DRAUC: An Instance-wise Distributionally Robust AUC Optimization Framework

Siran Dai[1,2] **Qianqian Xu**[3*] **Zhiyong Yang**[4]
Xiaochun Cao[5] **Qingming Huang**[4,3,6*]

[1] SKLOIS, Institute of Information Engineering, CAS
[2] School of Cyber Security, University of Chinese Academy of Sciences
[3] Key Lab. of Intelligent Information Processing, Institute of Computing Tech., CAS
[4] School of Computer Science and Tech., University of Chinese Academy of Sciences
[5] School of Cyber Science and Tech., Shenzhen Campus of Sun Yat-sen University
[6] BDKM, University of Chinese Academy of Sciences

daisiran@iie.ac.cn xuqianqian@ict.ac.cn
yangzhiyong21@ucas.ac.cn caoxiaochun@mail.sysu.edu.cn
qmhuang@ucas.ac.cn

Abstract

The Area Under the ROC Curve (AUC) is a widely employed metric in long-tailed classification scenarios. Nevertheless, most existing methods primarily assume that training and testing examples are drawn i.i.d. from the same distribution, which is often unachievable in practice. Distributionally Robust Optimization (DRO) enhances model performance by optimizing it for the local worst-case scenario, but directly integrating AUC optimization with DRO results in an intractable optimization problem. To tackle this challenge, methodically we propose an instance-wise surrogate loss of Distributionally Robust AUC (DRAUC) and build our optimization framework on top of it. Moreover, we highlight that conventional DRAUC may induce label bias, hence introducing distribution-aware DRAUC as a more suitable metric for robust AUC learning. Theoretically, we affirm that the generalization gap between the training loss and testing error diminishes if the training set is sufficiently large. Empirically, experiments on corrupted benchmark datasets demonstrate the effectiveness of our proposed method. Code is available at: https://github.com/EldercatSAM/DRAUC.

1 Introduction

The Area Under the ROC Curve (AUC) is an essential metric in machine learning. Owing to its interpretation equivalent to the probability of correctly ranking a random pair of positive and negative examples [11], AUC serves as a more suitable metric than accuracy for imbalanced classification problems. Research on AUC applications has expanded rapidly across various scenarios, including medical image classification [40, 51], abnormal behavior detection [5] and more.

However, current research on AUC optimization assumes that the training and testing sets share the same distribution [46], a challenging condition to satisfy when the testing environment presents a high degree of uncertainty. This situation is common in real-world applications.

Distributionally Robust Optimization (DRO) as a technique designed to handle distributional uncertainty, has emerged as a popular solution [38] in various applications, including machine learning [19], energy systems [1] and transportation [25]. This technique aims to develop a model that performs well, even under the most adversarial distribution within a specified distance from the original training distribution. However, existing DRO methods primarily focus on accuracy as a

[*] Corresponding authors.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

metric, making it difficult to directly apply current DRO approaches to AUC optimization due to its pairwise formulation. Consequently, it prompts the following question:

Can we optimize the Distributionally Robust AUC (DRAUC) using an end-to-end framework?

This task presents three progressive challenges: **1)**: The pairwise formulation of AUC necessitates simultaneous access to both positive and negative examples, which is computationally intensive and infeasible in online settings. **2)**: The naive integration of AUC optimization and DRO leads to an intractable solution. **3)**: Based on a specific observation, we find that the ordinary setting of DRAUC might lead to severe label bias in the adversarial dataset.

In this paper, we address the aforementioned challenges through the following techniques: For **1)**, we employ the minimax reformulation of AUC and present an early trail to explore DRO under the context of AUC optimization. For **2)**, we propose a tractable surrogate loss that is proved to be an upper bound of the original formulation, building our distribution-free DRAUC optimization framework atop it. For **3)**, we further devise distribution-aware DRAUC, to perform class-wise distributional perturbation. This decoupled formulation mitigates the label noise issue. This metric can be perceived as a class-wise variant of the distribution-free DRAUC.

It is worth noting that [56] also discusses the combination of DRO techniques with AUC optimization. However, the scope of their discussion greatly differs from this paper. Their approach focuses on using DRO to construct estimators for partial AUC and two-way partial AUC optimization with convergence guarantees, whereas this paper primarily aims to enhance the robustness of AUC optimization.

The main contributions of this paper include the following:

- **Methodologically**: We propose an approximate reformulation of DRAUC, constructing an instance-wise, distribution-free optimization framework based on it. Subsequently, we introduce the distribution-aware DRAUC, which serves as a more appropriate metric for long-tailed problems.
- **Theoretically**: We conduct a theoretical analysis of our framework and provide a generalization bound derived from the Rademacher complexity applied to our minimax formulation.
- **Empirically**: We assess the effectiveness of our proposed framework on multiple corrupted long-tailed benchmark datasets. The results demonstrate the superiority of our method.

2 Related Works

2.1 AUC Optimization

AUC is a widely-used performance metric. AUC optimization has garnered significant interest in recent years, and numerous research efforts have been devoted to the field. The researches include different formulations of objective functions, such as pairwise AUC optimization [8], instance-wise AUC optimization [49, 26, 50], AUC in the interested range (partial AUC [48], two-way partial AUC [47]), and area under different metrics (AUPRC [35, 44, 45], AUTKC [43], OpenAUC [42]. For more information, readers may refer to a review on AUC [46].

Some prior work investigates the robustness of AUC. For instance, [52] improves the robustness on noisy data and [15] studies the robustness under adversarial scenarios. In this paper, we further explore robustness under the local worst distribution.

2.2 Distributionally Robust Optimization

DRO aims to enhance the robustness and generalization of models by guaranteeing optimal performance even under the worst-case local distribution. To achieve this objective, an ambiguity set is defined as the worst-case scenario closest to the training set. A model is trained by minimizing the empirical risk on the ambiguity set. To quantify the distance between distributions, prior research primarily considers $\phi - divergence$ [2, 16, 4, 31] and the Wasserstein distance [39, 28, 19, 3, 7] as distance metrics. For more details, readers may refer to recent reviews on DRO [28, 23].

DRO has applications in various fields, including adversarial training [39], long-tailed learning [37], label shift [55], etc. However, directly optimizing the AUC on the ambiguity set remains an open problem.

3 Preliminaries

In this subsection, we provide a brief review of the AUC optimization techniques and DRO techniques employed in this paper. First, we introduce some essential notations used throughout the paper.

We use $z \in \mathcal{Z}$ to denote the example-label pair, and $f_{\boldsymbol{\theta}} : \mathcal{Z} \rightarrow [0, 1]$ to represent a model with parameters $\boldsymbol{\theta} \in \Theta$. This is typical when connecting a Sigmoid function after the model output. For datasets, $\widehat{P}$ denotes the nominal training distribution with n examples, while P represents the testing distribution. We use $\widehat{P}_+ = \{x_1^+, ..., x_{n^+}^+\}$ and $\widehat{P}_- = \{x_1^-, ..., x_{n^-}^-\}$ to denote positive/negative training set, respectively. To describe the degree of imbalance of the dataset, we define $\widehat{p} = \frac{n^+}{n^+ + n^-}$ as the imbalance ratio of training set, and $p = \Pr(y = 1)$ as the imbalance ratio of testing distribution. The notation $\mathbb{E}_P$ signifies the expectation on distribution P. We use $c(z, z') = ||z - z'||_2^2$ to denote the cost of perturbing example z to z'.

3.1 AUC Optimization

Statistically, AUC is equivalent to the Wilcoxon–Mann–Whitney test [11], representing the probability of a model predicting a higher score for positive examples than negative ones

$$AUC(f_{\boldsymbol{\theta}}) = \mathop{\mathbb{E}}_{P_+, P_-} \left[\ell_{0,1}(f_{\boldsymbol{\theta}}(\boldsymbol{x}^+) - f_{\boldsymbol{\theta}}(\boldsymbol{x}^-))\right] \tag{1}$$

where $\ell_{0,1}(\cdot)$ denotes the 0-1 loss, i.e., $\ell_{0,1}(x) = 1$ if $x < 0$ and otherwise $\ell_{0,1}(x) = 0$. Based on this formulation, maximizing AUC is equivalent to the following minimization problem

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{P_+, P_-} \left[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}^+) - f_{\boldsymbol{\theta}}(\boldsymbol{x}^-))\right] \tag{2}$$

where ℓ is a differentiable, consistent surrogate loss of $\ell_{0,1}$. However, the pairwise formulation of the above loss function is not applicable in an online setting. Fortunately, [49] demonstrates that using the square loss as a surrogate loss, the optimization problem (2) can be reformulated as presented in the following theorem.

Theorem 1 ([26]). *When using square loss as the surrogate loss, the AUC maximization is equivalent to*

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{P_+, P_-} \left[\ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}^+) - f_{\boldsymbol{\theta}}(\boldsymbol{x}^-)\right)\right] = \min_{\boldsymbol{\theta}, a, b} \max_{\alpha} \mathop{\mathbb{E}}_{P} \left[g(a, b, \alpha, \boldsymbol{\theta}, \boldsymbol{z})\right] \tag{3}$$

where

$$\begin{aligned} g(a, b, \alpha, \boldsymbol{\theta}, \boldsymbol{z}) = {} & (1 - p) \cdot (f_{\boldsymbol{\theta}}(\boldsymbol{x}) - a)^2 \cdot \mathbb{I}_{[y=1]} + p \cdot (f_{\boldsymbol{\theta}}(\boldsymbol{x}) - b)^2 \cdot \mathbb{I}_{[y=0]} \\ & + 2 \cdot (1 + \alpha) \cdot (p \cdot f_{\boldsymbol{\theta}}(\boldsymbol{x}) \cdot \mathbb{I}_{[y=0]} - (1 - p) \cdot f_{\boldsymbol{\theta}}(\boldsymbol{x}) \cdot \mathbb{I}_{[y=1]} - p(1 - p) \cdot \alpha^2). \end{aligned} \tag{4}$$

Moreover, with the parameter $\boldsymbol{\theta}$ fixed, the optimal solution of a, b, α, denoted as $a^\star, b^\star, \alpha^\star$, can be expressed as:

$$a^\star = \mathop{\mathbb{E}}_{P_+} \left[f_{\boldsymbol{\theta}}(\boldsymbol{x}^+)\right], b^\star = \mathop{\mathbb{E}}_{P_-} \left[f_{\boldsymbol{\theta}}(\boldsymbol{x}^-)\right], \alpha^\star = b^\star - a^\star. \tag{5}$$

Similar results hold if the true distribution P_+, P_- in the expressions are replaced with $\widehat{P}_+, \widehat{P}_-$.

Remark 1 (**The constraints on** a, b, α). *Given that the output of the model $f_{\boldsymbol{\theta}}$ is restricted to $[0, 1]$, a, b, α can be confined to the following domains:*

$$\begin{aligned} \Omega_{a,b} &= \{a, b \in \mathbb{R} | 0 \leq a, b, \leq 1\}, \\ \Omega_{\alpha} &= \{\alpha \in \mathbb{R} | -1 \leq \alpha \leq 1\}. \end{aligned} \tag{6}$$

So that the minimax problem can be reformulated as:

$$\min_{\boldsymbol{\theta}, (a,b) \in \Omega_{a,b}} \max_{\alpha \in \Omega_{\alpha}} \mathop{\mathbb{E}}_{P} \left[g(a, b, \alpha, \boldsymbol{\theta}, \boldsymbol{z})\right]. \tag{7}$$

3.2 Distributionally Robust Optimization

Distributionally Robust Optimization (DRO) aims to minimize the learning risk under the local worst-case distribution. Practically, since we can only observe empirical data points, our discussion is primarily focused on empirical distributions. Their extension to population-level is straightforward

$$\min_{\boldsymbol{\theta}} \sup_{\widehat{Q}:d(\widehat{Q},\widehat{P})\leq\epsilon} \mathop{\mathbb{E}}_{\widehat{Q}}[\ell(f_{\boldsymbol{\theta}},z)] \tag{8}$$

where $\widehat{P}$ is the original empirical distribution, $\widehat{Q}$ is the perturbed distribution and d is the metric of distributional distance. The constraint $d(\widehat{Q},\widehat{P})\leq\epsilon$ naturally expresses that the perturbation induced $\widehat{Q}$ should be small enough to be imperceptible.

As demonstrated in [7], when employing the Wasserstein distance $\mathcal{W}_c$ as the metric, a Lagrangian relaxation can be utilized to reformulate DRO into the subsequent minimax problem.

Theorem 2 ([7]). *With $\phi_\lambda(z,\boldsymbol{\theta})=\sup_{z'\in\mathcal{Z}}\{\ell(f_{\boldsymbol{\theta}},z')-\lambda c(z,z')\}$, for all distribution $\widehat{P}$ and $\epsilon>0$, we have*

$$\sup_{\widehat{Q}:\mathcal{W}_c(\widehat{Q},\widehat{P})\leq\epsilon} \mathop{\mathbb{E}}_{\widehat{Q}}[\ell(f(z))] = \inf_{\lambda\geq 0}\{\lambda\epsilon + \mathop{\mathbb{E}}_{\widehat{P}}[\phi_\lambda(z,\boldsymbol{\theta})]\}. \tag{9}$$

With the theorem above, one can directly get rid of the annoying Wasserstein constraint in the optimization algorithms. We will use this technique to derive an AUC-oriented DRO framework in this paper.

4 Method

4.1 Warm Up: A Naive Formulation for DRAUC

As a technical warm up, we first start with a straightforward approach to optimize AUC metric directly under the worst-case distribution. By simply incorporating the concept of the Wasserstein ambiguity set, we obtain the following definition of DRAUC in a pairwise style.

