Optimizing Two-Way Partial AUC With an End-to-End Framework

Zhiyong Yang, Qianqian Xu, Senior Member, IEEE, Shilong Bao, Yuan He, Fellow, IEEE, Xiaochun Cao, Senior Member, IEEE, and Qingming Huang, Fellow, IEEE

Abstract—The Area Under the ROC Curve (AUC) is a crucial metric for machine learning, which evaluates the average performance over all possible True Positive Rates (TPR) and False Positive Rates (FPRs). Based on the knowledge that a skillful classifier should simultaneously embrace a high TPR and a low FPR, we turn to study a more general variant called Two-way Partial AUC (TPAUC), where only the region with TPR ≥ α, FPR ≤ β is included in the area. Moreover, a recent work shows that the TPAUC is essentially inconsistent with the existing Partial AUC metrics where only the FPR range is restricted, opening a new problem to seek solutions to leverage high TPAUC. Motivated by this, we present the first trial in this article to optimize this new metric. The critical challenge along this course lies in the difficulty of performing gradient-based optimization with end-to-end stochastic training, even with a proper choice of surrogate loss. To address this issue, we propose a generic framework to construct surrogate optimization problems, which supports efficient end-to-end training with deep learning. Moreover, our theoretical analyses show that: 1) the objective function of the surrogate problems will achieve an upper bound of the original problem under mild conditions, and 2) optimizing the surrogate problems leads to good generalization performance in terms of TPAUC with a high probability. Finally, empirical studies over several benchmark datasets speak to the efficacy of our framework.

Index Terms—AUC Optimization, machine learning, partial AUC.

I. INTRODUCTION

ROC (Receiver Operating Characteristics) curve is a well-known tool to evaluate classification performance at varying threshold levels. More precisely, as shown in Fig. 1(a), it captures the relationship between True Positive Rate (TPR) and False Positive Rate (FPR) as a function of the classification thresholds. AUC (Area Under the ROC Curve), summarizes the average performance of a given classifier by calculating its area. More intuitively, as shown in [23], AUC is equivalent to the possibility that a positive instance has a higher predicted score to be positive than a negative instance. Any skillful classifier that can produce well-separated scores for positive and negative instances will enjoy a high AUC value, no matter how skewed the class distribution is. As a natural result, AUC is more appropriate than accuracy for long-tail classification problems such as disease prediction [24], [80] and rare event detection detection [37], [41], [69], [71], due to this appealing property [19], [22].

Over the past two decades, the importance of AUC has raised a new wave to directly optimize AUC, which has achieved tremendous success. A partial list of the related studies includes [2], [12], [21], [27], [28], [48], [50]. However, the vast majority of such studies only consider the area over the entire ROC curve. As argued by [49], for some applications, only the performance within a given range of FPR is of interest, as shown in Fig. 1(b). In this sense, the standard AUC tends to provide a biased estimation of the performance by including unrelated regions. This key investigation has motivated a series of successful studies to optimize the One-way Partial AUC (OPAUC) with an FPR range [α, β] [48], [49], [50]. Here we note that the choice to truncate FPR on the ROC curve is based on domain-specific prior knowledge for some specific fields such as biometric screening, and medical diagnosis [49].

Taking a step further, what should be a general rule to select the target region under the ROC curve for classification problems?
Since TPR and FPR evaluate complementary aspects of the model performance, we argue that a practical classifier in most applications must simultaneously have a high TPR and a low FPR. In other words, a high TPR is meaningless if the FPR is lower than a tolerance threshold, while a low FPR cannot compensate for a low TPR (say, one can hardly consider a model with FPR higher than 0.8 even if its TPR is as high as 0.99, and vice versa for a low TPR model). In this sense, we only need to pay attention to the upper-left head region under the ROC curve, as shown in Fig. 1(c).

A recent work [73] exactly realizes this idea, where a new metric called Two-Way Partial AUC (TPAUC) is proposed to measure the area of a partial region of ROC within an FPR range; (c) The Two-way Partial AUC (TPAUC).

Inspired by this fact, we present the first trial to optimize the TPAUC metric with an end-to-end framework.

The major challenge of this task is that the objective function is not differentiable even with a proper surrogate loss function, suggesting that there is no easy way to perform end-to-end training. Facing this challenge, we propose a generic framework to approximately optimize the TPAUC with the help of deep learning. Generally speaking, our contributions are as follows.

First, we reformulate the original optimization problem as a bi-level optimization problem, where the inner-level problem provides a sparse sample selection process and the outer-level problem minimizes the loss over the selected instances. On top of the reformulation, we propose a generic framework to construct surrogate optimization problems for the original problem. In the core of this framework lies the interplay of the surrogate penalty functions and surrogate weighting functions defined in this article. Moreover, we construct a dual correspondence between these two classes of functions, such that we can easily find a standard single-level surrogate optimization problem whenever a surrogate penalty or a surrogate weighting function is obtained. We also provide a minimax reformulation of the objective function to induce an instance-wise objective function to improve efficiency.

We then proceed to explore theoretical guarantees for the framework. First, by comparing the surrogate problem and the original problem, we provide a mild sufficient condition under which the surrogate objective function becomes an upper bound of the original one. Meanwhile, we also show that the concave weighting function tends to be a better choice than its convex counterparts. Second, we present a theoretical analysis of the Bayes-optimal scoring functions for TPAUC optimization. The result shows that the Bayes scoring function for TPAUC is much harder to obtain than the plain AUC. Finally, we show that optimizing the surrogate problems could leverage good generalization performance in TPAUC, with two upper bounds of the excess risk. One of our results attains much sharper order than previous results on AUC generalization.

This article extends our ICML 2021 long-talk article [75], where we proposed a generic framework to implement the end-to-end TPAUC optimization. In this version, we have included systematic improvements in theory, methodology, and experiments. The novelty of the extended version is summarized as follows:

1) New Theoretical Results. We derive the Bayes-optimal solution for TPAUC optimization in Theorem 2, Theorem 3. The result shows that the optimal scoring functions are required not only to be aligned with \( \eta(x) \) in a local region, but also to choose the optimal hard examples. Moreover, under Assumption 1, we derive a sharp upper bound for the excess risk upper bound for a general form of hypothesis classes in Theorem 5.

2) New Methodology. The original objective function is formed as a sum of loss functions of instance pairs, which might lead to a heavy computational burden and the difficulty of generating a mini-batch. To address this issue, we propose an instance-wise minimax reformulation of the original objective function in Theorem 1.

3) New Experiments. In the current version, we include a series of new experiments, including new competitors, new finer-grained analysis, and the validation of the upper bounds in Proposition 2.

4) Miscellaneous Contents. Besides, we also refine the details of some existing contents, including the review of the related works (throughout Section II), the analysis of the inconsistency between TPAUC and OPAUC (in Section III-C), and practical illustration of Proposition 2 (in Section IV-A).

II. PRIOR ART

A. General AUC Optimization

Back in 1954, ROC made its first appearance in the literature of signal processing theory [55] as a tool to study the detectability of signals in the presence of noise, which has thereafter become a fundamental tool for signal detection together with AUC [44], [60]. Two decades ago, it started to gain attention from machine learning community due to its elegant statistical properties [10], [11], [22], [36], [46], [70]. For example, [36] argues that ROC or AUC is a better metric than accuracy in terms of the discriminancy to differentiate model performance. At almost the same time, a motivating study [15] further points out that maximizing AUC should not be replaced with minimizing...
the error rate, which shows the necessity to study direct AUC optimization methods.

After that, a series of algorithms are designed for optimizing AUC. At the early stage, the majority of studies focus on a full-batch off-line setting. [2] optimizes AUC based on a logistic surrogate loss function and ordinary gradient descent method. [12] approximates the original 0-1 loss with a polynomial function and obtains an efficient optimization method. RankBoost [20] provides an efficient ensemble-based AUC learning method based on a ranking extension of the AdaBoost algorithm. The work of [28], [78] constructs SV $M^{struc}$-based frameworks that optimize a direct upper bound of the 0 - 1 loss version AUC metric instead of its surrogates. [43], [59] also study the joint effort of AUC optimization and regression.

Later on, to accommodate Big Data analysis, researchers start to explore online extensions of the AUC optimization methods. [79] provides an early trial for this direction based on the reservoir sampling technique. [21] provides a completely one-pass AUC optimization method for streaming data based on the squared surrogate loss. Moreover, [76] reformulates the squared-loss-based stochastic AUC maximization problem as a stochastic saddle point problem. The new saddle point problem’s objective function only involves the summations of instance-wise loss terms, which significantly reduces the burden from the pairwise formulation. [52], [53] further accelerate this framework with tighter convergence rates. Most recently, it also has been applied to structured models [58], deep learning [38] and federated learning [77].

In this article, we take a further step to optimize the TPAUC. From the optimization perspective, the difficulty comes from the much more complicated formulation of the surrogate risk of TPAUC. To address this issue, we derive an approximated surrogate risk to support end-to-end training. Moreover, we also employ the saddle point reformulation technique [76] to derive an instance-wise objective function.

### B. Generalization Analysis for AUC

It is well-known that the recipe of machine learning comes from generalization, namely, the ability of a well-trained model to predict unseen data points. Generally speaking, its theoretical foundation could be traced back at least to the PAC theory of Valiant [65] and the statistical learning theory of Vapnik [66]. Nowadays, understanding the mathematical foundation of generalization has emerged as one of the most long-standing goals for the machine learning community [7], [8], [16], [45]. It is noteworthy that the standard techniques for generalization analysis [7], [16], [45] require the empirical risk function to be expressed as a sum of independent terms. Unfortunately, the AUC-based empirical risks do not satisfy this assumption, which poses a big challenge toward understanding its generalization guarantee. To solve this problem, many efforts have been made against the interdependency of AUC loss terms. The first study on the generalization property of the AUC optimization problem could be traced back to [1], where a specific notion of VC-dimension is proposed for the AUC risks. Later, [63] derived a novel symmetrization technique for the pairwise AUC risk function, where a novel notion of Rademacher complexity is proposed for the AUC risks. However, both of the studies require significant revision of the standard technique. Thereafter, a new wave of ideas emerges where the AUC risks are reformulated into a (series of) standard risk function(s). For example, [3], [64] employ the graph coloring technique to divide interdependent sample sets into a sequence of independent subsets. Within each subset, the samples satisfy the standard assumption so that traditional techniques such as the McDiarmid’s inequality [9] can be extended to interdependent datasets. Meanwhile, [13] proposes a U-statistics-based framework to derive generalization bounds for bipartite ranking, which is closely related to the AUC optimization problem. The key idea is to reformulate the U-statistics as a sum of independent terms. Moreover, it also proposes sharp results under certain assumptions. Later on, [56] provides a novel Talagrand concentration inequality for sub-additive and fractionally self-bounding functions, making it possible to introduce the notion of local Rademacher complexity [6] into the AUC optimization problems.

Besides the binary setting of the AUC optimization, there are also new explorations toward the generalization ability of AUC-like metrics under more complicated scenarios such as multipartite ranking [62], [68], multi-class classification [74] and fairness learning [67].

In this work, we will present two generalization results under two different assumptions for TPAUC optimization problem. We could find an upper bound for the excess risk in order of $O\left(\frac{\text{polylog}(\max\{n_+, n_-, n\})}{\min\{n_+, n_-, n\}}\right)$ which is sharper than the state-of-art result $O\left(\frac{1}{\min\{n_+, n_-, n\}}\right)$ for AUC [56], [64] and pretty close to the $O(1/n)$ result for U-statistics [13] but with much weaker assumptions. Here, $n_+, n_-, n$ are the number of positive, negative and total instances for a binary classification dataset, respectively.

### C. Partial AUC Optimization

Generally speaking, directly optimizing the OPAUC requires techniques of combinatorial optimization. As a classical work, [48] proposes a cutting-plane-based framework to perform the optimization. Later on [48], [50] provide refinement of [48] by formulating a tight convex upper bound of the original objective function. Recently, [26], [72] also extend this framework to time-series prediction and semi-supervised learning. [29] proposes an online learning method for OPAUC optimization with sublinear regret and online to batch conversion bounds. However, it requires the convexity of the loss function and a memory buffer. Consequently, though scaling well for non-deep learning models, it still cannot be employed to perform deep learning.

