Class-Attribute Priors: Adapting Optimization to Heterogeneity and Fairness Objective

Xuechen Zhang[1], Mingchen Li[2], Jiasi Chen[2], Christos Thrampoulidis[3], Samet Oymak[2]

[1]University of California, Riverside
[2]University of Michigan, Ann Arbor
[3]University of British Columbia
xzhan394@ucr.edu, milii@umich.edu, jiasi@umich.edu, cthrampo@ece.ubc.ca, oymak@umich.edu

Abstract

Modern classification problems exhibit heterogeneities across individual classes: Each class may have unique attributes, such as sample size, label quality, or predictability (easy vs difficult), and variable importance at test-time. Without care, these heterogeneities impede the learning process, most notably, when optimizing fairness objectives. Confirming this, under a gaussian mixture setting, we show that the optimal SVM classifier for balanced accuracy needs to be adaptive to the class attributes. This motivates us to propose `CAP`: An effective and general method that generates a class-specific learning strategy (e.g. hyperparameter) based on the attributes of that class. This way, optimization process better adapts to heterogeneities. `CAP` leads to substantial improvements over the naive approach of assigning separate hyperparameters to each class. We instantiate `CAP` for loss function design and post-hoc logit adjustment, with emphasis on label-imbalanced problems. We show that `CAP` is competitive with prior art and its flexibility unlocks clear benefits for fairness objectives beyond balanced accuracy. Finally, we evaluate `CAP` on problems with label noise as well as weighted test objectives to showcase how `CAP` can jointly adapt to different heterogeneities.

1 Introduction

Contemporary machine learning problems arising in natural language processing and computer vision often involve large number of classes to predict. Collecting high-quality training datasets for all of these classes is not always possible, and realistic datasets (Menon et al. 2020; Feldman 2020; Hardt, Price, and Srebro 2016) suffer from class-imbalances, missing or noisy labels (among other application-specific considerations). Optimizing desired accuracy objectives with such heterogeneities poses a significant challenge and motivates the contemporary research on imbalanced classification, fairness, and weak-supervision. Additionally, besides distributional heterogeneities, we might have objective heterogeneity. For instance, the target test accuracy may be a particular weighted combination of individual classes, where important classes are upweighted.

A plausible approach to address these distributional and objective heterogeneities is designing optimization strategies that are tailored to individual classes. A classical example is assigning individual weights to classes during optimization. The recent proposals on imbalanced classification (Li et al. 2021; Chawla et al. 2002) can be viewed as generalization of weighting and can be interpreted as developing unique loss functions for individual classes. More generally, one can use class-specific data augmentation schemes, regularization or even optimizers (e.g. Adam, SGD, etc) to improve target test objective. While promising, this approach suffers when there are a large number of classes K: naively learning class-specific strategies would require $\mathcal{O}(K)$ hyperparameters ($\mathcal{O}(1)$ strategy hyperparameter per class). This not only creates computational bottlenecks but also raises concerns of overfitting for tail classes with small sample size.

To overcome such bottlenecks, we introduce the **Class-attribute Priors (`CAP`)** approach. Rather than treating hyperparameters as free variables, `CAP` is a meta-approach that treats them as a function of the class attributes. As we discuss later, example attributes $\mathcal{A}$ of a class include its frequency, label-noise level, training difficulty, similarity to other classes, test-time importance, and more. Our primary goal with `CAP` is building an **attribute-to-hyperparameter** function **A2H** that generates class-specific hyperparameters based on the attributes associated with that class. This process infuses high-level information about the dataset to accelerate the design of class-specific strategies. The **A2H** maps the attributes $\mathcal{A}$ to a class-specific strategy $\mathcal{S}$. The primary advantage is robustness and sample efficiency of **A2H**, as it requires $\mathcal{O}(1)$ hyperparameters to generate $\mathcal{O}(K)$ strategies. The main contribution of this work is proposing `CAP` framework and instantiating it for loss function design and post-hoc optimization which reveals its empirical benefits. Specifically, we make the following contributions:

1. **Introducing Class-attribute Priors (Sec 3).** We first provide theoretical evidence on the benefits of using multiple attributes (see Fig 2). This motivates `CAP`: A meta approach that utilizes the high-level attributes of individual classes to personalize the optimization process. Importantly, `CAP` is particularly favorable to tail classes which contain too few examples to optimize individually.
2. **`CAP` improves existing approaches (Sec 4).** By integrating `CAP` within existing label-imbalanced training methods, `CAP` not only improves their performance but also increases their stability, notably, AutoBalance (Li et al. 2021) and logit-adjustment loss (Menon et al. 2020).

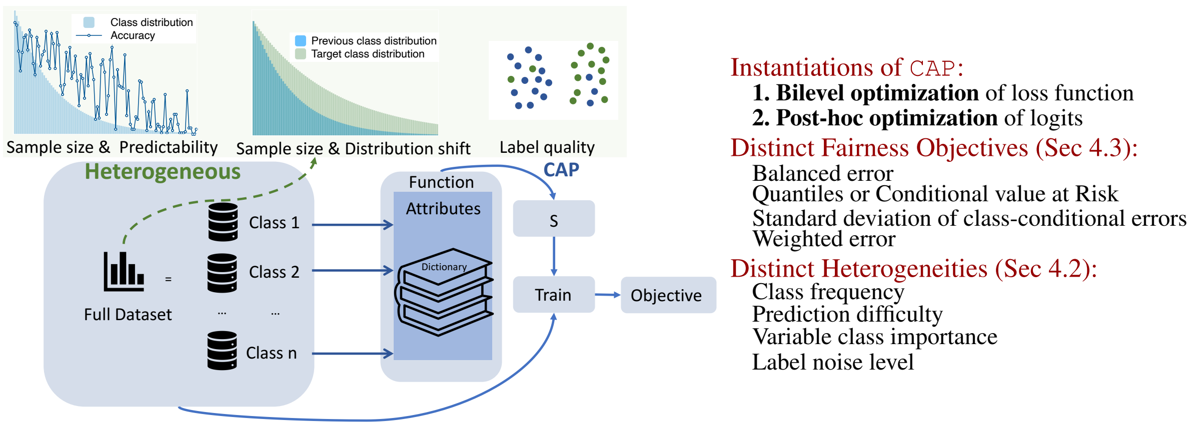


Figure 1: Left hand side: CAP views the global dataset as a composition of heterogeneous sub-datasets induced by classes. We extract high-level attributes from these classes and use these attributes to generate class-specific optimization strategies (which correspond to hyperparameters). Our proposal is efficiently generating these hyperparameters based on class-attributes through a meta-strategy. Right hand side: We demonstrate that CAP leads to state-of-the-art strategies for loss function design and post-hoc optimization. CAP can leverage multiple attributes to flexibly optimize a variety of test objectives under heterogeneities.