Definition 1 (**Pairwise Formulation of DRAUC**). *Let ℓ be a consistent loss of $\ell_{0,1}$, for any nominal distribution $\widehat{P}$ and $\epsilon>0$, we have*

$$DRAUC_\epsilon(f_{\boldsymbol{\theta}},\widehat{P}) = 1 - \max_{\widehat{Q}:\mathcal{W}_c(\widehat{Q},\widehat{P})\leq\epsilon} \mathop{\mathbb{E}}_{\widehat{Q}}\left[\ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}^+)-f_{\boldsymbol{\theta}}(\boldsymbol{x}^-)\right)\right]. \tag{10}$$

However, generating local-worst Wasserstein distribution $\widehat{Q}$ is loss-dependent, implying that we need to know all the training details to deliver a malicious attack. In our endeavor to secure a performance guarantee for our model, we cannot limit the scope of information accessible to an attacker. This pairwise formulation elevates the computational complexity from $O(n)$ to $O(n^+n^-)$, significantly increasing the computational burden. By a simple reuse of the trick in (7), one can immediately reach the following reformulation of the minimization of (10).

Proposition 1 (**A Naive Reformulation**). *When using square loss as the surrogate loss, The DRAUC minimization problem: $\min_{\boldsymbol{\theta}} DRAUC_\epsilon(f_{\boldsymbol{\theta}},\widehat{P})$, is equivalent to*

$$\textit{\textbf{(Ori)}}\quad \min_{\boldsymbol{\theta}} \max_{\widehat{Q}:\mathcal{W}_c(\widehat{Q},\widehat{P})\leq\epsilon} \min_{(a,b)\in\Omega_{a,b}} \max_{\alpha\in\Omega_\alpha} \mathop{\mathbb{E}}_{\widehat{Q}}\left[g(a,b,\alpha,\boldsymbol{\theta},z_i)\right]. \tag{11}$$

Unfortunately, the optimization operators adhere to a min-max-min-max fashion. There is no known optimization algorithm can deal with this kind of problems so far. Hence, in the rest of this section, we will present two tractable formulations as proper approximations of the problem.

4.2 DRAUC-Df: Distribution-free DRAUC

Let us take a closer look at the minimax problem $(\boldsymbol{Ori})$. It is straightforward to verify that, fix all the other variables, g is convex with respect to a,b and concave with respect to α within $\Omega_{a,b},\Omega_\alpha$. We

Algorithm 1 Algorithm for optimizing DRAUC-Df:

```
**Input:** the training data $\mathcal{Z}$, step number $K$, step size for inner $K$-step gradient ascent $\eta_z$, learning rates $\eta_\lambda, \eta_w, \eta_\alpha$ and maximal corrupt distance $\epsilon$.
**Initialize:** initialize $a^0, b^0, \alpha^0 = 0, \lambda^0 = \lambda_0$.
**for** $t = 1$ **to** $T$ **do**
    **Sample a batch of example** $z$ **from** $\mathcal{Z}$.
    **Generate Local Worst-Case Examples**:
    Initialize $z' = z$.
    **for** $k = 1$ **to** $K$ **do**
        $z' = \Pi_{\mathcal{Z}}(z' + \eta_z \cdot \nabla_z \phi_{\lambda^t,a,b,\alpha}(\boldsymbol{\theta}, z'))$.
    **end for**
    **Update Parameters**:
    Update $\alpha^{t+1} = \Pi_{\Omega_\alpha}(\alpha^t + \eta_\alpha \cdot \nabla_\alpha g^t(z'))$.
    Update $\lambda^{t+1} = \Pi_{\Omega_\lambda}(\lambda^t - \eta_l \cdot \nabla_\lambda[\lambda\epsilon + \phi_{\lambda^t,a,b,\alpha}(\boldsymbol{\theta}, z')])$.
    Update $\mathbf{w}^{t+1} = \Pi_{\Omega_\mathbf{w}}(\mathbf{w}^t - \eta_\mathbf{w} \cdot \nabla_\mathbf{w} g^t(z'))$.
**end for**
```

are able to interchange the inner $\min_{(a,b)\in\Omega_{a,b}}$ and $\max_{\alpha\in\Omega_\alpha}$ by invoking von Neumann's Minimax theorem [41], which results in

$$\min_{\boldsymbol{\theta}} \max_{\widehat{Q}:\mathcal{W}_c(\widehat{Q},\widehat{P})\le\epsilon} \max_{\alpha\in\Omega_\alpha} \min_{(a,b)\in\Omega_{a,b}} \mathbb{E}_{\widehat{Q}}[g(a,b,\alpha,\boldsymbol{\theta},z)]. \tag{12}$$

Moreover, based on the simple property that $\max_x \min_y f(x,y) \le \min_y \max_x f(x,y)$, we reach an upper bound of the objective function:

$$\underbrace{\max_{\widehat{Q}:\mathcal{W}_c(\widehat{Q},\widehat{P})\le\epsilon} \max_{\alpha\in\Omega_\alpha} \min_{(a,b)\in\Omega_{a,b}} \mathbb{E}_{\widehat{Q}}[g(a,b,\alpha,\theta,z)]}_{DRAUC_\epsilon(f_{\boldsymbol{\theta}},\widehat{P})} \le \underbrace{\min_{(a,b)\in\Omega_{a,b}} \max_{\alpha\in\Omega_\alpha} \max_{\widehat{Q}:\mathcal{W}_c(\widehat{Q},\widehat{P})\le\epsilon} \mathbb{E}_{\widehat{Q}}[g(a,b,\alpha,\theta,z)]}_{\widetilde{DRAUC}_\epsilon(f_{\boldsymbol{\theta}},\widehat{P})} \tag{13}$$

From this perspective, if we minimize $\widetilde{DRAUC}_\epsilon(f_{\boldsymbol{\theta}},\widehat{P})$ in turn, we can at least minimize an **upper bound** of $DRAUC_\epsilon(f_{\boldsymbol{\theta}},\widehat{P})$. In light of this, we will employ the following optimization problem as a surrogate for **(Ori)**:

$$(\boldsymbol{Df}) \quad \min_{\mathbf{w}} \max_{\alpha\in\Omega_\alpha} \max_{\widehat{Q}:\mathcal{W}_c(\widehat{Q},\widehat{P})\le\epsilon} \mathbb{E}_{\widehat{Q}}[g(\mathbf{w},\alpha,z)] \tag{14}$$

where $\mathbf{w} = \boldsymbol{\theta}, (a,b) \in \Omega_{a,b}$. Now, by applying the strong duality to the inner maximization problem

$$\max_{\widehat{Q}:\mathcal{W}_c(\widehat{Q},\widehat{P})\le\epsilon} \mathbb{E}_{\widehat{Q}}[g(\mathbf{w},\alpha,z)]$$

we have

$$(\boldsymbol{Df}) \min_{\mathbf{w}} \max_{\alpha\in\Omega_\alpha} \min_{\lambda\ge 0} \{\lambda\epsilon + \mathbb{E}_{\widehat{P}}[\phi_{\mathbf{w},\lambda,\alpha}(z)]\} \tag{15}$$

where $\phi_{\mathbf{w},\lambda,\alpha}(z) = \max_{z'\in\mathcal{Z}}[g(\mathbf{w},\alpha,z) - \lambda c(z,z')]$. This min-max-min formulation remains difficult to optimize, so we take a step similar to (13) that interchange the inner $\min_{\lambda\ge 0}$ and outer $\max_{\alpha\in\Omega_\alpha}$, resulting in a tractable **upper bound**

$$(\boldsymbol{Df\star}) \min_{\mathbf{w}} \min_{\lambda\ge 0} \max_{\alpha\in\Omega_\alpha} \{\lambda\epsilon + \mathbb{E}_{\widehat{P}}[\phi_{\mathbf{w},\lambda,\alpha}(z)]\}. \tag{16}$$

In this sense, we will use the $(\boldsymbol{Df\star})$ as the final optimization problem for **DRAUC-Df**.

4.3 DRAUC-Da: Distribution-aware DRAUC

Though AUC itself is inherently robust toward long-tailed distributions, we also need to examine whether DRAUC shares this resilience. We now present an analysis within a simplified feature space on the real line, where positive and negative examples are collapsed to their corresponding clusters. The choice of the feature space is simple yet reasonable since it is a 1-d special case of the well-accepted neural collapse phenomenon [32, 10, 17, 57, 27].

Specifically, the following proposition states that the distributional attacker in DRAUC can ruin the AUC performance easily by merely attacking the tail-class examples.

Proposition 2 (**Powerful and Small-Cost Attack on Neural Collapse Feature Space**). *Let the training set comprises n^+ positive examples and n^- negative examples in $\mathbb{R}^1$, i.e., $\mathcal{D} = \left\{x_1^+, ..., x_{n^+}^+, x_{n^++1}^-, ..., x_n^-\right\}$, with the empirical distribution $\widehat{P} = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ (δ_z represents the Dirac point mass at point z.). According to the neural collapse assume, we have: $x_i^+ = x^+$, $x_j^- = x^-$. Given a classifier $f(x) = x$, we assume that the maximization of perturb distribution $\widehat{Q}$ is further constrained on the subset:*

$$\mathcal{Q} = \left\{\widehat{Q} : \widehat{Q} = \frac{1}{n}\sum_{i=1}^n \delta_{x_i'}\right\}$$

where $x_i \to x_i'$ forms a discrete Monge map. Then, we have:

$$\inf_{\widehat{Q}\in\mathcal{Q}, AUC(f,\widehat{Q})=0} \mathcal{W}_c(\widehat{P}, \widehat{Q}) \leq \widehat{p}\cdot(1-\widehat{p})\cdot(x^+ - x^-)^2$$

where $\widehat{p} = \frac{n^+}{n}$ is the ratio of the positive examples in the dataset. Moreover, the cost $\widehat{p}\cdot(1-\widehat{p})\cdot(x^+ - x^-)^2$ is realized by setting:

$$x^{+\prime} = x^{-\prime} = \widehat{p}\cdot x^+ + (1-\widehat{p})\cdot x^-$$

the barycenter of the two-bodies system (x^+, x^-).

It is noteworthy that $\widehat{p}\cdot(1-\widehat{p})$ reflects the degree-of-imbalanceness, which is relatively small for long-tailed datasets. Moreover, the barycenter tends to be pretty close to the head-class examples. Therefore, only the tail-class examples are required to be revised heavily during the attack. In this sense, the attacker can always exploit the tail class examples as a backdoor to ruin the AUC performance with small Wasserstein cost. This is similar to the overly-pessimistic phenomena [6, 16] in DRO. The following example shows how small such cost could be in a numerical sense.

Example 1. *Consider a simplified setting in which the training set is comprised of only one positive example and 99 negative examples, i.e., $\widehat{P} = \{x_1^+, x_2^-, ..., x_{100}^-\}$ with $x^+ = 0.99$ and $x^- = 0.01$. The minimum distance required to perturb the AUC metric from 1 to 0 is 0.009702. This result is achieved by perturbing the positive example from 0.99 to 0.0198 and the negative examples from 0.01 to 0.0198, respectively.*

This perturbation strategy indicates a preference towards strong attack on tail-class examples. The resulting distribution $\widehat{Q}$ is always highly biased toward the original distribution, despite the small Wasserstein cost. In the subsequent training process, one has to minimize the expected loss over $\widehat{Q}$, resulting to label noises.

Therefore, it is natural to consider perturbations on the positive and negative distributions separately to avoid such a problem. Accordingly, we propose here a distribution-aware DRAUC formulation:

Definition 2 (**Distribution-aware DRAUC**). *Let ℓ be a consistent loss of $\ell_{0,1}$, for any nominal distribution $\widehat{P}$ and $\epsilon_+, \epsilon_- > 0$, we have*

$$DRAUC_{\epsilon_+,\epsilon_-}^{Da}(f_{\boldsymbol{\theta}}, \widehat{P}) = 1 - \max_{\substack{\widehat{Q}_+:\mathcal{W}_c(\widehat{Q}_+,\widehat{P}_+)\leq\epsilon_+ \\ \widehat{Q}_-:\mathcal{W}_c(\widehat{Q}_-,\widehat{P}_-)\leq\epsilon_-}} \mathop{\mathbb{E}}_{\widehat{Q}_+,\widehat{Q}_-}\left[\ell(f_{\boldsymbol{\theta}}(x_i^+) - f_{\boldsymbol{\theta}}(x_j^-))\right]. \quad (17)$$

For simplicity, let us denote

$$\widehat{\mathcal{Q}} = \{\widehat{Q}|\ \mathcal{W}_c(\widehat{Q}_+, \widehat{P}_+) \leq \epsilon_+, \mathcal{W}_c(\widehat{Q}_-, \widehat{P}_-) \leq \epsilon_-\} \quad (18)$$

Similar to **DRAUC-Df**, we construct our reformulation as follows:

$$\textbf{(Da)} \quad \min_{\mathbf{w}} \max_{\alpha\in\Omega_\alpha} \max_{\widehat{Q}\in\widehat{\mathcal{Q}}} \mathop{\mathbb{E}}_{\widehat{Q}_+,\widehat{Q}_-}[g(a, b, \alpha, \boldsymbol{\theta}, z_i)]. \quad (19)$$

Moreover, we conduct a similar derivation as **DRAUC-Df**, to construct a tractable upper bound:

$$\boldsymbol{(Da\star)} \min_{\mathbf{w}} \min_{\lambda_+,\lambda_-\geq 0} \max_{\alpha\in\Omega_\alpha} \{\lambda_+\epsilon_+ + \lambda_-\epsilon_- + \widehat{p} \mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] + (1-\widehat{p}) \mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\} \quad (20)$$

where $\phi_{\mathbf{w},\lambda_+,\alpha}(z) = \max_{z'\in\mathcal{Z}}[g(\mathbf{w}, \alpha, z) - \lambda_+ c(z, z')]$ and $\phi_{\mathbf{w},\lambda_-,\alpha}(z) = \max_{z'\in\mathcal{Z}}[g(\mathbf{w}, \alpha, z) - \lambda_- c(z, z')]$. Please see Appendix A for the details.