To make end-to-end training of deep models available, [4], [5] present meaningful attempts to apply OPAUC to the speaker verification task. Generally speaking, such algorithms adopt the following steps to perform the optimization: a) construct the pairwise training set by random sampling; b) pick the negative imposters over the FPR range $[\alpha, \beta]$; c) optimize the one-way partial AUC with the picked negatives and all positive samples.
In this way, one can perform end-to-end training on top of the generated dataset. However, their work is specifically designed for metric learning settings, which is not available for generic scenarios. Besides, there is no theoretical guarantee that the training on the generated dataset could generalize well on unseen data.

Compared with existing studies to optimize partial AUC, the key difference is two-fold. First, the previous studies only focus on one-way partial AUC, where only the FPR is restricted within $[\alpha, \beta]$; while we are the first to study TPAUC optimization, a new AUC metric where both TPR and FPR are truncated. Moreover, most related studies either adopt the cutting plane framework that is not scalable toward the training samples or adopt the generated dataset without a theoretical guarantee. In our work, getting rid of complicated combinatorial optimization techniques, we propose a general framework to construct much simpler surrogate optimization problems for TPAUC that supports end-to-end training based on systematic theoretical guarantees.

D. Learning With Constrained Optimization

According to the work [18], we can as well formulate the AUC maximization as a constrained optimization problem by binning the FPR range into a finite number of thresholds and constraining them to satisfy specific FPR/TPR values. In this sense, an alternative way to optimize partial AUC is to consider the constrained optimization framework proposed in [47], [51]. For example, based on the most recent work in this direction [30], it is possible to perform OPAUC optimization by considering a selected range of FPRs in the constraints.

By contrast, our article aims to formulate a partial AUC optimization problem on top of a simple optimization framework, which could be implemented at ease with the deep learning framework. Hence, our work is based on the unconstrained form of AUC risk. The constrained form of TPAUC optimization is considered an opening problem worth broad future effort.

E. Top Instance Ranking

Last but not least, we discuss the major difference between our work with another similar direction of top instance ranking, where only the ranking of the first few instances in the ranking list is optimized [14], [33]. First of all, in our setting, we are not optimizing the rank of the top instances but rather the relative ranking of the bottom positive instances and the top negative instances, which are much often harder to be ranked in the correct place. Naturally, our theoretical analysis also differs from what is carried out in these works. Second, our proposed framework directly supports end-to-end training of deep neural networks, which is not studied in the literature.

III. PRELIMINARIES

A. Standard AUC Metric

Before showing the formal definition of the two-way partial AUC, we first provide a quick review of the standard AUC metric. Under the context of binary classification problems, an instance is denoted as $(x, y)$, where $x \in \mathcal{X}$ is the input raw features and $y \in \{0, 1\}$ is the label. Taking a step further, given a dataset $\mathcal{D}$, denote $\mathcal{X}_P$ as the set of positive instances in our dataset, and $\mathcal{X}_N$ as the set of the negative ones, then the sampling process could be expressed as:

$$\mathcal{X}_P = \{x_i^+\}_{i=1}^{n_+} \sim \mathcal{P} : \mathbb{P} [x^+ | y = 1],$$

$$\mathcal{X}_N = \{x_i^-\}_{i=1}^{n_-} \sim \mathcal{N} : \mathbb{P} [x^- | y = 0],$$

where $n_+$, $n_-$ are the numbers of positive/negative instances, respectively; and $\mathcal{P}$, $\mathcal{N}$ are the corresponding conditional distributions. For binary class problems, our goal is to learn a score function $f : \mathcal{X} \to [0, 1]$, such that $f(x)$ is proportional to the possibility that $x$ belongs to the positive class. Based on the score function, we can further predict the label of an instance $x$ as $\mathbb{I}[f(x) > t]$, where $t$ is the decision threshold, $\mathbb{I}[\cdot]$ is the indicator function. Given a threshold $t$, we can define two elementary metrics known as True Positive Rate (TPR) and False Positive Rate (FPR), which are the probabilities that a positive/negative instance is predicted as a positive instance, i.e.:

$$\text{TPR}_f(t) = \mathbb{P}_{x^+ \sim \mathcal{P}} [f(x^+) > t],$$

$$\text{FPR}_f(t) = \mathbb{P}_{x^- \sim \mathcal{N}} [f(x^-) > t].$$

Based on the label predictions, AUC is defined as the Area under the Receiver Operating Characteristic (ROC) curve plotted by True Positive Rate (TPR) against False Positive Rate (FPR) with varying thresholds, which could be expressed mathematically as follows:

$$\text{AUC}(f) = \int_0^1 \text{TPR}_f \left( \text{FPR}_f^{-1}(t) \right) dt.$$  \hspace{1cm} (2)

When the possibility to observe a tied comparison is null, i.e.

$$\mathbb{P}_{x^+ \sim \mathcal{P}, x^- \sim \mathcal{N}} [f(x^+) = f(x^-)] = 0,$$

AUC is known [23] to enjoy a much simpler formulation as the possibility that correct ranking takes place between a positive and negative instance:

$$\text{AUC}(f) = 1 - \mathbb{E}_{x^+ \sim \mathcal{P}} \left[ \mathbb{E}_{x^- \sim \mathcal{N}} [\ell_{0,1}(f(x^+) - f(x^-))] \right],$$

where $\ell_{0,1}$ denotes the 0−1 loss with $\ell_{0,1}(x) = 1$ if $x < 0$, and $\ell_{0,1}(x) = 0$, otherwise. Given a finite dataset $\mathcal{S} = \mathcal{X}_P \cup \mathcal{X}_N$, the unbiased estimation of $\text{AUC}(f)$ could be expressed as:

$$\hat{\text{AUC}}(f) = 1 - \sum_{i=1}^{n_+} \sum_{j=1}^{n_-} \frac{\ell_{0,1}(f(x^+_i) - f(x^-_j))}{n_+ n_-}.$$  \hspace{1cm} (3)

B. Two-Way Partial AUC Metrics

Definitions. As presented in the introduction, instead of the complete area of ROC, we focus on the area of ROC in a partial region with $\text{TPR}_f(t) \geq t$, which is called two-way partial AUC in [73]. Here we define it as $\text{AUC}^\beta_{\alpha}$:

$$\text{AUC}^\beta_{\alpha}(f) = \int_{\text{FPR}_f(\text{TPR}_f^{-1}(1-\alpha))}^{\text{FPR}_f(\text{TPR}_f^{-1}(1-\beta))} \text{TPR}_f \left( \text{FPR}_f^{-1}(t) \right) dt - (1-\alpha) \cdot \left( \beta - \text{FPR}_f \left( \text{TPR}_f^{-1}(1-\alpha) \right) \right).$$

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Moreover, TPAUC could also be reformulated as a bipartite ranking accuracy across positive and negative classes:

\[ \text{TPAUC}(f) = 1 - \mathbb{E}_{x^+ \sim P} \left[ \mathbb{E}_{x^- \sim N} \left[ \delta_{x^+, \ell_{0.1}}(f(x^+) - f(x^-)) \right] \right], \]

where

\[ \delta_{x^+, \ell} = \mathbb{1} \{ f(x^+) \leq t_{1-\alpha}(f) \} \cdot \mathbb{1} \{ f(x^-) \geq t_{\beta}(f) \}, \]

and \( t_{1-\alpha}(f) \) and \( t_{\beta}(f) \) are two quantiles:

\[ t_{1-\alpha}(f) = \arg \min_{\delta \in \mathbb{R}} \left\{ \delta \in \mathbb{R} : \mathbb{E}_{x^+ \sim P} \left[ \mathbb{1} \{ f(x^+) \leq \delta \} \right] = \alpha \right\}, \]

\[ t_{\beta}(f) = \arg \min_{\delta \in \mathbb{R}} \left\{ \delta \in \mathbb{R} : \mathbb{E}_{x^- \sim N} \left[ \mathbb{1} \{ f(x^-) \geq \delta \} \right] = \beta \right\}. \]

Different from AUC, the extra term \( \delta_{x^+, \ell} \) allows TPAUC to attend to the hard positive (with a score below a quantile) and hard negative (with a score above a quantile).

Since the data distributions \( P, N \) are often unknown, it is necessary to study the empirical estimation of TPAUC based on an observed dataset \( S \). [73] derives an empirical version of \( \text{AUC}_\alpha^\beta(f, S) \), where both the expectation and the quantiles are replaced directly with their empirical version. We denote it as \( \text{AUC}_\alpha^\beta(f, S) \) in our article:

\[ \text{AUC}_\alpha^\beta(f, S) = 1 - \sum_{i=1}^{n^+_i} \sum_{j=1}^{n^-_j} \left( 0.1 \left( f(x^+_i) - f(x^-_j) \right) \right) \]

where \( x^+_i \) denotes the hard positive instance that achieves bottom-i score among all positive instances, and \( x^-_j \) denotes the hard negative instance achieves top-i score among all negative instances, \( n^+_i = |n_+ \cdot \alpha_i \| \), and \( n^-_j = |n_- \cdot \beta_j \| \), are the numbers of the chosen hard positive and negative examples.

### C. Inconsistency Between OPAUC and TPAUC

In this section, we show the inconsistency between TPAUC metric and the OPAUC metric. Mathematically, OPAUC calculates the partial AUC within FPR range \([\alpha, \beta]\), which could be defined as:

\[ \text{AUC}_\alpha^\beta_{\text{op}}(f) = \int_{\alpha}^{\beta} \text{TPR}_f \left( \text{FPR}_f^{-1}(t) \right) \, dt. \]

Recall that TPAUC could be defined as:

\[ \text{AUC}_\alpha^\beta_{\text{tp}}(f) = \int_{\text{FPR}_f^{-1}(1-\alpha)}^{\beta} \text{TPR}_f \left( \text{FPR}_f^{-1}(t) \right) \, dt \]

\[ - (1-\alpha) \cdot \left( \beta - \text{FPR}_f \left( \text{FPR}_f^{-1}(1-\alpha) \right) \right). \]

From the definitions, we can find that the TPAUC is intrinsically inconsistent with OPAUC. The source of the inconsistency is that both FTR\(_f\) and TPR\(_f\) are functions of \( f \). It is thus impossible to regard FPR\(_f^{-1}(1-\alpha)\) and TPR\(_f(\text{FPR}_f^{-1}(1-\alpha))\) as constants. Thus one cannot simply replace the FPR lower bound \( \text{FPR}_f^{-1}(1-\alpha) \) with any constant \( c \). Consequently, \( \text{AUC}_\alpha^\beta_{\text{tp}}(f) \) is in general not consistent with any OPAUC with FPR range \([c, \beta]\).

But can we estimate TPAUC with an OPAUC metric? A natural choice of such OPAUC could be defined as follows:

\[ \int_{\text{FPR}_f^{-1}(1-\alpha)}^{\beta} \text{TPR}_f \left( \text{FPR}_f^{-1}(t) \right) \, dt \]

which is called TPR-dependent OPAUC in our article since its lower limit is defined by a TPR lower bound. As suggested by [73], TPR-dependent OPAUC and TPAUC provide inconsistent results when the two ROC curves intersect. As shown in Fig. 2, we calculate TPAUC with \( \alpha = 0.4 \) and \( \beta = 0.6 \). It is easy to see that \( \text{TPAUC}_1 = S_6 + S_7 \) and \( \text{TPAUC}_2 = S_5 + S_7 \). Next, we calculate the TPR-dependent OPAUC. As shown in the figure, we have \( \text{OPAUC}_1 = S_6 + S_7 + S_9 \) and \( \text{OPAUC}_2 = S_5 + S_7 + S_8 + S_9 \). In this case, we have \( \text{OPAUC}_1 < \text{OPAUC}_2 \) but \( \text{TPAUC}_1 < \text{TPAUC}_2 \) whenever \( S_5 < S_5 + S_8 \) and \( S_5 > S_5 \). Moreover, \( \text{ROC}_1 \) tends to have better higher TPR than \( \text{ROC}_2 \) in the most points within the restricted ROC region. TPAUC is more consistent with a better performance curve than TPR-dependent OPAUC.

\[ \text{FPR lower bound: } p_0=0.60 \]

\[ \text{TPR lower bound: } q_0=0.60 \]

Fig. 2. TPR-dependent OPAUC versus TPAUC. Practically, OPAUC and TPAUC tend to exhibit different comparison results when two ROC curves intersect. For the two ROC curves in the figure, it is easy to see that \( \text{TPAUC}_1 = S_6 + S_7 \) and \( \text{TPAUC}_2 = S_5 + S_7 \). By calculating their tightest OPAUC upper bounds, we have \( \text{OPAUC}_1 = S_6 + S_7 + S_9 \) and \( \text{OPAUC}_2 = S_5 + S_7 + S_8 + S_9 \). In this case, we have \( \text{OPAUC}_1 < \text{OPAUC}_2 \) but \( \text{TPAUC}_1 < \text{TPAUC}_2 \) whenever \( S_5 < S_5 + S_8 \) and \( S_5 > S_5 \). Moreover, \( \text{ROC}_1 \) tends to have better higher TPR than \( \text{ROC}_2 \) in the most points within the restricted ROC region. TPAUC is more consistent with a better performance curve than TPR-dependent OPAUC.

