Statistical Learning Under Adversarial Distribution Shift

Chen Dan

CMU-CS-22-127

August 2022

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Pradeep Ravikumar, Chair
Zico Kolter
Zachary Lipton
Avrim Blum (Toyota Technological Institute in Chicago)
Yuting Wei (University of Pennsylvania)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was sponsored by UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN (Army) under award number 08883118409, the Air Force Research Laboratory under award number FA87501720152, the Office of Naval Research under award number N000141812861, the Defense Advanced Research Projects Agency under award number HR00112020006, and the National Science Foundation under award numbers CCF-1525971, CCF-1535967, DMS-1661802, and IIS-1664720. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Machine Learning, Statistical Learning Theory, Robustness

To my parents, my grandparents, my aunt and uncle, and Yifei.

Abstract

One of the most fundamental assumptions in statistical machine learning is that training and testing data should be sampled independently from the same distribution. However, modern real world applications require that the learning algorithm should perform robustly even when this assumption is no longer valid. Specifically, the training and testing distributions may shift slightly (yet adversarially) within a small neighborhood of each other. This formulation encompasses many new challenges in machine learning, including adversarial examples, outlier contaminated data, group fairness and label imbalance.

In this thesis, we seek to understand the statistical optimality and provide better algorithms under the aforementioned adversarial distribution shift. Our contributions include (1) the first near optimal minimax lower bound for the sample complexity of adversarially robust classification in a Gaussian setting. (2) introducing the framework of distributional and outlier robust optimization, which allowed us to apply distributionally robust optimization to large scale experiments with deep neural networks and outperformed existing methods in sub-population shift tasks. (3) margin sensitive group risk, a principled way of improving distributional robust generalization via group-asymmetric margin maximization.

Acknowledgments

It has been a fascinating 6-year adventure for my life in Carnegie Mellon. I am grateful for the support from my advisor, my collaborators, my friends, my family and many more. This thesis is dedicated to all of you.

First and foremost, I would like to thank my advisor Pradeep Ravikumar. I have learned a lot from Pradeep's diverse knowledge and broad vision. I also enjoyed the exceptional freedom to explore whatever I feel interestd in, and countless helpful discussions with him and all of other RAIL group members.

I would also like to thank my thesis committee members Avrim Blum, Zico Kolter, Yuting Wei and Zachary Lipton for their constructive suggestions. Avrim was the first professor that I have interacted with in the US, advised me for a year and hosted my summer visit to TTIC. I was always amazed by his sharpness and great intuition about everything. Zico and Yuting have been close collaborators of mine on many projects discussed in this thesis.

A huge thanks to all of my other collaborators (in reverse chronological order): Yuchen Zhou, Kristoffer Arnsfelt Hansen, He Jiang, Liwei Wang, Liu Leqi, Bryon Aragam, Eric Xing, Haris Angelidakis, Pranjal Awasthi, Vaggos Chatziafratis, Hong Wang, Hongyang Zhang, Saeed Seddighin, Runtian Zhai, Tianle Cai, Di He, Kun He, John Hopcroft, Huan Zhang, Boqing Gong, Cho-Jui Hsieh, Xun Zheng, Neil Xu, Justin Khim, Han Zhao, Tommi Jaakkola, Geoffrey Gordon, Arun Sai Suggala. Especially, I would like to thank Bryon Aragam, who helped me become a better researcher and influenced me in many ways.

I would also like to thank other faculty and staff in CMU. Thanks Deb Cavlovich for helping me with so many things over the six years. Thanks Ryan Tibshirani, Larry Wasserman and Ryan O'Donnell for their excellent lectures. Thanks David Woodruff, Pravesh Kothari, Sivaraman Balakrishnan and Aditi Raghunathan for serving as my skills test committee members.

I am grateful to all of my friends. First of all, I would like to thank Xun Zheng and Hongyang Zhang, I miss those dinners and fun conversations with you so much. I would also like to thank Yichong Xu, Han Zhao, Yifan Wu, Yao Liu and Bingyan Wang for their help during my job search, and many other friends at CMU (sorry if I missed any of your names): Fan Yang, Biwei Huang, Yanzhe Yang, Tiancheng Zhi, Ziqiang Feng, Haoxian Chen, Yun Meng, Simon Du, Ruosong Wang, Shuqi Dai, Tian Li, Shaojie Bai, Petar Stojanov and Adarsh Prasad; and my friends at Princeton: Kaizheng Wang, Guanhua He, Ping Wu, Yuling Yan, Ruiqi Gao, Anran Li and Zongjun Tan. Thanks to members of the learning theory reading group: Colin White, Nika Haghtalab, Vaishnavh Nagarajan, Travis Dick, Dravyansh Sharma and Mikhail Khodak. Thanks Zhiyi You, Xuwen Shi, Yangqinwei Shi, Qiang Gu, Yuanfeng Liu and Qizhe Yang for the wonderful trips and board games we had together.

Thanks to the wonderful restaurants in Pittsburgh: Sakura, Ka Mei, Green Pepper, Jian's Kitchen, Cafe 33, Sun Penang and more. Thanks to Jay Chou, Wei Dou, Black Panther and Tang Dynasty's music, along with the board game Wingspan and the lovely birds that came by my apartment, for saving me from depression. Thanks to Manchester United. Their horrible performance really helped me focus on my research.

Lastly and most importantly, I would like to thank Yifei Weng and my family, for your support and love. You made me a happier person. It would haven been impossible to finish my doctoral degree without your support. Words cannot adequately express my gratitude to you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the most fundamental assumptions in statistical machine learning is that training and testing data should be sampled independently from the *same* distribution. Mathematically speaking, in a supervised learning problem where we have feature vector $x \in \mathbb{R}^d$ and label $y \in \mathbb{R}$, it was assumed that there is an underlying distribution $\mathbb{P}$, such that the learner has access to n i.i.d. training samples

$$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n) \sim_{i.i.d.} \mathbb{P}^n, \tag{1.1}$$

and the learner's goal is to find a classifier $\hat{f}_n : \mathbb{R}^d \to \mathbb{R}$ based on observed training samples $\{(x_i, y_i)\}_{i=1}^n$, which makes the test error L_{test} as small as possible:

$$L_{test}(\hat{f}_n; \mathbb{P}) := \mathbb{E}_{(x,y)\sim\mathbb{P}} l(\hat{f}_n(x), y). \tag{1.2}$$

Here, $l(y', y)$ can be chosen as square loss $(y' - y)^2$ for regression problems, or zero-one loss $1[y' \neq y]$ for classification problems.

Here, we can see that both training and testing samples are sampled from the same underlying distribution $\mathbb{P}$. However, in recent years, modern real world applications require that the learning algorithm should perform robustly even when this assumption is no longer valid.

Here are some typical applications where the training and testing distributions differ from each other:

Adversarial Examples While deep learning algorithms have achieved tremendous success in a variety of different domains such as image classification, natural language processing and strategy games (e.g. Bahdanau et al. [10], Krizhevsky et al. [81], Silver et al. [124]), a crucial weakness, i.e. adversarial examples, has been observed by recent works Szegedy et al. [128] (among others e.g. Goodfellow et al. [53], Papernot et al. [107]). Namely, deep learning models often achieve extremely accurate performances yet are susceptible to small perturbations of the inputs, i.e. one can add small (nearly imperceptible) perturbation δ to image x , which cause neural network classifiers to make wrong predictions $\hat{f}_n(x)$ far from ground truth y, with very high confidence.

Outlier Contaminated Data Outlier robust estimation is a classic problem in statistics starting with the pioneering works of [65, 136]. The classic Huber's ε-contamination model assumes that at most ε fraction of training data can be contaminated i.e. sampled from any arbitrary distribution.

Efficient algorithms for very basic tasks, e.g. mean estimation, remains unsolved until late 2010s [42, 43, 85, 110].

Label Imbalance Datasets with class imbalance — that is, the number of samples of one class far exceeds the number of data of another class — are prevalent in cutting-edge data science applications [30, 60]. Take COVID-19 testing data for example: a dominant fraction of data samples often come from the negative class (i.e., non-targeted people who have not contracted the virus). The evaluation criterion in reality, however, might place equal, or even higher, emphasis on the minority class (e.g., the infected people in the COVID-19 case). The ability to generalize favorably in both majority and minority classes plays a pivotal role in critical scientific and societal issues (e.g., fairness/equity in machine learning, discovery of rare disease, transferability of knowledge to sample-starved tasks). It has been widely recognized, however, that the imbalanced availability of data can cause severe issues to modern data-limited learning algorithms including neural networks (e.g., [24, 60, 138]), particularly when reasoning about the underrepresented class.

Subpopulation Shift Another common type of distributional shift studied in this thesis is subpopulation shift, where the training and testing distributions are both a mixture of the same group of subpopulations, while the mixing weights can be different. Mathematically, we assume there are K subpopulations $P_1, P_2, \cdots, P_k$, and training/testing distributions can be written as

$$P_{\text{train}} = w_1 P_1 + w_2 P_2 + \cdots w_K P_K \tag{1.3}$$

$$P_{\text{test}} = w_1' P_1 + w_2' P_2 + \cdots w_K' P_K \tag{1.4}$$

It is commonly assumed that w and w' are similar, yet not exactly the same, from each other. Subpopulation shift is closely related to algorithmic fairness and class imbalance. The setting of subpopulation shift is most closely related to the algorithmic fairness notion of Rawlsian Max-Min fairness [58, 116].