Algorithm 2 Algorithm for optimizing DRAUC-Da:

```
Input: the training data $\mathcal{Z}$, step number $K$, step size for inner $K$-step gradient ascent $\eta_z$, learning rates $\eta_\lambda, \eta_w, \eta_\alpha$ and maximal corrupt distance $\epsilon_+, \epsilon_-$.
Initialize: initialize $a^0, b^0, \alpha^0 = 0, \lambda_+^0 = \lambda_-^0 = \lambda_0$.
for $t = 1$ to $T$ do
    Sample a batch of example $z$ from $\mathcal{Z}$.
    Generate Local Worst-Case Examples:
    Initialize $z'_+ = z_+, z'_- = z_-$.
    for $k = 1$ to $K$ do
        $z'_+ = \Pi_{\mathcal{Z}}(z'_+ + \eta_z \cdot \nabla_z \phi_{\lambda_+^t, a, b, \alpha}(\boldsymbol{\theta}, z'_+))$.
        $z'_- = \Pi_{\mathcal{Z}}(z'_- + \eta_z \cdot \nabla_z \phi_{\lambda_-^t, a, b, \alpha}(\boldsymbol{\theta}, z'_-))$.
    end for
    Update Parameters:
    Update $\alpha^{t+1} = \Pi_{\Omega_\alpha}(\alpha^t + \eta_\alpha \cdot (p \nabla_a g^t(z'_+) + (1-p) \nabla_a g^t(z'_-)))$.
    Update $\lambda_+^{t+1} = \Pi_{\Omega_\lambda}(\lambda_+^t - \eta_l \cdot \nabla_{\lambda_+}[\lambda_+ \epsilon_+ + \phi_{\lambda_+^t, a, b, \alpha}(\boldsymbol{\theta}, z'_+)])$.
    Update $\lambda_-^{t+1} = \Pi_{\Omega_\lambda}(\lambda_-^t - \eta_l \cdot \nabla_{\lambda_-}[\lambda_- \epsilon_- + \phi_{\lambda_-^t, a, b, \alpha}(\boldsymbol{\theta}, z'_-)])$.
    Update $\mathbf{w}^{t+1} = \Pi_{\Omega_{\mathbf{w}}}(\mathbf{w}^t - \eta_{\mathbf{w}} \cdot (\widehat{p} \nabla_{\mathbf{w}} g^t(z'_+) + (1-\widehat{p}) \nabla_{\mathbf{w}} g^t(z'_-)))$.
end for
```

4.4 Algorithm

4.4.1 DRAUC Optimization

Motivated by the above reformulation, we propose our DRAUC optimization framework, where we solve this optimization problem alternatively.

Inner maximization problem : K-step Gradient Ascent: Following [39], we consider accessing K-step gradient ascent with learning rate η_z to solve the inner maximization problem, which is widely used in DRO and can be considered as a variance of PGM. For α, we use SGA with a step size η_α.

Outer minimization problem: Stochastic Gradient Descent: On each iteration, we apply stochastic gradient descent over w with learning rate η_w and over λ with learning rate η_λ.

See Algorithms 1,2 for more details.

4.5 Generalization Bounds

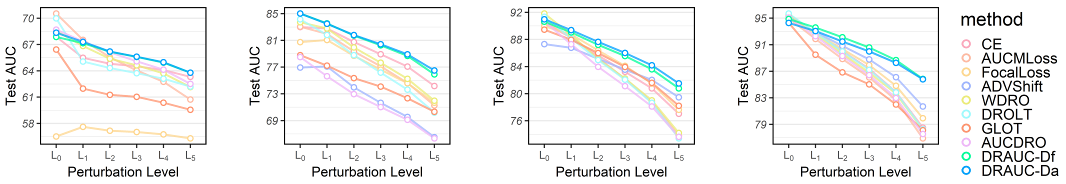
In this section, we theoretically show that the proposed algorithm demonstrates robust generalization in terms of DRAUC-Da metric, even under local worst-case distributions. That is, we show that a model sufficiently trained under our approximate optimization $(\boldsymbol{Da\star})$ enjoys a reasonable performance guarantee in DRAUC-Da metric. Our analysis based on the standard assumption that the model parameters $\boldsymbol{\theta}$ are chosen from the hypothesis set Θ(such as neural networks of a specific structure). To derive the subsequent theorem, we utilize the results analyzed in Section 4.3 and perform a Rademacher complexity analysis of DRAUC-Da. The proof for DRAUC-Df follows a similar proof and is much simpler, thus we omit the result here. For additional details, please refer to Appendix A.

Theorem 3 (Informal Version). *For all $\boldsymbol{\theta} \in \Theta, \lambda_+, \lambda_- \geq 0, (a,b) \in \Omega_{a,b}, \alpha \in \Omega_\alpha$ and $\epsilon_+, \epsilon_- > 0$, the following inequality holds with a high probability*

$$\underbrace{DRAUC^{Da}_{\epsilon_+,\epsilon_-}(f_{\boldsymbol{\theta}}, P)}_{(\boldsymbol{a})} \leq \underbrace{\widehat{\mathcal{L}}}_{(\boldsymbol{b})} + \underbrace{\mathcal{O}(\sqrt{1/\tilde{n}})}_{(\boldsymbol{c})} \tag{21}$$

where $\tilde{n}$ is some normalized sample size and $\widehat{\mathcal{L}} = \min_{\mathbf{w}} \min_{\lambda_+,\lambda_- \geq 0} \max_{\alpha \in \Omega_\alpha} \{\lambda_+ \epsilon_+ + \lambda_- \epsilon_- + \widehat{p}\, \mathbb{E}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] + (1-\widehat{p})\, \mathbb{E}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\}$.

In Thm.3, $(\boldsymbol{a})$ represents the robust AUC loss in terms of expectation, $(\boldsymbol{b})$ denotes the training loss that we use to optimize our model parameters, and $(\boldsymbol{c})$ is an error term that turns to zero when the



(a) Imratio **0.01** (b) Imratio **0.05** (c) Imratio **0.1** (d) Imratio **0.2**

Figure 1: Overall Performance of ResNet32 Across Perturbation Levels on CIFAR10. This graph illustrates the performance of various methods at different corruption levels, with Level 0 indicating no corruption and Level 5 representing the most severe corruption. In each figure, the seven lines depict the test AUC for CE, AUCMLoss, FocalLoss, ADVShift, WDRO, DROLT, GLOT, AUCDRO, DRAUC-Da and DRAUC-Df, respectively. Best viewed in colors.

sample size turns to infinity. In this sense, if we train our model sufficiently within a large enough training set, we can achieve a minimal generalization error.

5 Experiments

In this section, we demonstrate the effectiveness of our proposed framework on three benchmark datasets with varying imbalance ratios.

5.1 Experiment Settings

We evaluate our framework using the following approach. First, we conduct a binary, long-tailed training set. Then, we proceed to train the model on the long-tailed training set with varying imbalance ratios, tune hyperparameters on the validation set, and evaluate the model exhibiting the highest validation AUC on the corrupted testing set. For instance, we train our model on binary long-tailed MNIST [22], CIFAR10, CIFAR100 [18], and Tiny-ImageNet [21], and evaluate our proposed method on the corrupted version of corresponding datasets [30, 13, 14]. Furthermore, we compare our results with multiple competitors including the baseline (CE), typical methods for long-tailed problems [24, 52, 56] and DRO methods [55, 20, 37, 34]. Please see Appendix B for more details.

5.2 Results and Analysis

5.2.1 Overall Performance

The overall performances on CIFAR10 and Tiny-ImageNet are presented in Table 1 and Table 2, respectively. We further compare model performances by altering the level of perturbation, with results displayed in Figure 1. Due to the space limitation, we attach results on MNIST and CIFAR100 in Appendix B. Based on these findings, we make the following observations:

Effectiveness. Our proposed method outperforms all competing approaches across Corrupted MNIST, CIFAR10, CIFAR100 and Tiny-ImageNet datasets for all imbalance ratios, thereby substantiating its effectiveness. Additionally, our approach exhibits enhanced performance as the level of perturbation intensifies, indicating its robustness in challenging testing scenarios.

Ablation results. Given that our method is modified on AUCMLoss [52], the results presented in Figure 1 can be treated as ablation results. Under the same hyperparameters of AUCMLoss, our method exhibits significant improvement over the baseline, indicating enhanced model robustness.

Advantage of Distribution-awareness. As presented in Table 1, DRAUC-Da attains higher scores than DRAUC-Df across almost all corrupted scenarios. This supports our hypothesis that a strong attack on tail-class examples can potentially compromise model robustness.

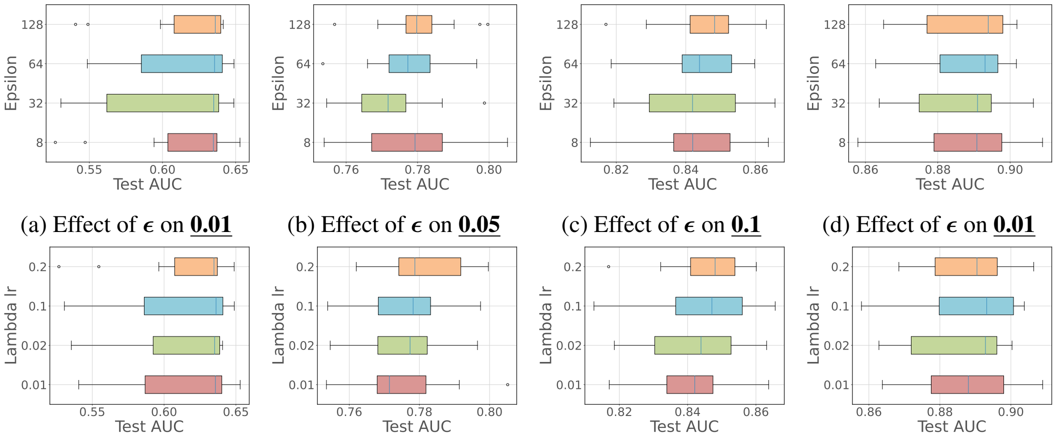
Performances on non-corrupted data. Within non-corrupted datasets, our approach continues to exhibit competitive performance under conditions of extreme data imbalance, specifically when the imbalance ratio equals to 0.01. However, with less imbalanced training data, our method may suffer performance degradation, attributable to the potential trade-off between model robustness and clean performance, which is an unavoidable phenomenon in Adversarial Training [54].

Table 1: Overall Performance on CIFAR10-C and CIFAR10-LT with different imbalance ratios and different models. The highest score on each column is shown with **bold**, and we use darker color to represent higher performance.

Model	Methods	CIFAR10-C				CIFAR10-LT			
		0.01	0.05	0.10	0.20	0.01	0.05	0.10	0.20
ResNet20	CE	62.48	75.87	83.13	86.20	65.43	84.12	**92.32**	**95.68**
	AUCMLoss	63.93	76.77	81.75	85.26	**68.88**	84.74	90.97	94.40
	FocalLoss	56.56	74.44	81.81	84.97	57.63	81.62	91.33	94.62
	ADVShift	61.36	75.97	83.78	87.35	64.97	82.91	87.87	95.46
	WDRO	63.19	78.90	80.59	86.02	68.80	**88.54**	91.04	94.04
	DROLT	59.92	77.51	81.09	86.46	60.99	85.76	91.35	95.17
	GLOT	63.98	77.19	83.33	87.57	65.95	88.37	90.51	94.62
	AUCDRO	63.35	76.19	81.82	85.96	67.14	84.00	90.92	94.88
	DRAUC-Df	65.58	**80.18**	85.71	88.83	68.12	86.47	90.57	94.17
	DRAUC-Da	**66.06**	80.13	**85.91**	**89.51**	68.71	84.43	90.30	93.76
ResNet32	CE	64.43	78.79	83.12	86.89	66.05	84.40	90.44	**95.61**
	AUCMLoss	64.00	76.98	81.87	85.66	**68.90**	84.94	**91.52**	95.16
	FocalLoss	56.96	76.53	83.82	87.42	58.04	82.99	91.02	95.16
	ADVShift	55.74	72.42	83.47	88.32	56.73	79.36	87.88	94.95
	WDRO	64.51	78.45	83.87	88.03	68.16	**86.48**	90.11	95.23
	DROLT	63.66	76.71	83.93	88.42	65.40	84.68	90.11	95.51
	GLOT	62.59	77.21	83.67	87.30	64.53	82.62	89.59	94.62
	AUCDRO	65.10	71.23	81.45	86.23	68.69	78.51	90.67	95.07
	DRAUC-Df	65.44	80.27	85.70	**90.62**	67.11	85.03	90.63	94.86
	DRAUC-Da	**65.50**	**80.57**	**86.25**	90.15	68.51	85.03	90.98	94.27

Table 2: Overall Performance on Tiny-ImageNet-C and Tiny-ImageNet-LT with different imbalance ratios and different models. The highest score on each column is shown with **bold**, and we use darker color to represent higher performance.