--

[73]
D. Covering Number as a Measure of Complexity

To obtain an upper bound of the Rademacher complexity, we need to recall the notion of covering number and its chaining bound. First, we provide the definition of \(\epsilon\)-covering and the covering number, which is elaborated in Definitions 1 and 2.

Definition 1 (\(\epsilon\)-covering). [31] Let \((\mathcal{H}, d)\) be a (pseudo)metric space, and \(\Theta \in \mathcal{H}, \{h_1, \ldots, h_K\}\) is said to be an \(\epsilon\)-covering of \(\Theta\) if \(\Theta \subseteq \bigcup_{i=1}^{K} B(h_i, \epsilon)\), i.e., \(\forall \theta \in \Theta, \exists i \text{ s.t. } d(\theta, h_i) \leq \epsilon\).

Definition 2 (Covering Number). [31] Based on the notations in Definition 1, the covering number of \(\Theta\) with radius \(\epsilon\) is defined as:

\[
\mathcal{N}(\Theta, \epsilon, d) = \min \{ n : \exists \epsilon - \text{covering over } \Theta \text{ with size } n \}.
\]

To calculate the covering number, one has to first clarify the metric \(d\). In our proof, we will employ the following norms to induce a metric. Given an input set \(\mathcal{X}\), for functions \(f()\), \(\tilde{f}() : \mathcal{X} \rightarrow \mathbb{R}\) we have:

\[
\|f - \tilde{f}\|_{2,n+} = \sqrt{\frac{1}{n+} \sum_{i=1}^{n+} (f(x_i) - \tilde{f}(x_i))^2},
\]

\[
\|f - \tilde{f}\|_{2,n-} = \sqrt{\frac{1}{n-} \sum_{j=1}^{n-} (f(x_j) - \tilde{f}(x_j))^2}.
\]

IV. THE PROPOSED FRAMEWORK

A. A Generic Framework to Construct Surrogate Optimization Problems

Based on the empirical estimation shown in Section III-B, it is clear that optimizing TPAUC over a finite dataset \(S\) requires minimizing the following quantity:

\[
1 - \text{AUC}_\alpha(f, S) = \sum_{i=1}^{n^+} \sum_{j=1}^{n^-} \frac{\ell(v_{(i)}^+ - f(x_{(j)}^-))}{n^+ n^-}.
\]

Following the framework of surrogate loss [45], we replace the non-differential 0-1 loss with a convex loss function \(\ell\), such that \(\ell(t)\) is an upper bound of \(\ell_{0,1}(t)\). Note that if the scores live in \([0, 1]\), standard loss functions such as \(\ell_{\exp}(t) = \exp(-t)\), \(\ell_{sq}(t) = (1 - t)^2\) often satisfy this constraint. Hence given a feasible surrogate loss \(\ell\), our goal is then to solve the following problem:

\[
\text{(OP\#)} \min_\theta \tilde{R}_{\alpha, \beta}(S, f) = \sum_{i=1}^{n^+} \sum_{j=1}^{n^-} \frac{\ell(f(x_{(i)}^+), f(x_{(j)}^-))}{n^+ n^-},
\]

where \(f\) is parameterized by \(\theta\). Unfortunately, even with the choice of differentiable surrogate losses, the objective function \(\tilde{R}_{\alpha, \beta}(S, f)\) is still not differentiable. This is because calculating \(x_{(i)}^+, x_{(j)}^-(\tilde{f})\) requires sorting the scores of positive and negative instances. Nonetheless, the objective function is essentially a composition of a sparse sample selection operation and the original loss. This is shown in the following proposition, where \(\text{(OP\#)}\) is reformulated as a so-called bi-level optimization problem [39], [40]. The inner-level problems provide a sparse sample selection process, and the outer-level problem performs the optimization based on the chosen instances. Please see Appendix A, available online, for the proof.

Proposition 1. For any \(\alpha, \beta \in (0, 1)\), if scores \(f(x) \in [0, 1]\), and there are no ties in the scores, the original optimization problem is equivalent to the following problem:

\[
\min_\theta \frac{1}{n^+ n^-} \sum_{i=1}^{n} v_i^+ \cdot v_j^- \cdot \ell(f(x_i^+, x_j^-))
\]

s.t. \(v_+ = \arg\max_{v_i^+ \in [0,1], \sum_{i=1}^{n} v_i^+ \leq n^+} \sum_{j=1}^{n^-} (v_i^+ \cdot (1 - f(x_i^+)))\)

\(v_- = \arg\max_{v_j^- \in [0,1], \sum_{j=1}^{n} v_j^- \leq n^-} \sum_{i=1}^{n^+} (v_j^- \cdot f(x_j^-))\)

where

\[
\ell(f(x_i^+, x_j^-)) = \ell(f(x_i^+) - f(x_j^-)).
\]

Based on the proposition, the source of the intractability of \((\text{OP\#})\) comes from the \(\ell_1\) ball constraints \(n^+ \sum_{i=1}^{n^+} v_i^+ \leq n^+\), \(\sum_{j=1}^{n^-} v_j^- \leq n^-\) in the inner-level problem. Moreover, we can find out \(\lambda_+, \lambda_-\) such that the \(\ell_1\) constraints can be equivalently expressed as a penalty function:

\(v_+ = \arg\max_{v_i^+ \in [0,1]} \sum_{i=1}^{n^+} (v_i^+ \cdot (1 - f(x_i^+)) - \lambda^+ \cdot v_i^+\)

\(v_- = \arg\max_{v_j^- \in [0,1]} \sum_{j=1}^{n^-} (v_j^- \cdot f(x_j^-) - \lambda^- \cdot v_j^-)\)

To establish an efficient approximation of the original problem, we leave \(\lambda^+, \lambda^-\) as hyperparameters to be tuned. Furthermore, to avoid sparsity, we replace the sparsity-inducing \(\ell_1\) penalty with a smooth surrogate \(\phi_\gamma\). This naturally leads to a smooth problem:

\[
\text{(OP1)} \min_\theta \frac{1}{n^+ n^-} \sum_{i=1}^{n^+} \sum_{j=1}^{n^-} v_i^+ \cdot v_j^- \cdot \ell(f(x_i^+, x_j^-))
\]

s.t. \(v_+ = \arg\max_{v_i^+ \in [0,1]} \sum_{i=1}^{n^+} (v_i^+ \cdot (1 - f(x_i^+)) - \phi_\gamma(v_i^+))\)

\(v_- = \arg\max_{v_j^- \in [0,1]} \sum_{j=1}^{n^-} (v_j^- \cdot f(x_j^-) - \phi_\gamma(v_j^-))\)

To ensure that the chosen \(\phi_\gamma\) provides an effective approximation of the \(\ell_1\) penalty, we pose several regularities on such functions. In the following, we define this class of functions as the calibrated smooth penalty function.

Definition 3. A penalty function \(\phi_\gamma(x) : \mathbb{R}_{+} \rightarrow \mathbb{R}\) is called a calibrated smooth penalty function, if it satisfies the following regularities:

A) \(\phi_\gamma\) has continuous third-order derivatives.
B) \(\phi_\gamma\) is strictly increasing in the sense that \(\phi'_\gamma(x) > 0\).
C) \(\phi_\gamma\) is strictly convex in the sense that \(\phi''_\gamma(x) > 0\).
D) \(\phi_\gamma\) has positive third-order derivatives in the sense that \(\phi'''_\gamma(x) > 0\).
Note that since we restrict the output of $f$ to $[0, 1]$, condition (B) is inherited from the $\ell_1$ norm for non-negative inputs. While the other conditions improve the smoothness of the function. Moreover, the last condition is to ensure that the weighting function is strictly concave (see the arguments about the weights).

Given the penalty functions, we turn to explore a corresponding factor in the framework. According to the inner level problem, the sample weights $v_i^+$, $v_j^-$ have a dual correspondence with the penalty functions. More precisely, given a fixed $\phi$-one can can the corresponding weighting function $\psi_\gamma$ as a function of $f(x)$ such that:

$$v_i^+ = \psi_\gamma(1 - f(x_i^+)), \quad v_j^- = \psi_\gamma(f(x_j^-)), \quad v_i^+, v_j^- \in [0, 1].$$

Moreover, if $\psi_\gamma$ has a closed-form expression, then we can cancel the inner optimization problem and instead minimize the following weighted empirical risk $\hat{R}_\psi^f$:

$$\hat{R}_\psi^f(S, f) = \frac{1}{n_n + n_+} \sum_{i=1}^{n_n} \psi_\gamma(1 - f(x_i^+)) + \sum_{j=1}^{n_+} \psi_\gamma(f(x_j^-)) \cdot \ell(f, x_i^+, x_j^-). \quad (3)$$

In this sense, adopting a smooth penalty function ends up with a dual soft weighting strategy over the hard instances. Again, to reach a proper weight function, we also require it to satisfy some necessary regularities. In the following, we define this class of functions as the calibrated weighting function.

**Definition 4.** A weighting function $\psi_\gamma(x) : [0, 1] \rightarrow \text{Rng}$, where $\text{Rng} \subseteq [0, 1]$, is called a calibrated weighting function, if it satisfies the following regularities:

A) $\psi_\gamma$ has continuous second-order derivatives.

B) $\psi_\gamma$ is strictly increasing in the sense that $\psi_\gamma'(x) > 0$.

C) $\psi_\gamma$ is strictly concave in the sense that $\psi_\gamma''(x) < 0$.

In this definition, (B) is a natural requirement to make the weight proportional to the target instance’s difficulty. Condition (C) is an interesting trait in our framework. To see why this is necessary, let us note that the weight functions $v_i^+$, $v_j^-$ are continuous surrogates residing in $[0, 1]$ for threshold function

$$1 \left[ 1 - f(x_i^+) > 1 - f(x_i^{(n_n)}) \right],$$

$$1 \left[ f(x_j^-) > f(x_j^{(n_+)} \right],$$

respectively. To be simple, we continue our discussion with a general form $1[x > 0]$. Obviously, weight decay for large $x$ should be smooth such that the loss could attend at the top $f(x^+)$ and $(1 - f(x^-))$ scores. Moreover, to avoid over-fitting, the model should as well have sufficient memory of the easy examples. Hence the weights for such examples should not be too close to zero. These observations are exactly typical traits for a concave function. As shown in Fig. 3, we visualize the difference between a convex function $y = x^2$ and $y = 1[x > 0.5]$, and the difference between concave function $y = x^{0.05}$ and $y = 1[x > 0.5]$. Another reason to choose concave functions is that they can benefit the optimization process. More precisely, we expect the loss function $\hat{R}_\psi^f$ in (3) to be an upper bound of $\hat{R}_\psi^f$, such that minimizing $\hat{R}_\psi^f$ could also minimize the original loss. Back to Fig. 3, this is more likely to happen if $S_1/S_2$ is large. In fact, this is a condition which is much easier for concave functions to satisfy. This could be proved theoretically from the following proposition. From a quantitative perspective, we can provide a sufficient condition under which $\hat{R}_\psi^f \geq \hat{R}_\psi^{f, \alpha, \beta}$. Moreover, we show that a similar sufficient condition also holds in the population-level. Please see Appendix B, available online, for the proof.