3. **CAP adapts to fairness objective (Sec 4.2).** CAP's flexibility is particularly powerful for non-standard settings that prior works do not account for: CAP achieves significant improvement when optimizing fairness objectives other than balanced accuracy, such as standard deviation, quantile errors, or Conditional Value at Risk (CVaR).
4. **CAP adapts to class heterogeneities (Sec 4.3).** CAP can also effortlessly combine multiple attributes (such as frequency, noise, class importance) to boost accuracy by adapting to problem heterogeneity.

Finally, while we instantiate CAP for the problems of loss-function design and post-hoc optimization, CAP can be applied for the design of class-specific augmentations, regularization schemes, and optimizers. This work makes key contributions to fairness and heterogeneous learning problems in terms of methodology, as well as practical impact. An overview of our approach is shown in Fig. 1

1.1 Related Work

The existing literature establishes a series of algorithms, including sample weighting (Alshammari et al. 2022; Maldonado et al. 2022; Kubat, Matwin et al. 1997; Wallace et al. 2011; Chawla et al. 2002), post-hoc tuning (Menon et al. 2020; Zhang et al. 2019; Kim and Kim 2020; Kang et al. 2019; Ye et al. 2020), and loss functions tuning (Cao et al. 2019; Kini et al. 2021; Menon et al. 2020; Khan et al. 2017; Cui et al. 2019; Li et al. 2022; Tan et al. 2020; Zhang et al. 2017), and more (Zhang et al. 2023). This work aims to establish a principled approach for designing a loss function for imbalanced datasets. Traditionally, a Bayes-consistent loss function such as weighted cross-entropy (Xie et al. 2018; Morik, Brockhausen, and Joachims 1999) has been used. However, recent work shows it only adds marginal benefit to the over-parameterized model due to overfitting during training. (Menon et al. 2020; Ye et al. 2020; Kini et al. 2021) propose a family of loss functions formulated as

$$\ell(y, f(\boldsymbol{x})) = \log\left(1+\sum_{k\neq y} e^{\boldsymbol{l}_k-\boldsymbol{l}_y} \cdot e^{\boldsymbol{\Delta}_k f_k(\boldsymbol{x})-\boldsymbol{\Delta}_y f_y(\boldsymbol{x})}\right)$$

with theoretical insights, where $f(\boldsymbol{x})$ denotes the output logits of $\boldsymbol{x}$ and $f_y(\boldsymbol{x})$ represents the entry that corresponds to label y. Above methods determine the value of $\boldsymbol{l}$ and $\boldsymbol{\Delta}$ to reweight the loss function so the optimization generates a class-balanced model. In addition to these methods, (Li et al. 2021) proposes a bilevel training scheme that directly optimizes $\boldsymbol{l}$ and $\boldsymbol{\Delta}$ on a sufficient small imbalanced validation data without the prior theoretical insights. However, the theory-based methods require expertise and trial and error to tune one temperature variable, making it time-consuming and challenging to achieve a fine-grained loss function that carefully handles each class individually. Although the bilevel-based method consider each class separately and personalizes the weight using validation data, optimizing the bilevel problem is typically time-consuming due to the Hessian computations. Bilevel optimization is also brittle, especially when (Li et al. 2021) optimizes the inner loss function, which continually changes the inner optima during the training.

Regarding the broader goal of fairness with respect to protected groups, the literature contains several proposals (Sahu et al. 2018; Kleinberg, Mullainathan, and Raghavan 2016). Balanced error and standard deviation (Calmon et al. 2017; Alabi, Immorlica, and Kalai 2018) between subgroup predictions are widely used metrics. However, they may be insensitive to certain types of imbalances. The Difference of Equal Opportunity (DEO) (Kini et al. 2021; Hardt, Price, and Srebro 2016) was proposed to measure true positive rates across groups. (Zafar et al. 2017) focus on disparate mistreatment in both false positive rate and false negative rate. Many modern ML tasks necessitate models with good tail performance, focusing on underrepresented groups. Recent works have introduced techniques that promote better tail accuracy (Hashimoto et al. 2018; Sagawa et al. 2019, 2020; Li et al. 2021; Kini and Thrampoulidis 2020). The worst-case subgroup error is commonly used in recent papers (Kini et al. 2021; Sagawa et al. 2019, 2020). Another popular metric to evaluate the model's tail performance is the CVaR (Conditional Value at Risk) (Williamson and Menon 2019; Zhai et al. 2021; Michel, Hashimoto, and Neubig 2021), which computes the average error over the tails. Previous works (Hashimoto et al. 2018; Duchi and Namkoong 2021; Hu et al. 2018; Michel, Hashimoto, and Neubig 2021; Lahoti et al. 2020) also measure tail behaviour using Distributionally Robust Optimization (DRO).

2 Problem Setup

This paper investigates the advantages of utilizing attribute-based personalized training approaches for addressing heterogeneous classes in the context of class imbalance, label noise, and fairness objective problems. We begin by presenting the general framework, followed by an examination of specific fairness issues, which encompass both distributional and objective heterogeneities. Consider a multi-class classification problem for a dataset $(\boldsymbol{x}_i, y_i)_{i=1}^N$ sampled i.i.d from a distribution with input space $\mathcal{X}$ and K classes. Let $[K]$ denote the set $\{1..K\}$ and for the training sample $(\boldsymbol{x}, y)$, $\boldsymbol{x} \in \mathcal{X}$ is the input and $y \in [K]$ is the output. $f : \mathcal{X} \to \mathbb{R}^K$ represents the model and $\boldsymbol{o}$ is the output logits. $\hat{y}_{f(\boldsymbol{x})} = \arg\max_{k \in [K]} \boldsymbol{o}_k$ is the predicted label of the model $f(\boldsymbol{x})$. We also denote $K \times K$ identity matrix by $\boldsymbol{I}_K$. Moreover, in the post-hoc setup, a logit adjustment function $g : \mathbb{R}^K \to \mathbb{R}^K$ is employed to modify the logits, resulting in adjusted logits $\hat{\boldsymbol{o}} = g(\boldsymbol{o})$.

Our goal is to train a model that minimizes a specific classification error metric. The class-conditional errors are defined over the data distribution as $\text{Err}_k = \mathbb{P}\left[y \neq \hat{y}_f(\boldsymbol{x}) \mid y = k\right]$. The standard classification error is denoted by $\text{Err}_{\text{plain}} = \mathbb{P}\left[y \neq \hat{y}_f(\boldsymbol{x})\right]$. In situations with label imbalance, $\text{Err}_{\text{plain}}$ is dominated by the majority classes. To this end, balanced classification error $\text{Err}_{\text{bal}} = \frac{1}{K}\sum_{k=1}^K \text{Err}_k$ is widely employed as a fairness metric. We will later introduce other objectives that aim to achieve different fairness goals. A complete list of the objectives we examine can be found in Appendix.

