (109) gives

$$\begin{aligned} &\left|Q_{\widetilde{\pi},s}(\widetilde{t}_{\alpha,s}, \widetilde{t}_{\beta,s}) - \mathrm{pAUC}_s(\alpha,\beta)\right| \\ &\leq \frac{3\epsilon_\pi}{\beta-\alpha} = \frac{3|\widetilde{\pi}-\pi|}{(\beta-\alpha)(1-\widetilde{\pi})}, \end{aligned} \tag{117}$$

which proves Eq. (101). □

Theorem A.8 shows that the population-level error caused by an inaccurate class-prior is controlled by $|\widetilde{\pi}-\pi|/(1-\widetilde{\pi})$.

B Experimental Settings

For MNIST, FMNIST, SUSY, CODRNA, IJCNN, EPSILON, we used a three-layered feed-forward neural network with ReLU activations for the classifier (score function), where each hidden layer consisted of 128 units. For SVHN and CIFAR10, we used a convolutional neural network composed of two convolutional blocks followed by a three-layer feed-forward neural network for the classifier (score function). The first (second) convolutional block comprised a 6 (16) filter 5×5 convolution, the ReLU activation, and a 2×2 max-pooling layer. The numbers of nodes in the two hidden layers were 120 and 84, and ReLU activations were used. For all comparison methods, the same neural network architecture was used. The PU mini-batch sizes P and U were set to 50 and $1,000$, respectively. For all methods, we used the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 10^{-3}. The maximum number of epochs was 200. The loss with validation PU data was used for early-stopping to mitigate overfitting. All methods were implemented using PyTorch (Paszke et al., 2017), and all experiments were conducted on a Linux server with an Intel Xeon CPU and A100 GPU.

C Additional Experimental Results

C.1 Results with $\alpha = 0.1$ and $\beta = 0.2$

Table 4 shows the test average pAUCs and their standard deviations with $\alpha = 0.1$, $\beta = 0.2$. The proposed method performed well in many cases. Compared to case $\alpha = 0$ (Tables 1 and 2), PAUC and 2step tended to perform well. We describe possible reasons for their good performance below. These methods treat unlabeled data (or pseudo N data) whose scores fall within a certain range

Table 4: Test average pAUCs and their standard deviations with $\alpha = 0.1$, $\beta = 0.2$ over different labeled positive data sizes $N^{\mathrm{P}} = 30$, 50, and 100. Values in bold are not statistically different at the 5% level from the best performing method in each row according to a paired t-test. '# Best' row represents the number of results of each method that are the best or comparable to it.

Data	π	Ours	NNPU	PUAUC	PAUC	PUpAUCh	2step
MNIST	0.1	**0.838 ± 0.043**	0.698 ± 0.126	0.823 ± 0.041	**0.839 ± 0.048**	**0.839 ± 0.042**	0.826 ± 0.036
	0.2	**0.821 ± 0.048**	0.776 ± 0.063	0.800 ± 0.046	0.776 ± 0.041	**0.818 ± 0.050**	0.808 ± 0.050
FMNIST	0.1	0.958 ± 0.021	0.795 ± 0.011	0.955 ± 0.020	**0.964 ± 0.019**	0.958 ± 0.021	**0.963 ± 0.018**
	0.2	**0.954 ± 0.024**	0.806 ± 0.034	0.943 ± 0.022	0.910 ± 0.037	0.951 ± 0.025	**0.958 ± 0.018**
SVHN	0.1	0.423 ± 0.098	0.154 ± 0.006	0.402 ± 0.092	**0.451 ± 0.104**	0.377 ± 0.092	0.410 ± 0.097
	0.2	0.360 ± 0.104	0.154 ± 0.007	0.353 ± 0.074	**0.391 ± 0.085**	0.258 ± 0.063	0.351 ± 0.081
CIFAR10	0.1	**0.744 ± 0.063**	0.127 ± 0.077	0.731 ± 0.069	0.721 ± 0.062	**0.743 ± 0.056**	**0.752 ± 0.063**
	0.2	**0.722 ± 0.071**	0.162 ± 0.118	0.698 ± 0.080	0.623 ± 0.081	0.697 ± 0.069	**0.712 ± 0.081**
SUSY	0.1	**0.647 ± 0.029**	0.557 ± 0.074	0.629 ± 0.035	**0.651 ± 0.024**	**0.642 ± 0.030**	0.632 ± 0.034
	0.2	**0.629 ± 0.038**	0.578 ± 0.047	0.610 ± 0.043	0.606 ± 0.044	**0.628 ± 0.035**	0.617 ± 0.040
CODRNA	0.1	0.959 ± 0.025	0.454 ± 0.319	0.951 ± 0.030	0.949 ± 0.033	0.959 ± 0.027	**0.965 ± 0.023**
	0.2	0.946 ± 0.037	0.892 ± 0.056	0.935 ± 0.037	0.853 ± 0.061	0.949 ± 0.031	**0.959 ± 0.028**
IJCNN	0.1	**0.792 ± 0.100**	0.469 ± 0.108	0.778 ± 0.086	**0.803 ± 0.078**	0.760 ± 0.125	**0.811 ± 0.086**
	0.2	0.730 ± 0.136	0.636 ± 0.098	**0.750 ± 0.094**	0.714 ± 0.086	0.620 ± 0.217	**0.769 ± 0.101**
EPSILON	0.1	**0.411 ± 0.065**	0.177 ± 0.008	**0.404 ± 0.067**	**0.410 ± 0.060**	**0.406 ± 0.068**	**0.399 ± 0.079**
	0.2	**0.360 ± 0.063**	0.184 ± 0.015	**0.368 ± 0.062**	**0.363 ± 0.048**	0.295 ± 0.066	**0.368 ± 0.071**
# Best		10	0	3	8	6	10