Model	Methods	Tiny-ImageNet-C			Tiny-ImageNet-LT		
		Dogs	Birds	Vehicles	Dogs	Birds	Vehicles
ResNet20	CE	78.46	85.19	87.53	93.72	94.49	97.72
	AUCMLoss	77.35	85.98	82.37	93.35	94.11	97.34
	FocalLoss	78.34	81.48	86.55	93.25	92.87	97.66
	ADVShift	81.20	80.94	86.65	93.70	93.53	97.66
	WDRO	82.20	85.23	85.92	94.46	95.50	**98.19**
	DROLT	80.44	86.91	86.76	93.89	**96.40**	97.86
	GLOT	81.96	85.89	86.80	**94.67**	96.14	98.05
	AUCDRO	75.97	83.26	79.46	92.58	93.04	96.29
	DRAUC-Df	**84.11**	87.30	88.67	93.39	95.58	97.50
	DRAUC-Da	83.96	**87.61**	**89.06**	93.76	95.94	97.25
ResNet32	CE	82.55	84.64	86.26	94.31	94.49	97.76
	AUCMLoss	77.25	85.20	81.12	93.19	95.19	97.57
	FocalLoss	77.96	79.80	85.33	93.41	92.85	97.78
	ADVShift	84.30	84.56	86.43	92.92	94.71	97.59
	WDRO	80.08	85.58	86.94	94.39	95.51	97.67
	DROLT	79.25	85.75	86.79	91.68	**96.06**	97.82
	GLOT	81.70	83.09	88.24	94.08	95.16	**97.92**
	AUCDRO	78.21	80.55	85.26	91.56	93.15	96.33
	DRAUC-Df	**85.79**	**88.00**	88.32	**94.43**	95.29	97.37
	DRAUC-Da	84.56	87.60	**88.46**	94.03	95.96	97.65



(a) Effect of $\boldsymbol{\epsilon}$ on **<u>0.01</u>** (b) Effect of $\boldsymbol{\epsilon}$ on **<u>0.05</u>** (c) Effect of $\boldsymbol{\epsilon}$ on **<u>0.1</u>** (d) Effect of $\boldsymbol{\epsilon}$ on **<u>0.01</u>**

(e) Effect of $\boldsymbol{\eta_\lambda}$ on **<u>0.01</u>** (f) Effect of $\boldsymbol{\eta_\lambda}$ on **<u>0.05</u>** (g) Effect of $\boldsymbol{\eta_\lambda}$ on **<u>0.1</u>** (h) Effect of $\boldsymbol{\eta_\lambda}$ on **<u>0.2</u>**

Figure 2: Sensitivity analysis of $\boldsymbol{\epsilon}$ and $\boldsymbol{\eta_\lambda}$ on different **<u>imbalance ratios</u>**.

5.2.2 Sensitivity Analysis

The Effect of ϵ. In Figure 2-(a)-(d), we present the sensitivity of ϵ. The results demonstrate that when the training set is relatively balanced (i.e., the imbalance ratio $p \geq 0.1$), the average robust performance improves as ϵ increases. Nonetheless, when the training set is highly imbalanced, the trend is less discernible due to the instability of the training process in these long-tailed settings.

The Effect of η_λ. In Figure 2-(e)-(h), we present the sensitivity of η_λ. η_λ governs the rate of change of λ and serves as a similar function to the warm-up epochs in AT. When η_λ is small, λ remains large for an extended period, so the adversarial example is regularized to be less offensive. In cases where the training set is extremely imbalanced, a large η_λ introduces strong examples to the model while it struggles to learn, increasing the instability of the training process and explaining why the smallest η_λ performs best with an imbalance ratio of 0.01. Conversely, when the model does not face difficulty fitting the training data, an appropriately chosen η_λ around 0.1 enhances the model's robustness.

6 Conclusion and Future Works

This paper presents an instance-wise, end-to-end framework for DRAUC optimization. Due to the pairwise formulation of AUC optimization, a direct combination with DRO is intractable. To address this issue, we propose a tractable surrogate reformulation on top of the instance-wise formulation of AUC risk. Furthermore, through a theoretical investigation on the neural collapse feature space, we find that the distribution-free perturbation is a scheme that might induce heavy label noise into the dataset. In this sense, we propose a distribution-aware framework to handle class-wise perturbation separately. Theoretically, we show that the robust generalization error is small if both the training error and $(1/\sqrt{\tilde{n}})$ is small. Finally, we conduct experiments on three benchmark datasets employing diverse model structures, and the results substantiate the superiority of our approach.

Owing to space constraints, not all potential intersections between AUC optimization and distributionally robustness can be exhaustively explored in this paper. Numerous compelling aspects warrant further investigation. We offer a detailed, instance-wise reformulation of DRAUC, primarily evolving from an AUC optimization standpoint. Future discussions could benefit from initiating dialogue from the angle of DRO. Additionally, integrating various formulations of AUC such as partial AUC and AUPRC with distributional robustness presents a fertile ground for exploration. The existence of a potentially overly-pessimistic phenomenon is yet to be conclusively determined, which paves the way for future inquiries and discoveries.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0102000, in part by National Natural Science Foundation of China: 62236008, U21B2038, U2001202, 61931008, 62122075, 61976202, and 62206264, in part by the Fundamental Research Funds for the Central Universities, in part by Youth Innovation Promotion Association CAS, in part by the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDB28000000) and in part by the Innovation Funding of ICT, CAS under Grant No. E000000.

References

[1] K. Baker, E. Dall'Anese, and T. Summers. Distribution-agnostic stochastic optimal power flow for distribution grids. In *2016 North American Power Symposium (NAPS)*, pages 1–6. IEEE, 2016.

[2] A. Ben-Tal, D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

[3] J. Blanchet, Y. Kang, and K. Murthy. Robust wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019.

[4] J. C. Duchi, T. Hashimoto, and H. Namkoong. Distributionally robust losses against mixture covariate shifts. *Under review*, 2:1, 2019.

[5] A. Feizi. Hierarchical detection of abnormal behaviors in video surveillance through modeling normal behaviors based on auc maximization. *Soft Computing*, 24(14):10401–10413, 2020.

[6] C. Frogner, S. Claici, E. Chien, and J. Solomon. Incorporating unlabeled data into distributionally robust learning. *arXiv preprint arXiv:1912.07729*, 2019.

[7] R. Gao and A. Kleywegt. Distributionally robust stochastic optimization with wasserstein distance. *Mathematics of Operations Research*, 2022.

[8] W. Gao and Z.-H. Zhou. On the consistency of auc pairwise optimization. *arXiv preprint arXiv:1208.0645*, 2012.

[9] N. Golowich, A. Rakhlin, and O. Shamir. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pages 297–299. PMLR, 2018.

[10] X. Han, V. Papyan, and D. L. Donoho. Neural collapse under mse loss: Proximity to and dynamics on the central path. *arXiv preprint arXiv:2106.02073*, 2021.

[11] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[14] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[15] W. Hou, Q. Xu, Z. Yang, S. Bao, Y. He, and Q. Huang. Adauc: End-to-end adversarial auc optimization against long-tail problems. In *International Conference on Machine Learning*, pages 8903–8925. PMLR, 2022.

[16] W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR, 2018.

[17] V. Kothapalli, E. Rasromani, and V. Awatramani. Neural collapse: A review on modelling principles and generalization. *arXiv preprint arXiv:2206.04041*, 2022.

[18] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[19] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh. Wasserstein distributionally robust optimization: Theory and applications in machine learning. In *Operations research & management science in the age of analytics*, pages 130–166. Informs, 2019.

[20] Y. Kwon, W. Kim, J.-H. Won, and M. C. Paik. Principled learning method for wasserstein distributionally robust optimization with local perturbations. In *International Conference on Machine Learning*, pages 5567–5576. PMLR, 2020.

[21] Y. Le and X. Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[23] F. Lin, X. Fang, and Z. Gao. Distributionally robust optimization: A review on theory and applications. *Numerical Algebra, Control and Optimization*, 12(1):159–212, 2022.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[25] H. Liu, K. Han, V. V. Gayah, T. L. Friesz, and T. Yao. Data-driven linear decision rule approach for distributionally robust optimization of on-line signal control. *Transportation Research Part C: Emerging Technologies*, 59:260–277, 2015.

[26] M. Liu, Z. Yuan, Y. Ying, and T. Yang. Stochastic auc maximization with deep neural networks. *arXiv preprint arXiv:1908.10831*, 2019.

[27] J. Lu and S. Steinerberger. Neural collapse with cross-entropy loss. *arXiv preprint arXiv:2012.08465*, 2020.

[28] P. Mohajerin Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1-2):115–166, 2018.

[29] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[30] N. Mu and J. Gilmer. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019.

[31] H. Namkoong and J. C. Duchi. Variance-based regularization with convex objectives. *Advances in neural information processing systems*, 30, 2017.

[32] V. Papyan, X. Han, and D. L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[34] H. Phan, T. Le, T. Phung, A. T. Bui, N. Ho, and D. Phung. Global-local regularization via distributional robustness. In *International Conference on Artificial Intelligence and Statistics*, pages 7644–7664. PMLR, 2023.

[35] Q. Qi, Y. Luo, Z. Xu, S. Ji, and T. Yang. Stochastic optimization of areas under precision-recall curves with provable convergence. *Advances in Neural Information Processing Systems*, 34:1752–1765, 2021.

[36] L. Rice, E. Wong, and Z. Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

[37] D. Samuel and G. Chechik. Distributional robustness loss for long-tail learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9495–9504, 2021.

[38] A. Shapiro. Tutorial on risk neutral, distributionally robust and risk averse multistage stochastic programming. *European Journal of Operational Research*, 288(1):1–13, 2021.

[39] A. Sinha, H. Namkoong, R. Volpi, and J. Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

[40] J. Sulam, R. Ben-Ari, and P. Kisilev. Maximizing auc with deep learning for classification of imbalanced mammogram datasets. In *VCBM*, pages 131–135, 2017.

[41] J. v. Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

[42] Z. Wang, Q. Xu, Z. Yang, Y. He, X. Cao, and Q. Huang. Openauc: Towards auc-oriented open-set recognition. *Advances in Neural Information Processing Systems*, 35:25033–25045, 2022.

[43] Z. Wang, Q. Xu, Z. Yang, Y. He, X. Cao, and Q. Huang. Optimizing partial area under the top-k curve: Theory and practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[44] P. Wen, Q. Xu, Z. Yang, Y. He, and Q. Huang. When false positive is intolerant: End-to-end optimization with low fpr for multipartite ranking. *Advances in Neural Information Processing Systems*, 34:5025–5037, 2021.

[45] P. Wen, Q. Xu, Z. Yang, Y. He, and Q. Huang. Exploring the algorithm-dependent generalization of auprc optimization with list stability. *Advances in Neural Information Processing Systems*, 35:28335–28349, 2022.

[46] T. Yang and Y. Ying. Auc maximization in the era of big data and ai: A survey. *ACM Computing Surveys*, 55(8):1–37, 2022.

[47] Z. Yang, Q. Xu, S. Bao, Y. He, X. Cao, and Q. Huang. When all we need is a piece of the pie: A generic framework for optimizing two-way partial auc. In *International Conference on Machine Learning*, pages 11820–11829. PMLR, 2021.

[48] Y. Yao, Q. Lin, and T. Yang. Large-scale optimization of partial auc in a range of false positive rates. *arXiv preprint arXiv:2203.01505*, 2022.

[49] Y. Ying, L. Wen, and S. Lyu. Stochastic online auc maximization. *Advances in neural information processing systems*, 29, 2016.

[50] Z. Yuan, Z. Guo, N. Chawla, and T. Yang. Compositional training for end-to-end deep auc maximization. In *International Conference on Learning Representations*, 2021.

[51] Z. Yuan, Y. Yan, M. Sonka, and T. 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*, pages 3040–3049, 2021.

[52] Z. Yuan, Y. Yan, M. Sonka, and T. 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*, pages 3040–3049, 2021.

[53] Z. Yuan, D. Zhu, Z.-H. Qiu, G. Li, X. Wang, and T. Yang. Libauc: A deep learning library for x-risk optimization. In *29th SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.

[54] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

[55] J. Zhang, A. Menon, A. Veit, S. Bhojanapalli, S. Kumar, and S. Sra. Coping with label shift via distributionally robust optimisation. *arXiv preprint arXiv:2010.12230*, 2020.

[56] D. Zhu, G. Li, B. Wang, X. Wu, and T. Yang. When auc meets dro: Optimizing partial auc for deep learning with non-convex convergence guarantee. In *International Conference on Machine Learning*, pages 27548–27573. PMLR, 2022.

[57] Z. Zhu, T. Ding, J. Zhou, X. Li, C. You, J. Sulam, and Q. Qu. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34:29820–29834, 2021.

Appendices

Contents

A Proofs

A.1 Proof of Proposition 2

Proof. We first give a description of the problem. Our objective is to identify the corrupted distribution that minimizes the Wasserstein distance to the original distribution, while simultaneously perturbing the AUC from 1 to 0. Specifically,

$$\min \mathcal{W}_c(\widehat{Q}, \widehat{P}) \tag{22}$$

$$s.t. AUC(f_{\boldsymbol{\theta}}, \widehat{Q}) = 0 \tag{23}$$

From the definition of Wasserstein distance, we have

$$\mathcal{W}_c(\widehat{P}, \widehat{Q}) = \min_{\boldsymbol{\Gamma}} \sum_{i=1}^{n} \sum_{j=1}^{n} \Gamma_{i,j} c_x(z_i, z_j') \tag{24}$$

$$s.t. \quad \Gamma_{i,j} \geq 0, \boldsymbol{\Gamma}^T \mathbf{1} = \frac{1}{n}\mathbf{1}, \boldsymbol{\Gamma}\mathbf{1} = \frac{1}{n}\mathbf{1}, AUC(f_{\boldsymbol{\theta}}, \widehat{Q}) = 0 \tag{25}$$

where $\boldsymbol{\Gamma}$ is the optimal transportation matrix between $\widehat{P}, \widehat{Q}$ and $c_x(z, z') = (x - x')^2 + \infty \cdot \mathbb{I}(y \neq y')$ is a metric of distance between sample z and z'.