**Proposition 2.** Given a strictly increasing weighting function $\psi_\gamma : [0, 1] \rightarrow [0, 1]$, such that $v_i^+ = \psi_\gamma(1 - f(x_i^+)), v_j^- = \psi_\gamma(f(x_j^-))$, denote:

$$I_{1}^+ = \{x_+ : x_+ \in \mathcal{X}_P, f(x_+) \geq f(x_+^{(n_n)})\},$$

$$I_{1}^- = \{x_- : x_- \in \mathcal{X}_N, f(x_-) \leq f(x_-^{(n_+)})\},$$

denote $I_2$ as $(\mathcal{X}_P \times \mathcal{X}_N) \setminus (I_{11}^+ \times I_{11}^-)$; denote $\mathbb{E}_{x^+ \in I_{11}^+}[x]$ as the empirical expectation of $x$ over the set $I_{11}^+$, and $\mathbb{E}_{x^+ \in I_{11}^+}, \mathbb{E}_{x^+ \in I_{11}^-}, \mathbb{E}_{x^+, x^- \in I_{11}}$ are defined similarly; define $\xi_{i,j} = \ell(f, x_i^+, x_j^-)$. We assume that

$$n_n^+ \in \mathbb{N}, n_n^- \in \mathbb{N}, f(x^+), f(x^-) \in (0, 1),$$

then:

a) A sufficient condition for $\hat{R}_{\psi, \alpha, \beta}^f(S, f) \leq \hat{R}_\psi^f(S, f)$ is that:

$$\sup_{p \in (0, 1), q = \frac{1}{p}} \left[ \rho_p - \xi_q \right] \geq 0,$$

where

$$\rho_p = \frac{\left( \mathbb{E}_{x^+ \in I_{11}^+} \left[ v_i^+ \cdot p^p \right] \right)^{1/p}}{\left( \mathbb{E}_{x^+ \in I_{11}^+} \left[ (1 - v_i^+ v_j^-)^2 \right] \right)^{1/2}},$$

$$\xi_q = \frac{\frac{\alpha \beta}{1 - \alpha \beta} \cdot \left( \mathbb{E}_{x^+ \in I_{11}^+} \left[ f_{i,j}^+ \right] \right)^{1/2}}{\left( \mathbb{E}_{x^+ \in I_{11}^+} \left[ f_{i,j}^- \right] \right)^{1/2}}.$$

b) A sufficient condition for $R_{\psi, \alpha, \beta}^f(f) \leq R_{\psi}^f(f)$ is that:

$$\sup_{p \in (0, 1), q = \frac{1}{p}} \left[ \rho_p - \xi_q \right] \geq 0,$$
where

$$
\rho_p = \frac{\left( \mathbb{E}_{x^+, x^+ \sim \mathcal{D}_x} \left[ \left( v^p_+ \cdot v^p_+ \right)^{1/p} \right] \right)^{1/p}}{\left( \mathbb{E}_{x^+, x^+ \sim \mathcal{D}_x} \left[ \left( 1 - v^+ v^+ \right)^2 \right] \right)^{1/2}},
$$

$$
\xi_q = \frac{\alpha \beta}{1 - \alpha \beta} \cdot \frac{\left( \mathbb{E}_{x^+, x^+ \sim \mathcal{D}_x} \left[ \left( f^q_{x^+} \right)^2 \right] \right)^{1/2}}{\left( \mathbb{E}_{x^+, x^+ \sim \mathcal{D}_x} \left[ \left( f^q_{x^+} \right)^2 \right] \right)^{1/2}}.
$$

c) If there exists at least one strictly concave $\psi_\gamma$ such that the $\hat{R}_{\alpha, \beta}^\alpha(S, f) > \hat{R}_{\alpha, \beta}^\beta(S, f)$, then $\hat{R}_{\alpha, \beta}^\alpha(S, f) > 0$ holds for all convex $\psi_\gamma$.

According to Proposition 2, $\hat{R}_{\alpha, \beta}^\alpha$ becomes an upper bound of $\hat{R}_{\alpha, \beta}$ if $\alpha, \beta$ are small, and the empirical distribution has significant masses at instances with moderate difficulty. To validate the result empirically, we conduct the following two sets of experiments.

1) First, we validate the effectiveness of the sufficient condition. To do this, we construct a distribution that $f_\theta(x^+) \sim N(0.5, 0.08), f_\theta(x^-) \sim N(0.3, 0.08)$, and sample 100 points for each class, and repeat it for 50 trials. For all these trials, we can find $p, q$ that satisfies the inequality: $\rho_p - \xi_q > 0$. Thus the sufficient condition holds for these trials.

2) Second, in Fig. 4(b), we plot the training curve of the relaxed surrogate empirical risk $\hat{R}_{\alpha, \beta}^\alpha$, the surrogate empirical risk $\hat{R}_{\alpha, \beta}^\beta$, and the $0-1$ empirical risk $\hat{R}_{0,1}^{\alpha, \beta}$-TPAUC for CIFAR-10-LT-Subset-1, with a concave polynomial weighting function. As shown in the figure, the relaxed risk function $\hat{R}_{\alpha, \beta}^\alpha$ is always an upper bound of the other two risk functions.

**Dual Correspondence Theory.** Now with the penalty function and weighting function clarified, we establish their dual correspondence with the following proposition. Please see Appendix C, available online, for the proof.

**Proposition 3.** Given a strictly convex function $\varphi_\gamma$, and define $\psi_\gamma(t)$ as:

$$
\psi_\gamma(t) = \arg \max_{v \in [0, 1]} v \cdot t - \varphi_\gamma(v),
$$

then we can draw the following conclusions:

a) If $\varphi_\gamma$ is a calibrated smooth penalty function, we have $\psi_\gamma(t) = \varphi_\gamma^{-1}(t)$, which is a calibrated weighting function.

b) If $\psi_\gamma$ is a calibrated weighting function such that $v = \psi_\gamma(t)$, we have $\varphi_\gamma(v) = \int \psi_\gamma^{-1}(v) dv + \text{const.}$, which is a calibrated smooth penalty function.

According to Proposition 3, given a calibrated smooth penalty function, one can obtain an implicit soft weighting strategy via Proposition 3(a). Likewise, given a calibrated weighting function, one can find an implicit regularizer over the sample weights via Proposition 3(b). Based on the regularities of the two components, both $\psi$ and $\varphi$ have closed-form formulations if the other one is known. This means that we can solve the bi-level optimization framework in (OP1) by simply minimizing the resulting $\hat{R}_{\alpha, \beta}^\alpha(S, f)$, leading to a much simpler optimization that can be solved directly by an end-to-end training framework. Consequently, the dual correspondence theory provides a simple way to establish a surrogate optimization problem of TPAUC, once a weighting function or a penalty function is at hand.

**B. An Instance-Wise Minimax Reformulation of the Relaxed Objective Function**

It is easy to see that the proposed objective function is formed as a sum of pairwise loss functions, which brings extra computational burden and the difficulty to generate a mini-batch. To address this issue, inspired by [76], we propose a minimax reformulation of the original objective function, where the objective function turns out to be an instance-wise loss sum. Note that, for the sake of convenience, we adopt the shorthand notation:

$$
v^+_i = \psi_\gamma(1 - f(x^+_i)), \quad v^-_i = \psi_\gamma(f(x^-_i)),
$$

in this subsection. Moreover, denote:

$$
c_+ = \frac{1}{n_+} \sum_{i=1}^{n_+} v^+_i, \quad c_- = \frac{1}{n_-} \sum_{j=1}^{n_-} v^-_j,
$$

$$
f^+ = \frac{1}{n_+} \sum_{i=1}^{n_+} v^+_i \cdot f(x^+_i), \quad f^- = \frac{1}{n_-} \sum_{j=1}^{n_-} v^-_j \cdot f(x^-_j),
$$

$$
\hat{f}^{+2} = \frac{1}{n_+} \sum_{i=1}^{n_+} v^+_i \cdot f^2(x^+_i), \quad \hat{f}^{-2} = \frac{1}{n_-} \sum_{j=1}^{n_-} v^-_j \cdot f^2(x^-_j),
$$

we come to the following theorem, please see Section ?? for the proof:

**Theorem 1.** Denote $v^\alpha_\infty = \sup_x |\psi_\alpha(x)|, f^\infty = \sup_x |f(x)|$, assume that $v^\alpha_\infty < \infty, f^\infty < \infty, \ell(t) = (1-t)^2$, (OP1) could
Algorithm 1: The Minimax Optimization Algorithm of the TPAUC Framework.

Input: \( D, \gamma \)
Output: best parameters \( \theta \)

1. Initialize \( a = b = 0 \).
2. \( C_\alpha = (1/\sqrt{2}) \cdot [1, 1, \sqrt{2}, \sqrt{2}, 1, 1, 1, \sqrt{2}] \);
3. \( C_b = (1/\sqrt{2}) \cdot [1, \sqrt{2}, \sqrt{2}, 1, 1, \sqrt{2}, \sqrt{2}] \);
4. Setup the stepsizes \( \eta_1, \eta_2, \eta_3 \).
5. while Not Converged do
6. Random sample a mini-batch \( B \) from \( D \);
7. Calculate \( \zeta_1, \zeta_2, a, b \), according to \( B \);
8. \( \theta = \theta - \eta_1 \cdot \nabla \theta r_i(\theta) \);
9. \( a = a - \eta_2 \cdot (\zeta_1 + 2 \cdot C_\alpha \odot \tilde{a}) \);
10. \( b = b + \eta_3 \cdot (\zeta_2 - 2 \cdot C_b \odot \tilde{b}) \);
11. return \( \theta \);

be reformulated as: \(^1\)

\[
\min_{\theta, a, b} \max_{a, b, \in [a_\min, a_\max]} \left| a \right|^2 + \left| b \right|^2,
\]

where

\[
\zeta_1 = -[c_+ + c_-, 2(f^+ + c_-), 2c_+, 2f^-, c_-, f^+ + c_+, f^- - 2],
\]

\[
\zeta_2 = [c_+ + c_-, 2c_-, 2f^+ + 2c_+, c_-, f^- - 2],
\]

\[
\tilde{a} = (1/\sqrt{2}) \cdot [a_1, a_2, \sqrt{2} \cdot a_3, \sqrt{2} a_4, \sqrt{2} a_5, a_6, a_7, a_8, a_9, \sqrt{2} \cdot a_{10}],
\]

\[
\tilde{b} = (1/\sqrt{2}) \cdot [b_1, \sqrt{2} \cdot b_2, \sqrt{2} b_3, \sqrt{2} b_4, b_5, b_6, \sqrt{2} b_7, \sqrt{2} b_8],
\]

\[
c_\alpha = [v^\alpha, v^\alpha_2, v^\alpha_3, f_\infty, v^\beta, v^\beta_2, v^\beta_3, f_\infty, f'^2, 2 f_\infty],
\]

\[
c_b = [2v^\beta_2, v^\beta_3, f_\infty, v^\beta_4 + f_\infty, v^\beta_5 + f_\infty, v^\beta_6 + f_\infty, f_\infty, f_\infty].
\]

C. Two Instantiations of the Generic Framework

Based on the generic framework, in this subsection, we provide two practical instantiations.

Polynomial Surrogate Model. From the penalty function perspective, the original \( \ell_1 \) penalty realizes \( \psi_\gamma = \gamma \cdot t \). In this way, it is a natural choice to adopt a polynomial penalty \( \varphi_\gamma \)

\[
C \cdot t^\gamma \text{ as a dense surrogate for } \ell_1. \text{ Inspired by this, we propose a polynomial surrogate model as example 1.}
\]

Example 1 (Polynomial Surrogate Model). In the polynomial surrogate model, we set:

\[
\varphi_\gamma^\text{poly}(t) = \frac{1}{\gamma} \cdot t^\gamma, \quad \psi_\gamma^\text{poly}(t) = t^{-\frac{1}{\gamma}}, \quad \gamma > 2
\]

The visualizations of the weights are shown in Fig. 5.