3 Our Approach: Class-attribute Priors (`CAP`)

We start with a motivating question:

Q: *Does utilizing multiple class attributes provably help?*

To answer this, we consider the benefits of multiple attributes for a binary gaussian mixture model (GMM) and provide a simple theoretical justification why synergistically leveraging attributes can help balanced accuracy. Consider a GMM where data from the two classes are generated as

$$y = \begin{cases} +1 & \text{, with prob. } \pi \\ -1 & \text{, with prob. } 1-\pi \end{cases} \quad \text{and} \quad \mathbf{x}|y \sim \mathcal{N}(y\boldsymbol{\mu}, \sigma_y \mathbf{I}_d).$$

Note here that the two classes are **imbalanced** as a function of the value of $\pi \in (0, 1)$, which models class frequency. Also, the two classes are allowed to have different noise variances $\sigma_{\pm 1}$. This is our model for the **difficulty** attribute: examples generated from the class with larger variance are "more difficult" to classify as they fall further apart from their mean. Intuitively, a "good" classifier should account for both attributes. We show here that this is indeed the case for the model above. Our setting is as follows: Draw n IID samples $(\mathbf{x}_i, y_i)$ from the GMM distribution above. Without loss of generality, assume class $y = +1$ is minority, i.e. $\pi < 1/2$. We train linear classifier $(\mathbf{w}, b)$ by solving the following cost-sensitive support-vector-machines (CS-SVM) problem:

$$(\hat{\mathbf{w}}_\delta, \hat{b}_\delta) := \arg\min_{\mathbf{w}, b} \|\mathbf{w}\|_2 \text{ s.t. } y_i(\mathbf{x}_i^T\mathbf{w} + b) \geq \begin{cases} \delta & y_i = +1 \\ 1 & y_i = -1 \end{cases}.$$

Here, δ is a hyperparameter that when taking values larger than one, it pushes the classifier towards the majority, thus assigning larger margin to the minorities. In particular, setting $\delta = 1$ recovers the vanilla SVM. CS-SVM is particularly relevant to our setting because it has rigorous connections to the generalized cross-entropy loss in Sec 3.2 (see Appendix F for details). Given CS-SVM solution $(\hat{\mathbf{w}}_\delta, \hat{b}_\delta)$, we measure the balanced error as follows:

$$\mathcal{R}_{\text{bal}}(\delta) := \mathbb{P}_{(\mathbf{x},y)\sim\text{GMM}}\left\{y(\mathbf{x}^T\hat{\mathbf{w}}_\delta + \hat{b}_\delta) > 0\right\}.$$

We ask: How does the optimal CS-SVM classifier (i.e, the optimal hyperparameter δ) depend on the data attributes, i.e. on the frequency π and on the difficulty σ_{+1}/σ_{-1}? To answer this we consider a high-dimensional asymptotic setting in which $n, d \to \infty$ at a linear rate $d/n =: \bar{d}$. This regime is convenient as previous work has shown that the limiting behavior of the balanced error $\mathcal{R}_{\text{bal}}(\delta)$ can be captured precisely by analytic formulas (Montanari et al. 2019). Specifically, (Kini et al. 2021) computes formulas for the optimal hyperparameter δ when π is variable but both classes are equally difficult, i.e. $\sigma_{+1} = \sigma_{-1}$. Here, we derive risk curves for arbitrary $\sigma_{\pm 1}$ by extending their study and investigate the synergistic effect of frequency and difficulty.

Figure 2 confirms our intuition: **the optimal hyperparameter δ_* (y-axis) does indeed depend on the frequency and difficulty** (as well as the training sample size, x-axis). Specifically, we observe in both Figures 2(a,b) that as the minority class becomes easier (aka, the smaller ratio σ_{+1}/σ_{-1}), δ decreases. That is, there is less need to assign an even larger margin for the minority. Conversely, as σ_{+1}/σ_{-1} increases and minority becomes more difficult, its margin gets a boost. Finally, comparing Figures 2(a) to 2(b), note that δ_* takes larger values for larger imbalance ratio (i.e., smaller frequency π), again aggreeing with commonsense.

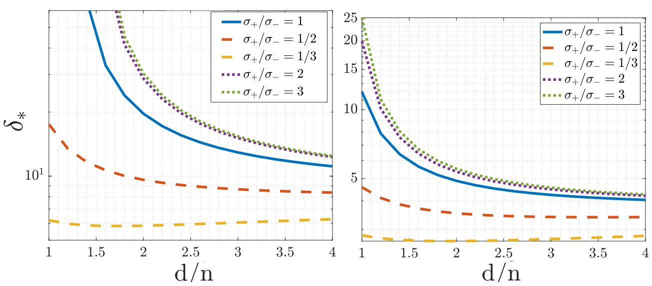
Since GMM data is synthetic, δ_* can be computed analytically. Our approach `CAP` will facilitate realizing such benefits systematically and efficiently for arbitrary class attributes.

3.1 Attributes and Adaptation to Heterogeneity

To proceed, we introduce our `CAP` approach at a conceptual level and provide concrete applications of `CAP` to loss function design in the next section. Recall that our high-level goal is designing a map from **A2H** that takes attributes $\mathcal{A}_k$ of class k and generates the hyperparameters of the optimization strategy $\mathcal{S}_k$. Each coordinate $\mathcal{A}_k[i]$ characterizes a specific attribute of class k such as label frequency, label noise ratio, training difficulty shown in Table 1. To model **A2H**, one can use any hypothesis space including deep nets. However, since **A2H** will be optimized over the validation loss, depending on the application scenario, it is often preferable to use a simpler linearized model.

Attributes	Definition	Notation	Application scenario
$\mathcal{A}_{\text{FREQ}}\star$	Class frequency	$\pi_k = \mathbb{P}(y = k)$	Imbalanced classes
$\mathcal{A}_{\text{DIFF}}\star$	Class-conditional error	$\mathbb{P}(y \neq \hat{y})$	Difficult vs easy classes
$\mathcal{A}_{\text{WEIGHTS}}\star$	Test-time class weights	ω_k^{test} of (3)	Weighted test accuracy
$\mathcal{A}_{\text{NOISE}}$	Label noise ratio	$\mathbb{P}(y^{\text{CLEAN}} \neq y \mid y^{\text{clean}} = k)$	Datasets with label noise
$\mathcal{A}_{\text{NORM}}$	Norm of classifier weights	See (Cao et al. 2019)	Imbalanced classes

Table 1: Definition of example attributes and associated application scenarios. Attributes $\mathcal{A}_{\text{DIFF}}$ and $\mathcal{A}_{\text{NORM}}$ are computed during the training (for post-hoc optimization, it is pre-training). For bilevel training they are computed at the end of warm-up. The upper attributes with $\star$ are those we utilize in our experiments. Also we use $\mathcal{A}_{\text{ALL}}$ to denote combined attributes.