Step 1): Separating positive and negative distance. From the definition of c_x, it is easy to check that $c_x(z_i, z_j') = \infty$ if $i \leq n^+, j > n^+$ or $i > n^+, j \leq n^+$. Consequently, the Wasserstein distance goes infinity if $\Gamma_{i,j} > 0$ in the corresponding area, resulting in

$$\boldsymbol{\Gamma} = \left[\begin{array}{cccccc} \Gamma_{1,1} & \cdots & \Gamma_{1,n^+} & & & \\ \vdots & \ddots & \vdots & & \mathbf{0} & \\ \Gamma_{n^+,1} & \cdots & \Gamma_{n^+,n^+} & & & \\ & & & \Gamma_{n^++1,n^++1} & \cdots & \Gamma_{n^++1,n} \\ & \mathbf{0} & & \vdots & \ddots & \vdots \\ & & & \Gamma_{n,1} & \cdots & \Gamma_{n,n} \end{array}\right] = \begin{bmatrix} \boldsymbol{\Gamma}^+ & 0 \\ 0 & \boldsymbol{\Gamma}^- \end{bmatrix} \tag{26}$$

Now, we can rewrite the Wasserstein distance by separating positive and negative examples

$$\mathcal{W}_c(\widehat{P}, \widehat{Q}) = \min_{\boldsymbol{\Gamma}} \underbrace{\sum_{i=1}^{n_+} \sum_{j=1}^{n_+} \Gamma_{i,j}(x_i^+ - x_j^{+'})^2}_{positive} + \underbrace{\sum_{i=n_+ + 1}^{n} \sum_{j=n_+ + 1}^{n} \Gamma_{i,j}(x_i^- - x_j^{-'})^2}_{negative} \tag{27}$$

$$s.t. \quad \Gamma_{i,j} \geq 0, \boldsymbol{\Gamma}^T \mathbf{1} = \frac{1}{n}\mathbf{1}, \boldsymbol{\Gamma}\mathbf{1} = \frac{1}{n}\mathbf{1}, AUC(f_{\boldsymbol{\theta}}, \widehat{Q}) = 0 \tag{28}$$

Step 2): Cancelling $\boldsymbol{\Gamma}$. Plugging in $x_i^+ = x^+, x_j^- = x^-, \forall i, j$, yields the Wasserstein distance of positive class can be considered as

$$\sum_{i=1}^{n_+} \sum_{j=1}^{n_+} \Gamma_{i,j}(x_i^+ - x_j^{+'})^2 \tag{29}$$

$$= \sum_{j=1}^{n_+} \left(\sum_{i=1}^{n_+} \Gamma_{i,j}(x^+ - x_j^{+'})^2 \right) \tag{30}$$

$$= \sum_{j=1}^{n_+} \left(\sum_{i=1}^{n_+} \Gamma_{i,j} \right) (x^+ - x_j^{+'})^2 \tag{31}$$

$$= \sum_{j=1}^{n_+} \frac{1}{n}(x^+ - x_j^{+'})^2 \tag{32}$$

Taking a similar step toward the negative Wasserstein distance, yields that

$$\mathcal{W}_c(\widehat{P},\widehat{Q}) = \sum_{j=1}^{n_+} \frac{1}{n}(x^+ - x_j^{+\prime})^2 + \sum_{j=n^++1}^{n} \frac{1}{n}(x^- - x_j^{-\prime})^2 \tag{33}$$

Hence, we only need to analysis the problem:

$$\min_{\boldsymbol{x}^{+\prime},\boldsymbol{x}^{-\prime}} \sum_{j=1}^{n_+} \frac{1}{n}(x^+ - x_j^{+\prime})^2 + \sum_{j=n^++1}^{n} \frac{1}{n}(x^- - x_j^{-\prime})^2 \tag{34}$$

$$s.t. \quad \max \boldsymbol{x}^{+\prime} \leq \min \boldsymbol{x}^{-\prime} \tag{35}$$

where $\boldsymbol{x}^{+\prime} = \{x_1^{+\prime}, ..., x_{n^+}^{+\prime}\}, \boldsymbol{x}^{-\prime} = \{x_{n^++1}^{-}{}^{\prime}, ..., x_n^{-\prime}\}$. The constraint comes from the definition of AUC [11].

Step 3): Solving the optimal perturbations.

We now show that, the optimal $\boldsymbol{x}^{+\prime}, \boldsymbol{x}^{-\prime}$ consists of same element, and we construct the proof by contradiction. Assume that the optimal perturbation of positive class $\boldsymbol{x}^{+,\star}$ and $\boldsymbol{x}^{-,\star}$. For the positive examples, we assume that the vector $\boldsymbol{x}^{+,\star}$ has at least two different values. Moreover, we check the simple solution $\widehat{x}^{+,\star}$ such that:

$$\widehat{x}^{+,\star} = \arg\min_{x \in \boldsymbol{x}^{+,\star}} (x^+ - x)^2$$

and denote $\widehat{\boldsymbol{x}}^{+\prime} = \{\widehat{x}^{+,\star}, ..., \widehat{x}^{+,\star}\}$. It is easy to check that

$$\sum_{j=1}^{n_+} \frac{1}{n}(x^+ - \widehat{x}^{+,\star})^2 \leq \sum_{j=1}^{n_+} \frac{1}{n}(x^+ - x_j^{+\prime})^2 \tag{36}$$

Furthermore, since

$$\max \widehat{\boldsymbol{x}}^{+\prime} \leq \max \boldsymbol{x}^{+,\star} \leq \min \boldsymbol{x}^{-,\star}, \tag{37}$$

we see that $\widehat{\boldsymbol{x}}^{+\prime}$ is also a feasible solution of the problem. Hence, $\widehat{\boldsymbol{x}}^{+\prime}$ should be the optimal solution instead of $\boldsymbol{x}^{+,\star}$. Following a similar spirit, we can also show that $\boldsymbol{x}^{-,\star}$ is not the optimal solution. In this since, the optimal solution of both $\boldsymbol{x}^{+\prime}$ and $\boldsymbol{x}^{-\prime}$ must be a vector containing the same value.

In this sense, we can further simplfy the targeted optimization problem as:

$$\min_{x^{+\prime},x^{-\prime}} \widehat{p}(x^+ - x^{+\prime})^2 + (1-\widehat{p})(x^- - x^{-\prime})^2 \tag{38}$$

$$s.t. \quad x^{+\prime} \leq x^{-\prime} \tag{39}$$

where $\widehat{p} = \frac{n^+}{n}$ is the ratio of the positive examples in the dataset.

Step 4): Calculating an upper bound of the objective function. To obtain an upper bound, we can instead check the solution of the following problem:

$$\min_{x'} \ \widehat{p}(x^+ - x')^2 + (1-\widehat{p})(x^- - x')^2 \tag{40}$$

It achieves an upper bound since $x^{+\prime} \leq x^{-\prime}$ is automatically satisfied by setting $x^{+\prime} = x^{-\prime} = x'$. By solving this problem, we can see that the optimal solution is:

$$x' = \widehat{p}x^+ + (1-\widehat{p})x^-, \tag{41}$$

and an upper bound of minimal Wasserstein distance to perturb AUC from 1 to 0 is $\widehat{p}(1-\widehat{p})(x^+ - x^-)^2$. □

A.2 Derivations of Optimization Problem (20)

Remark 2. *The original optimization of Distribution-aware DRAUC is*

$$(\mathbf{DRAUC-Da}) \quad \min_{\boldsymbol{\theta}} \max_{\widehat{Q} \in \widehat{\mathcal{Q}}} \min_{(a,b)\in\Omega_{a,b}} \max_{\alpha\in\Omega_\alpha} \mathop{\mathbb{E}}_{\widehat{Q}_+,\widehat{Q}_-}[g(a,b,\alpha,\boldsymbol{\theta},z_i)]. \tag{42}$$

Similar to what we have done in Section 4.2, for a fixed $\boldsymbol{\theta}, \widehat{Q}_+, \widehat{Q}_-$, we are able to interchange the inner $\min_{(a,b)\in\Omega_{a,b}}$ and $\max_{\alpha\in\Omega_\alpha}$ by invoking von Neumann's Minimax theorem [41], which results in

$$\min_{\boldsymbol{\theta}} \max_{\widehat{Q}\in\widehat{\mathcal{Q}}} \max_{\alpha\in\Omega_\alpha} \min_{(a,b)\in\Omega_{a,b}} \mathop{\mathbb{E}}_{\widehat{Q}_+,\widehat{Q}_-}[g(a,b,\alpha,\boldsymbol{\theta},z)]. \tag{43}$$

Subsequently, based on the property that $\max_x \min_y f(x,y) \leq \min_y \max_x f(x,y)$, we reach an upper bound of the objective function:

$$\underbrace{\max_{\widehat{Q}\in\widehat{\mathcal{Q}}} \max_{\alpha\in\Omega_\alpha} \min_{(a,b)\in\Omega_{a,b}} \mathop{\mathbb{E}}_{\widehat{Q}_+,\widehat{Q}_-}[g(a,b,\alpha,\theta,z)]}_{DRAUC^{Da}_{\epsilon_+,\epsilon_-}(f_{\boldsymbol{\theta}},\widehat{P})} \leq \underbrace{\min_{(a,b)\in\Omega_{a,b}} \max_{\alpha\in\Omega_\alpha} \max_{\widehat{Q}\in\widehat{\mathcal{Q}}} \mathop{\mathbb{E}}_{\widehat{Q}_+,\widehat{Q}_-}[g(a,b,\alpha,\theta,z)]}_{\widetilde{DRAUC}^{Da}_{\epsilon_+,\epsilon_-}(f_{\boldsymbol{\theta}},\widehat{P})} \tag{44}$$

From this perspective, if we minimize $\widetilde{DRAUC}^{Da}_{\epsilon_+,\epsilon_-}(f_{\boldsymbol{\theta}},\widehat{P})$ in turn, we can at least minimize an **upper bound** of $DRAUC^{Da}_{\epsilon_+,\epsilon_-}(f_{\boldsymbol{\theta}},\widehat{P})$. In light of this, we will employ the following optimization problem as a surrogate for $(\mathbf{DRAUC-Da})$:

$$(\boldsymbol{Da}) \quad \min_{\mathbf{w}} \max_{\alpha\in\Omega_\alpha} \max_{\widehat{Q}\in\widehat{\mathcal{Q}}} \mathop{\mathbb{E}}_{\widehat{Q}_+,\widehat{Q}_-}[g(\mathbf{w},\alpha,z)] \tag{45}$$

where $\mathbf{w} = \boldsymbol{\theta}, (a,b) \in \Omega_{a,b}$. To further derive a simplified upper bound, one should note that

$$\mathop{\mathbb{E}}_{\widehat{Q}_+,\widehat{Q}_-}[g(\mathbf{w},\alpha,z)] = \widehat{p} \mathop{\mathbb{E}}_{\widehat{Q}_+}[g(\mathbf{w},\alpha,z)] + (1-\widehat{p}) \mathop{\mathbb{E}}_{\widehat{Q}_-}[g(\mathbf{w},\alpha,z)]$$

Hence the inner maximization admits an upper bound:

$$\max_{\widehat{Q}\in\widehat{\mathcal{Q}}} \mathop{\mathbb{E}}_{\widehat{Q}_+,\widehat{Q}_-}[g(\mathbf{w},\alpha,z)] \leq \widehat{p} \max_{\widehat{Q}_+\leq\epsilon_+} \mathop{\mathbb{E}}_{\widehat{Q}_+}[g(\mathbf{w},\alpha,z)] + (1-\widehat{p}) \max_{\widehat{Q}_-\leq\epsilon_-} \mathop{\mathbb{E}}_{\widehat{Q}_-}[g(\mathbf{w},\alpha,z)]$$

By adopting Thm.2, we reach the correspding upper bound:

$$(\boldsymbol{Da}) = \min_{\mathbf{w}} \max_{\alpha\in\Omega_\alpha} \min_{\lambda_+,\lambda_-\geq 0} \{\lambda_+\epsilon_+ + \lambda_-\epsilon_- + \widehat{p} \mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] + (1-\widehat{p}) \mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\} \tag{46}$$

where $\phi_{\mathbf{w},\lambda,\alpha}(z) = \max_{z'\in\mathcal{Z}}[g(\mathbf{w},\alpha,z) - \lambda c(z,z')]$. This min-max-min formulation remains difficult to optimize, so we take a step similar to (13) that interchange the inner $\min_{\lambda_+,\lambda_-\geq 0}$ and outer $\max_{\alpha\in\Omega_\alpha}$, resulting in a tractable **upper bound**

$$(\boldsymbol{Da\star}) \min_{\mathbf{w}} \min_{\lambda_+,\lambda_-\geq 0} \max_{\alpha\in\Omega_\alpha} \{\lambda_+\epsilon_+ + \lambda_-\epsilon_- + \widehat{p} \mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] + (1-\widehat{p}) \mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\} \tag{47}$$

A.3 Proof of Theorem 3

Since we optimize DRAUC-Da in a class-wise manner, we now give our definition of Rademacher Complexity on positive dn negative distributions, respectively.