Example 2 (Exponential Surrogate Model). In the exponential surrogate model, we set:

\[
\varphi_\gamma^\text{exp}(t) = \left(1 - t \right) \left(\log(1 - t) - 1\right) + \frac{1}{\gamma}, \quad \psi_\gamma^\text{exp}(t) = 1 - e^{-\gamma t}
\]

Exponential Surrogate Model. Considering the properties of the weighting functions, we expect \( \psi_\gamma \) will have a flat landscape for large \( t \). Motivated by this, we adopt an exponential weighting function \( \psi_\gamma^\text{exp}(t) = 1 - e^{-\gamma t} \) (the landscape is shown in Fig. 5(c)(d)). The resulting model is then shown as Exp.2. The visualizations of the weights are shown in Fig. 5.

V. THEORETICAL ANALYSIS

In this section, we will present an investigation into the theoretical guarantees of the proposed framework. Specifically, our study includes two aspects. In the first subsection, we will derive the Bayes optimal scoring function to see what are the best solutions under TPAUC optimization. In the second subsection, we turn to analyze the generalization ability of the proposed framework, where we provide excess risk upper bounds under two different assumptions.

Note that all the proofs for the results shown in the Appendix: please see Appendix E, available online, for the details associated with Theorems 2 and 3; please see Appendix F, available online, for the details associated with Theorem 4; please see Appendix G, available online, for the details associated with Theorem 5.

A. Bayes Scoring Functions for TPAUC

In this subsection, we derive the set of Bayes learners realizing the optimized TPAUC in expectation.

First, given a scoring function \( f \), we define \( C_{f}^{\alpha, \beta}, C_{f}^{\alpha, \beta, \dagger}, C_{f}^{\alpha, \beta, \ddagger} \) as:

\[
C_{f}^{\alpha, \beta} = \{ x \in X : t_\beta(f(x)) \leq f(x) \leq t_{1-\alpha}(f) \};
\]

\[
C_{f}^{\alpha, \beta, \dagger} = \{ x \in X : f(x_1) > t_{1-\alpha}(f), f(x_2) \leq t_{1-\alpha}(f) \};
\]

\[
C_{f}^{\alpha, \beta, \ddagger} = \{ x \in X : f(x_1) < t_\beta(f), f(x_2) \geq t_\beta(f) \};
\]

where

\[
t_{1-\alpha}(f) = \arg\min_{x \sim P} \mathbb{E}_{x \sim P} \left[ \mathbb{I} \left[ f(x) \geq \delta \right] \right] = 1 - \alpha
\]

\[
t_\beta(f) = \arg\min_{x \sim \mathcal{N}} \mathbb{E}_{x \sim \mathcal{N}} \left[ \mathbb{I} \left[ f(x) \geq \delta \right] \right] = \beta
\]

and \( X \) is the domain of the input feature.
The following theorem reveals the set of all Bayes optimal scoring functions:

**Theorem 2.** Assume that \( t_\beta(f) \leq t_{1-\alpha}(f) \), and that there are no tied comparisons, \( f \) is a Bayes scoring function for \( \text{TPAUC}_\beta \) if it is a solution to the following problem:

\[
\min_{\hat{f}} \int_{x_1,x_2 \in \mathcal{C}_f} p(x_1) \cdot p(x_2) \cdot \min \{ \eta_{1,2}, \eta_{2,1} \} \, dx_1 \, dx_2 \\
+ 2 \int_{\mathcal{C}_f} p(x_1) \cdot p(x_2) \cdot \eta_{1,2} \, dx_1 \, dx_2 \\
+ 2 \int_{\mathcal{C}_f} p(x_1) \cdot p(x_2) \cdot \eta_{1,2} \, dx_1 \, dx_2
\]

Specifically, for all \( x_1, x_2 \in \mathcal{C}_f \), we have:

\[
(\eta(x_1) - \eta(x_2)) \cdot (f(x_1) - f(x_2)) > 0,
\]

where \( \eta(x) = \mathbb{P}[y = 1|x] \), \( \eta_{1,2} = \eta(x_1)(1 - \eta(x_2)) \), \( \eta_{2,1} = \eta(x_2)(1 - \eta(x_1)) \), and \( p(x) \) is the p.d.f. of the marginal distribution of \( x \).

To optimize TPAUC, we need to make sure that, within the set \( \mathcal{C}_f \), the scoring function \( f \) should preserve the ranking result of \( \eta \). Moreover, we need to choose the sets \( \mathcal{C}_f^{\alpha,\beta}, \mathcal{C}_f^{\alpha,\beta,1}, \mathcal{C}_f^{\alpha,\beta,2} \) to minimize the Bayes error. This is a much tougher condition than that for AUC (One does not need to choose any instance at all) and for OPAUC (Only FPR has a constraint). Unfortunately, the optimization problem in Theorem 2 does not have a closed-form solution in general. Moreover, we can see that \( \eta \) may not be a solution of the Bayes classifier since the choice of \( C_f^{\alpha,\beta}, C_f^{\alpha,\beta,1}, C_f^{\alpha,\beta,2} \) with \( f = \eta \) might not be optimal. This observation is similar to [14], where \( \eta \) is proved suboptimal for metric like OPAUC. However, if we restrict the choice of \( f \) in a simple subclass of the hypothesis, it is possible to find a much simpler solution. To do this, we first define the \((\alpha, \beta)\)-weakly separable \( \eta \) functions.

**Definition 5** \((\alpha, \beta)\)-Weakly Separable \( \eta \). \( \eta(x) = \mathbb{P}[y = 1|x] \) is said to be \((\alpha, \beta)\)-weakly separable w.r.t. \( f \), if:

\[
\eta(x) = \begin{cases} 
1, & f(x) \geq t_{1-\alpha}(f) \\
0, & f(x) \leq t_\beta(f) \\
\in (0,1), & \text{otherwise}
\end{cases}
\]

Then the following theorem shows that if we focus on the subclass \( \mathcal{F}_a \), then the optimal scoring functions would have a simpler expression.

**Theorem 3.** Under the same setting as Theorem 2, if we restrict our choice of \( f \) in \( \mathcal{F}_a \), where

\[
\mathcal{F}_a = \{ f : \eta \text{ is } (\alpha, \beta)\text{-weakly separable w.r.t } f, \\
f(x_1) \neq f(x_2), \forall x_1 \neq x_2, x_1, x_2 \in \mathcal{X} \},
\]

Assume that when \( 1 > \eta(x_1) > 0, 1 > \eta(x_2) > 0 \), we have:

\[
\eta(x_1) \neq \eta(x_2), \forall x_1 \neq x_2, x_1, x_2 \in \mathcal{X}
\]

Then \( f \) minimizes the expected risk in \( \mathcal{F}_a \), if:

\[
(f(x_1) - f(x_2)) \cdot (\eta(x_1) - \eta(x_2)) > 0, (x_1, x_2) \in \mathcal{C}_\eta
\]

where

\[
\mathcal{C}_\eta = \{ x : \eta(x) \in (0,1) \}.
\]

### B. Generalization Analysis

In this subsection, we explore how generalization error behaves away from the training error in terms of the TPAUC metric. In other words, we will show when a well-trained model will lead to a reasonable generalization performance. Our analysis is based on a standard assumption that the classifiers are chosen from a hypothesis class \( \mathcal{F} \) (e.g., the class of a specific type of deep neural network). To do this, we provide uniform upper bounds for the excess risk \( \mathcal{R}^\ell_{\alpha,\beta} - \mathcal{R}^\ell_{\psi} \). Such that

\[
\sup_{f \in \mathcal{F}} |\mathcal{R}^\ell_{\alpha,\beta} - \mathcal{R}^\ell_{\psi}| \leq \epsilon
\]

holds with high probability. Specifically, we present two results using two different assumptions.

**Assumption 1.** We focus on scoring functions satisfying either of the following assumptions:

a) The sufficient condition in Proposition 2 (a) holds, such that \( \mathcal{R}^\ell_{\alpha,\beta}(S, f) \leq \mathcal{R}^\ell_{\psi}(S, f) \).

b) The sufficient condition in Proposition 2 (c) holds, such that \( \mathcal{R}^\ell_{\alpha,\beta}(f) \leq \mathcal{R}^\ell_{\psi}(f) \).

Moreover, since we are dealing with imbalanced datasets, we adopt the following assumption without loss of generality.

**Assumption 2.** We assume that the training data satisfies:

\[
n_+ < n_-
\]

Note that the assumption above will be employed throughout Section V-B2; we will omit in the claim of the lemmas and theorems therein.
Finally, since only the order of the bounds is of interest, we will use the asymptotic notation \( \lesssim \) such that constants and undominated terms could be omitted from the expressions. Specifically, we have:

\[
f(x) \lesssim g(x) \iff \exists \text{ a constant } C \text{ s.t. } f(x) \leq C \cdot g(x).
\]

1) Results Under Assumption (a): When assumption (a) is available, we have:

\[
R_{\alpha, \beta}^\ell - \hat{R}_{\psi}^\ell \leq R_{\alpha, \beta}^{\ell} - \hat{R}_{\alpha, \beta}^\ell.
\]

This allows us to instead focus on the excess risk \( R_{\alpha, \beta}^{\ell} - \hat{R}_{\alpha, \beta}^\ell \).

The key challenge here is that \( \hat{R}_{\alpha, \beta}^\ell(S, f) \) is not an unbiased estimation of \( \text{AUC}_\alpha^\beta(f, S) \), making standard generalization analysis [45] unavailable. Here we extend the error decomposition technique for OPAUC [50] and employ the result in Proposition 2 to reach the following theorem. More precisely, we have the following decomposition lemma which allows one to decompose the upper bound of the excess risk into the sum of the excess risk of the threshold functions.

**Lemma 1 (Excess Risk Decomposition).** For \( \forall f \in \mathcal{F} \), we have:

\[
R_{\alpha, \beta}^{\text{AUC}}(f, S) - \hat{R}_{\alpha, \beta}^{\text{AUC}}(f, S) \leq 2(\Delta_+ + \Delta_-)
\]

where

\[
\Delta_+ = \sup_{\delta \in \mathbb{R}} \left| \frac{1}{n_+} \sum_{i=1}^{n_+} 1 \left[ f(x_i^+) \leq \delta \right] - \mathbb{E}_{x \sim p} \left[ 1 \left[ f(x^+) \leq \delta \right] \right] \right|
\]

\[
\Delta_- = \sup_{\delta \in \mathbb{R}} \left| \frac{1}{n_-} \sum_{j=1}^{n_-} 1 \left[ f(x_j^-) \geq \delta \right] - \mathbb{E}_{x \sim \mathcal{S}} \left[ 1 \left[ f(x^-) \geq \delta \right] \right] \right|
\]

**Theorem 4 (Excess Risk Bound).** Assume that there are no ties in the datasets, and the surrogate loss function \( \ell \) with range \([0, 1]\), is an upper bound of the 0-1 loss, then, for all \( f \in \mathcal{F} \), and all \( (\alpha, \beta) \in \mathcal{T}_{\text{surf}}(S) \), the following inequality holds with probability at least \( 1 - \delta \) over the choice of \( S \):

\[
R_{\alpha, \beta}^{\text{AUC}}(f, S) - \hat{R}_{\psi}^\ell(f, S) \lesssim \sqrt{\text{VC} \cdot \log(n_+) + \log(1/\delta)}
\]

\[
+ \sqrt{\text{VC} \cdot \log(n_-) + \log(1/\delta)},
\]

where \( \text{VC} \) is the VC dimension of the hypothesis class:

\[
\mathcal{T}(\mathcal{F}) \triangleq \{ \text{sign}(f(\cdot) - \delta) : f \in \mathcal{F}, \delta \in \mathbb{R} \}
\]

and

\[
\mathcal{T}_{\text{surf}}(S) = \{ (\alpha, \beta) : \alpha \in (0, 1), \beta \in (0, 1), n_+^\alpha, n_-^\beta \in \mathbb{N}_+, \beta \in \mathbb{N}_+ \text{, condition (a) in Proposition 2 holds} \}.
\]

According to the theorem, for all \( \alpha, \beta \) satisfying condition (a) of Proposition 2 and any model in \( \mathcal{F} \), the generalization error represented by the loss version of TPAUC \( R_{\alpha, \beta}^{\text{AUC}}(f, S) = 1 - \text{AUC}_\alpha^\beta(f, S) \) is no larger than the empirical loss \( \hat{R}_{\psi}^\ell(f, S) \) plus a complexity term. The complexity term is affected by two factors. On one hand, it vanishes with large enough training datasets. On the other hand, it remains moderate if the model hypothesis class’s VC dimension is not too large. Moreover, moderate upper bounds for the VC dimension are now available for typical models ranging from linear models to deep neural networks. Finally, for a well-trained model, the empirical loss \( \hat{R}_\psi(f, S) \) is restricted to be small in our framework; one can then reach reasonable generalization results with high probability.