Table 5: The results with $\pi = 0.01$ and $\pi = 0.3$: test average pAUCs and their standard deviations with $\alpha = 0$, $\beta = 0.2$ over different labeled positive data sizes $N^{\mathrm{P}} = 30$, 50, and 100. Values in bold are not statistically different at the 5% level from the best performing method in each row according to a paired t-test. '# Best' row represents the number of results of each method that are the best or comparable to it.

Data	π	Ours	NNPU	PUAUC	PAUC	PUpAUCh	2step
MNIST	0.01	**0.799 ± 0.054**	0.338 ± 0.009	0.768 ± 0.055	0.794 ± 0.052	**0.799 ± 0.053**	0.792 ± 0.055
FMNIST	0.01	0.949 ± 0.021	0.731 ± 0.112	0.945 ± 0.021	0.947 ± 0.022	0.949 ± 0.021	**0.954 ± 0.017**
SVHN	0.01	**0.411 ± 0.089**	0.105 ± 0.005	0.363 ± 0.089	**0.410 ± 0.089**	**0.412 ± 0.083**	0.398 ± 0.087
CIFAR10	0.01	0.661 ± 0.067	0.081 ± 0.053	0.634 ± 0.079	0.662 ± 0.072	0.656 ± 0.070	**0.679 ± 0.063**
SUSY	0.01	**0.571 ± 0.022**	0.295 ± 0.038	0.528 ± 0.041	**0.570 ± 0.022**	**0.569 ± 0.022**	0.552 ± 0.033
CODRNA	0.01	**0.882 ± 0.025**	0.124 ± 0.031	0.875 ± 0.032	**0.883 ± 0.024**	**0.882 ± 0.024**	**0.881 ± 0.027**
IJCNN	0.01	**0.766 ± 0.070**	0.188 ± 0.023	0.714 ± 0.085	**0.765 ± 0.067**	0.763 ± 0.070	0.753 ± 0.077
EPSILON	0.01	**0.353 ± 0.055**	0.131 ± 0.005	0.324 ± 0.059	0.350 ± 0.052	**0.354 ± 0.054**	0.338 ± 0.064
MNIST	0.3	**0.688 ± 0.068**	0.665 ± 0.067	0.663 ± 0.051	0.565 ± 0.061	**0.689 ± 0.060**	0.664 ± 0.052
FMNIST	0.3	**0.923 ± 0.029**	0.824 ± 0.070	0.883 ± 0.042	0.747 ± 0.083	**0.922 ± 0.029**	0.888 ± 0.042
SVHN	0.3	**0.239 ± 0.079**	0.105 ± 0.005	**0.233 ± 0.056**	**0.237 ± 0.045**	0.145 ± 0.032	**0.229 ± 0.056**
CIFAR10	0.3	**0.544 ± 0.079**	0.229 ± 0.148	0.513 ± 0.083	0.373 ± 0.058	**0.522 ± 0.080**	0.518 ± 0.085
SUSY	0.3	**0.502 ± 0.050**	0.465 ± 0.053	0.478 ± 0.051	0.366 ± 0.051	0.483 ± 0.043	0.474 ± 0.049
CODRNA	0.3	**0.824 ± 0.045**	0.804 ± 0.050	0.802 ± 0.060	0.623 ± 0.094	**0.819 ± 0.053**	0.808 ± 0.060
IJCNN	0.3	**0.580 ± 0.134**	0.518 ± 0.136	**0.593 ± 0.087**	0.480 ± 0.066	0.415 ± 0.226	**0.593 ± 0.134**
EPSILON	0.3	0.231 ± 0.054	0.180 ± 0.057	**0.245 ± 0.044**	0.213 ± 0.032	0.169 ± 0.046	**0.245 ± 0.046**
# Best		13	0	3	5	9	6