(a) Imbalance $\pi = 0.1$ (b) Imbalance $\pi = 0.2$

Figure 2: The optimal hyperparameter δ_* depends on both attributes: frequency (π) and difficulty (σ_+/σ_-).

Linearized approach. Suppose each class has n attributes with $\mathcal{A}_k \in \mathbb{R}^n$. We will use a nonlinear feature map $\boldsymbol{\mathcal{F}}(\cdot) : \mathbb{R}^n \to \mathbb{R}^M$ where M is the embedding space. Suppose the class-specific strategy $\mathcal{S}_k \in \mathbb{R}^s$. Then, **A2H** can be parameterized by a weight matrix $\boldsymbol{W} \in \mathbb{R}^{s\times M}$ so that

$$\mathcal{S}_k = \textbf{A2H}(\mathcal{A}_k) := \boldsymbol{W}\boldsymbol{\mathcal{F}}(\mathcal{A}_k). \tag{1}$$

Our goal becomes finding $\boldsymbol{W}$ so that the resulting strategies maximize the target validation objective. Observe that $\boldsymbol{W}$ has $s \times M$ parameters rather than $s \times K$ parameters which is the naive approach that learns individual strategies. In practice, K can be significantly large, so for typical problems, $M \ll K$. Moreover, $\boldsymbol{W}$ ties all classes together during training through weight-sharing whereas the naive approach would be brittle for tail classes that contain very limited data.

3.2 CAP for Loss Function Design

Consider the generalized cross-entropy loss

$$\ell(y, f(\boldsymbol{x})) = \omega_y \log(1 + \sum_{k\neq y} e^{\boldsymbol{l}_k - \boldsymbol{l}_y} \cdot e^{\boldsymbol{\Delta}_k f_k(\boldsymbol{x}) - \boldsymbol{\Delta}_y f_y(\boldsymbol{x})}).$$

Here, $(\omega_k, \boldsymbol{l}_k, \boldsymbol{\Delta}_k)_{k=1}^K$ are hyperparameters that can be tuned to optimize the desired test objective. For class k, we get to choose the tuple $\mathcal{S}_k := [\omega_k, \boldsymbol{l}_k, \boldsymbol{\Delta}_k]$ which can be considered as its training strategy. Here elements of $\mathcal{S}_k$ arise from existing imbalance-aware strategies, namely weighting ω_k, additive logit-adjustment $\boldsymbol{l}_k$ and multiplicative adjustment $\boldsymbol{\Delta}_k$.

Example: LA and CDT losses viewed as CAP. For label imbalanced problems, (Menon et al. 2020; Ye et al. 2020) propose to set hyperparameters $\boldsymbol{l}_k$ and $\boldsymbol{\Delta}_k$ as a function of frequency $\pi_k = \mathbb{P}(y = k)$. Concretely, they propose $\boldsymbol{l}_k = -\gamma\log(\pi_k)$ (Menon et al. 2020) and $\boldsymbol{\Delta}_k = \pi_k^\gamma$ (Ye et al. 2020) for some scalar γ. These can be viewed as special instances of CAP where we have a single attribute $\mathcal{A}_k = \pi_k$ and **A2H**(x) is $-\gamma\log(x)$ or x^γ respectively.

Our approach can be viewed as an extension of these to attributes beyond frequency and general class of **A2H**. In light of (1), hyperparameters of a specific element of $\mathcal{S}_k = [\omega_k, \boldsymbol{l}_k, \boldsymbol{\Delta}_k]$ correspond to a particular row of $\boldsymbol{W} \in \mathbb{R}^{3\times M}$ since $\boldsymbol{W} = [\boldsymbol{w}_{\boldsymbol{\omega}}, \boldsymbol{w}_{\boldsymbol{l}}, \boldsymbol{w}_{\boldsymbol{\Delta}}]^\top$. Our goal is then tuning the $\boldsymbol{W}$ matrix over validation data. In practical implementation, we define a feature dictionary

$$\boldsymbol{\mathcal{D}} = [\boldsymbol{\mathcal{F}}(\mathcal{A}_1) \ \cdots \ \boldsymbol{\mathcal{F}}(\mathcal{A}_K)]^\top \in \mathbb{R}^{K\times M}. \tag{2}$$

Each row of this dictionary is the features associated to the attributes of class k. We generate the strategy vectors $\boldsymbol{\Delta}, \boldsymbol{l}, \boldsymbol{\omega} \in \mathbb{R}^K$ (for all classes) via $\boldsymbol{\omega} = \boldsymbol{\mathcal{D}}\boldsymbol{w}_{\boldsymbol{\omega}}$, $\boldsymbol{\Delta} = \text{sigmoid}(\sqrt{K}\frac{\boldsymbol{\mathcal{D}}\boldsymbol{w}_{\boldsymbol{\Delta}}}{\|\boldsymbol{\mathcal{D}}\boldsymbol{w}_{\boldsymbol{\Delta}}\|})$, $\boldsymbol{l} = \boldsymbol{\mathcal{D}}\boldsymbol{w}_{\boldsymbol{l}}$.

For both loss function design and post-hoc optimization, we use a decomposable feature map $\boldsymbol{\mathcal{F}}$. Concretely, suppose we have basis functions $(\mathcal{F}_i)_{i=1}^m$. These functions are chosen to be poly-logarithms or polynomials inspired by (Menon et al. 2020; Ye et al. 2020). For ith attribute $\mathcal{A}_k[i] \in \mathbb{R}$, we generate $\boldsymbol{\mathcal{F}}(\mathcal{A}_k[i]) \in \mathbb{R}^m$ obtained by applying $(\mathcal{F}_i)_{i=1}^m$. We then stitch them together to obtain the overall feature vector $\boldsymbol{\mathcal{F}}(\mathcal{A}_k) = [\boldsymbol{\mathcal{F}}(\mathcal{A}_k[1])^\top \ \cdots \ \boldsymbol{\mathcal{F}}(\mathcal{A}_k[m])^\top] \in \mathbb{R}^{M:=m\times n}$. We emphasize that prior approaches are special instances where we choose a single basis function and single attribute π_k.