In this thesis, we seek to understand the statistical optimality and provide better algorithms under aforementioned types of distrbution shift. In fact, we can formulate all of these applications as certain realization of adversarial distribution shift, defined in detail below.

Under the framework of adversarial distribution shift, we assume that P_{train} and P_{test} can be perturbed slightly, yet adversarially, from the underlying distribution $\mathbb{P}$, and the performance of the classifier is evaluated as the worst possible outcome from such perturbations. Mathematically, we assume there exists a collection of distributions $B_{train}(P)$, $B_{test}(P)$, both close to P in certain sense, and $P_{\text{train}} \in B_{train}(P), P_{\text{test}} \in B_{test}(P)$. The learner has access to n i.i.d. training samples

$$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n) \sim_{i.i.d.} \mathbb{P}_{train}^n, \tag{1.5}$$

and the learner's goal is to find a classifier $\hat{f}_n : \mathbb{R}^d \to \mathbb{R}$ based on observed training samples $\{(x_i, y_i)\}_{i=1}^n$, which makes the *worst-case* test error L_{test} as small as possible:

$$\sup_{P_{\text{train}} \in B_{train}(P), P_{\text{test}} \in B_{test}(P)} L_{test}(\hat{f}_n; \mathbb{P}_{\text{test}}) := \mathbb{E}_{(x,y)\sim \mathbb{P}_{\text{test}}} l(\hat{f}_n(x), y). \tag{1.6}$$

We can see that all of the four applications: adversarial examples, outlier contaminated data, label imbalance and sub-population shift, are realizations of this framework. (For the mathematical details, we refer the readers to the corresponding chapters.)

Adversarial Examples: The testing distribution P_{test} can be written as a sum of a vector $x \in P$ and a perturbation vector δ where $\|\delta\| \leq \varepsilon$.

Outlier contaminated data: The training distribution P_{train} can be written as a mixture $P_{\text{train}} = (1-\varepsilon)P + \varepsilon P'$, where P' can be arbitrary.

Label Imbalance: Denote the class conditional distributions as $P_0 := P(X|Y=0)$ and $P_1 := P(X|Y=1)$. We assume that the testing distribution is a balanced mixture of P_0 and P_1, while the training distribution can be imbalanced.

Subpopulation Shift: Assume that $P_{train} = \sum_{j=1}^{K} w_j P_j$ and $P_{test} = \sum_{j=1}^{K} w'_j P_j$. We assume that the testing distribution has the weight w' satisfying a *bounded divergence* condition:

$$D_f(w'||w) = \sum_{j=1}^{K} w_j f(\frac{w'_j}{w_j}) \leq t. \tag{1.7}$$

1.1 Background and Related Works

1.1.1 Sample Complexity in Adversarial Robustness

Achieving adversarial robustness has been a very challenging task. One of the main obstacle is sample complexity: training a classifier with adversarial robustness typically requires more training data. [120] showed that, in the setting where an adversary is present, the generalization gap is much larger compared to the standard supervised learning setting. They also showed that in a gaussian mixture model, adversarial robustness provably need more data. Carmon et al. [28], Stanforth et al. [125], Zhai et al. [161] showed that with the help of unlabeled data, it is possible to achieve high robust accuracy with the same number of labeled data required for standard learning. However, it was unclear about the optimal sample complexity in these models.

Another line of research study the sample complexity of adversarially robust learning under the PAC framework, using extensions of Rademacher complexity or VC dimension, including Attias et al. [5], Khim and Loh [73], Yin et al. [158], Cullina et al. [33], Montasser et al. [102], Awasthi et al. [8]. These works analyzed the difference between training and testing robust accuracy (i.e. the robust generalization gap) over a family of classifiers. Informally, these works show that the robust generalization gap scales as $O(\sqrt{\frac{C(F)}{n}})$, where $C(F)$ is a complexity measure for the family of classifiers. Again, it was not clear whether the $n^{-1/2}$ dependency is optimal.

In Chapter 2, we analyze the sample complexity of adversarially robust learning in the same Gaussian Mixture setting as [120], and provided optimal upper and lower bounds for it.

1.1.2 Subpopulation Shift

Distributionally robust optimization (DRO) [47, 103] is a popular approach to train classifiers which generalizes under subpopulation shift. distribution in a neighborhood of the observed training distribution. Generally speaking, DRO trains the model on the worst-off subpopulation,

and when the subpopulation membership is unknown, it focuses on the worst-off training instances, that is, the tail performance of the model. Previous work has shown effectiveness of DRO in subpopulation shift settings, such as algorithmic fairness [58] and class imbalance [155]. However, while DRO has been effective in small datasets or simple linear classifiers, it suffers from poor performance and severe instability during training when applied to large scale datasets and deep neural networks. One of the obstacle is that DRO is sensitive to outliers, as noted by several previous papers [58, 63, 169]. However, there has been no existing works on how to fix DRO in presense of outliers.

In Chapter 3, we propose an modification to DRO which provably generalize well even when the training set has a small fraction of outliers. The algorithm is inspired by the field of robust statistics, which has been an active field of research in statistics since 1960s [65, 136].

1.1.3 High Dimensional and Overparameterized Models

In statistical learning, it was typically assumed that the number of samples n is much larger than the dimensionality d (or the number of parameters p). This is due to the fact that the most common statistical error bounds scales as $\sqrt{\frac{d}{n}}$ or $\frac{d}{n}$, which becomes vacuous when d is much larger than n. However, in recent years, it became a common practice that practitioners train huge deep neural networks where the number of parameters is much larger than n, and these classifiers with huge amount of parameters outperformed the ones trained with convention wisdom. This discrepency got a lot of attention in the learning theory community in recent years.

Specifically, in [14], it was shown that the *algorithmic regularization* plays an important role in the overparameterized regime. For overparameterized models, there could be many model parameters which all minimize the training loss. Traditional analyzes do not distinguish between those minimizers. However, it was shown that the popular algorithms used in practice, like gradient descent, favors certain special solutions when there are multiple minimizers. Furthermore, in the context of deep neural networks, [1, 46] proved that when the neural networks are sufficiently wide, gradient descent can converge to zero training loss despite the non-convexity of training loss, and the converged classifier is closely related to Neural Tangent Kernel [67]. In short, the success of deep neural network is (at least partially) due to the blessing of algorithmic regularization.

However, in the context of subpopulation shift, algorithmic regularization actually brings more negative effect. This is because the most popular approaches, like DRO or sample reweighting, are aimed at improving over ERM - however, the algorithmic regularization actually makes them behave very similar to (or even the same as) ERM and overfit to training data. Sagawa et al. [117, 119] showed empiricaly that DRO overfits to the training data in subpopulation shift tasks. However, there hasn't been theoretical works that proves this observation. In chapter 5, we analyze the implicit bias of Distributionally Robust Optimization methods for overparameterized and interpolating models and showed that under various settings, DRO converges to the same solution of ERM.

1.2 Organization of this thesis

- In Chapter 2, we provide the first minimax lower bounds for adversarially robust classification in a Gaussian setting, along with an algorithm that achieves this lower bound. This is based on our paper published in ICML 2020.
- In Chapter 3, we introduced the framework of distributional and outlier robust optimization (DORO), which allowed us to apply distributionally robust optimization to large scale experiments with deep neural networks and outperformed existing methods in sub-population shift tasks. This is based on our paper published in ICML 2021.
- In chapter 4, we study the problem of imbalanced classification in a high dimensional Gaussian setting, where the number of samples n scale linearly with dimension d. This is a regime where classical theory breaks down (due to the requirement of $n >> d$). Our analysis reveals a surprising phenomenon: more samples can hurt the performance of M-estimators, even when popular heuristic of re-weighting is applied. We also derived a new lower bound which remains tight in the $d = \Theta(n)$ regime, showing that a bias-correcting estimator first proposed in Deev, 1970 is optimal.
- In chapter 5, we analyze the implicit bias of Distributionally Robust Optimization methods for overparameterized and interpolating models. We show that the implicit bias of gradient-based DRO leads to the convergence to the same solution of ERM in many settings, including linear regression and heavily overparameterized neural networks.
- In Chapter 6, we propose a new risk function, the margin sensitive group risk (MSG), as a risk upper bound for group sensitive generalization error based on margin theory. While this risk function is non-convex, we designed a alternate minimization algorithm to optimize MSG, which performs very well in practice. Using MSG to fine-tune the final layer of the neural network is both effective and efficient. We achieved higher worst group accuracy comparing with group DRO based methods on several datasets, without retraining the representation.

1.3 Excluded Research

To keep this thesis concise, a significant portion of this author's work during Ph.D. has been excluded from the document. These works include:

- Adversarial examples: [161, 162].
- Subpopulation shift and class imbalance: [155, 165].
- Matrix low rank approximation:[34, 36].
- Nonparametric mixture and graphical models:[4, 35, 168].
- Beyond worst case analysis of algorithms: [2, 19].
- Invariant representation learning: [167].

Chapter 2

Statistical Minimax Guarantees for Adversarially Robust Classification

2.1 Introduction

Recent years, machine learning algorithms have revolutionized our life due to their tremendous success in a variety of different domains such as image classification, natural language processing and strategy games (e.g. Bahdanau et al. [10], Krizhevsky et al. [81], Silver et al. [124]). These algorithms often achieve extremely accurate performances yet are susceptible to small perturbations of the inputs. In particular, Szegedy et al. [128] (among others e.g. Goodfellow et al. [53], Papernot et al. [107]) noticed that small perturbations (nearly imperceptible) to images could cause neural network classifiers to make wrong predictions with high confidence. While a growing amount of effort has been made in order to empirically improve the robustness of these learning algorithms against adversarial attacks, the problems of assessing statistical optimality, understanding generalization and statistical significance are important but far less understood. In this paper, we take a step towards this end.