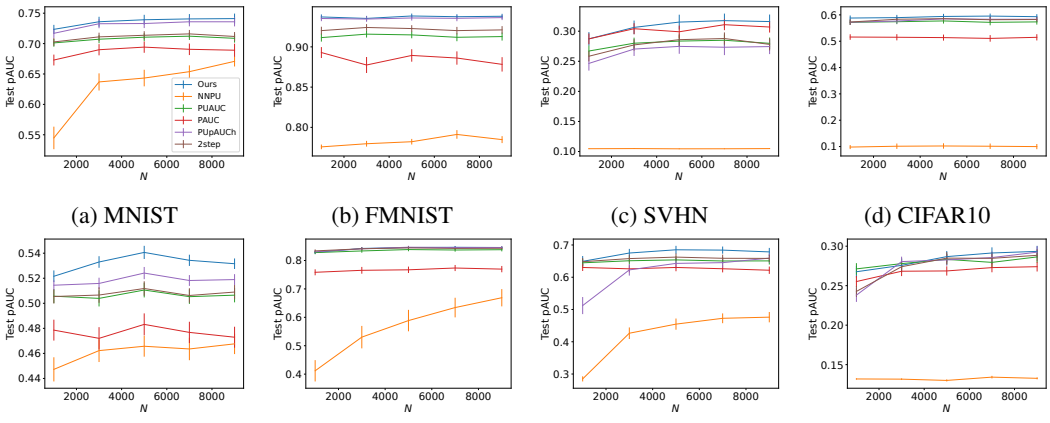
defined by α and β as negatives for training. In general, unlabeled data with high scores are more likely to be positive. Therefore, when $\alpha = 0$, such high score positive data tend to fall into the range, making it difficult for these methods to perform strongly. In contrast, when $\alpha > 0$, these positive data in the unlabeled data are less likely to be included in the range, which can lead to better performance. Nevertheless, the proposed method performed comparably to or better than these methods. Note that the proposed method outperformed these methods by a large margin in the case of $\alpha = 0$ as shown in Tables 1 and 2.

C.2 Results with $\pi = 0.01$ and $\pi = 0.3$

In the main paper, we evaluated the proposed method with $\pi = 0.1$ and $\pi = 0.2$. Here, we investigated the performance of the proposed method under smaller and larger class-prior values. Table 5 shows the average test pAUCs and their standard deviations with $\alpha = 0$, $\beta = 0.2$. The proposed method performed well. These results show that the proposed method is effective across a wide range of class-prior values.

Table 6: The results using the logistic loss: test average pAUCs and their standard deviations with $\alpha = 0$, $\beta = 0.2$ over different labeled positive data sizes $N^{\mathrm{p}} = 30$, 50, and 100. Values in bold are not statistically different at the 5% level from the best performing method in each row according to a paired t-test. '# Best' row represents the number of results of each method that are the best or comparable to it.