Definition 3 (Definition of Rademacher Complexity of Robust AUC). *Given a hypothesis class Θ and empirical distribution $\widehat{P}$, for all $t \in \Theta, \lambda_+ \geq 0, \lambda_- \geq 0, \alpha \in \Omega_\alpha, (a,b) \in \Omega_{a,b}$, the Positive/Negative Empirical Rademacher Complexity of Robust AUC is defined as*

$$\widehat{\mathfrak{R}}^+_{\widehat{P}_+}(\Theta) = \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\mathbf{w},\lambda_+\geq\mathbf{0},\alpha\in\boldsymbol{\Omega}_\alpha} \frac{1}{n_+}\sum_{i=1}^{n_+} \sigma_i \cdot \phi_{\mathbf{w},\lambda_+,\alpha}(z)\right], \tag{48}$$

$$\widehat{\mathfrak{R}}^-_{\widehat{P}_-}(\Theta) = \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\mathbf{w},\lambda_-\geq\mathbf{0},\alpha\in\boldsymbol{\Omega}_\alpha} \frac{1}{n_-}\sum_{i=1}^{n_-} \sigma_i \cdot \phi_{\mathbf{w},\lambda_-,\alpha}(z)\right], \tag{49}$$

where σ is the Rademacher random variable, and Positive/Negative Rademacher Complexity of Robust AUC on hypothesis class Θ is

$$\mathfrak{R}^+_m(\Theta) = \mathop{\mathbb{E}}_{\widehat{P}_+}[\widehat{\mathfrak{R}}^+_{\widehat{P}_+}(\Theta)], \mathfrak{R}^-_m(\Theta) = \mathop{\mathbb{E}}_{\widehat{P}_-}[\widehat{\mathfrak{R}}^-_{\widehat{P}_-}(\Theta)]. \tag{50}$$

The main result could be restated formally in the following sense.

Theorem 4 (**Restate of Theorem 3**). *If the samples of the training drawn* `i.i.d.`, *then for all* $\boldsymbol{\theta} \in \Theta, (a,b) \in \Omega_{a,b}, \alpha \in \Omega_\alpha, \lambda_+ \geq 0, \lambda_- \geq 0$, *the following holds with probability at least* $1-\delta$ *over the randomness of the sample:*

$$\begin{aligned}
DRAUC^{Da}_{\epsilon_+,\epsilon_-}(f_{\boldsymbol{\theta}}, P) \leq \widehat{\mathcal{L}} + 2 \cdot \widehat{p} \cdot \widehat{\mathfrak{R}}^+_{\widehat{P}_+}(\Theta) + 2 \cdot (1-\widehat{p}) \cdot \widehat{\mathfrak{R}}^-_{\widehat{P}_-}(\Theta) + \\
C_+ \cdot \widehat{p} \cdot \sqrt{\frac{\log(8/\delta)}{2n_+}} + C_- \cdot (1-\widehat{p}) \cdot \sqrt{\frac{\log(8/\delta)}{2n_-}} + \\
2 \cdot C_\infty \cdot \sqrt{\frac{\log(1/\delta)}{2n}}
\end{aligned}$$

where C_+, C_-, C_∞ *are universal constants,* $p = \mathbb{P}[y=1], \widehat{p} = \widehat{\mathbb{P}}[y=1]$ *and*

$$\widehat{\mathcal{L}} = \min_{\mathbf{w}} \min_{\lambda_+,\lambda_- \geq 0} \max_{\alpha \in \Omega_\alpha} \left[\lambda_+\epsilon_+ + \lambda_-\epsilon_- + \widehat{p} \mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] + (1-\widehat{p}) \mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)] \right]$$

is the saddle point of the training loss.

Remark 3. *The claim of Thm.3 holds since the Rademacher complexity of training data with size* n *is known to be scaled like* $O(\sqrt{1/n})$ *for many hypothesis classes such as linear classifiers [29] and neural networks [9].*

We now give a detailed proof of Theorem 3. As the begining, we give some useful lemmas in proving the result.

Lemma 1. *The following inequality holds for all* $\boldsymbol{\theta} \in \Theta, \epsilon_+, \epsilon_- > 0$

$$DRAUC^{Da}_{\epsilon_+,\epsilon_-}(f_{\boldsymbol{\theta}}, P) \tag{51}$$

$$\leq \min_{(a,b)\in\Omega_{a,b}} \min_{\lambda_+,\lambda_- \geq 0} \max_{\alpha \in \Omega_\alpha} \{ \lambda_+\epsilon_+ + \lambda_-\epsilon_- + p \mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] + (1-p) \mathop{\mathbb{E}}_{P_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)] \} \tag{52}$$

Proof. The proof is the similar to the proof derivations in the last subsection, except dropping the outer $\min_{\boldsymbol{\theta}}$ and changing the empirical distribution $\widehat{P}$ to the real distribution P. □

Lemma 2. *For any real valued function continuous function:* $f : \mathbb{R} \to \mathbb{R}$, $g : \mathbb{R} \to \mathbb{R}$, *and for any tight set* $\mathcal{X} \subset \mathbb{R}$*:*

$$\begin{aligned}
\max_{x\in\mathcal{X}} f(x) - \max_{x'\in\mathcal{X}} g(x') \leq \max_{x\in\mathcal{X}} f(x) - g(x) \\
\min_{x\in\mathcal{X}} f(x) - \min_{x'\in\mathcal{X}} g(x') \leq \max_{x\in\mathcal{X}} f(x) - g(x)
\end{aligned}$$

Proof. Since both f and g are continuous, and $\mathcal{X}$ is tight, we now that the maximum and the minimum in the lemma exists. From the basic property of the maxima, we have:

$$\max_{x\in\mathcal{X}} f(x) - \max_{x'\in\mathcal{X}} g(x') = \max_{x\in\mathcal{X}} \min_{x'\in\mathcal{X}} f(x) - g(x') \leq \max_{x\in\mathcal{X}} f(x) - g(x').$$

Similarly, for the minimum, we have:

$$\begin{aligned}
\min_{x\in\mathcal{X}} f(x) - \min_{x'\in\mathcal{X}} g(x') = \min_{x\in\mathcal{X}} \max_{x'\in\mathcal{X}} f(x) - g(x') \\
= \max_{x'\in\mathcal{X}} \min_{x\in\mathcal{X}} f(x) - g(x') \leq \max_{x\in\mathcal{X}} f(x) - g(x).
\end{aligned}$$

□

Lemma 3. *Assume that for each $\boldsymbol{x} \in \mathcal{X}$, there exist a sample pair $(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X} \times \mathcal{X}$, such that $d(\boldsymbol{x}, \boldsymbol{x}') < \infty$, we have the following result holds for the risk function:*

$$\begin{aligned}
&\sup_{\boldsymbol{\theta}\in\Theta}\Bigg[DRAUC^{Da}_{\epsilon_+,\epsilon_-}(f_{\boldsymbol{\theta}}, P) \\
&\qquad - \min_{(a,b)\in\Omega_{a,b}} \min_{\lambda_+,\lambda_-\geq 0} \max_{\alpha\in\Omega_\alpha}\{\lambda_+\epsilon_+ + \lambda_-\epsilon_- + \widehat{p}\cdot \mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] + (1-\widehat{p})\cdot \mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\Bigg] \\
&\leq \widehat{p}\cdot \sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_+\geq 0}\left[\mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] - \mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]\right] + \\
&\quad (1-\widehat{p})\cdot \sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_-\geq 0}\left[\mathop{\mathbb{E}}_{P_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)] - \mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\right] + 2\cdot C_\infty \cdot |p-\widehat{p}|
\end{aligned}$$

where:

$$0 \leq C_\infty = \sup_{\boldsymbol{z}\in\mathcal{Z},\boldsymbol{w},\alpha\in\Omega_\alpha,\lambda\geq 0} \phi_{\mathbf{w},\lambda,\alpha}(z) < \infty.$$

Proof. First, we proof the claim that:

$$0 \leq C_\infty < \infty$$

We have:

$$\max_{z'\in\mathcal{Z},\mathbf{w},\alpha\in\Omega_\alpha,\lambda\geq 0}[g(\mathbf{w},\alpha,z) - \lambda c(z,z')] \geq \max_{\mathbf{w},\alpha\in\Omega_\alpha,\lambda\geq 0}[g(\mathbf{w},\alpha,z) - \lambda c(z,z)] = \max_{\mathbf{w},\alpha\in\Omega_\alpha}[g(\mathbf{w},\alpha,z)].$$

Moreover, since the output of the scoring function resides in $[0,1]$, we know that $g(\mathbf{w},\alpha,z)$ is bounded from below uniformally by 0. Hence, $0 \leq C_\infty$.

Similarly,

$$\max_{z'\in\mathcal{Z},\mathbf{w},\alpha\in\Omega_\alpha,\lambda\geq 0}[g(\mathbf{w},\alpha\in\Omega_\alpha,z) - \lambda c(z,z')] \leq \max_{\mathbf{w},\alpha}[g(\mathbf{w},\alpha,z)]$$

since $c(z,z) \geq 0$. Moreover, since the output of the scoring function resides in $[0,1]$, we know that $g(\mathbf{w},\alpha,z)$ is bounded from above uniformally by some finite constant $B < \infty$. Hence, $C_\infty < \infty$.

From Lem.1, we have:

$$\begin{aligned}
&\sup_{\boldsymbol{\theta}\in\Theta}\Bigg[DRAUC^{Da}_{\epsilon_+,\epsilon_-}(f_{\boldsymbol{\theta}}, P) \\
&\qquad - \min_{(a,b)\in\Omega_{a,b}} \min_{\lambda_+,\lambda_-\geq 0} \max_{\alpha\in\Omega_\alpha}\{\lambda_+\epsilon_+ + \lambda_-\epsilon_- + \widehat{p}\mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] + (1-\widehat{p})\mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\Bigg] \\
&\leq \sup_{\boldsymbol{\theta}\in\Theta}\Bigg[\min_{(a,b)\in\Omega_{a,b}} \min_{\lambda_+,\lambda_-\geq 0} \max_{\alpha\in\Omega_\alpha}\{\lambda_+\epsilon_+ + \lambda_-\epsilon_- + p\mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] + (1-p)\mathop{\mathbb{E}}_{P_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)] \\
&\qquad - \min_{(a,b)\in\Omega_{a,b}} \min_{\lambda_+,\lambda_-\geq 0} \max_{\alpha\in\Omega_\alpha}\{\lambda_+\epsilon_+ + \lambda_-\epsilon_- + \widehat{p}\mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] + (1-\widehat{p})\mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\Bigg]
\end{aligned}$$

By applying Lem.2 three times for $(a,b)\in\Omega_{a,b}$, $\min_{\lambda_+\geq 0,\lambda_-\geq 0}$ and $\alpha\in\Omega_\alpha$, we have:

$$\begin{aligned}
&\sup_{\boldsymbol{\theta}\in\Theta}\Bigg[DRAUC^{Da}_{\epsilon_+,\epsilon_-}(f_{\boldsymbol{\theta}}, P) \\
&\qquad - \min_{(a,b)\in\Omega_{a,b}} \min_{\lambda_+,\lambda_-\geq 0} \max_{\alpha\in\Omega_\alpha}\{\lambda_+\epsilon_+ + \lambda_-\epsilon_- + \widehat{p}\cdot \mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] + (1-\widehat{p})\mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\Bigg] \\
&\leq \sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_+\geq 0,\lambda_-\geq 0}\Bigg[p\cdot \mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] - \widehat{p}\cdot \mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] \\
&\qquad\qquad + (1-p)\cdot \mathop{\mathbb{E}}_{P_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)] - (1-\widehat{p})\cdot \mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\Bigg] \\
&\leq \sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_+\geq 0}\left[p\cdot \mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)] - \widehat{p}\cdot \mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]\right] + \\
&\quad \sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_-\geq 0}\left[(1-p)\cdot \mathop{\mathbb{E}}_{P_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)] - (1-\widehat{p})\cdot \mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\right]
\end{aligned}$$

For the positive part, we have:

$$\begin{aligned}
&\sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_+\geq 0}\left[p\cdot\mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]-\widehat{p}\cdot\mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]\right]\\
\leq&\sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_+\geq 0}\left[p\cdot\mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]-\widehat{p}\cdot\mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]+\widehat{p}\cdot\mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]-\widehat{p}\cdot\mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]\right]\\
\leq&\sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_+\geq 0}\left[(p-\widehat{p})\cdot\mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]\right]+\widehat{p}\cdot\sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_+\geq 0}\left[\mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]-\mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]\right]\\
\leq& C_\infty\cdot|p-\widehat{p}|+\widehat{p}\cdot\sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_+\geq 0}\left[\mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]-\mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]\right].
\end{aligned}$$

Similar, we have for the negative part:

$$\begin{aligned}
&\sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_-\geq 0}\left[(1-p)\cdot\mathop{\mathbb{E}}_{P_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]-(1-\widehat{p})\cdot\mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\right]\\
\leq& C_\infty\cdot|p-\widehat{p}|+(1-\widehat{p})\cdot\sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_-\geq 0}\left[\mathop{\mathbb{E}}_{P_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]-\mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\right].
\end{aligned}$$

The result then follows directly.

□

Proof ***of Thm.4***. For the sake of simplicity, we denote:

$$\begin{aligned}
\textbf{(a)}&=\sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_+\geq 0}\left[\mathop{\mathbb{E}}_{P_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]-\mathop{\mathbb{E}}_{\widehat{P}_+}[\phi_{\mathbf{w},\lambda_+,\alpha}(z)]\right]\\
\textbf{(b)}&=\sup_{\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\lambda_-\geq 0}\left[\mathop{\mathbb{E}}_{P_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]-\mathop{\mathbb{E}}_{\widehat{P}_-}[\phi_{\mathbf{w},\lambda_-,\alpha}(z)]\right]\\
\textbf{(c)}&=|p-\widehat{p}|
\end{aligned}$$

From the Rademacher-complexity-based uniform convergence result, we have, with probability at least $1-\frac{\delta}{4}$:

$$\textbf{(a)}\leq 2\cdot\widehat{\mathfrak{R}}^{+}_{\widehat{P}_+}(\Theta)+C_+\cdot\sqrt{\frac{\log(8/\delta)}{2n_+}}$$

where C_+ is a universal constant.