In this work, we consider the adversarially robust classification problem under the Gaussian mixture model proposed by Schmidt et al. [120]. While the classification for mixture of Gaussian distributions — which is also referred to as discriminant analysis — has now been standard in statistics and computer science literature (see, e.g. McLachlan and Peel [96]), it is only until recently that researchers start to consider what can go wrong in the adversarial scenarios for this simple problem. It turns out (and as is shown in the sequel) that this simple yet instructive model demonstrates clear tradeoffs between adversarially robustness and the statistical complexities, and at the same time, capturing some of the features one would encounter in real applications.

Under minimal assumptions of the adversarial perturbations, we provide optimal minimax lower bounds, and show that a natural computationally efficient estimator achieves these minimax lower bounds in terms of the adversarial signal to noise ratio. Putting these together gives a sharp characterization of the intrinsic hardness of this problem in terms of how far one can push towards a robust estimator without any essential loss of statistical accuracy. These optimal lower and upper bounds are useful since that they provide a comprehensive view of the adversarially robust sample complexity of the conditional Gaussian model, which could then be contrasted with that

of the rates of the classical conditional Gaussian model.

Despite of an extensive line of work considering this problem, Schmidt et al. [120] and Bhagoji et al. [15] lie most closely to this paper. In order to obtain tight statistical characterizations of the risk, they made a number of simplifications, which thus do not directly provide answers to the minimax sample complexity of the original problem. As one main contrast, they consider the Bayesian setting where the means of the conditional Gaussians have as prior an independent standard Gaussian distribution. For other simplifications, Schmidt et al. [120] considered the spherical models so that the covariance is identity and also made additional simplifications such as large separation between two Gaussians and an upper bound on the noise level. These additional assumptions made it hard to compare with that of the adversary-free scenario. More detailed comparisons and discussions are provided after our main results.

2.1.1 Our contributions

The main contributions of this paper are summarized below, all of which are built upon a careful analysis of the classification error for linear classifiers.

- We develop the first minimax lower bounds for the classification excess risk in the conditional Gaussian model, stated in Theorem 2.4.1. In terms of the Adversarial Signal-to-Noise Ratio (AdvSNR), this excess risk scales as $\Omega_P(\exp(-(\frac{1}{8}+o(1))r^2)\frac{d}{n})$ for $\mathrm{AdvSNR}=r$, dimension d and sample size n.
- We construct a computationally efficient estimator based on the solution of a constrained quadratic optimization problem that has excess risk of order $O_P(\exp(-(\frac{1}{8}+o(1))r^2)\frac{d}{n})$. This result is given in Theorem 2.3.1. Hence, the upper bound is nearly tight (up to lower order terms in r) with the minimax lower bound in our regime of interest in terms of AdvSNR r, dimension d and sample size n .
- The recipe provided herein, works for a wide range of adversarial perturbations, generalizing the result by Schmidt et al. [120] who focus only on the ℓ_∞-type perturbations.
- Finally, our results are built upon minimum set of assumptions, without assuming strong separations between two classes, allowing for unknown and arbitrary covariance structure and the rates are naturally adaptive to the true signal.

Our findings unveil new insights into the adversarially robust sample complexity of the conditional Gaussian model which goes beyond of what the current theory has to offer.

2.1.2 Other related works

The conditional Gaussian models or mixture of Gaussians has been studied a lot in statistics and computer science literature. An incomplete and more recent list includes Azizyan et al. [9], Cai and Zhang [26], Kim et al. [74], Li et al. [91, 92]. In the context of adversarial robustness, since the seminal work of [120], there are several other papers that studied the sample complexity issue in conditional Gaussian models. Bhagoji et al. [15] also provided a slightly improved bound in the same setting. Carmon et al. [28], Stanforth et al. [125], Zhai et al. [161] showed that with the help of unlabeled data, it is possible to achieve high robust accuracy with the same number of labeled data required for standard learning.

Another line of research study the sample complexity of adversarially robust learning under the PAC framework, using extensions of Rademacher complexity or VC dimension, including Attias et al. [5], Khim and Loh [73], Yin et al. [158], Cullina et al. [33], Montasser et al. [102], Awasthi et al. [8]. The tradeoff in standard and robust accuracy has been theoretically and empirically studied in Zhang et al. [166], Suggala et al. [126], Tsipras et al. [135], Raghunathan et al. [114] and Javanmard et al. [69].

Several previous works analyzed the robustness of specific family of classifiers. The early work of Xu et al. [152, 153] estabilished the connections between robust optimization for linear models and certain types of regularization in classification and regression settings. Subsequently, Xu and Mannor [154] also showed that under certain notion of robustness, robust algorithms can generalize well. Wang et al. [146] studied the robustness of nearest neighbor classifiers.

From the aspect of computational complexity, some recent works showed that learning a robust model or even verifying robustness of a given model can be computationally hard, including [22, 23] and [7, 147].

2.1.3 Notations

For the reader's convenience, we list here our notational conventions.

For positive semi-definite matrix A, we use $\|x\|_A := \sqrt{x^T A x}$. Let $\Phi(\cdot)$ be the CDF of standard Gaussian distribution $\mathcal{N}(0,1)$ and $\bar{\Phi}(x) := 1 - \Phi(x)$. The notation $f(n,d) = O\left(g(n,d)\right)$ means that there exists a universal constant $c > 0$ that does not depend on the problem parameters such as n, d etc, such that $|f(n,d)| \leq c|g(n,d)|$. Similarly, we define $f(n,d) = \Omega\left(g(n,d)\right)$ when there exist constants $c_1, c_2 > 0$ such that $c_1|g(n,d)| \leq |f(n,d)| \leq c_2|g(n,d)|$. Notation O_P, Ω_P are used if the corresponding relations happen with probability converges to 1 as $n \to \infty$ (see e.g. Chapter 2 of [137]). We define the ℓ_p norm $\|x\|_p = (\sum_{i=1}^d x_i^p)^{1/p}$ and the corresponding ℓ_p-ball as $\{x \in \mathbb{R}^d | \|x\|_p \leq 1\}$.

2.2 Preliminaries

This section is devoted to setting up the adversarial robust classification problem that is considered in this paper. Along the way, we introduce necessary background and state several preliminary results for future comparisons.

Conditional Gaussian Model We consider the binary classification problem with data pair (x, y) generated from the mixture of two Gaussian distributions $P_{\mu,\Sigma}$,

$$p(y=1) = \frac{1}{2}, \quad p(y=-1) = \frac{1}{2},$$
$$p(x|y) = \mathcal{N}(x; y\mu, \Sigma).$$

Here $\mu \in \mathbb{R}^d$, $\Sigma \in \mathbb{R}^{d\times d}, \Sigma \succeq 0$ denote the mean and covariance of the Gaussian distribution. Given n training samples $(x_i, y_i) \sim_{i.i.d.} P_{\mu,\Sigma}$ for $1 \leq i \leq n$, the goal is to learn a classifier $\hat{f}(x)$ for predicting the class of a future data point that is drawn from the same distribution $P_{\mu,\Sigma}$.

Adversarially Robust Classification In the standard setting of classification, the optimal classifier is defined as the one that which minimizes the population classification error

$$R^{\text{std}}_{\mu,\Sigma}(f) := \mathbb{E}_{(x,y)\sim P_{\mu,\Sigma}}\left[\mathbb{I}(f(x) \neq y)\right].$$

which we refer to the standard error throughout. In this paper, we consider the classification problem under conditional Gaussian generative model in presence of an adversary — which is to say — at the testing stage, an adversary is allowed to add any perturbation δ to the input x, that has bounded magnitude $\|\delta\|_B \leq \varepsilon$. The norm defined here is the standard Minkowski functional that associated with a convex set [131]. Formally, given a closed and origin-symmetric convex set B, the Minkowski functional is defined as

$$\|x\|_B := \inf\{\lambda \in \mathbb{R}_{>0} : x \in \lambda B\}.$$

For instance, when B is the ℓ_p unit ball, then $\|x\|_B$ boils down to the classical ℓ_p norm of x. In practice, the most widely considered norm for the adversary are ℓ_∞ and ℓ_2 norms.

In the adversarially robust setting, a mapping $f : \mathbb{R}^d \to \{-1, +1\}$ classifies a sample (x, y) correctly, if and only if the prediction agrees with the true label for *all* possible perturbations of the adversary. To put it in mathematical form,

$$\ell_{B,\varepsilon}(f; x, y) := \mathbb{I}\left(\exists \delta : \|\delta\|_B \leq \varepsilon,\ f(x+\delta) \neq y\right).$$

Our goal is to obtain a classifier with minimal expected robust classification error, i.e. finding mapping f that minimizes

$$\begin{aligned} R^{B,\varepsilon}_{\mu,\Sigma}(f) &= \mathbb{E}_{(x,y)\sim P_{\mu,\Sigma}}[\ell_{B,\varepsilon}(f; x, y)] \\ &= \mathbb{E}_{(x,y)\sim P_{\mu,\Sigma}}[\mathbb{I}\left(\exists \|\delta\|_B \leq \varepsilon, f(x+\delta) \neq y\right)]. \end{aligned} \tag{2.1}$$

The optimal risk is then defined as the classification error regarding the optimal classifier, namely

$$R^{B,\varepsilon}_{\mu,\Sigma}* := R^{B,\varepsilon}_{\mu,\Sigma}(f_*), \tag{2.2}$$

and accordingly, we define the excess risk of any classifier f as

$$R^{B,\varepsilon}_{\mu,\Sigma}(f) - R^{B,\varepsilon}_{\mu,\Sigma}*, \tag{2.3}$$

which by definition is always non-negative.