Data	π	Ours	PUAUC	PAUC	PUpAUCh	2step
MNIST	0.1	**0.759 ± 0.058**	0.727 ± 0.047	0.743 ± 0.057	0.752 ± 0.055	0.732 ± 0.048
	0.2	**0.727 ± 0.066**	0.699 ± 0.050	0.652 ± 0.060	**0.719 ± 0.057**	0.700 ± 0.049
FMNIST	0.1	**0.945 ± 0.023**	0.915 ± 0.034	0.933 ± 0.027	**0.944 ± 0.024**	0.929 ± 0.029
	0.2	**0.939 ± 0.024**	0.907 ± 0.036	0.854 ± 0.057	**0.936 ± 0.026**	0.915 ± 0.034
SVHN	0.1	**0.346 ± 0.095**	0.302 ± 0.070	0.328 ± 0.071	0.330 ± 0.096	0.308 ± 0.073
	0.2	**0.285 ± 0.091**	0.262 ± 0.061	**0.268 ± 0.050**	0.227 ± 0.069	0.259 ± 0.062
CIFAR10	0.1	**0.613 ± 0.070**	0.592 ± 0.072	0.552 ± 0.078	0.602 ± 0.066	**0.606 ± 0.068**
	0.2	**0.579 ± 0.072**	0.559 ± 0.079	0.452 ± 0.069	**0.572 ± 0.065**	**0.564 ± 0.079**
SUSY	0.1	**0.551 ± 0.033**	0.516 ± 0.042	0.514 ± 0.048	0.541 ± 0.028	0.521 ± 0.040
	0.2	**0.525 ± 0.044**	0.493 ± 0.043	0.427 ± 0.053	0.509 ± 0.042	0.495 ± 0.042
CODRNA	0.1	**0.853 ± 0.034**	0.849 ± 0.043	0.824 ± 0.048	**0.854 ± 0.035**	**0.856 ± 0.038**
	0.2	**0.837 ± 0.040**	0.828 ± 0.049	0.707 ± 0.078	**0.839 ± 0.039**	**0.834 ± 0.047**
IJCNN	0.1	**0.712 ± 0.081**	0.663 ± 0.089	0.685 ± 0.071	0.688 ± 0.100	0.680 ± 0.085
	0.2	**0.652 ± 0.122**	0.628 ± 0.092	0.572 ± 0.069	0.595 ± 0.145	**0.633 ± 0.094**
EPSILON	0.1	**0.303 ± 0.058**	**0.297 ± 0.055**	0.290 ± 0.047	**0.305 ± 0.054**	0.295 ± 0.056
	0.2	**0.272 ± 0.049**	**0.269 ± 0.048**	0.249 ± 0.040	**0.258 ± 0.054**	**0.269 ± 0.055**
# Best		16	2	1	8	6



(a) MNIST (b) FMNIST (c) SVHN (d) CIFAR10

(e) SUSY (f) CODRNA (g) IJCNN (h) EPSILON

Figure 5: The average test pAUCs and their standard errors with $\alpha = 0$, $\beta = 0.2$ over different class-priors $\pi \in \{0.1, 0.2\}$ and positive data sizes $N^{\mathrm{p}} \in \{30, 50, 100\}$ when changing the number of unlabeled data N.

C.3 Results using the Logistic Loss

In the main paper, we used the sigmoid function as the surrogate function in Eq. (15). Here, we evaluated the (p)AUC-based methods including the proposed method using the logistic loss as the surrogate function. Table 6 shows the results. The results are similar to those obtained with the sigmoid function in Table 2, and the proposed method still outperforms the other methods. These results suggest that the proposed method is effective with multiple surrogate functions.

C.4 Results with Different Numbers of Unlabeled Training Data

Figure 5 shows the average test pAUCs with their standard errors when changing the number of unlabeled training data N. The proposed method tended to perform better as N increased and consistently performed well compared to other methods across different N.

Table 7: Training time [sec] of each method on the MNIST and CIFAR10 datasets, where $(\alpha, \beta) = (0, 0.2)$, $\pi = 0.2$, $N^{\mathrm{P}} = 100$.

Data	N	Ours	NNPU	PUAUC	PAUC	PUpAUCh	2step
MNIST	5,000	90.460	90.054	90.036	90.978	89.618	184.040
CIFAR10	9,000	305.871	304.184	302.897	305.471	302.469	608.849

Table 8: The results on Mammography and Online-shoppers: test average pAUCs and their standard deviations with $\alpha = 0$, $\beta = 0.1$ and $\alpha = 0$, $\beta = 0.2$. Values in bold are not statistically different at the 5% level from the best performing method in each row according to a paired t-test.