2) Results Under Assumption (b): When assumption (b) is available, we have:

\[
R_{\alpha, \beta}^\ell - \hat{R}_{\psi}^\ell \leq R_{\alpha, \beta}^\ell - \hat{R}_{\alpha, \beta}^\ell.
\]

This allows us to instead focus on the excess risk \( R_{\alpha, \beta}^\ell - \hat{R}_{\alpha, \beta}^\ell \). Before elaborating on the details, we first provide the preliminaries.

Since \( \hat{R}_{\alpha, \beta}^\ell(f, S) \) is an unbiased estimation of \( R_{\alpha, \beta}^\ell \), we can now get rid of the error decomposition. Typically, the standard technique for deriving uniform upper bound of the excess risks requires the objective function to be a finite sum of independent terms. However, in our case, the terms in \( R_{\alpha, \beta}^\ell \) exert certain degrees of interdependency. For instance, the terms associated with \((x_1^+, x_1^-)\) and \((x_2^+, x_2^-)\) are interdependent as long as \( x_1^+ = x_2^+ \) or \( x_1^- = x_2^- \). To address this issue, previous work applies coloring based Rademacher complexity [3, 56], making the calculation of the (local) Rademacher complexity [32] much more complicated. To avoid this trick, we instead derive the following error decomposition lemma:

**Lemma 2.** The following error decomposition inequality holds:

\[
\sup_{f \in \mathcal{F}} \left[ \mathbb{E}_{x^+, x^-} \left[ g_f(x^+, x^-) \right] \right]
\]

\[
- \frac{1}{n_+ n_-} \sum_{i=1}^{n_+} \sum_{j=1}^{n_-} \frac{K^2}{(K - 1)^2} g_f(x_i^+, x_j^-)
\]

\[
\leq \sup_{f \in \mathcal{F}} \left[ \mathbb{E}_{x^+, x^-} \left[ g_f(x^+, x^-) \right] \right]
\]

\[
- \frac{K}{(K - 1)} \frac{1}{n_+} \sum_{i=1}^{n_+} \mathbb{E} g_f(x_i^+, x^-)
\]

\[
+ \sup_{f \in \mathcal{F}} \left[ \frac{K}{(K - 1)} \frac{1}{n_-} \sum_{j=1}^{n_-} \mathbb{E} g_f(x_i^+, x_j^-) \right]
\]

\[
- \frac{1}{n_+ n_-} \sum_{i=1}^{n_+} \sum_{j=1}^{n_-} \frac{K^2}{(K - 1)^2} g_f(x_i^+, x_j^-)
\]

where \( g_f(x^+, x^-) = \psi_x(1 - f(x^+))\psi_x(f(x^-))\ell(f(x^+) - f(x^-)) \).

With the lemma above, the generalization bound could be divided into two parts. Both of them could be expressed as a sum over i.i.d terms. This allows us to employ the standard local Rademacher complexity with proper extensions.
With all the preliminaries mentioned above, we first provide a generic result under Assumption 2 as the following theorem:

**Theorem 5.** Assume that the weighting function $\psi$ is $L_\infty$ Lipschitz continuous, the loss function $\ell$ is $L_\ell$ Lipschitz continuous with $||\psi||_\infty = \psi_\infty$, $||\ell||_\infty = \ell_\infty$, and the covering number of $\mathcal{F}$ has the following form with respect to $|| \cdot ||_\infty$ norm:

$$\log (\mathcal{N}(\mathcal{F}, \epsilon, || \cdot ||_\infty)) \leq \text{Alog}(R/\epsilon),$$

then for all $f \in \mathcal{F}$ and $(\alpha, \beta) \in \mathcal{T}_{\text{suff}}^2$, the following inequality holds with probability at least $1 - \delta$:

$$R^{\alpha, \beta}(f, S) - \frac{K^2}{(K - 1)^2} \hat{R}_\psi(f, S) \leq C \cdot \text{Alog}(\Gamma_{n_+} + \log(2/\delta)) + \frac{2K^2}{K - 1} \log(\Gamma_{n_+}) + \frac{4K^2}{K - 1} \log(\Gamma_{n_+}) \sum_{j=1}^{N_{\text{conv}}}(\text{op}(K^{(j)}) - \text{op}(\tilde{K}^{(j)}))_2 + \sum_{i=1}^{N_{\text{conv}}} ||V^{(i)} - \tilde{V}^{(i)}||_2.$$

where $C, \Gamma$ are universal constants depending on $\nu_\infty, \ell_\infty, L_\ell, L_\nu, K, R$, and

$$\mathcal{T}_{\text{suff}}^2(S) = \{(\alpha, \beta) : \alpha \in (0, 1), \beta \in (0, 1), n_+^\alpha \in \mathbb{N}, n_+^\beta \in \mathbb{N}, \text{condition (b) in Proposition 2 holds}\}.$$

We have the following crucial remarks on the theorem above:

**Remark 1.** It is easy to show that when $\psi(\cdot) \equiv 1$, TPAC objective function degenerates to a normal AUC risk function. Moreover, if we set the weight of the positive examples to $1$ and the weight of the negative examples to $0$, TPAC objective function degenerates to an OPAUC approximated risk function. In this sense, our result could be extended generally to the case of AUC and OPAUC optimization settings.

**Remark 2.** Note that our upper bound scales as $O\left(\text{Polylog}(n_+\cdot n_-)/n_+\right)$ with respect to the sample size, that is much sharper than the existing results such as [1], [57], [63], [64], if we degenerate our result to ordinary AUC. Taking a step further, if the dataset is roughly balanced in the sense that $n_+ = \theta(n_+ + n_-)$, we can even recover the result of $O(1/n)$ for U-statistics-based analysis [13] with much weaker assumptions.

Next, we present a concrete upper bound for deep convolutional neural networks. First, we introduce the setting of the deep neural networks employed in the forthcoming theoretical analysis, which is adopted from [42].

**Settings.** We focus on the deep neural networks with $N_{\text{conv}}$ fully-connected layers and $N_{\text{conv}}$ convolutional layers. The $i$-th convolutional layer has a kernel $K^{(i)} \in \mathbb{R}^{k_i \times k_i \times c_i \times c_{i-1}}$. Recall that convolution is a linear operator. For a given kernel $K$, we denote its associated matrix as $\text{op}(K)$, such that $K(x) = \text{op}(K)x$. Moreover, we assume that, each time, the convolution layer is followed by a componentwise non-linear activation function and an optional pooling operation. We assume that the activation functions and the pooling operations are all 1-Lipschitz. For the $i$-th fully-connected layer, we denote its weight as $V^{(i)}$. Above all, the complete parameter set of a given deep neural network could be represented as $P = \{K^{(1)}, \ldots, K^{(N_{\text{conv}})}, V^{(1)}, \ldots, V^{(N_{\text{conv}})}\}$. Again, we also assume that the loss function is $\phi_{\psi}$-Lipschitz and $\text{Range}(\ell) \subseteq [0, B]$. Finally, given two deep neural networks with parameters $P$ and $\tilde{P}$, we adopt a metric $d_{NN}(\cdot, \cdot)$ to measure their distance:

$$d_{NN}(P, \tilde{P}) = \sum_{i=1}^{N_{\text{conv}}} ||\text{op}(K^{(i)}) - \text{op}(\tilde{K}^{(i)})||_2 + \sum_{i=1}^{N_{\text{conv}}} ||V^{(i)} - \tilde{V}^{(i)}||_2.$$

**Constraints Over the Parameters.** First, we define $\mathcal{P}_{\nu}^{(0)}$ as the class for initialization of the parameters:

$$\mathcal{P}_{\nu}^{(0)} = \left\{ P : \left(\max_{i \in \{1, \ldots, N_{\text{conv}}\}} ||\text{op}(K^{(i)})||_2\right) \leq 1 + \nu \right\}.$$

Now we further assume that the learned parameters should be chosen from a class denoted by $\mathcal{P}_{\beta, \nu}$, where the distance between the learned parameter and the fixed initialization residing in $\mathcal{P}_{\nu}^{(0)}$ is no bigger than $d$:

$$\mathcal{P}_{\beta, \nu} = \left\{ P : d_{NN}(P, \tilde{P}_0) \leq d, \tilde{P}_0 \in \mathcal{P}_{\nu}^{(0)} \right\}.$$

Now we can apply Theorem 5 to obtain the following corollary based on the result in [42].

**Corollary 1.** Under the assumption of Theorem 5, assume that for all $x \in \mathcal{X}, ||x||_\nu \leq R X$. Define $\mathcal{F}_{d, \nu}$ as the hypothesis class of the scoring function induced by the above-mentioned CNN:

$$\mathcal{F}_{d, \nu} = \left\{ s_P : \mathbb{R}^{N_{\text{L}} - 1} \rightarrow [0, 1] | P \in \mathcal{P}_{\beta, \nu} \right\}.$$

For all $f \in \mathcal{F}_{d, \nu}$, and $(\alpha, \beta) \in \mathcal{T}_{\text{suff}}^2$, with the same assumption in Theorem 5, the following inequality holds with probability at least $1 - \delta$:

$$R^{\alpha, \beta}(f, S) \leq \hat{R}_\psi(f, S) + \frac{\text{Npar} \log(C L \cdot n_-) + \log(1/\delta)}{n_+}.$$
B. Dataset Description

Note that AUC is aimed at dealing with binary classification problems, hence we construct long-tail binary datasets as follows.

Binary CIFAR-10-LT Dataset. The original CIFAR-10 dataset consists of 60,000 $32 \times 32$ color images in 10 classes, with 6,000 images per class. There are 50,000 and 10,000 images in the training set and the test set, respectively. We create a long-tailed CIFAR-10 where the sample sizes across different classes decay exponentially, and the ratio of sample sizes of the least frequent to the most frequent class $\rho$ is set to 0.01. We then create binary long-tailed datasets based on CIFAR-10-LT by selecting one category as positive examples and the others as negative examples. We construct three binary subsets, in which the positive categories are 1) birds, 2) automobiles, and 3) cats, respectively. The datasets are split into training, validation and test sets with ratios 70\% : 15\% : 15\%, respectively. More details are provided in Table I.

Binary CIFAR-100-LT Dataset. The original CIFAR-100 dataset is similar to CIFAR-10, except it has 100 classes with each containing 600 images. The 100 classes in the CIFAR-100 are grouped into 20 superclasses. We create CIFAR-100-LT in the same way as CIFAR-10-LT, and transform it into three binary long-tailed datasets by selecting a superclass as positive class examples each time. Specifically, the positive superclasses are 1) fruits and vegetables, 2) insects and 3) large omnivores and herbivores, respectively. More details are provided in Table I.

Implementation Details on CIFAR Datasets. We utilize the ResNet-20 [25] as the backbone, which takes images with size $32 \times 32 \times 3$ as input and outputs 64-d features. Then the features are mapped into $224 \times 224 \times 3$ as input and outputs 512-d features.

C. Competitors

To validate the effectiveness of our proposed methods, we consider two types of competitors in our experiments. On one hand, we compare our proposed methods with other methods dealing with imbalanced data:

1) CE: Here use a class-wise reweighted version of the CE loss as one of our competitors, the sample weight is set to $1/n_y$, where $n_y$ the frequency of the class the sample belongs to.
2) Focal: [35] It tackles the imbalance problem by adding a modulating factor to the cross-entropy loss to highlight the hard and minority samples during the training process.
3) CB-CE: It refers to the loss function that applies the reweighting scheme proposed in [17] on the cross-entropy loss.
4) CB-Focal: It refers to the loss function that applies the reweighting scheme proposed in [17] on the Focal loss.