Robust Bayes Optimal Classifier To motivate the robust optimal classifiers, we start our discussion with the optimal risk and optimal classifier in the conditional Gaussian Model. We note that when $\varepsilon = 0$, i.e. there is no adversary, the classification problem reduces to the well-known *Fisher's Linear Discriminant Analysis* problem, where the Bayes optimal classifier is a simple linear classifier

$$f_{\text{Bayes}}(x) = \text{sign}(\mu^T x),$$

known as Fisher's linear discriminant rule (see, e.g. Johnson et al. [71]). The Bayes optimal classifier minimizes the misclassification rate. However, the classifier that minimizes the *robust* classification error is not known until recently, where [15] provided a tight lower bound on the minimal robust classification error via optimal transport techniques. It is also proved that the optimal risk can be written as the optimal value of a convex program, and the *oracle* optimal classifier is a linear classifier that has a closed form given the solution of the convex program.

We find it is useful to first simplify and restate this result in order to set the stage for our main result.

Theorem 2.2.1 (Restated and simplified from Bhagoji et al. [15])**.** *Let $z_\Sigma(\mu)$ be the solution of the following convex program:*

$$z_\Sigma(\mu) = \underset{\|z\|_B \le \varepsilon}{\operatorname{argmin}} \|\mu - z\|_{\Sigma^{-1}}^2, \tag{2.4}$$

where $\|x\|_A = \sqrt{x^T A x}$. [1] *Then, the optimal robust classifier for $P_{\mu,\Sigma}$ is a linear classifier $f_*(x) = \operatorname{sign}(w_0^T x)$, where*

$$w_0 := \Sigma^{-1}(\mu - z_\Sigma(\mu)), \tag{2.5}$$

and the optimal robust classification error is

$$R^{B,\varepsilon}_{\mu,\Sigma}* := \bar{\Phi}(\|w_0\|_\Sigma) = \bar{\Phi}(\|\mu - z_\Sigma(\mu)\|_{\Sigma^{-1}}).$$

We remark that the above mentioned classifier is indeed an oracle classifier since it is constructed using the unknown parameters μ and Σ.

Adversarial Signal-To-Noise Ratio (AdvSNR). In the context of standard classification in the conditional Gaussian model, the notion of Signal-To-Noise Ratio was introduced to measure the effective separation which is defined as the Mahalanobis distance between the means of two conditional distributions.

Definition 2.2.1 (Standard Signal-To-Noise Ratio)**.** *The Standard Signal-To-Noise Ratio (StdSNR) of conditional Gaussian model $P_{\mu,\Sigma}$ is defined as*

$$\mathrm{StdSNR}(\mu, \Sigma) := 2\|\mu\|_{\Sigma^{-1}}.$$

Here, the constant 2 is introduced to be consistent with the literature in Fisher's LDA, e.g. [26], where SNR is defined as the Mahalanobis distance between means of two mixture components. We make the note that the StdSNR measures the difficulty of standard classification in the conditional Gaussian model, since the minimal misclassification error equals to $\bar{\Phi}(\frac{1}{2}\mathrm{StdSNR}(\mu, \Sigma))$ [26]. In fact, the misclassification error decreases exponentially as the StdSNR increases.

When it comes to the adversarial setting, StdSNR, however, is no longer a proper metric for the classification difficulty. Specifically, conditional Gaussian models with the same StdSNR can have very different levels of hardness in the adversarially robust classification problem. In order to illustrate this, we demonstrate a simple example.

[1] Note that this notation is different with [15], where in their notation $\|x\|_A = \sqrt{x^T A^{-1} x}$.

Example 2.2.1. *Consider an adversary which is allowed to perturb the input with budget $\varepsilon = \frac{6}{\sqrt{d}}$ in terms the ℓ_∞ norm. Set the covariance Σ to be the identity matrix I_d. We examine two conditional Gaussian models, $P_{\mu_1,\Sigma}$ and $P_{\mu_2,\Sigma}$ with different means μ_1 and μ_2, where*

$$\mu_1 = \frac{6}{\sqrt{d}} \cdot (1, 1, 1, \cdots, 1)^T, \quad \mu_2 = (6, 0, 0, \cdots, 0)^T.$$

It is easily seen that $\|\mu_1\|_{\Sigma^{-1}} = \|\mu_2\|_{\Sigma^{-1}} = 6$, therefore $P_{\mu_1,\Sigma}$ and $P_{\mu_2,\Sigma}$ have the same StdSNR. However, by Theorem 2.2.1, these two distributions actually exhibit completely different minimal robust classification error, indeed,

$$R^{B,\varepsilon}_{\mu_1,\Sigma} = \bar{\Phi}(0) = \frac{1}{2}, \quad R^{B,\varepsilon}_{\mu_2,\Sigma} = \bar{\Phi}(6 - \frac{6}{\sqrt{d}}).$$

When the dimension d is sufficiently large, the optimal risk $R^{B,\varepsilon}_{\mu_2,\Sigma}$ approaches $\bar{\Phi}(6) \approx 10^{-8}$, which means there exists a very good robust classifier for $P_{\mu_2,\Sigma}$. In contrast, the optimal risk $R^{B,\varepsilon}_{\mu_1,\Sigma} = \frac{1}{2}$, i.e. no classifier can achieve a robust accuracy better than a uninformative predictor that classifies everything as the same class. From this simple example, it is safe to conclude that StdSNR is not an ideal measurement for the difficulty in the adversarially robust classification problem.

To address the above issue, one need a proper definition of the signal-to-noise-ratio that is suitable for the adversarial robust setting. Therefore we introduce the Adversarial Signal-To-Noise Ratio (AdvSNR) for any (B, ε) adversary.

Definition 2.2.2 (Adversarial Signal-To-Noise Ratio)**.** *Define the (B, ε) Adversarial Signal-To-Noise Ratio (AdvSNR) of conditional Gaussian model $P_{\mu,\Sigma}$ as*

$$\mathrm{AdvSNR}_{B,\varepsilon}(\mu, \Sigma) := 2\|\mu - z_\Sigma(\mu)\|_{\Sigma^{-1}} = 2\|w_0\|_\Sigma,$$

where w_0 is defined in (2.5).

As a consequence of Theorem 2.2.1, the minimal robust classification error satisfies

$$R^{B,\varepsilon}_{\mu,\Sigma}* = \bar{\Phi}\left(\frac{1}{2}\mathrm{AdvSNR}(\mu, \Sigma)\right). \tag{2.6}$$

Consequently, the AdvSNR fully characterizes the difficulty for the adversarially robust setting as the StdSNR in the standard setting. We also note that when $\varepsilon = 0$, i.e. there is no adversary, the AdvSNR reduces to the traditional definition of the StdSNR. Thus, AdvSNR is a reasonable generalization for StdSNR.

Naturally, for every $r > 0$, one can consider a class of distributions where each of them has the same (B, ε)-AdvSNR equal to r. Within each class, they should enjoy the same hardness of the classification problem. Formally, let us define the class $D_{B,\varepsilon}(r)$.

Definition 2.2.3. *The family of conditional Gaussian models with (B, ε)-AdvSNR value of r, is defined as:*

$$D_{B,\varepsilon}(r) := \{(\mu, \Sigma) | \mathrm{AdvSNR}_{B,\varepsilon}(\mu, \Sigma) = r\}.$$

In the sequel, we develop our minimax lower bounds over these classes of distributions. To assist our analysis, we also define the family of conditional Gaussian models with a standard SNR value of r similarly.

Definition 2.2.4. *The family of conditional Gaussian models with a standard SNR value of r, is defined as:*

$$D_{\text{std}}(r) := \{(\mu, \Sigma) | \text{StdSNR}(\mu, \Sigma) = r\}.$$

In the derivations of our upper bounds and minimax lower bounds, we make the assumption that the AdvSNR r is strictly bounded away from zero by a universal constant [2], otherwise as a result of Theorem 2.2.1, no classifier can achieve accuracy much better than $\frac{1}{2}$, the robust risk of a constant classifier $f(x) \equiv 1$.

2.3 A Coputationally Efficient Estimator and Risk Upper Bound

Thus far, we introduce the notion of AdvSNR which is known to characterize the minimal robust classification error as in expression (2.6). However, whether there exists a computation-efficient classifier that behaves similarly to the oracle best classifier is still unclear.

This section, we aim to answer this question in the affirmative by constructing such a classifier. For the classifier that we shall define in the sequel, we give an exact characterization of its excess robust classification error compared with the oracle best classifier. Motivated by the fact that the optimal robust classifier has the form of (2.5), we design a "plug-in" estimator for w_0. The estimator is described in the following algorithm.

Algorithm 1 A plug-in estimator of w_0

Input: Data pairs $\{(x_i, y_i)\}_{i=1}^n$.
Output: $\widehat{w}$.
Step 1: Define $\widehat{\mu}$ and $\widehat{\Sigma}$ as

$$\widehat{\mu} := \frac{1}{n}\sum_{i=1}^n y_i x_i, \qquad \widehat{\Sigma} := \frac{1}{n}\sum_{i=1}^n x_i x_i^T - \widehat{\mu}\widehat{\mu}^T.$$

Step 2: Solve for $\widehat{z}$ in the following

$$\widehat{z} := z_{\widehat{\Sigma}}(\widehat{\mu}) = \underset{\|z\|_B \le \varepsilon}{\text{argmin}} \|\widehat{\mu} - z\|_{\widehat{\Sigma}^{-1}}^2.$$

Step 3: Define $\widehat{w} := \widehat{\Sigma}^{-1}(\widehat{\mu} - \widehat{z})$.