Data	β	Ours	NNPU	PUAUC	PAUC	PUpAUCh	2step
Mammography	0.1	**0.784 ± 0.041**	0.001 ± 0.001	0.754 ± 0.041	**0.780 ± 0.045**	**0.786 ± 0.037**	**0.784 ± 0.037**
Mammography	0.2	**0.821 ± 0.039**	0.004 ± 0.004	0.810 ± 0.032	**0.831 ± 0.039**	**0.827 ± 0.036**	**0.834 ± 0.029**
Online-shoppers	0.1	**0.329 ± 0.081**	0.222 ± 0.072	0.259 ± 0.069	0.226 ± 0.032	0.308 ± 0.085	0.263 ± 0.071
Online-shoppers	0.2	**0.442 ± 0.079**	0.331 ± 0.079	0.371 ± 0.073	0.386 ± 0.063	0.425 ± 0.079	0.379 ± 0.073

C.5 Computation Costs

We investigated the training time of the proposed method on the MNIST and CIFAR10 datasets. We used a Linux server with a 2.20 GHz CPU. Table 7 shows the results. Since 2step additionally learns the classifier to create pseudo PN data, its training time was long. For the remaining methods, the training times were all similar. This result indicates that the proposed method is sufficiently practical in terms of the computation cost.

C.6 Evaluation on Real-World Medical and Advertising Datasets

In Section 1, we presented examples from the medical and advertising domains to motivate our problem setting. In this section, we evaluated the proposed method on Mammography (Vanschoren et al., 2014) and Online-shoppers (Sakar et al., 2019), which are real-world medical and advertising-related datasets, respectively.

Mammography is a dataset for detecting microcalcifications, which are treated as positives, with a positive class-prior of 0.023. Due to the limited (positive) data size, we used 30 and 20 positive data for training and validation, respectively; $5,000$ and $1,000$ unlabeled data for training and validation, respectively; and $3,000$ test data. The unlabeled and test data were created to preserve the original class-prior value. Online-shoppers is a dataset for predicting whether an online browsing session results in a purchase, with purchase sessions treated as positives. The positive class-prior is 0.155. This task is closely related to the advertising scenario motivating our work: it is important to identify sessions likely to result in purchases while maintaining a low FPR, since falsely targeting non-purchase sessions can waste advertising resources and degrade user experience. For this dataset, we changed the positive training data size within $\{30, 50, 100\}$ as in the other datasets and used $1,000$ test data. The sizes of other datasets were the same as those for Mammography. The unlabeled data were created to preserve the original class-prior. Table 8 shows the results. The proposed method performed well for all cases. These results demonstrate the effectiveness of the proposed method on both medical and advertising-related datasets.

C.7 Empirical Analysis of FPRs Induced by the Estimated Thresholds

We tracked the FPRs with the estimated thresholds, $\mathrm{FPR}(\hat{t}_\alpha)$ and $\mathrm{FPR}(\hat{t}_\beta)$, during training to empirically illustrate their behavior.[3] In this experiment, $\hat{t}_\alpha$ and $\hat{t}_\beta$ were estimated from training PU mini-batch as described in Algorithm 1, and the FPRs induced by the estimated thresholds were calculated on different fixed $2,500$ negative instances.

Table 9 shows the average FPRs induced by the estimated thresholds at selected training epochs for FMNIST and CODRNA, providing examples from an image dataset and a tabular dataset, respectively. For both datasets, $\mathrm{FPR}(\hat{t}_\beta)$ remained close to the target value β throughout training. In

[3]We did not track the estimated thresholds themselves because they may change substantially when the scale or offset of the score function changes, even when the resulting pAUC remains unchanged.

Table 9: FPRs induced by the estimated thresholds during training.