Which attributes to use and why multiple attributes help? Attributes should be chosen to reflect the heterogeneity across individual classes. These include class frequency, difficulty of prediction, noise level and more. We list such potential attributes $\mathcal{A}$ in Table 1. The frequency $\mathcal{A}_{\text{FREQ}}$ is widely used to mitigate label imbalance, and $\mathcal{A}_{\text{NORM}}$ is inspired by the imbalanced learning literature (Cao et al. 2019). However, these may not fully capture the heterogenous nature of the problem. As discussed above for GMMs (Fig 2), some classes can be more difficult to learn and require more upweighting despite containing sufficient training examples. This motivates the use of $\mathcal{A}_{\text{DIFF}}$. Moreover, rather than balanced accuracy, we may wish to optimize general test objectives including weighted accuracy with varying class importance. We can declare these test-time weights as an attribute $\mathcal{A}_{\text{WEIGHTS}}$. Appendix provides theoretical justification for incorporating $\mathcal{A}_{\text{WEIGHTS}}$ by showing CAP can accomplish Bayes optimal logit adjustment for weighted error. More broadly, any class-specific meta-feature can be used as an attribute within CAP.

Reduced search space and increased stability. Searching $\boldsymbol{l}$ and $\boldsymbol{\Delta}$ on $\mathbb{R}^K$ with very few validation samples raises the problem of unstable optimization. (Li et al. 2021) indicates the bilevel optimization is brittle and hard to optimize. They introduce a long warm-up phase and aggregate classes with similar frequency into g groups, reducing the search space to k/g dimensions. However, to achieve a fine-grained loss function, g cannot be very large, so the search space remains large. In our method, with a good design of $\mathcal{D}$ (normally $n \approx 2$ and $m \approx 3$), we can utilize a constant $2mn \ll K$ that efficiently reduces the search space and provides better convergence and stability.

We remark that dictionary is a general and efficient design that can recover multiple existing successful imbalanced loss function design algorithms. For example, (Menon et al. 2020) and (Ye et al. 2020) both utilize the frequency as $\mathcal{A}$ and apply logarithm and polynomial functions as $\mathcal{F}$ on frequency to determine $\boldsymbol{l}$ and $\boldsymbol{\Delta}$ respectively. Moreover, let $\mathcal{A} = \boldsymbol{I}_k$ and $\mathcal{F}$ be an identity function, then training $\boldsymbol{w_l}, \boldsymbol{w_\Delta}$ is equivalent to train $\boldsymbol{l}, \boldsymbol{\Delta}$ which recovers the algorithm of (Li et al. 2021). Despite the ability to generalize, the dictionary is more flexible and powerful since the attributes can be chosen based on the scenarios. For example, naturally, class frequency is a critical criterion in an imbalanced dataset, but classification error in early training can also be a good criterion for evaluating class training difficulty. Furthermore, some specific attributes can be introduced to noisy or partial-labeled datasets to help design a better loss function. Our empirical study elucidates the benefit of combining multiple attributes and the dictionary performance on the noisy imbalanced dataset.

3.3 Class-specific Learning Strategies: Bilevel Optimization and Post-hoc optimization

To instantiate CAP as a meta-strategy, we focus on two important class-specific optimization problems: loss function design via bilevel optimization and post-hoc logit adjustment. We describe them in this section and demonstrate that both methods outperform the state-of-the-art approaches. The figure in Appendix illustrates how CAP is implemented under bi-level optimization and post-hoc optimization in detail.

• Strategy 1: Loss function design via bilevel optimization. Inspired by (Li et al. 2021) and following our exposition in Section 3.1, we formalize the meta-strategy optimization problem as

$$\min_{\boldsymbol{w_l}, \boldsymbol{w_\Delta}} \mathcal{L}_{\text{val}}(\boldsymbol{w_l}, \boldsymbol{w_\Delta}, f) \quad \text{s.t.} \quad \min_{f} \mathcal{L}_{\text{train}}(\boldsymbol{w_l}, \boldsymbol{w_\Delta}, f)$$

where f is the model and $\mathcal{L}_{\text{val}}$, $\mathcal{L}_{\text{train}}$ are validation and training losses respectively. Our goal is finding CAP parameters $\boldsymbol{w_l}, \boldsymbol{w_\Delta}$ that minimize the validation loss which is the target fairness objective. Following the implementation of (Li et al. 2021), we split the training data to 80% training and 20% validation to optimize $\mathcal{L}_{\text{train}}$ and $\mathcal{L}_{\text{val}}$. The optimization process is split to two phases: the search phase that finds CAP parameters $\boldsymbol{w_l}, \boldsymbol{w_\Delta}$ and the retraining phase that uses the outcome of search and entire training data to retrain the model. We note that, during initial search phase, (Li et al. 2021) employs a long *warm-up* phase where they only train f while fixing $\boldsymbol{w_l}, \boldsymbol{w_\Delta}$ to achieve better stability. In contrast, we find that CAP either needs very short warm-up or no warm-up at all pointing to its inherent stability.

• Strategy 2: Post-hoc optimization. In (Menon et al. 2020; Feldman et al. 2015; Hardt, Price, and Srebro 2016), the author displays that the post-hoc logit adjustment can efficiently address the bias when training with imbalanced datasets. Formally, given a model f, a post-hoc function $g : \mathbb{R}^K \to \mathbb{R}^K$ adjusts the output of f to minimize the fairness objective. Thus the final model of post-hoc optimization is $g \circ f(\boldsymbol{x})$.

4 Experiments and Main Results

In this section, we present our experiments in the following way. Firstly, we demonstrate the performance of CAP on both loss function design via bilevel optimization and post-hoc logit adjustment in Sec. 4.1. Sec. 4.2 demonstrates that CAP provides noticeable improvements for fairness objectives beyond balanced accuracy. Then Sec. 4.3 discusses the advantage of utilizing attributes and how CAP leverages them in noisy, long-tailed datasets through perturbation experiments. Lastly, we defer the experiment details including hyper-parameters, number of trails, and other reproducibility information to appendix.