The main theorem of this section is to characterize the excess risk bound of the classifier induced by $\widehat{w}$.

Theorem 2.3.1. *For the $(\|\cdot\|_B, \varepsilon)$ adversary, suppose the adversarial signal-to-noise ratio $\text{AdvSNR}_{B,\varepsilon}(\mu, \Sigma) = r$, then the excess risk of $f_{\widehat{w}}$ is upper bounded by*

$$R_{\mu,\Sigma}^{B,\varepsilon}(f_{\widehat{w}}) - R_{\mu,\Sigma}^{B,\varepsilon *} \le O_P\Big(e^{-\frac{1}{8}r^2} \cdot r \cdot \frac{d}{n}\Big).$$

[2] for instance, $r \ge 10^{-9}$

We take a moment to make several remarks. First recall that the AdvSNR is defined as a measurement for the hardness of the classification problem. Indeed, as the above result shows, the excess risk vanishes exponentially with the AdvSNR. Moreover, our estimator is *adaptive* in the sense that it does not require knowing any information about the value of r, but the theoretical guarantee improves automatically with larger AdvSNRs. We also note that the dependency with sample size n is $O\left(\frac{1}{n}\right)$, which is the same as the rate of Fisher's LDA, but faster than the typical $O\left(\frac{1}{\sqrt{n}}\right)$ rate.

Comparisons to [120] We note that our result generalizes the one showed in [120] in many different aspects:

1. In terms of the perturbations, Schmidt et al. [120] considered perturbations in ℓ_∞ balls, while ours allow for any convex, closed and origin-symmentric perturbaion set B, including all ℓ_p balls for $p \geq 1$.
2. Our upper and lower bounds hold for both spherical and non-spherical Gaussians, without the knowledge of the population covariance structure.
3. We impose no restrictions on the separation between Gaussian distributions. Schmidt et al. [120] studied a very specific regime, where the budget of ℓ_∞ adversary is bounded by $\frac{1}{4}$, the separation between the means of two Gaussians is $\sqrt{d}$, and the spherical covariance matrix $\Sigma = \sigma^2 I$ satisfies $\sigma \leq \frac{1}{32} d^{1/4}$. This regime is low-noise by design, while our analysis applies to any regime whenever there exists a classifier with robust accuracy slightly better than $\frac{1}{2}$.
4. Our estimator is consistent, i.e. the excess risk converges to zero as sample size $n \to \infty$. The classifier used in Schmidt et al. [120] is actually $\mathrm{sign}(\hat{\mu}^T x)$. While this classifier achieve near-optimal classification error in the regime of their interst (the low noise regime mentioned above with Gaussian prior on μ), the excess risk does not converge to zero in general. This is due to the fact that the large-sample limit of their classifier is actually $\mathrm{sign}(\mu^T x)$, i.e. the Bayes optimal classifier for the standard setting. As we can see from Theorem 2.2.1 and a simple simulation in Figure 2.1, the excess risk of their algorithm saturates at a level above zero, which is very different from the behavior of Algorithm 1.

Proof Sketch: Here we provide a brief sketch of the proof. More details can be found in the Section 2.6.

Step 1: First order approximation of the risk. Since both the learned $f_{\hat{w}}$ and the optimal robust classifier f_* are linear classifiers, we can calculate the robust excess risk in closed form using Lemma 2.6.2 (also shown in [15]):

$$R^{B,\varepsilon}_{\mu,\Sigma}(f_{\widehat{w}}) - R^{B,\varepsilon}_{\mu,\Sigma}* = \bar{\Phi}\left(\frac{\widehat{w}^T\mu - \varepsilon\|\widehat{w}\|_{B*}}{\|\widehat{w}\|_\Sigma}\right) - \bar{\Phi}(\frac{1}{2}r).$$

By the Taylor expansion of $\bar{\Phi}(\cdot)$, we have

$$\bar{\Phi}\left(\frac{\widehat{w}^T\mu - \varepsilon\|\widehat{w}\|_{B*}}{\|\widehat{w}\|_\Sigma}\right) - \bar{\Phi}(\frac{1}{2}r) \approx \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{8}r^2}\delta_n,$$

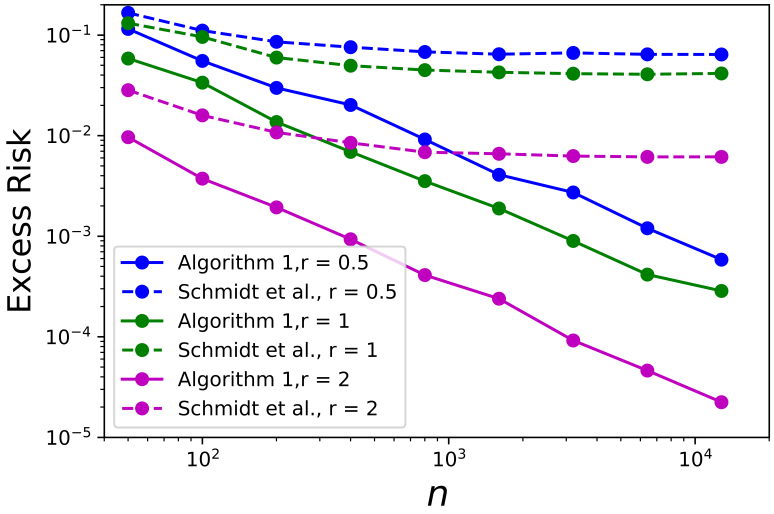


Figure 2.1: A simple simulation on the performance of Algorithm 1 and the algorithm proposed in [120] is shown here with different values of AdvSNR r. Here we consider a 50-dimensional example under ℓ_∞ adversary with $\varepsilon = 0.1$. The covariance matrix is fixed to be $\Sigma = I$, and the mean parameter μ is set as $\mu = (r + \varepsilon, \varepsilon, \varepsilon, \cdots, \varepsilon)$ for $r \in \{0.5, 1.0, 2.0\}$. We evaluate the excess risk $R^{B,\varepsilon}_{\mu,\Sigma}(f_{\widehat{w}}) - R^{B,\varepsilon}_{\mu,\Sigma}*$ returned by the two algorithms using n i.i.d. training data pairs, where $n \in \{50, 100, 200, 400, 800, 1600, 3200, 6400, 12800\}$. For each combination of (n, r), the averaged excess risk over 10 random repetitions is reported respectively.

where

$$\delta_n = \frac{1}{2}r - \frac{\widehat{w}^T\mu - \varepsilon\|\widehat{w}\|_{B*}}{\|\widehat{w}\|_\Sigma}.$$

Therefore, it is sufficient to show that $\delta_n = O_P(r \cdot \frac{d}{n})$.

Step 2: Controlling δ_n. To give an upper bound of δ_n, we will use the fact that sample mean $\widehat{\mu}$ and sample covariace $\widehat{\Sigma}$ converge to μ and Σ respectively. Furthermore, the convergence rate is well known as $O_P(\sqrt{\frac{d}{n}})$.

From a high level, the upper bound of δ_n is estabished (see Lemma 2.6.3) by carefully decomposing δ_n into four terms and each term is in the form of the differences between population and sample quantities like Σ vs $\widehat{\Sigma}$, $\widehat{\mu}$ vs μ. Invoking the convergence rates of $\widehat{\mu}$ and $\widehat{\Sigma}$, we are able to bound each of these terms and complete the proof.

2.4 Minimax Lower Bounds

This section is dedicated to developing minimax excess risk lower bounds for the adversarially robust classification with conditional Gaussian models.

As is mentioned above, we consider a class of distributions $D_{B,\varepsilon}(r)$ which have the same $\mathrm{AdvSNR}_{B,\varepsilon} = r$, as in Definition 2.2.3. As quantity AdvSNR characterizes the minimal robust classification error, this class of distributions $D_{B,\varepsilon}(r)$ all share the same adversarially robust classification error. Therefore, our lower bounds here measure the fundamental information-theoretic limit of this problem, namely, no estimator can achieve an essential improvement in terms of the adversarial classification error.

Theorem 2.4.1. *Let $\widehat{f}$ be any estimator based on n samples $(x_1, y_1), \cdots, (x_n, y_n) \sim_{i.i.d.} P_{\mu,\Sigma}$. We have the following lower bound on the minimax excess risk:*

$$\min_{\widehat{f}} \max_{(\mu,\Sigma)\in D_{B,\varepsilon}(r)} [R^{B,\varepsilon}_{\mu,\Sigma}(\widehat{f}) - R^{B,\varepsilon}_{\mu,\Sigma}*] \geq \Omega_P\Big(e^{-(\frac{1}{8}+o(1))r^2}\frac{d}{n}\Big).$$

Putting together with the upper bound in Theorem 2.3.1, this lower bound matches almost exactly with the upper bound in the regime of interest, therefore they are both optimal up to lower order terms.