Data	α	β	N^p	FPR	Epoch 1	Epoch 10	Epoch 50	Epoch 100
FMNIST	0.0	0.1	30	$\mathrm{FPR}(\hat{t}_\alpha)$	0.025	0.031	0.031	0.031
	0.0	0.1	100	$\mathrm{FPR}(\hat{t}_\alpha)$	0.017	0.020	0.022	0.023
	0.0	0.1	30	$\mathrm{FPR}(\hat{t}_\beta)$	0.118	0.115	0.115	0.116
	0.0	0.1	100	$\mathrm{FPR}(\hat{t}_\beta)$	0.107	0.109	0.108	0.109
	0.0	0.2	30	$\mathrm{FPR}(\hat{t}_\alpha)$	0.022	0.030	0.031	0.031
	0.0	0.2	100	$\mathrm{FPR}(\hat{t}_\alpha)$	0.013	0.021	0.021	0.022
	0.0	0.2	30	$\mathrm{FPR}(\hat{t}_\beta)$	0.217	0.208	0.208	0.208
	0.0	0.2	100	$\mathrm{FPR}(\hat{t}_\beta)$	0.210	0.204	0.200	0.200
CODRNA	0.0	0.1	30	$\mathrm{FPR}(\hat{t}_\alpha)$	0.061	0.064	0.058	0.058
	0.0	0.1	100	$\mathrm{FPR}(\hat{t}_\alpha)$	0.027	0.033	0.044	0.046
	0.0	0.1	30	$\mathrm{FPR}(\hat{t}_\beta)$	0.140	0.138	0.129	0.130
	0.0	0.1	100	$\mathrm{FPR}(\hat{t}_\beta)$	0.114	0.115	0.113	0.115
	0.0	0.2	30	$\mathrm{FPR}(\hat{t}_\alpha)$	0.058	0.058	0.056	0.057
	0.0	0.2	100	$\mathrm{FPR}(\hat{t}_\alpha)$	0.027	0.027	0.041	0.041
	0.0	0.2	30	$\mathrm{FPR}(\hat{t}_\beta)$	0.223	0.216	0.214	0.215
	0.0	0.2	100	$\mathrm{FPR}(\hat{t}_\beta)$	0.208	0.201	0.204	0.204

contrast, $\mathrm{FPR}(\hat{t}_\alpha)$ showed somewhat larger deviations from the target $\alpha = 0$, which corresponds to an extreme tail of the negative-score distribution. Nevertheless, increasing the number of positive training instances from $N^p = 30$ to $N^p = 100$ consistently reduced the deviations from the target FPRs for both thresholds. These results suggest that the amount of positive training data can be important for accurate threshold estimation.

C.8 Empirical Analysis of the Heuristic Scoring Assumption

The existing pAUC maximization method, PUpAUCh, trains the score function by heuristically treating all unlabeled instances whose ranks based on their current scores fall within a specified range as negatives (Iwata et al., 2020). Here, we evaluate the quality of these pseudo-negatives to investigate to what extent their heuristic assumption that the selected unlabeled instances are negative holds during training.

Specifically, we first define

$$e = \frac{|\{\mathbf{x} \in B : y = +1\}|}{|B|}, \tag{118}$$

where B denotes the set of selected pseudo-negative instances within the mini-batch unlabeled data. Thus, e represents the proportion of selected instances that are actually positive. If instances are selected uniformly at random from the unlabeled data, the expected proportion of positives among them is π. Thus, we define

$$c = \frac{e}{\pi}, \tag{119}$$

where $c = 1$ indicates that the selection is random, $c < 1$ indicates that it is better than the random selection, and $c > 1$ indicates that it is worse than the random selection. A larger c means a greater violation of the PUpAUCh's heuristic scoring assumption.

Table 10 reports the average c during PUpAUCh training, together with the test pAUC gap from an oracle method that explicitly uses the true negative instances contained in the unlabeled data, i.e., Oracle − PUpAUCh. Here, we fixed $\pi = 0.1$, $\alpha = 0$, and $\beta = 0.1$. We first observe that $c > 1$ for most datasets, indicating that the pseudo-negatives selected by PUpAUCh often contain more positive instances than expected under random selection. This result empirically shows that the heuristic scoring assumption of PUpAUCh is frequently violated during training. Moreover, we observe a positive association between c and the performance gap between Oracle and PUpAUCh (Spearman's $\rho = 0.714$). This suggests that poorer pseudo-negative selection tends to be associated with a larger performance degradation relative to the oracle.

Table 10: Pseudo-negative selection quality of PUpAUCh (Iwata et al., 2020) and the test pAUC gap between Oracle and PUpAUCh. Larger c indicates greater positive contamination in the selected pseudo-negatives relative to random selection.

Data	c	Oracle − PUpAUCh
MNIST	1.5421	0.08379
FMNIST	0.6666	0.01786
SVHN	1.5432	0.09771
CIFAR10	1.5502	0.09186
SUSY	1.1720	0.04063
CODRNA	2.2437	0.04952
IJCNN	1.6922	0.13128
EPSILON	1.5265	0.03976