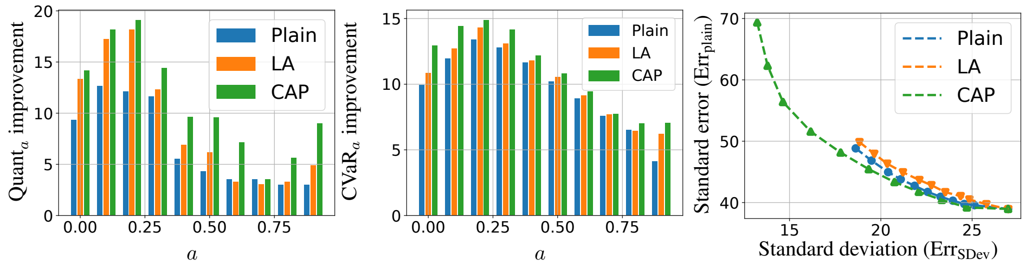
Dataset. In line with previous research (Menon et al. 2020; Ye et al. 2020; Li et al. 2021), we conduct the experiments on CIFAR-LT and ImageNet-LT datasets. The CIFAR-LT modifies the original CIFAR10 or CIFAR100 by reducing the number of samples in tail classes. The imbalance factor, represented as $\rho = N_{max}/N_{min}$, is determined by the number of samples in the largest (N_{max}) and smallest (N_{min}) classes. To create a dataset with the imbalance factor, the sample size decreases exponentially from the first to the last class. We use $\rho = 100$ in all experiments, consistent with previous literature. The ImageNet-LT, a long-tail version of ImageNet, has 1000 classes with an imbalanced ratio of $\rho = 256$. The maximum and minimum samples per class are 1280 and 5, respectively. During the search phase for bilevel CAP, we split the training set into 80% training and 20% validation to obtain the optimal loss function design. We remark that the validation set is imbalanced, with tail classes containing very few samples, making it challenging to find optimal hyper-parameters without overfitting. For all other post-hoc experiments (Sec. 4.2 and 4.3), we follow the setup of (Menon et al. 2020; Hardt, Price, and Srebro 2016) by training a model on entire training dataset as the pre-train model, and optimizing a logit adjustment g on a balanced validation dataset. Additionally, all CIFAR-LT experiments use ResNet-32 (He et al. 2016), and ImageNet-LT experiments use ResNet-50.

4.1 CAP Improves Prior Methods Using Post-hoc or Bilevel Optimization

This section presents our loss function design experiments on imbalanced datasets by incorporating CAP into the training scheme of (Li et al. 2021; Menon et al. 2020; Ye et al. 2020),

Method	CIFAR10-LT	CIFAR100-LT	ImageNet-LT
Cross entropy	30.45(±0.49)	61.94(±0.28)	55.59(±0.26)
Logit adjustment (LA)(Menon et al. 2020)	21.29†(±0.43)	58.21†(±0.31)	52.46♯
CDT(Ye et al. 2020)	21.57†(±0.50)	58.38†(±0.33)	53.47♯
$\text{Plain}_{\text{Bilevel}}$ (AutoBalance(Li et al. 2021))	21.15♯	56.70♯	50.91♯
$\texttt{CAP}_{\text{Bilevel}} : \mathcal{A}_{\text{ALL}}$	**20.22**(±0.35)	**56.38**(±0.19)	**49.31**(±0.34)
$\texttt{CAP}_{\text{Post-hoc}} : \mathcal{A}_{\text{ALL}}$	20.87(±0.38)	57.63(±0.26)	51.46(±0.20)

Table 2: Balanced error on long-tailed data using loss function designed via bilevel optimization. ♯: best reported results taken from (Li et al. 2021).†: Reproduced results.



(a) Errors of quantile classes (b) Conditional value of errors (c) Aggregation of errors

Figure 3: Benefit of CAP for optimizing different Fairness Objectives. We compare among plain post-hoc, LA post-hoc and $\texttt{CAP}_{\text{post-hoc}}$. (a): Results of optimizing quantile class performance $\text{Quant}_a = \mathbb{P}\left[y \neq \hat{y}_f(\boldsymbol{x}) \mid y = \text{K}_a\right]$, where K_a denotes the class index with the worst $\lceil \text{K} \times a \rceil$-th error. (b): Results of optimizing tail performance CVaR_a. (c): Results of optimizing $\mathcal{R}(\text{Err}) = \lambda \cdot \text{Err}_{\text{plain}} + (1-\lambda) \cdot \text{Err}_{\text{SDev}}$. The plot shows the trade-off between standard deviation of class-conditional errors Err_{SDev} and Standard misclassification error $\text{Err}_{\text{plain}}$ as λ varies. See Sec.4.2 for detailed definition and discussions.

as discussed in Sec. 3.3. Table 2 demonstrates our results. The first part displays the outcomes of various existing methods with their optimal hyper-parameters. It is worth noting that the original best results for single-level methods ((Menon et al. 2020; Ye et al. 2020)) are obtained from grid search on the test dataset, which leads to much better performance than our reproduced results using validation grid search in Table. 2. Moreover, both of the grid search methods demand substantial computation budgets. As illustrated in the second part of Table 2, bilevel and post-hoc CAP significantly improve the balanced error across all datasets.

4.2 Benefits of CAP for Optimizing Distinct Fairness Objectives

Recent works on label-imbalance places a significant emphasis on the balanced accuracy evaluations (Menon et al. 2020; Li et al. 2021; Cao et al. 2019). However, in practice, there are many different fairness criteria and balanced accuracy is only one of them. In fact, as we discuss in (3), we might even want to optimize arbitrary weighted test objectives. In this section, we demonstrate the flexibility and merits of CAP when optimizing fairness objectives other than balanced accuracy. The experiments are conducted on the CIFAR100-LT dataset using the post-hoc approaches. For the fairness objectives, we mainly focus on three objectives: quantile class error Quant_a, conditional value at risk (CVaR) CVaR_a, and the combined risk $\mathcal{R}(\text{Err})$, which consists of standard deviation of error and the regular classification error.

We first demonstrate the performance on quantile class error $\text{Quant}_a = \mathbb{P}\left[y \neq \hat{y}_f(\boldsymbol{x}) \mid y = \text{K}_a\right]$, where K_a denotes the class index with the worst $\lceil \text{K} \times a \rceil$-th error. For instance, in CIFAR100-LT, where $K = 100$, $\text{Quant}_{0.2}$ denotes the test error of the worst 20 percentile class. That is, we sort the classes in descending order of test error and return the error of the class 20%th class ID. Thus, each selection of a raises a new objective. Fig. 3a shows the improvement over the pre-trained model when optimizing Quant_a with multiple selections of a. We observe that CAP significantly outperforms both logit adjustment and plain post-hoc.

$\text{CVaR}_a = \mathbb{E}\left[\text{Err}_k \mid \text{Err}_k > \text{Quant}_a\right]$ measures the average error of $\lceil \text{K} \times a \rceil$ classes with worst errors. Instead of Quant_a, which only focuses on the specific quantile class error, optimizing the CVaR_a tend to improve the tail behavior of the classifier, which is a more general fairness objective. Fig. 3b shows the test improvements over three approaches, and CAP is consistently better than all other methods.

Finally, for the combined risk $\mathcal{R}(\text{Err})$, we define $\mathcal{R}(\text{Err}) = \lambda \cdot \text{Err}_{\text{plain}} + (1-\lambda) \cdot \text{Err}_{\text{SDev}}$ where $\text{Err}_{\text{plain}}$ is the regular classification error and Err_{SDev} denotes the standard deviation of classification errors. We plot the error-deviation curve by varying λ from 0 to 1 with stepsize 0.1 on three approaches in Fig. 3c, each point corresponds to a different λ. We observe that plain post-hoc cannot achieve a small standard deviation, and post-hoc LA degrades when achieving smaller Err_{SDev}, CAP accomplish the best performance and are flexible to adapt to different objectives.