The main technique we used for this lower bound is with a flavor of black-box reduction. In particular, we show that the minimax *robust* excess risk in $D_{B,\varepsilon}(r)$ cannot be smaller than the minimax *standard* excess risk in $D_{\mathrm{std}}(r)$. In other words,

Lemma 2.4.1. *The minimax excess error satisfies*

$$\min_{\widehat{f}} \max_{(\mu,\Sigma)\in D_{B,\varepsilon}(r)} [R^{B,\varepsilon}_{\mu,\Sigma}(\widehat{f}) - R^{B,\varepsilon}_{\mu,\Sigma}*] \geq \min_{\widehat{f}} \max_{(\mu',\Sigma)\in D_{\mathrm{std}}(r)} [R^{\mathrm{std}}_{\mu',\Sigma}(\widehat{f}) - R^{\mathrm{std}}_{\mu',\Sigma}*].$$

The right hand side of (2.7), i.e. the minimax rate for standard classification, is well-studied in the existing literature of Fisher's LDA. For example, [92] proved the following lower bound:

Theorem 2.4.2 (Theorem 1 of [92])**.** *Suppose the covariance matrix satisfies $\Sigma = I$ and is known to the learner, then we have the minimax lower bound*

$$\min_{\widehat{f}} \max_{(\mu',I)\in D_{\mathrm{std}}(r)} [R^{\mathrm{std}}_{\mu',\Sigma}(\widehat{f}) - R^{\mathrm{std}}_{\mu',\Sigma}*] \geq \Omega_P\left(e^{-\frac{1}{8}r^2}\cdot\frac{1}{r}\cdot\frac{d}{n}\right).$$

Since the parameter space considered in [92] is a subset of $D_{\mathrm{std}}(r)$, we have (2.7) is also lower bounded by $\Omega_P\left(e^{-\frac{1}{8}r^2}\cdot\frac{1}{r}\cdot\frac{d}{n}\right)$, therefore proves Theorem 2.4.1.

Comparisons to [120] and [15] To the best of our knowlege, Theorem 2.4.1 is the first minimax-type lower bound in adversarially robust classification. Existing works [120] and [15] also studied the sample complexity of robust learning in conditional Gaussian model. However, both of them simplified the problem and considered the case when μ follows from a prior distribution $\mathcal{N}(0, I)$. This assumption is crucial to their analysis, otherwise the posterior distribution of μ given training data is intractable. Hence, the technical tool used in prior works is not sufficient for developing such a minimax lower bound of our interest.

Proof Sketch: Here we also provide a proof sketch to Lemma 2.4.1. More details can be found in the Section 2.6.

Step 1: Connecting standard and robust risks In Lemma 2.6.4, we prove that for any classifier f and a perturbed distribution $P_{\mu',\Sigma}$, where $\|\mu' - \mu\|_B \leq \varepsilon$, the robust risk of f on $P_{\mu,\Sigma}$ is always lower bounded by the standard risk on $P_{\mu',\Sigma}$.

As a consequence, in Corollary 2.6.1 we show that if we choose $\mu' = \mu - z_\Sigma(\mu)$, then the robust excess risk of f on $P_{\mu,\Sigma}$ is always lower bounded by the standard excess risk on $P_{\mu',\Sigma}$.

Step 2: A mapping fron $D_{\text{std}}(r)$ to $D_{B,\varepsilon}(r)$ To prove Lemma 2.4.1, we only need to answer the following question: for any $(\mu', \Sigma) \in D_{\text{std}}(r)$, can we find a $(\mu, \Sigma) \in D_{B,\varepsilon}(r)$, so that the robust excess risk on $P_{\mu,\Sigma}$ is always lower bounded by the standard excess risk on $P_{\mu',\Sigma}$? We give an affirmative answer to this question. The proof in a combination of Corollary 2.6.1 showed in Step 1 and an examination of optimality condition in the optimization problem 2.4.

2.5 Comparing Adversarial and Standard Rates

Putting the upper and lower bounds together provides a comprehensive view of the statistical aspect of the adversarially robust classification. A key question to ask is that: How much does the classification error blows up as the price of being adversarially robust?

To answer this question, it is sufficient to compare the optimal risks in both cases. Informally, one can write the logarithm ratio between two rates as

$$\log\left(\frac{\text{AdvRate}}{\text{StdRate}}\right) \approx \frac{1}{2}\left(\|\mu - z_\Sigma(\mu)\|_{\Sigma^{-1}}^2 - \|\mu\|_{\Sigma^{-1}}^2\right). \tag{2.7}$$

From the definition of $z_\Sigma(\mu)$ in (2.4), we can see that $\|\mu - z_\Sigma(\mu)\|_{\Sigma^{-1}}^2 \leq \|\mu\|_{\Sigma^{-1}}^2$, hence adversarial rate is always slower.

To analyze this difference quantitively and interpretably, we consider the special case where $\Sigma = I$ and the adversary is ℓ_2 bounded. Similar results hold for other adversaries as well. The key observation is that depending on the different scale of $\|\mu\|_2$ and the budget of perturbation ε, this difference can be as small as $O(1)$, or as large as $\Omega(\exp(d))$.

Proposition 2.5.1. *When $\Sigma = I$ and the adversarial perturbation satisfies $\|\delta\|_2 \leq \varepsilon$, then*

- *When $\varepsilon \leq O(\frac{1}{\|\mu\|_2})$, the adversarial rate is at most $O(1)$ times slower than the standard rate.*
- *When $\|\mu\|_2 \geq \Omega(\log d)$ and $\varepsilon \geq \Omega(\frac{\log d}{\|\mu\|_2})$, the adversarial rate can be slower than the standard rate by a $poly(d)$ factor.*
- *When $\|\mu\|_2 \geq \Omega(\sqrt{d})$ and $\varepsilon \geq \Omega(\frac{d}{\|\mu\|_2})$, the adversarial rate can be slower than the standard rate by an $\exp(d)$ factor.*

In general, the difference is more significant when ε or $\|\mu\|_2$ is larger. This example demonstrates a clear tradeoff between being adversarial robust and obtaining the optimal accuracy, in particular in the case of large perturbations.

2.6 Proofs and further details

In this section, we provide detailed proofs for our main results. The proof details of some lemmas are deferred to our supplementary file.

2.6.1 Proof of Theorem 2.3.1

Before presenting our analysis, we first state a standard lemma about the convergence of empirical mean and covariance.

Lemma 2.6.1 (Convergence of the empirical mean and covariance (see, e.g. Wainwright [141]))**.** *The convergence rates of the empirical mean $\widehat{\mu}$ and $\widehat{\Sigma}$ to the corresponding ground truth satisfy*

$$\|\widehat{\mu} - \mu\|_{\Sigma^{-1}} = O_P\left(\sqrt{\frac{d}{n}}\right),$$

and

$$\|\Sigma^{-\frac{1}{2}}\widehat{\Sigma}\Sigma^{-\frac{1}{2}} - I\|_{op} = O_P\left(\sqrt{\frac{d}{n}}\right).$$

The following lemma about the classification error of linear classifiers will also be useful for us.

Lemma 2.6.2 (Robust classification error of linear classifier, (see e.g. in [15], Appendix B.3))**.** *For a linear classifier $f_w(x) = \mathrm{sign}(w^T x)$, the robust classification error with a B, ε adversary is*

$$R^{B,\varepsilon}_{\mu,\Sigma}(f_w) = \bar{\Phi}\left(\frac{w^T\mu - \varepsilon\|w\|_{B*}}{\|w\|_\Sigma}\right).$$

Here, $\|\cdot\|_{B*}$ is the dual norm of $\|\cdot\|_B$. We use $R^{B,\varepsilon}_{\mu,\Sigma}(w)$ as a shorthand for $R^{B,\varepsilon}_{\mu,\Sigma}(f_w)$ when the meaning is clear from context.

Proof of Theorem 2.3.1. By Lemma 2.6.2 and Taylor expansion of $\bar{\Phi}(t)$ around $t = \frac{1}{2}r = \|w_0\|_\Sigma$, the excess risk can be written as:

$$\begin{aligned} R^{B,\varepsilon}_{\mu,\Sigma}(\widehat{w}) - R^{B,\varepsilon}_{\mu,\Sigma}* &= \bar{\Phi}\left(\frac{\widehat{w}^T\mu - \varepsilon\|\widehat{w}\|_{B*}}{\|\widehat{w}\|_\Sigma}\right) - \bar{\Phi}(\|w_0\|_\Sigma) \\ &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{8}r^2}\delta_n + O(\delta_n^2), \end{aligned}$$

where

$$\delta_n = \|w_0\|_\Sigma - \frac{\widehat{w}^T\mu - \varepsilon\|\widehat{w}\|_{B*}}{\|\widehat{w}\|_\Sigma}.$$

Therefore, to analyze the convergence rate of the excess risk, we only need to analyze the convergence rate of δ_n. We would like to prove that

$$\delta_n = O_P\left(r \cdot \frac{d}{n}\right).$$

The following lemma is the key of our analysis: it decomposes δ_n into four terms, each in the form of the difference between population and sample quantities like Σ vs $\widehat{\Sigma}$, $\widehat{\mu}$ vs μ.

Lemma 2.6.3. *We have the following decomposition for δ_n:*

$$\|\widehat{w}\|_\Sigma \delta_n = \underbrace{-\frac{1}{2}\left(\|w_0\|_\Sigma - \|\widehat{w}\|_\Sigma\right)^2}_{T_1} \underbrace{+w_0^T(\widehat{z} - z_\Sigma(\mu))}_{T_2} \underbrace{-\frac{1}{2}\|\widehat{z} - z_\Sigma(\mu)\|_{\Sigma^{-1}}^2}_{T_3} \underbrace{+\frac{1}{2}\|(\Sigma - \widehat{\Sigma})\widehat{w} + (\widehat{\mu} - \mu)\|_{\Sigma^{-1}}^2}_{T_4}.$$

where $\hat{z}$ is the shorthand for $\hat{z} = z_{\widehat{\Sigma}}(\widehat{\mu})$.