Similarly, we have, with probability at least $1-\frac{\delta}{4}$:

$$\textbf{(b)}\leq 2\cdot\widehat{\mathfrak{R}}^{-}_{\widehat{P}_-}(\Theta)+C_-\cdot\sqrt{\frac{\log(8/\delta)}{2n_-}}$$

where C_- is a universal constant. From the Chernoff bound, we have, with probability at least $1-\frac{\delta}{2}$:

$$\textbf{(c)}\leq\sqrt{\frac{\log(1/\delta)}{2n}}$$

Following the union bound and Lem.3, we have the following result holds for all $\boldsymbol{\theta}\in\Theta,(a,b)\in\Omega_{a,b},\alpha\in\Omega_\alpha,\lambda_+\geq 0,\lambda_-\geq 0$, the following holds with probability at least $1-\delta$:

$$\begin{aligned}
DRAUC^{Da}_{\epsilon_+,\epsilon_-}(f_{\boldsymbol{\theta}},P)\leq&\widehat{\mathcal{L}}+2\cdot\widehat{p}\cdot\widehat{\mathfrak{R}}^{+}_{\widehat{P}_+}(\Theta)+2\cdot(1-\widehat{p})\cdot\widehat{\mathfrak{R}}^{-}_{\widehat{P}_-}(\Theta)+\\
&C_+\cdot\widehat{p}\cdot\sqrt{\frac{\log(8/\delta)}{2n_+}}+C_-\cdot(1-\widehat{p})\cdot\sqrt{\frac{\log(8/\delta)}{2n_-}}+\\
&2\cdot C_\infty\cdot\sqrt{\frac{\log(1/\delta)}{2n}}
\end{aligned}$$

□

B Experiments

B.1 Datasets

We first introduce the dataset used in the following section:

- **MNIST** [22]: The MNIST dataset comprises 60,000 images of digits, each with a resolution of 28x28, and includes 6,000 images for each digit from 0 to 9. This dataset is partitioned into a training set containing 50,000 images and a testing set with 10,000 images. We also allocate 10,000 images from the training set to create a validation set.
- **CIFAR-10/CIFAR-100** [18]: CIFAR-10/CIFAR-100 features 60,000 images, each having a resolution of 32x32x3, equally distributed across 10/100 classes and containing 6,000/600 images per class. The dataset is separated into a training set of 50,000 images and a testing set of 10,000 images. In addition, we extract 10,000 images from the training set to form a validation set.
- **Tiny-ImageNet** [21]: The Tiny-ImageNet dataset comprises 110,000 images in 200 classes, including 100,000 training examples and 10,000 testing examples. We further split off a validation set containing 20,000 examples from the training set. We find that generating a binary Tiny-ImageNet-200 by assigning the first half of the classes as positive and the rest as negative makes this dataset too challenging to learn. The methods struggle to learn good features and reach a testing AUC no larger than 0.6. As a result, we assign the binary version of Tiny-ImageNet by utilizing the hyper-class information. As detailed, we construct three subsets as follows:
  - Tiny-ImageNet-200-Dogs: Classes [11, 39, 78, 135, 182, 194] are assigned as positive, with the remainder designated as negative.
  - Tiny-ImageNet-200-Birds: Classes [35, 41, 67, 115] are assigned as positive, with the remainder designated as negative.
  - Tiny-ImageNet-200-Vehicles: Classes [15, 64, 69, 75, 90, 108, 114, 117, 147, 152, 157, 163] are assigned as positive, with the remainder designated as negative.
- **MNIST-C** [30]: MNIST-C is a corrupted variant of the original MNIST testing set, consisting of 160,000 testing examples generated through 16 distinct perturbation techniques (including identity transform) tailored for handwritten digits.
- **CIFAR-10-C/CIFAR-100-C** [13]: The CIFAR-10-C/CIFAR-100-C datasets are corrupted versions of the original CIFAR-10/CIFAR-100 testing sets, encompassing 950,000 images obtained by applying five intensity levels of 19 different corruption types, such as noises, blurs, and transformations. We analyze the average performance for each corruption level.
- **Tiny-ImageNet-C** [14]: The Tiny-ImageNet-C is the corrupted version of Tiny-ImageNet. 5 levels of 15 different corruptions including brightness, compression and blurs are applied to 10,000 images to generate 950,000 testing images.

B.2 Dataset Constructions

We construct our binary long-tailed dataset following a manner similar to [48]. First, we construct a binary version of the dataset by assigning the former half of the classes as positive and the latter half as negative. Then, utilizing the imbalance ratio, i.e. $\{0.01, 0.05, 0.1, 0.2\}$ in our configuration, we randomly eliminate a portion of positive samples to create the long-tailed version. For instance, to produce CIFAR10-LT with an imbalance ratio of 0.1, we designate classes 0-4 as positive and classes 5-9 as negative. Subsequently, we randomly remove $\approx 89\%$ of the training samples to achieve the desired long-tailed dataset.

B.3 Competitors

To confirm the robustness of our proposed method in imbalanced scenarios, we compare it with the following competitors, each corresponding to one row in Table 1:

- **Baseline**: Cross-entropy loss (**CE**).
- **Typical methods for long-tailed problems**:

  - **FocalLoss** [24]: A classical reweighting method for long-tailed problems.
  - **AUCMLoss** [51]: An instance-wise binary AUC optimization technique.
  - **AUCDRO** [56, 53]: A method that integrates DRO technique with partial AUC optimization.

- **DRO methods**:
  - **ADVShift** [55]: A DRO method addressing label shift.
  - **WDRO** [20]: A Wasserstein DRO technique incorporating local perturbations.
  - **DROLT** [37]: A loss function designed to learn low-variance representations.
  - **GLOT** [34]: A regularization method based on optimal transport distributional robustness.

- **Our approach**:
  - Our Algorithm 1 (**DRAUC-Df**).
  - Our Algorithm 2 (**DRAUC-Da**).

B.4 Implementation Details

We conducted all experiments on a `Ubuntu 20.04.5` server, equipped with an `Intel(R) Xeon(R) Gold 6230R CPU` and an `RTX 3090 GPU`. All codes were implemented using `PyTorch` (v-1.8.2) [33], `TorchVision` (v-0.9.2), and `Numpy` (v-1.21.4) under a `Python` 3.8 and `CUDA` 11.1 environment.

For our models, we selected ResNet20 [12], ResNet32, and Small CNN as the backbone architectures. The models were trained for 100 epochs across all datasets. On the CIFAR10, CIFAR100 and Tiny-ImageNet datasets, we applied random cropping with padding and random horizontal flipping as data augmentation techniques. However, for the MNIST dataset, we refrained from applying any data augmentation because the horizontal flip could alter the semantic meaning of the digits. For all experiments, we set the weight decay to 5×10^{-4} and the batch size to 128. During training, we utilized a sampler to ensure that at least one positive example was included in each batch.

B.5 Choices of Hyperparameters

Initial Learning Rate and Learning Rate Scheduler. We selected the initial learning rate from the set $0.01, 0.05, 0.1, 0.2$. In the majority of cases that are not extremely imbalanced, $lr = 0.1$ is a favorable choice. However, in situations where the dataset is extremely imbalanced, careful tuning of the initial learning rate is necessary. We chose the learning rate scheduler from a step scheduler, which decays the learning rate by $10\times$ at the 50-th and 75-th epochs, and the Cosine Annealing scheduler.

Robust Diameter ϵ. We selected ϵ from the set $\{8/255, 32/255, 64/255, 128/255\}$, considering l_2 distance. A sensitivity analysis regarding ϵ is presented in Section 5.2.2. For distribution-aware DRAUC, we chose ϵ_+ and ϵ_- using the following approach: Given an overall diameter ϵ and a tunable parameter $k \in \{0.5, 0.8, 1, 1.2, 1.5\}$, we set $\epsilon_+ = k\epsilon$ and $\epsilon_- = (1 - k\widehat{p})\epsilon/(1 - \widehat{p})$.

Learning Rates for Tunable Parameters. We selected $\eta_\alpha = \eta_w = lr$ and $\eta_z = 15/255$, which aligns with the standard settings in Adversarial Training [36]. For η_λ, we chose from the set $\{0.01, 0.02, 0.1, 0.2\}$. A sensitivity analysis regarding η_λ is detailed in Section 5.2.2.

Initialization of Tunable Parameters. For initialization, we set $\lambda_0 = 1$, $a^0 = 0$, $b^0 = 0$, $\alpha^0 = 0$, and PGD steps $K = 10$.

B.6 Additional Empirical Results

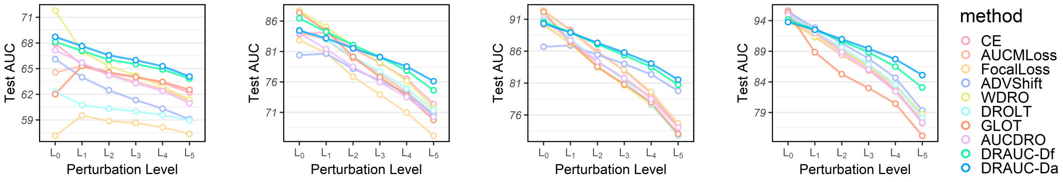
In Table 3, we display the overall performance metrics for CIFAR100-C and CIFAR100-LT, while Table 4 illustrates the performance for MNIST-C and MNIST-LT. Additionally, the overall performance under varying perturbation levels is presented in Figures 3-5. We have not included the results for MNIST-C due to the original MNIST-C [30] only providing a single perturbation level. These comprehensive results facilitate several observations, as detailed in Section 5.2.1:

Table 3: Overall Performance on CIFAR100-C and CIFAR100-LT with different imbalance ratios and different models. The highest score on each column is shown with **bold**, and we use darker color to represent higher performance.

Model	Methods	CIFAR100-C				CIFAR100-LT			
		0.01	0.05	0.10	0.20	0.01	0.05	0.10	0.20
ResNet20	CE	53.86	60.71	65.11	69.59	54.22	62.30	67.77	74.58
	AUCMLoss	55.70	61.91	65.73	69.58	57.38	64.20	**69.33**	74.92
	FocalLoss	52.10	62.28	65.91	70.36	52.46	64.46	69.24	75.05
	ADVShift	54.08	61.33	66.13	69.25	54.73	63.71	68.77	73.43
	WDRO	55.31	61.38	65.92	70.70	56.79	64.12	68.91	**76.37**
	DROLT	55.61	61.36	63.83	69.13	56.49	63.72	67.74	74.82
	GLOT	55.58	60.62	63.53	68.13	57.13	62.43	65.88	71.89
	AUCDRO	55.96	61.65	62.67	65.72	57.14	**64.74**	66.59	70.66
	DRAUC-Df	**57.47**	**62.32**	**66.25**	**71.36**	**58.97**	63.95	68.78	74.14
	DRAUC-Da	57.42	62.28	66.11	71.14	58.94	63.91	68.44	74.07
ResNet32	CE	52.90	60.74	64.57	69.51	53.08	62.03	67.14	74.43
	AUCMLoss	56.19	61.87	63.64	69.81	57.62	63.67	67.99	73.85
	FocalLoss	50.27	59.70	62.91	68.52	50.41	61.53	66.30	72.20
	ADVShift	50.15	59.35	64.00	69.37	50.20	61.97	65.70	73.64
	WDRO	55.90	61.17	65.41	68.98	57.19	63.32	68.55	73.10
	DROLT	56.43	61.10	64.02	69.61	57.27	63.22	67.37	73.18
	GLOT	57.04	60.34	63.76	65.64	58.33	62.53	66.30	70.99
	AUCDRO	56.93	61.41	64.08	68.93	58.33	64.02	**68.71**	73.86
	DRAUC-Df	**57.17**	62.02	65.83	**71.22**	**58.38**	63.90	68.51	**74.57**
	DRAUC-Da	56.81	**62.48**	**66.12**	70.62	57.98	**64.39**	68.70	74.26

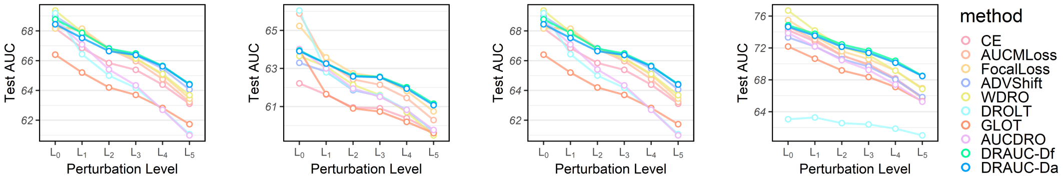
Table 4: Overall Performance on MNIST-C and MNIST-LT with different imbalance ratios and different models. The highest score on each column is shown with **bold**, and we use darker color to represent higher performance.