Post-hoc methods	Err_{bal}	Err_{SDev}	$\text{CVaR}_{0.2}$	$\text{Quant}_{0.2}$	$\text{Err}_{\text{weighted}}$
Pretrained	61.94 (±0.28)	27.13(±0.35)	96.95(±0.15)	93.01(±0.58)	62.53(±0.53)
$\text{Plain}_{\text{Post-hoc}}$	-1.62(±0.36)	-8.51(±0.75)	-11.48(±0.81)	-12.79(±0.43)	-2.82(±0.56)
$\text{LA}_{\text{Post-hoc}}$	-3.73(±0.29)	-8.72(±0.66)	-12.21(±0.50)	-15.01(±0.35)	-3.62(±0.37)
$\texttt{CAP}_{\text{Post-hoc}}$	**-4.36**(±0.25)	**-13.92**(±0.24)	**-14.75**(±0.87)	**-18.34**(±0.47)	**-6.21**(±0.49)

Table 3: The error difference between other approaches compared to pre-trained model. The first line shows the performance of Pretrained model, and the following line shows the error difference of other methods (smaller is better). For objectives with a, we set $a = 0.2$. This is commonly used for difficult or few classes in other papers(Zhai et al. 2021; Liu et al. 2019).

	CIFAR100-LT		ImageNet-LT		CIFAR10-LT+Noise	
	Err_{bal}	Err_{SDev}	Err_{bal}	Err_{SDev}	Err_{bal}	Err_{SDev}
Cross entropy	61.94(±0.28)	27.13(±0.35)	55.59(±0.26)	29.10(±0.64)	43.76(±0.74)	31.69(±0.81)
$\text{Plain}_{\text{Bilevel}}$ (AutoBalance (Li et al. 2021))	56.70(±0.32)	20.13(±0.68)	50.93(±0.16)	26.06(±0.61)	40.04(±0.79)	36.30(±0.89)
$\texttt{CAP}_{\text{Bilevel}}$: $\mathcal{A}_{\text{FREQ}}$	56.64 (±0.21)	19.10(±0.67)	50.82(±0.13)	24.36(±0.49)	39.91(±0.66)	26.54(±0.80)
$\texttt{CAP}_{\text{Bilevel}}$: $\mathcal{A}_{\text{DIFF}}$	58.27(±0.24)	**17.62**(±0.65)	52.97(±0.30)	**21.28**(±0.58)	40.61(±0.61)	**14.49**(±0.72)
$\texttt{CAP}_{\text{Bilevel}}$: $\mathcal{A}_{\text{FREQ}}$+$\mathcal{A}_{\text{DIFF}}$	**56.38**(±0.19)	18.53(±0.63)	**49.31**(±0.34)	22.14(±0.46)	**38.36**(±0.79)	19.78(±0.75)

Table 4: Attributes help optimization adapt to dataset heterogeneity. We conduct experiments using bilevel loss design and report the balanced misclassification error,and standard deviation of class-conditional errors with different class-specific attributes.

When using plain post-hoc (without CAP), each class parameter is updated individually. Thus, optimizing for specific classes (e.g., Quant_a) dramatically hurts the performance of other classes which are ignored. Confirming this, we found that optimizing plain post-hoc is unstable and does not achieve decent results. On the other hand, while post-hoc LA outperforms plain post-hoc, optimizing only one temperature variable lacks fine-grained adaptation to various objectives. In contrast, CAP achieves a noticeably better performance on all objectives thanks to its best of both worlds design.

Table 3 shows more results. $\text{Err}_{\text{weighted}}$ denotes a weighted test objective induced by weights $\boldsymbol{\omega}_k^{\text{test}} \in \mathbb{R}^K$ given by

$$\text{Err}_{\text{weighted}} = \sum_{k=1}^{K} \boldsymbol{\omega}_k^{\text{test}} \text{Err}_k \quad \text{where} \quad \sum_{k=1}^{K} \boldsymbol{\omega}_k^{\text{test}} = K. \quad (3)$$

Overall, Table 3 shows that CAP consistently achieves the best results on multiple fairness objectives. An important conclusion is that, the benefit of CAP is more significant for objectives beyond balanced accuracy and improvements are around 2% or more (compared to (Menon et al. 2020) or plain post-hoc). This is perhaps natural given that prior works put an outsized emphasis on balanced accuracy in their algorithm design (Menon et al. 2020; Li et al. 2021).

4.3 Benefits of CAP for Adapting to Distinct Class Heterogeneities

Continuing the discussion in Sec. 3.1, we investigate the advantage of different attributes in the context of dataset heterogeneity adoption. In Table 4, we conduct loss function design CAP experiments on CIFAR-LT and ImageNet-LT dataset. Specifically, besides using regular CIFAR100-LT and ImageNet-LT, we introduce label noise into CIFAR10-LT following (Tanaka et al. 2018; Reed et al. 2014) to extend the heterogeneity of the dataset. To add the label noise, firstly, we split the training dataset to 80% train and 20% validation to accommodate bilevel optimization. Then we randomly generate a noise ratio $\boldsymbol{r} \in \mathbb{R}^K, \boldsymbol{r}_i \sim U(0, 0.5)$ that denotes the label noise ratio for each class. Finally, keeping the validation set clean, we add label noise into the train set by randomly flipping the labels of selected training samples (according to the noise ratio) to all possible labels. As a result, all classes contain an unknown fraction of label noise in the noisy CIFAR10-LT dataset, which raises more heterogeneity and challenge in optimization. Through bilevel optimization, we optimize the balanced classification loss and report the balanced test error and its standard deviation after the retraining phase in Table 4. As shown in Table 4, we employ label frequency $\mathcal{A}_{\text{FREQ}}$ which is designed for sample size heterogeneity and $\mathcal{A}_{\text{DIFF}}$ which is designed for class predictability as the attributes in CAP approach. Table 4 highlights that CAP consistently outperforms other methods while different attributes can shape the optimization process differently. Importantly, CAP is particularly favorable to tail classes which contain too few examples to optimize individually. Only using $\mathcal{A}_{\text{DIFF}}$ achieves smallest Err_{SDev} demonstrating that optimization with $\mathcal{A}_{\text{DIFF}}$ tends to keep better class-wise fairness because $\mathcal{A}_{\text{DIFF}}$ is directly related to class predictability. The combination of $\mathcal{A}_{\text{FREQ}}$ and $\mathcal{A}_{\text{DIFF}}$ shows that incorporating multiple class-specific attributes provides additional information about the dataset and jointly enhances performance. Overall, the results indicate that CAP establishes a principled approach to adapt to multiple kinds of heterogeneity.