On the other hand, we also include standard AUC optimization methods as our baseline.

1) $\beta$AUC: Perform a standard AUC optimization with the surrogate loss function $\ell_{sq}(t) = (1 - t)^2$.

To show the effect of OPAUC optimizing on TPAUC, we implement the following algorithms:

1) Truncated-base method, which performs OPAUC optimization with the following objective function:

$$\sum_{i=1}^{n_+} \sum_{j=1}^{n^-} \frac{\ell_{sq}(f(x^+_i) - f(x^-_j))}{n_+ n^-}$$

where $x^+_i$ denotes the $i$-th positive instance. In each epoch, we first rank the negative instances among $X_N$ in ascending order based on their prediction scores. Then, the hard negative samples are generated from the top 1-th to $n^-$-th positions and all positive samples are involved in the optimization process.

### Table I

| Dataset         | Pos. Class ID | Pos. Class Name     | # Pos. Examples | # Neg. Examples |
|-----------------|---------------|---------------------|-----------------|-----------------|
| CIFAR-10-LT-1   | 2             | birds               | 2,155           | 12,729          |
| CIFAR-10-LT-2   | 1             | automobiles         | 3,596           | 11,288          |
| CIFAR-10-LT-3   | 3             | cats                | 1,292           | 13,592          |
| CIFAR-100-LT-1  | 6, 7, 14, 18, 24 | insects              | 2,756           | 18,938          |
| CIFAR-100-LT-2  | 0.5, 53, 77, 83 | fruits and vegetables | 1,266         | 20,428          |
| CIFAR-100-LT-3  | 15, 19, 21, 32, 38 | large omnivores and herbivores | 1,678 | 20,016 |
| Tiny-ImageNet-200-LT-1 | 24, 25, 26, 27, 28, 29 | dogs | 3,000 | 97,000 |
| Tiny-ImageNet-200-LT-2 | 11, 20, 21, 22 | birds | 2,000 | 98,000 |
| Tiny-ImageNet-200-LT-3 | 70, 81, 94, 107, 111, 116, 121, 133, 145, 153, 164, 166 | vehicles | 6,000 | 94,000 |
2) \textit{OPAUC-Poly}: Perform OPAUC optimization with the objective function:

\[
\frac{1}{n_+ n_-} \sum_{i=1}^{n_+} \sum_{j=1}^{n_-} \psi_{\text{poly}}(f(v_j^-)) \cdot \ell(f, x_i^+, x_j^-)
\]

3) \textit{OPAUC-Exp}: Perform TPAUC optimization with the objective function:

\[
\frac{1}{n_+ n_-} \sum_{i=1}^{n_+} \sum_{j=1}^{n_-} \psi_{\text{Exp}}(f(v_j^-)) \cdot \ell(f, x_i^+, x_j^-)
\]
10242 IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 45, NO. 8, AUGUST 2023

Fig. 6. Empirical validations for Proposition 2 (a).

Moreover, we include the truncation-based TPAUC method as a baseline of TPAUC optimization:

1) **TruncTPAUC**: We implement a Truncation-based TPAUC method, which performs TPAUC optimization with the following objective function:

\[
\frac{1}{n_+ n_-} \sum_{i=1}^{n_+} \sum_{j=1}^{n_-} \psi_{\text{poly}}(1 - f(x_i^+)) \cdot \psi_{\text{poly}}(f(v_j^-)) \cdot \ell(f, x_i^+, x_j^-)
\]

2) **TPAUC-Exp**: Perform TPAUC optimization with the objective function:

\[
\frac{1}{n_+ n_-} \sum_{i=1}^{n_+} \sum_{j=1}^{n_-} \psi_{\text{Exp}}(1 - f(x_i^+)) \cdot \psi_{\text{Exp}}(f(v_j^-)) \cdot \ell(f, x_i^+, x_j^-)
\]

3) **TPAUC-Minmax-Poly**: Perform the TPAUC-Poly algorithm with the minimax reformulation in Theorem 1.

4) **TPAUC-Minmax-Exp**: Perform the TPAUC-Exp algorithm with the minimax reformulation in Theorem 1.

D. General Implementation Details

All the experiments are carried out on a ubuntu 16.04.1 server equipped with Intel(R) Xeon(R) Gold 5218 CPU@2.30 GHz and an RTX 3090 GPU, and all codes are implemented with PyTorch (v-1.8.1) [54] under python 3.7 environment.

Fig. 7. Sensitivity analysis of Exp on γ. Experiments are conducted on CIFAR-10-LT. For each box, γ is fixed as the y-axis value, and the scattered points along the box show the variation of γ.

and the other settings follow from TruncOPAUC.
Fig. 8. Sensitivity analysis of $\gamma$ on $\gamma$. Experiments are conducted on CIFAR-10-LT. For each box, $(\gamma - 1)^{-1}$ is fixed as the y-axis value, and the scattered points along the box show the variation of $\gamma$.

Stochastic Gradient Descent (SGD) [61] with Nesterov momentum is adopted to optimize the objective function. Empirically, for all datasets, the learning rate is $10^{-3}$; the $l_2$ regularization term is set as $10^{-5}$, and the Nesterov momentum is 0.9. We also employ an exponential learning rate decay scheduler to adjust the learning rate after each training epoch, where the learning rate decay rate is set as 0.99 for all methods. The training batch size is 128, and we restrict the ratio of positive and negative samples by 1:10 in each batch. The batch size of validation/test examples is 256. Specifically, $E_k$ is searched in $\{0, 3, 5, 8, 10, 15, 20, 30, 35\}$. ForPoly, $\gamma$ is searched in $\{0.01, 0.03, 0.05, 0.08, 0.1, 1, 5\}$. ForExp, $\gamma$ is searched in $\{0.1, 1, 10, 20, 25, 30, 35\}$. Finally, we select the model based on the best validation performance and report the test set results.

E. Warm-Up Training Phase With Delay Epochs

Focusing on the hard examples at the beginning of the training process brings a high risk of over-fitting. It is thus necessary to focus on the entire dataset to capture the global information. Inspired by this investigation, we adopt a warm-up training strategy. Specifically, the model will go through a warm-up phase with $E_k$ Epochs of ordinary AUC optimization training. Afterward, we start the TPAUC training phase by optimizing our proposed surrogate problems. We will show its effect in the next subsection.

F. Validation of the Upper Bound Result in Proposition 2

In this subsection, we validate the sufficient condition in Proposition 2 (a). To do this, we plot the training curve for
the three subsets of CIFAR-10-LT under three TPAUC metrics with polynomial weighting. Here $R_{\alpha,\beta}^\phi$, $R_{\alpha,\beta}^k$, $R_{\alpha,\beta}^{N_{k-1}}$ represent the approximated, surrogate, and 0-1 empirical risk function over the training data. We can observe that our proposed risk $R_{\alpha,\beta}^\phi$ is an upper bound of the other two quantities in most cases. Even for the failure case Fig. 6(e), (f), we can also observe that the three curves at least exert a consistent decreasing trend. Hence the overall results show the effectiveness of our theoretical result.

G. Overall Performance

The performance comparisons for CIFAR-10-LT, CIFAR-100-LT and tiny-ImageNet-200-LT are shown in Tables II, III, and IV, respectively. We have the following observations from the results:

1) The best performance of our proposed methods consistently surpasses all the competitors significantly on all metrics. Moreover, the improvements are significant in most cases.

2) In most cases, the AUC-based algorithms often tend to outperform other algorithms. This shows that optimizing the AUC-like metric is more effective in improving the TPAUC metric. This observation could also be supported by Theorem 2. Theorem 2 states optimizing TPAUC is equivalent to partially generate a consistent ranking with $\eta(x)$. Meanwhile, optimizing AUC or OPAUC could also generate a consistent ranking $\eta(x)$ globally or in a different partial region from TPAUC. In this sense, optimizing AUC or OPAUC shares similar (but different) behavior to optimizing TPAUC.

3) Sometimes, we see that the OPAUC-based algorithms could not outperform the sqAUC, specifically on the CIFAR-100-LT dataset. This suggests that OPAUC and TPAUC tend to provide inconsistent comparison results, which could also be supported by the analysis in Section III-C.

4) Comparing different ways to implement the partial AUC optimization, we observe that the weighting-based algorithms, i.e., OPAUC-Poly, OPAUC-Exp, TPAUC-Poly, TPAUC-Exp, outperform the truncated version, i.e., TruncOPAUC, TruncTPAUC. This suggests that our proposed weighting function is more effective than sample manipulation.

5) For our algorithms, the ordinary and minimax implementations tend to show different performances. This is not surprising since the reformulated objective function has a different landscape from the original one. Hence when they have to be solved with different optimization algorithms, they often end up with different local optimal solutions practically, though their global optimal solutions are proved to be the same in Theorem 1.

H. Effect of $\gamma$

In this setting, we will use one same $\gamma$ for both the positive example weight $w_+$ and the negative weight $w_-$. In Figs. 7 and 8, we show the sensitivity in terms of $\gamma$ on CIFAR-10-LT for Exp and Poly, respectively. One can observe very different trends in these two methods. This is because that Exp and Poly have different characteristics in terms of the landscape of the weight function. As shown in Fig. 5(c), (d), the weight landscape of Exp is flat within a large subset of the domain. In this sense, it does not have a strong dependency on $\gamma$. As shown in Fig. 5(a), (b), the weight landscape of Poly is more sensitive toward $\gamma$. Moreover, the weighting function of Poly changes from a concave function $(\gamma - 1)^{-1} \leq 1)$ to a concave function $(\gamma - 1)^{-1} \geq 1$ with increasing $\gamma$. It leads to a clear trend with various $\gamma$. Moreover, we see that concave weight functions own significantly better performance. This validates our theoretical analysis in Proposition 2. In Appendix H, available online, we will present more analysis on the effect of $\gamma$.

VII. CONCLUSION

In this article, we initiate the study on TPAUC optimization. Since the original optimization problem could not be solved directly with an end-to-end framework, we propose a general framework to construct surrogate optimization problems for TPAUC. Following our dual correspondence theory, we can establish a surrogate problem once a calibrated penalty function or a calibrated weighting function is found. To see how and when our framework could provide efficient approximations of the original problem, we show that the surrogate objective function could reach the upper bound of the original one and that concave weighting functions are better choices than their convex counterparts. Moreover, we also provide high probability uniform upper bounds for the generalization error. The experiments on three datasets consistently show the advantage of our framework.