Model	Methods	MNIST-Origin				MNIST-Corrupted			
		0.01	0.05	0.10	0.20	0.01	0.05	0.10	0.20
SmallCNN	CE	95.54	97.01	98.09	98.41	99.38	99.84	99.94	99.95
	AUCMLoss	95.52	98.16	98.04	98.60	99.26	99.87	99.92	99.96
	FocalLoss	55.10	91.39	92.61	96.35	67.05	98.64	99.39	99.73
	ADVShift	94.06	97.66	98.09	98.09	99.21	99.84	**99.95**	99.96
	WDRO	95.98	97.62	98.48	98.40	99.26	99.89	99.95	99.96
	DROLT	88.90	92.36	96.33	98.29	**99.46**	99.79	99.88	99.96
	GLOT	95.78	97.81	97.76	98.56	99.39	**99.91**	99.94	**99.97**
	AUCDRO	94.00	97.80	97.76	98.54	99.12	99.82	99.92	99.95
	DRAUC-Df	96.06	**98.38**	**98.69**	98.84	99.19	99.9	99.94	99.96
	DRAUC-Da	**96.35**	98.04	98.59	**98.92**	99.34	99.86	99.94	99.97
ResNet20	CE	91.88	97.49	97.14	97.88	99.48	**99.97**	**99.98**	99.98
	AUCMLoss	89.09	97.82	96.26	97.74	99.47	99.82	99.96	99.98
	FocalLoss	70.78	94.45	95.83	97.28	98.90	99.85	99.95	99.97
	ADVShift	87.43	90.12	96.74	97.36	99.46	99.83	99.97	99.98
	WDRO	93.87	97.81	97.66	98.47	99.17	99.94	99.97	**99.99**
	DROLT	88.82	94.49	96.17	97.97	**99.73**	99.81	99.79	99.98
	GLOT	84.46	95.90	97.46	97.07	98.80	99.88	99.96	99.98
	AUCDRO	89.11	94.40	95.71	96.73	98.65	99.87	99.89	99.95
	DRAUC-Df	95.96	98.21	98.44	**98.80**	99.45	99.93	99.97	99.97
	DRAUC-Da	**96.70**	**98.37**	**98.57**	98.79	99.56	99.91	99.94	99.96



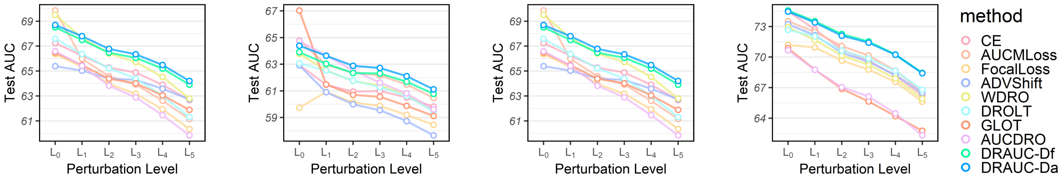
(a) Imratio **0.01** (b) Imratio **0.05** (c) Imratio **0.1** (d) Imratio **0.2**

Figure 3: Overall Performance of ResNet20 Across Perturbation Levels on CIFAR10. This graph illustrates the performance of various methods at different corruption levels, with Level 0 indicating no corruption and Level 5 representing the most severe corruption. In each figure, the seven lines depict the test AUC for CE, AUCMLoss, FocalLoss, ADVShift, WDRO, DROLT, GLOT, AUCDRO, DRAUC-Da and DRAUC-Df, respectively. Best viewed in colors.



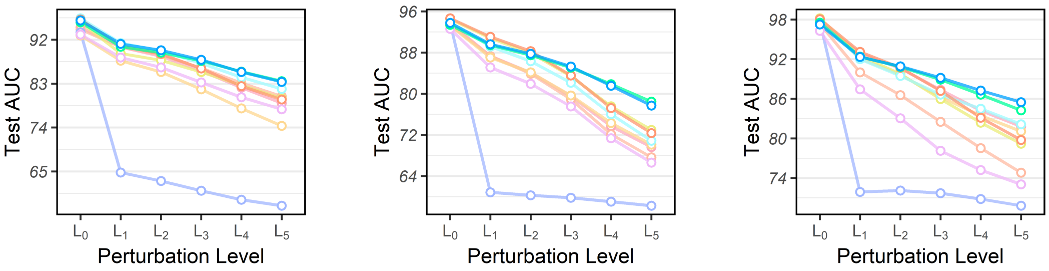
(a) Imratio **0.01** (b) Imratio **0.05** (c) Imratio **0.1** (d) Imratio **0.2**

Figure 4: Overall Performance of ResNet32 Across Perturbation Levels on CIFAR100. This graph illustrates the performance of various methods at different corruption levels, with Level 0 indicating no corruption and Level 5 representing the most severe corruption. In each figure, the seven lines depict the test AUC for CE, AUCMLoss, FocalLoss, ADVShift, WDRO, DROLT, GLOT, AUCDRO, DRAUC-Da and DRAUC-Df, respectively. Best viewed in colors.



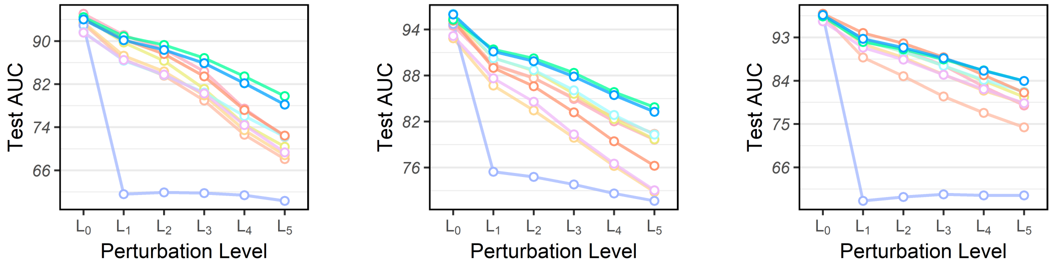
(a) Imratio **0.01** (b) Imratio **0.05** (c) Imratio **0.1** (d) Imratio **0.2**

Figure 5: Overall Performance of ResNet32 Across Perturbation Levels on CIFAR100. This graph illustrates the performance of various methods at different corruption levels, with Level 0 indicating no corruption and Level 5 representing the most severe corruption. In each figure, the seven lines depict the test AUC for CE, AUCMLoss, FocalLoss, ADVShift, WDRO, DROLT, GLOT, AUCDRO, DRAUC-Da and DRAUC-Df, respectively. Best viewed in colors.



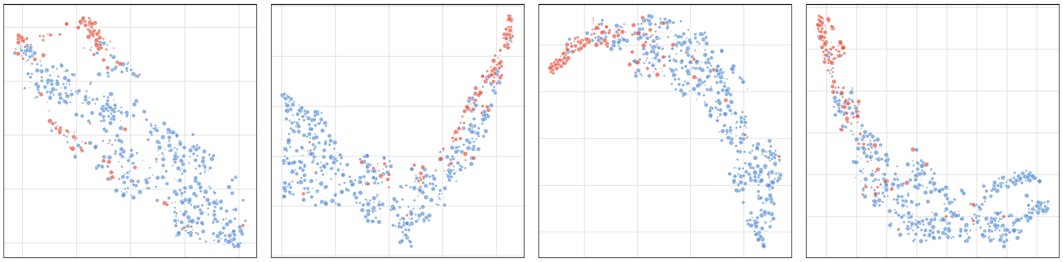
(a) Imratio **0.02** (b) Imratio **0.03** (c) Imratio **0.06**

Figure 6: Overall Performance of ResNet20 Across Perturbation Levels on Tiny-ImageNet. This graph illustrates the performance of various methods at different corruption levels, with Level 0 indicating no corruption and Level 5 representing the most severe corruption. In each figure, the seven lines depict the test AUC for CE, AUCMLoss, FocalLoss, ADVShift, WDRO, DROLT, GLOT, AUCDRO, DRAUC-Da and DRAUC-Df, respectively. Best viewed in colors.



(a) Imratio **0.02** (b) Imratio **0.03** (c) Imratio **0.06**

Figure 7: Overall Performance of ResNet32 Across Perturbation Levels on Tiny-ImageNet. This graph illustrates the performance of various methods at different corruption levels, with Level 0 indicating no corruption and Level 5 representing the most severe corruption. In each figure, the seven lines depict the test AUC for CE, AUCMLoss, FocalLoss, ADVShift, WDRO, DROLT, GLOT, AUCDRO, DRAUC-Da and DRAUC-Df, respectively. Best viewed in colors.



(a) **CE** (b) **AUCMLoss** (c) **DRAUC-Df(ours)** (d) **DRAUC-Da(ours)**

Figure 8: t-SNE plots of model embeddings on CIFAR10-C.

- Our methods consistently outperform all competitors on the corrupted datasets, across varying imbalance ratios and model architectures, confirming the effectiveness of our proposed method.
- Our methods achieve superior performance under stronger perturbations, thereby substantiating that our proposed methods enhance model robustness. This inference can be considered an ablation result.
- In most cases, distribution-aware contributes to improving model robustness.

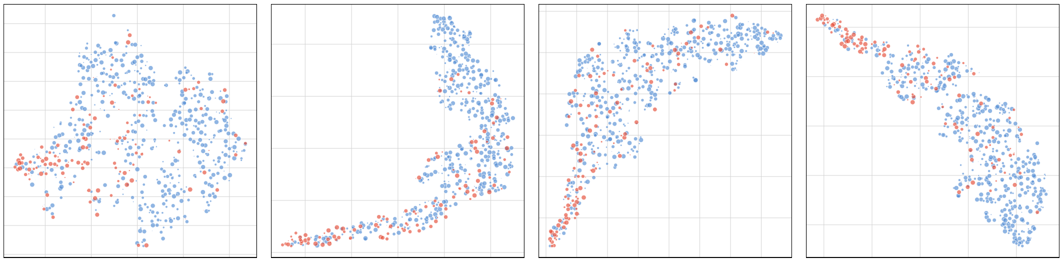
B.7 Visualizations

In this section, we provide more visualization results.

t-SNE Plots. We display the t-SNE plots for CIFAR10-C, CIFAR100-C and MNIST-C in Figures 8, 9 and 10. As evident from the plots, the embeddings on CIFAR100-C are more challenging to separate than those on CIFAR10-C and MNIST-C. This outcome primarily due to two factors: **a)** The number of patterns in CIFAR100 exceeds those in CIFAR10 and MNIST. **b)** When we create our binary version datasets, we designate the first half of classes as positive and the remaining half as negative. Consequently, the positive class of CIFAR100 comprises 50 original classes, making it more complex to learn, and we should anticipate a larger inner-class variance of its embeddings.

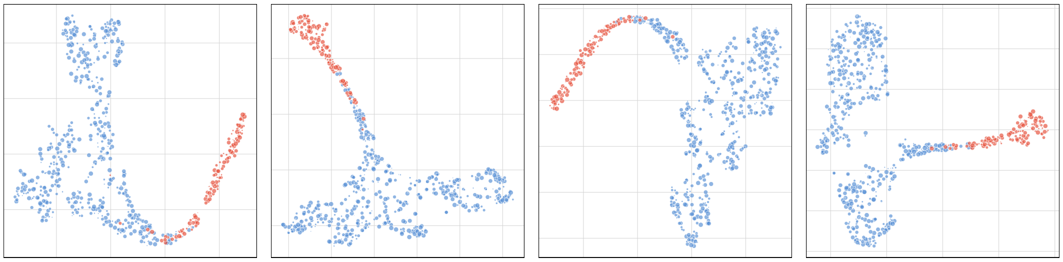
However, as demonstrated by the plots, our proposed method offers a more separable embedding space compared to the baselines.

An Interpretation of DRAUC's Improvement on Model Generalization for Corrupted Data. We provide several examples generated by our method in Figure 11. The results demonstrate that even without prior knowledge of the corruptions in the testing distribution, our DRAUC method generates adversarial examples closely resembling the test corruptions, thereby enhancing the model's resistance to them.



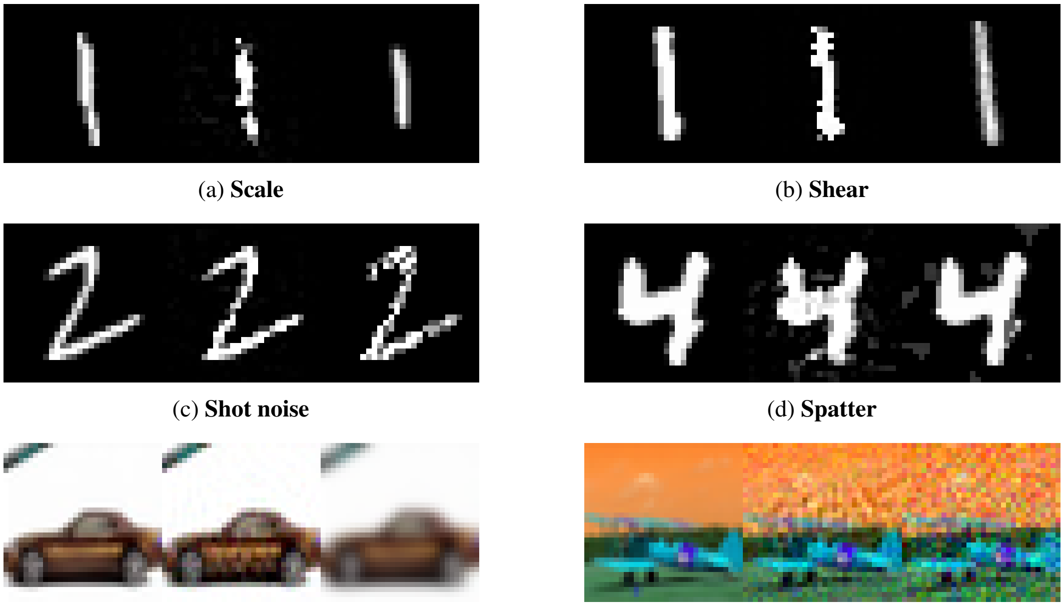
(a) **CE** (b) **AUCMLoss** (c) **DRAUC-Df(ours)** (d) **DRAUC-Da(ours)**

Figure 9: t-SNE plots of model embeddings on CIFAR100-C.



(a) **CE** (b) **AUCMLoss** (c) **DRAUC-Df(ours)** (d) **DRAUC-Da(ours)**

Figure 10: t-SNE plots of model embeddings on MNIST-C.



(a) **Scale** (b) **Shear**

(c) **Shot noise** (d) **Spatter**

(e) **Gaussian blur** (f) **Gaussian noise**

Figure 11: **Visualizations of adversarial examples generated by DRAUC-Df.** Each group of images represent original image, adversarial image generated by DRAUC-Df and the corrupted image in the testing set, respectively.