5 Discussion

We proposed CAP as a flexible method to tackle class heterogeneities and general fairness objectives. CAP achieves high performance by efficiently generating class-specific strategies based on their attributes. We presented strong theoretical and empirical evidence on the benefits of multiple attributes. Evaluations on post-hoc optimization and loss function design revealed that CAP substantially improves multiple types of fairness objectives as well as general weighted test objectives.

Acknowledgements

This work was supported by the NSF grants CCF-2046816 and CCF-2212426, NSF CAREER 1942700, NSERC Discovery Grant RGPIN-2021-03677, Google Research Scholar award, Adobe Data Science Research award, and Army Research Office grant W911NF2110312.

References

Alabi, D.; Immorlica, N.; and Kalai, A. 2018. Unleashing linear optimizers for group-fair learning and optimization. In *Conference On Learning Theory*, 2043–2066. PMLR.

Alshammari, S.; Wang, Y.-X.; Ramanan, D.; and Kong, S. 2022. Long-tailed recognition via weight balancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6897–6907.

Calmon, F.; Wei, D.; Vinzamuri, B.; Natesan Ramamurthy, K.; and Varshney, K. R. 2017. Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems*, 30.

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *arXiv preprint arXiv:1906.07413*.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16: 321–357.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9268–9277.

Duchi, J. C.; and Namkoong, H. 2021. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3): 1378–1406.

Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 259–268.

Feldman, V. 2020. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, 954–959.

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of opportunity in supervised learning. *arXiv preprint arXiv:1610.02413*.

Hashimoto, T.; Srivastava, M.; Namkoong, H.; and Liang, P. 2018. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, 1929–1938. PMLR.

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

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

Kang, B.; Xie, S.; Rohrbach, M.; Yan, Z.; Gordo, A.; Feng, J.; and Kalantidis, Y. 2019. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*.

Khan, S. H.; Hayat, M.; Bennamoun, M.; Sohel, F. A.; and Togneri, R. 2017. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8): 3573–3587.

Kim, B.; and Kim, J. 2020. Adjusting decision boundary for class imbalanced learning. *IEEE Access*, 8: 81674–81685.

Kini, G.; and Thrampoulidis, C. 2020. Analytic study of double descent in binary classification: The impact of loss. *arXiv preprint arXiv:2001.11572*.

Kini, G. R.; Paraskevas, O.; Oymak, S.; and Thrampoulidis, C. 2021. Label-Imbalanced and Group-Sensitive Classification under Overparameterization. *accepted to the Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS)*.

Kleinberg, J.; Mullainathan, S.; and Raghavan, M. 2016. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*.

Kubat, M.; Matwin, S.; et al. 1997. Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, volume 97, 179–186. Citeseer.

Lahoti, P.; Beutel, A.; Chen, J.; Lee, K.; Prost, F.; Thain, N.; Wang, X.; and Chi, E. 2020. Fairness without demographics through adversarially reweighted learning. *Advances in neural information processing systems*, 33: 728–740.

Li, M.; Zhang, X.; Thrampoulidis, C.; Chen, J.; and Oymak, S. 2021. AutoBalance: Optimized Loss Functions for Imbalanced Data. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.

Li, T.; Cao, P.; Yuan, Y.; Fan, L.; Yang, Y.; Feris, R. S.; Indyk, P.; and Katabi, D. 2022. Targeted supervised contrastive learning for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6918–6928.

Liu, Z.; Miao, Z.; Zhan, X.; Wang, J.; Gong, B.; and Yu, S. X. 2019. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2537–2546.

Maldonado, S.; Vairetti, C.; Fernandez, A.; and Herrera, F. 2022. FW-SMOTE: A feature-weighted oversampling approach for imbalanced classification. *Pattern Recognition*, 124: 108511.

Menon, A. K.; Jayasumana, S.; Rawat, A. S.; Jain, H.; Veit, A.; and Kumar, S. 2020. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*.

Michel, P.; Hashimoto, T.; and Neubig, G. 2021. Modeling the second player in distributionally robust optimization. *arXiv preprint arXiv:2103.10282*.

Montanari, A.; Ruan, F.; Sohn, Y.; and Yan, J. 2019. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *arXiv preprint arXiv:1911.01544*.

Morik, K.; Brockhausen, P.; and Joachims, T. 1999. Combining statistical learning with a knowledge-based approach: a case study in intensive care monitoring. Technical report, Technical Report.

Reed, S.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; and Rabinovich, A. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*.

Sagawa, S.; Koh, P. W.; Hashimoto, T. B.; and Liang, P. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

Sagawa, S.; Raghunathan, A.; Koh, P. W.; and Liang, P. 2020. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, 8346–8356. PMLR.

Sahu, A. K.; Li, T.; Sanjabi, M.; Zaheer, M.; Talwalkar, A.; and Smith, V. 2018. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 3.

Tan, J.; Wang, C.; Li, B.; Li, Q.; Ouyang, W.; Yin, C.; and Yan, J. 2020. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11662–11671.

Tanaka, D.; Ikami, D.; Yamasaki, T.; and Aizawa, K. 2018. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5552–5560.

Wallace, B. C.; Small, K.; Brodley, C. E.; and Trikalinos, T. A. 2011. Class imbalance, redux. In *2011 IEEE 11th international conference on data mining*, 754–763. Ieee.

Williamson, R.; and Menon, A. 2019. Fairness risk measures. In *International Conference on Machine Learning*, 6786–6797. PMLR.

Xie, S.; Zheng, H.; Liu, C.; and Lin, L. 2018. SNAS: stochastic neural architecture search. *arXiv preprint arXiv:1812.09926*.

Ye, H.-J.; Chen, H.-Y.; Zhan, D.-C.; and Chao, W.-L. 2020. Identifying and compensating for feature deviation in imbalanced deep learning. *arXiv preprint arXiv:2001.01385*.

Zafar, M. B.; Valera, I.; Gomez Rodriguez, M.; and Gummadi, K. P. 2017. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, 1171–1180.

Zhai, R.; Dan, C.; Suggala, A.; Kolter, J. Z.; and Ravikumar, P. 2021. Boosted CVaR Classification. *Advances in Neural Information Processing Systems*, 34.

Zhang, J.; Liu, L.; Wang, P.; and Shen, C. 2019. To Balance or Not to Balance: A Simple-yet-Effective Approach for Learning with Long-Tailed Distributions. *arXiv preprint arXiv:1912.04486*.

Zhang, X.; Fang, Z.; Wen, Y.; Li, Z.; and Qiao, Y. 2017. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, 5409–5418.

Zhang, Y.; Kang, B.; Hooi, B.; Yan, S.; and Feng, J. 2023. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.