REFERENCES

[1] S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth, “Generalization bounds for the area under the ROC curve,” J. Mach. Learn. Res., vol. 6, pp. 393–425, 2005.
[2] A. H. Alan and B. Raskutti, “Optimising area under the ROC curve using gradient descent,” in Proc. Int. Conf. Mach. Learn., 2004, pp. 49–56.
[3] M.-R. Amini and N. Usunier, Learning With Partially Labeled and Interdependent Data. Berlin, Germany: Springer, 2015.
[4] Z. Bai, X. Zhang, and J. Chen, “Partial AUC optimization based on peak detector embeddings with class-center learning for text-independent speaker verification,” in Proc. Int. Conf. Acoust. Speech Signal Process., 2020, pp. 6819–6823.
[5] Z. Bai, X. Zhang, and J. Chen, “Speaker verification by partial AUC optimization with Mahalanobis distance metric learning,” IEEE/ACM Trans. Audio Speech Lang. Process., vol. 28, pp. 1533–1548, 2020.
[6] P. L. Bartlett, O. Bousquet, and S. Mendelson, “Local Rademacher complexities,” Ann. Statist., vol. 33, no. 4, pp. 1497–1537, 2005.
[7] P. L. Bartlett and S. Mendelson, “Rademacher and Gaussian complexities: Risk bounds and structural results,” J. Mach. Learn. Res., vol. 3, pp. 463–482, 2002.
[8] P. L. Bartlett, A. Montanari, and A. Rakhlin, “Deep learning: A statistical viewpoint,” 2021, arXiv:2103.09177.
[9] S. Boucheron, G. Lugosi, and P. Massart, Concentration Inequalities: A Nonasymptotic Theory of Independence, Oxford, U.K.: Oxford Univ. Press, 2013.
[10] K. W. Bowyer, C. Kranenburg, and S. Dougherty, “Edge detector evaluation using empirical ROC curves,” in Proc. Conf. Comput. Vis. Pattern Recognit., 1999, pp. 1354–1359.
[11] A. P. Bradley, “The use of the area under the ROC curve in the evaluation of machine learning algorithms,” Pattern Recognit., vol. 30, no. 7, pp. 1145–1159, 1997.
[12] T. Calders and S. Jaroszewicz, “Efficient AUC optimization for classification,” in Proc. Eur. Conf. Princ. Data Mining Knowl. Discov., 2007, pp. 42–53.
T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for
K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image
J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a
T. Lin, C. Jin, and M. Jordan, “On gradient descent ascent for nonconvex-
N. Li, R. Jin, and Z. Zhou, “Top rank optimization in linear time,” in
M. Ledoux and M. Talagrand, [38] T.-Y. Lin, C. Jin, and M. Jordan, “On gradient descent ascent for nonconvex-
C. Cortes and M. Mohri, “AUC optimization versus error rate minimization,” in Proc. Adv. Neural Inf. Process. Syst., 2003, pp. 313–320.
F. Cucker and S. Smale, “On the mathematical foundations of learning,” Bull. Amer. Math. Soc., vol. 39, pp. 1–49, 2002.
Y. Cui, M. Jia, Y.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9268–9277.
E. Eban, M. Schapire, A. Mackey, A. Gordon, R. Rifkin, and G. Eilidan, “Scalable learning of non-composable objectives,” in Proc. 20th Int. Conf. Artif. Intell. Statist., 2017, pp. 832–840.
T. Fawcett, “An introduction to ROC analysis,” Pattern Recognit. Lett., vol. 27, no. 8, pp. 861–874, 2006.
Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, “An efficient boo
ing algorithm for combining preferences,” J. Mach. Learn. Res., vol. 4, pp. 933–969, 2003.
W. Gao, L. Wang, R. Jin, S. Zhu, and Z. Zhou, “One-pass AUC optimization,” in Proc. Int. Conf. Mach. Learn., 2013, pp. 906–914.
D. J. Hand and R. J. Till, “A simple generalisation of the area under the ROC curve for multiple class classification problems,” Mach. Learn., vol. 45, no. 2, pp. 171–186, 2001.
J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (ROC) curve,” Radiology, vol. 143, no. 1, pp. 29–36, 1982.
H. Hao et al., “Open-narrow-synecchia anterior chamber angle classification in AS-OCT sequences,” 2020, arXiv:2006.05367.
K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015, arXiv:1512.03385.
T. Iwata, A. Fujino, and N. Ueda, “Semi-supervised learning for maximi
T. Joachims, “A support vector method for multivariate performance
T. Joachims, “Training linear SVMs in linear time,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2006, pp. 217–226.
P. Kar, H. Narasimhan, and P. Jain, “Online and stochastic gradient methods for non-composable loss functions,” in Proc. Adv. Neural Inf. Process. Syst., 2014.
A. Kumar, H. Narasimhan, and A. Cotter, “Implicit rate-constrained optimi
R. Liu, Z. Li, Y. Zhang, X. Fan, and Z. Luo, “Bi-level probabilistic feature learning for deformable image registration,” in Proc. Int. Joint Conf. Artif. Intell., 2020, pp. 723–730.
R. Liu, P. Mu, X. Yuan, S. Zeng, and J. Zhang, “A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 6305–6315.
W. Liu, W. Luo, D. Lian, and S. Gao, “Future frame prediction for anomaly detection - a new baseline,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 6536–6545.
YANG et al.: OPTIMIZING TWO-WAY PARTIAL AUC WITH AN END-TO-END FRAMEWORK 10245

Optimizing Two-Way Partial AUC with an End-to-End Framework

P. Long and H. Sedghi, “Generalization bounds for deep convolutional neural networks,” in Proc. Int. Conf. Learn. Representations, 2020.
A. K. Menon, X. J. Jiang, S. Vembu, C. Elkan, and L. Ohno-Machado, “Predicting accurate probabilities with a ranking loss,” in Proc. Int. Conf. Mach. Learn., 2012, Art. no. 703.
K. H. B. Metz C, “Statistical significance tests for binormal ROC curves,” J. Math. Psychol., vol. 22, no. 3, pp. 218–243, 1980.
M. Mohri, A. Storozhuk, and A. Talwalkar, Foundations of Machine Learning. Cambridge, MA, USA: MIT Press, 2018.
M. C. Mozer, R. H. Dodier, M. D. Colagrosso, C. Guerra-Salcedo, and R. H. Wolfe, “Priming the ROC curve: Constraining optimization of classifier performance,” in Proc. Adv. Neural Inf. Process. Syst., 2001, pp. 1409–1415.
H. Narasimhan, “Learning with complex loss functions and constraints,” in Proc. 21st Int. Conf. Artif. Intell. Statist., 2018, pp. 1646–1654.
H. Narasimhan and S. Agarwal, “A structural SVM based approach for optimizing partial AUC,” in Proc. Int. Conf. Mach. Learn., 2013, pp. 516–524.
H. Narasimhan and S. Agarwal, “SVMPauc-tight: A new support vec
tor method for optimizing partial AUC based on a tight convex upper bound,” in Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining, 2013, pp. 167–175.
H. Narasimhan and S. Agarwal, “Support vector algorithms for optimizing the partial area under the ROC curve,” Neural Computation, vol. 29, no. 7, pp. 1919–1963, 2017.
H. Narasimhan, A. Cotter, and M. R. Gupta, “Optimizing generalized rate metrics with three players,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 10747–10758.
M. Natale, Y. Ying, and S. Lyu, “Stochastic proximal algorithms for auc
maximization,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 3707–3716.
M. A. Natale, Y. Ying, and S. Lyu, “Stochastic AUC optimization algo
rithms with linear convergence,” Front. Appl. Math. Statist., vol. 5, 2019, Art. no. 30.
A. Paszke et al., “PyTorch: An imperative style, high-performance deep
learning library,” in Proc. 33rd Int. Conf. Neural Inf. Process. Syst., 2019, pp. 8024–8035.
W. Peterson, T. Birdsell, and W. Fox, “The theory of signal detectability,” Trans. IRE Professional Group Inf. Theory, vol. 4, no. 4, pp. 171–212, 1954.
L. Ralaivola and M.-R. Amini, “Entropy-based concentration inequali
ities for dependent variables,” in Proc. Int. Conf. Mach. Learn., 2015, pp. 2436–2444.
L. Ralaivola, M. Szafranski, and G. Stempfel, “Chromatic PAC-bayes bounds for non-iID data: Applications to ranking and stationary -mixing processes,” J. Mach. Learn. Res., vol. 11, pp. 1927–1956, 2010.
N. Rosenfeld, O. Meshi, D. Tarlow, and A. Globerson, “Learning struc
tured models with the AUC loss and its generalizations,” in Proc. 17th Int. Conf. Artif. Intell. Statist., 2014, pp. 841–849.
D. Sculley, “Combined regression and ranking,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2010, pp. 979–988.
A. J. Simpson and M.J. Fitter, “What is the best index of detectability?,” Psychol. Bull., vol. 80, no. 4, pp. 481–488, 1973.
I. Sutskever, J. Martens, G. E. Dahl, and G. H.inton, “On the importance of initialization and momentum in deep learning,” in Proc. Int. Conf. Mach. Learn., 2013, pp. 1139–1147.
K. Uematsu and Y. Lee, “Statistical optimality in multipartite ranking and ordinal regression,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 5, pp. 1080–1094, May 2015.
N. Usunier, M.-R. Amini, and P. Gallinari, “A data-dependent generalisation error bound for the AUC,” in Proc. ICML Workshop ROC Anal. Mach. Learn., 2019.
N. Usunier, M. R. Amini, and P. Gallinari, “Generalization error bounds for classifiers trained with interdependent data,” in Proc. Adv. Neural Inf. Process. Syst., 2006, pp. 1369–1376.
G. Valiant, “A theory of the learnable,” Commun. ACM, vol. 27, no. 11, pp. 1134–1142, 1984.
V. Vapnik, The Nature of Statistical Learning Theory, Berlin, Germany: Springer, 1995.
R. Vogel, A. Bellet, and S. Clémençon, “Learning fair scoring functions: Bipartite ranking under ROC-based fairness constraints,” in Proc. Int. Conf. Artif. Intell. Statist., 2021.
Zhiyong Yang received the MSc degree in computer science and technology from the University of Science and Technology Beijing (USTB), in 2017, and the PhD degree from the University of Chinese Academy of Sciences (UCAS), in 2021. He is currently a postdoctoral research fellow with the University of Chinese Academy of Sciences. His research interests include multimedia, image processing, and computer vision. He has authored or coauthored 32 academic papers in top-tier international conferences and journals including TPAMI/ICML/NeurIPS/CVPR. He served as a TPC member for IJCAI 2021 and a reviewer for several top-tier journals and conferences, such as IEEE Transactions on Pattern Analysis and Machine Intelligence, Transactions on Machine Learning Research, IJCAI, NeurIPS, and ICLR.

Qianqian Xu (Senior Member, IEEE) received the BS degree in computer science from the China University of Mining and Technology, in 2007, and the PhD degree in computer science from the University of Chinese Academy of Sciences, in 2013. She is currently an associate professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. Her research interests include statistical machine learning, with applications in multimedia and computer vision. She has authored or coauthored more than 50 academic papers in prestigious international journals and conferences (including IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Multimedia, TCST, ICML, NeurIPS, CVPR, AAAI, IJCAI, ACM Multimedia, and ICLR).

Shilong Bao received the BS degree from the College of Computer Science and Technology, Qingdao University, in 2019. He is currently working toward the PhD degree with the University of Chinese Academy of Sciences. His research interests include machine learning and data mining.

Yuan He received the BS and PhD degrees in computer science from Tsinghua University, China, in 2001 and 2007, respectively. He is a senior staff engineer with the Security Department, Alibaba Group, and working on artificial intelligence-based content moderation and intellectual property protection systems. Before joining Alibaba, he was a research manager with Fujitsu working on document analysis system. He has published more than 30 papers in computer vision and machine learning related conferences and journals including CVPR, ICCV, ICML, NeurIPS, AAAI, and ACMMM. His research interests include computer vision, machine learning, and AI security.

Xiaochun Cao (Senior Member, IEEE) received the BE and ME degrees in computer science from Beihang University (BUAA), China, in 1999 and 2002, respectively, and the PhD degree in computer science from the University of Central Florida, USA, in 2006, with his dissertation nominated for the university level Outstanding Dissertation Award. He is a professor with the School of Cyber Science and Technology, Shenzhen Campus, Sun Yat-sen University. After graduation, he spent about three years with ObjectVideo Inc. as a research scientist. From 2008 to 2012, he was a professor with Tianjin University. From 2012 to 2020, he was a professor with the Institute of Information Engineering, Chinese Academy of Sciences. He has authored and coauthored more than 200 journal and conference papers. In 2004 and 2010, he was the recipients of the Piero Zampieri best student paper award with the International Conference on Pattern Recognition. He is a fellow with the IET. He is an associate editor of IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology and IEEE Transactions on Multimedia.

Qingming Huang (Fellow, IEEE) received the bachelor’s degree in computer science from the Harbin Institute of Technology, China, in 1994, and the PhD degree in computer engineering from the Harbin Institute of Technology, China, in 1997. He is a chair professor with the University of Chinese Academy of Sciences and an adjunct research professor with the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include multimedia computing, image processing, computer vision, and pattern recognition. He has authored or coauthored more than 400 academic papers in prestigious international journals and top-level international conferences. He was the associate editor of IEEE Transactions on Circuits and Systems for Video Technology and Acta Automatica Sinica, and the reviewer of various international journals including IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, etc. He has served as general chair, program chair, track chair and TPC member for various conferences, including ACM Multimedia, CVPR, ICCV, ICME, ICPR, PCM, BigMM, PSIVT, etc.