The proof of Lemma 2.6.3 is provided in Appendix 2.10. Based on this decomposition, our goal is to establish the following relations.

$$T_1 \leq 0,\ T_2 \leq 0,\ T_3 \leq 0,\ T_4 \leq O_P\left(r^2 \frac{d}{n}\right).$$

It is obvious that $T_1 \leq 0, T_3 \leq 0$. For the second term T_2, consider $\phi(z) = \|\mu - z\|_{\Sigma^{-1}}^2$. Since $z_\Sigma(\mu) = \operatorname{argmin}_{\|z\|_B \leq \varepsilon} \|\mu - z\|_{\Sigma^{-1}}^2 = \operatorname{argmin}_{\|z\|_B \leq \varepsilon} \phi(z)$, by the first order optimality condition, we have $(z' - z_\Sigma(\mu))^T \nabla\phi(z_\Sigma(\mu)) \leq 0$ holds for any $\|z'\|_B \leq \varepsilon$. Choosing $z' = \hat{z}$ gives:

$$(\mu - z_\Sigma(\mu))^T \Sigma^{-1}(\widehat{z} - z_\Sigma(\mu)) \leq 0 \Leftrightarrow w_0^T(\widehat{z} - z_\Sigma(\mu)) \leq 0.$$

Therefore, $T_2 \leq 0$ as we desired.

The remaining work is to prove that $T_4 \leq O_P\left((1+r)^2 \frac{d}{n}\right)$. By triangle's inequality,

$$\|(\Sigma - \widehat{\Sigma})\widehat{w} + (\widehat{\mu} - \mu)\|_{\Sigma^{-1}} \leq \|(\Sigma - \widehat{\Sigma})\widehat{w}\|_{\Sigma^{-1}} + \|\widehat{\mu} - \mu\|_{\Sigma^{-1}}.$$

Both terms can be controled using covergence of sample mean and covariance. By Lemma 2.6.1, one has

$$\|\widehat{\mu} - \mu\|_{\Sigma^{-1}} \leq O_P\left(\sqrt{\frac{d}{n}}\right),$$

and direct calculations give

$$\begin{aligned}
\|(\Sigma - \widehat{\Sigma})\widehat{w}\|_{\Sigma^{-1}} &= \|(I - \Sigma^{-\frac{1}{2}}\widehat{\Sigma}\Sigma^{-\frac{1}{2}})(\Sigma^{\frac{1}{2}}\widehat{w})\|_2 \\
&\leq \|I - \Sigma^{-\frac{1}{2}}\widehat{\Sigma}\Sigma^{-\frac{1}{2}}\|_{op}\|\Sigma^{\frac{1}{2}}\widehat{w}\|_2 \\
&= O_P\left(\sqrt{\frac{d}{n}}\right)\|\widehat{w}\|_\Sigma.
\end{aligned}$$

Combined pieces together, triangle's inequality further guarantees that

$$\|(\Sigma - \widehat{\Sigma})\widehat{w} + (\widehat{\mu} - \mu)\|_{\Sigma^{-1}} \leq O_P\left(\sqrt{\frac{d}{n}}(\|\widehat{w}\|_\Sigma + 1)\right).$$

Since $\widehat{\mu} \to \mu, \widehat{\Sigma} \to \Sigma$, we have $\widehat{w} \to w_0$, therefore $\|\widehat{w}\|_\Sigma = (1+o(1))\|w_0\|_\Sigma = (\frac{1}{2}+o(1))r$, hence,

$$\begin{aligned} T_4 &= \frac{1}{2}\|(\Sigma - \widehat{\Sigma})\widehat{w} + (\widehat{\mu} - \mu)\|_{\Sigma^{-1}}^2 \\ &\leq \frac{1}{2}\left(O_P(\sqrt{\frac{d}{n}})(\|\widehat{w}\|_\Sigma + 1)\right)^2 \\ &= O_P\left(r^2 \cdot \frac{d}{n}\right). \end{aligned}$$

Putting things together and recall that $r = \Omega(1)$, we have

$$\delta_n = O_P\left(r \cdot \frac{d}{n}\right).$$

Therefore we have completed the proof. □

2.6.2 Proof of Lemma 2.4.1

To prove Lemma 2.4.1, we start with a simple observation: for any classifier f, its standard error on any perturbed distribution $P_{\mu',\Sigma}$ is always a lower bound on robust error of the original distribution $P_{\mu,\Sigma}$, as long as the perturbation has bounded B-norm $\|\mu' - \mu\|_B \leq \varepsilon$:

Lemma 2.6.4. *For any classifier $f : \mathbb{R}^d \to \{-1, +1\}$ and any $\mu' \in \mathbb{R}^d, \|\mu' - \mu\|_B \leq \varepsilon$*

$$R_{\mu,\Sigma}^{B,\varepsilon}(f) \geq R_{\mu',\Sigma}^{\text{std}}(f).$$

Proof. By the definition of robust classification error (2.1), we can decompose the error into two parts: the error on positive class ($y = 1$) and negative class ($y = -1$), namely,

$$\begin{aligned} R_{\mu,\Sigma}^{B,\varepsilon}(f) =& \mathbb{E}_{(x,y)\sim P_{\mu,\Sigma}}[\mathbb{I}\left(\exists\|\delta\|_B \leq \varepsilon, f(x+\delta) \neq y\right)] \\ =& \frac{1}{2}\mathbb{E}_{x\sim N(\mu,\Sigma)}[\mathbb{I}\left(\exists\|\delta\|_B \leq \varepsilon, f(x+\delta) \neq 1\right)] + \frac{1}{2}\mathbb{E}_{x\sim N(-\mu,\Sigma)}[\mathbb{I}\left(\exists\|\delta\|_B \leq \varepsilon, f(x+\delta) \neq -1\right)]. \end{aligned} \tag{2.8}$$

By choosing the adversarial perurbation as $\delta = \mu' - \mu$, we have the error on positive class is lower bounded by:

$$\begin{aligned} \mathbb{E}_{x\sim N(\mu,\Sigma)}[\mathbb{I}\left(\exists\|\delta\|_B \leq \varepsilon, f(x+\delta) \neq 1\right)] \geq& \mathbb{E}_{x\sim N(\mu,\Sigma)}[\mathbb{I}\left(f(x-\mu+\mu') \neq 1\right)] \\ =& \mathbb{E}_{x'\sim N(\mu',\Sigma)}[\mathbb{I}\left(f(x') \neq 1\right)]. \end{aligned} \tag{2.9}$$

Similarly, by choosing $\delta = \mu - \mu'$, we have the error on negative class is lower bounded by:

$$\mathbb{E}_{x\sim N(-\mu,\Sigma)}[\mathbb{I}\left(\exists\|\delta\|_B \leq \varepsilon, f(x+\delta) \neq 1\right)] \geq \mathbb{E}_{x'\sim N(-\mu',\Sigma)}[\mathbb{I}\left(f(x') \neq -1\right)].$$

Hence, combining (2.8) , (2.9) and (2.10), we get

$$\begin{aligned} R_{\mu,\Sigma}^{B,\varepsilon}(f) \geq& \frac{1}{2}\mathbb{E}_{x'\sim N(\mu',\Sigma)}[\mathbb{I}\left(f(x') \neq 1\right)] + \frac{1}{2}\mathbb{E}_{x'\sim N(-\mu',\Sigma)}[\mathbb{I}\left(f(x') \neq -1\right)] \\ =& R_{\mu',\Sigma}^{\text{std}}(f), \end{aligned}$$

where the last step is by the definition of standard error (2.2). Therefore we have completed the proof. □

Next, we show more connections between robust and standard classification. Namely, the robust Bayes classifier of $P_{\mu,\Sigma}$ coincides with the standard Bayes classifier of $P_{\mu-z_\Sigma(\mu),\Sigma}$, as stated in the following Lemma:

Lemma 2.6.5. *Let $z_\Sigma(\mu)$ be the solution of (2.4), then the robust Bayes classifier of $P_{\mu,\Sigma}$, $f_*(x) = \text{sign}(w_0^T x)$, satisfies the following conditions:*

1. $R^{B,\varepsilon}_{\mu,\Sigma}(f_*) = R^{\text{std}}_{\mu-z_\Sigma(\mu),\Sigma}(f_*)$.
2. *f_* is the standard Bayes Optimal Classifier of $P_{\mu-z_\Sigma(\mu),\Sigma}$.*

Proof. Note that by setting $\varepsilon = 0$ in Theorem 2.2.1, we get the characterization of the standard Bayes error and Bayes optimal classifier for conditional Gaussian models. Applying this result for the distribution $P_{\mu-z_\Sigma(\mu),\Sigma}$, we have

1. The standard Bayes Optimal Classifier of $P_{\mu-z_\Sigma(\mu),\Sigma}$ is $\text{sign}((\mu - z_\Sigma(\mu))^T\Sigma^{-1}x)$, which is exactly $f_*(x)$.
2. The standard Bayes error of of $P_{\mu-z_\Sigma(\mu),\Sigma}$ is $\bar{\Phi}(\sqrt{(\mu - z_\Sigma(\mu))^T\Sigma^{-1}(\mu - z_\Sigma(\mu))})$, which is exactly $R^{B,\varepsilon}_{\mu,\Sigma}*$.

Hence we have completed the proof. □

As a direct consequence of Lemma 2.6.4 and Lemma 2.6.5, we have the robust excess risk under $P_{\mu,\Sigma}$ is lower bounded by the standard excess risk under $P_{\mu-z_\Sigma(\mu),\Sigma}$:

Corollary 2.6.1. *For any classifier $f : \mathbb{R}^d \to \{-1, +1\}$,*

$$\begin{aligned} R^{B,\varepsilon}_{\mu,\Sigma}(f) - R^{B,\varepsilon}_{\mu,\Sigma}* &\geq R^{\text{std}}_{\mu-z_\Sigma(\mu),\Sigma}(f) - R^{\text{std}}_{\mu-z_\Sigma(\mu),\Sigma}(f_*) \\ &= R^{\text{std}}_{\mu-z_\Sigma(\mu),\Sigma}(f) - R^{\text{std}}_{\mu-z_\Sigma(\mu),\Sigma}*, \end{aligned}$$

where

$$R^{\text{std}}_{\mu',\Sigma}* = \inf_g R^{\text{std}}_{\mu',\Sigma}(g)$$

is the optimal standard risk.

The last piece of tool needed for proving Lemma 2.4.1 is a mapping from $D_{\text{std}}(r)$ to $D_{B,\varepsilon}(r)$ that keeps the excess risk non-decreasing. This is established via the following lemma:

Lemma 2.6.6. *For any $(\mu', \Sigma) \in D_{\text{std}}(r)$, there exists $(\mu, \Sigma) \in D_{B,\varepsilon}(r)$, such that $\mu - z_\Sigma(\mu) = \mu'$, here $z_\Sigma(\mu)$ is the optimal solution of (2.4).*

Proof. The proof is constructive: we choose $\mu = \mu' + \widetilde{z}_\Sigma(\mu')$, where $\widetilde{z}_\Sigma(\mu')$ is the maximizer of the following convex program (which is maximizing a linear function over a convex set):

$$\widetilde{z}_\Sigma(\mu') = \underset{\|z\|_B \leq \varepsilon}{\text{argmax}}\, \mu' \cdot \Sigma^{-1} z. \tag{2.10}$$

We want to prove that $\mu - z_\Sigma(\mu) = \mu'$. By our choice of μ, we also have $\mu = \mu' + \widetilde{z}_\Sigma(\mu')$. Hence, we only need to prove that

$$\widetilde{z}_\Sigma(\mu') = z_\Sigma(\mu).$$

In other words, we only need to show that $\widetilde{z}_\Sigma(\mu')$ is the minimizer of (2.4).

Since (2.4) is a convex program with a strongly convex objective, it suffices to prove the following first order optimality condition holds for any $\forall \|z'\|_B \leq \varepsilon$:

$$(\mu - \widetilde{z}_\Sigma(\mu'))^T \Sigma^{-1}(z' - \widetilde{z}_\Sigma(\mu')) \leq 0.$$

Since $\mu - \widetilde{z}_\Sigma(\mu') = \mu'$, the inequality is equivalent to:

$$\mu' \cdot \Sigma^{-1} z' \leq \mu' \cdot \Sigma^{-1} \widetilde{z}_\Sigma(\mu'),$$

which is correct by the definition of $\widetilde{z}_\Sigma(\mu')$. Hence we have completed the proof. □

Equipped with Lemma 2.6.6, now we can prove the important lemma:

Proof of Lemma 2.4.1. By Lemma 2.6.6, for any $(\mu', \Sigma) \in D_{\text{std}}(r)$, there exists $(\mu, \Sigma) \in D_{B,\varepsilon}(r)$, such that $\mu - z_\Sigma(\mu) = \mu'$, where $z_\Sigma(\mu)$ is the optimal solution of (2.4). By Corollary 2.6.1, we have the following inequality holds for any fixed $\widehat{f}$:

$$R^{\text{std}}_{\mu',\Sigma}(\widehat{f}) - R^{\text{std}}_{\mu',\Sigma} * \leq R_{\mu,\Sigma}(\widehat{f}) - R^{B,\varepsilon}_{\mu,\Sigma} * .$$

Therefore,

$$R^{\text{std}}_{\mu',\Sigma}(\widehat{f}) - R^{\text{std}}_{\mu',\Sigma} * \leq \max_{(\mu,\Sigma)\in D_{B,\varepsilon}(r)} [R_{\mu,\Sigma}(\widehat{f}) - R^{B,\varepsilon}_{\mu,\Sigma} *].$$

holds for all $(\mu, \Sigma) \in D_{B,\varepsilon}(r)$, which means

$$\max_{(\mu,\Sigma)\in D_{B,\varepsilon}(r)} [R^{B,\varepsilon}_{\mu,\Sigma}(\widehat{f}) - R^{B,\varepsilon}_{\mu,\Sigma} *] \geq \max_{(\mu',\Sigma)\in D_{\text{std}}(r)} [R^{\text{std}}_{\mu',\Sigma}(\widehat{f}) - R^{\text{std}}_{\mu',\Sigma} *].$$

Then, taking minimum over $\widehat{f}$ on both sides proves the theorem.

□

Acknowledgements

Y.W. is supported in part by the NSF grant DMS-2015447 and CCF-2007911. C.D. and P.R. are supported by DARPA via HR00112020006, and NSF via IIS1909816.

The authors would also like to thank Kaizheng Wang for many helpful discussions, Tianle Cai and Justin Khim for pointing us toward the work of [15, 26], and annonymous reviewer for many suggestions about improving the presentation of the paper.

2.7 Proof of Theorem 2.2.1

For completeness, in this section, we present the proof of Theorem 2.2.1. This result follows from combining Theorem 1, Theorem 2 and Lemma 1 in [15]. The proof is mainly a simplified presentation of their proofs (e.g. without using the language of optimal transport) which make some of their results explicit to interpret for our case (e.g. they did not provide the expression for optimal linear classifier, which is useful to our algorithmic results).

To start with, let us define $w_1 := \frac{w_0}{\|w_0\|_\Sigma} = \frac{\Sigma^{-1}(\mu - z_\Sigma(\mu))}{\|\mu - z_\Sigma(\mu)\|_{\Sigma^{-1}}}$ be the normalized version of w_0 so that $\|w_1\|_\Sigma = 1$. The following lemma is implicit in [15]:

Lemma 2.7.1. *Suppose we define*

$$G(z, w) = w^T(\mu - z),$$

then $(z_\Sigma(\mu), w_1)$ *is solution of the following minimax optimization problem:*

$$\min_{\|z\|_B \le \varepsilon} \max_{\|w\|_\Sigma \le 1} G(z, w). \tag{2.11}$$

Proof. We first show that the optimal value of the inner maximization problem can be written as:

$$\max_{\|w\|_\Sigma \le 1} w^T(\mu - z) = \|\mu - z\|_{\Sigma^{-1}}, \tag{2.12}$$

and the maximum is achieved when

$$w = \frac{\Sigma^{-1}(\mu - z)}{\|\mu - z\|_{\Sigma^{-1}}}. \tag{2.13}$$

In fact, for any w such that $\|w\|_\Sigma \le 1$, Cauchy-Schwarz inequality gives

$$\begin{aligned} w^T(\mu - z) = (\Sigma^{1/2}w)^T\Sigma^{-1/2}(\mu - z) &\le \|\Sigma^{1/2}w\|_2\|\Sigma^{-1/2}(\mu - z)\|_2 \\ &= \|w\|_\Sigma\|\mu - z\|_{\Sigma^{-1}} \\ &\le \|\mu - z\|_{\Sigma^{-1}}. \end{aligned}$$

Furthermore, it is easy to check that the choice $w = \frac{\Sigma^{-1}(\mu - z)}{\|\mu - z\|_{\Sigma^{-1}}}$ directly yields $w^T(\mu - z) = \|\mu - z\|_{\Sigma^{-1}}$ achieving the equality. Therefore we have proved (2.12) and (2.13).

Using (2.12), the minimax problem (2.11) therefore simplifies to:

$$\min_{\|z\|_B \le \varepsilon} \|\mu - z\|_{\Sigma^{-1}}.$$

Recall that we define $z_\Sigma(\mu)$ (cf. (2.4)) as

$$z_\Sigma(\mu) = \operatorname*{argmin}_{\|z\|_B \le \varepsilon} \|\mu - z\|^2_{\Sigma^{-1}},$$

which is the optimal solution to this outer minimization problem. Combining with the optimality condition for the inner maximization (2.13), we conclude that $(z_\Sigma(\mu), w_1)$ is solution of the minimax problem (2.11) and complete the proof. □

Corollary 2.7.1. *The following relation is satisfied for quantities w_1 and $z_\Sigma(\mu)$:*

$$w_1^T\mu - \varepsilon\|w_1\|_{B*} = \|\mu - z_\Sigma(\mu)\|_{\Sigma^{-1}}.$$

Proof. Since $G(z,w)$ is linear in both z and w and both constraint sets $\{\|z\|_B \leq \varepsilon\}$ and $\{\|w\|_\Sigma \leq 1\}$ are convex, the minimax problem (2.11) satisfies strong duality by Von Neumann's Minimax Theorem. In other words, we can switch the order of the min and max, namely,

$$\min_{\|z\|_B\leq\varepsilon}\max_{\|w\|_\Sigma\leq 1} G(z,w) = \max_{\|w\|_\Sigma\leq 1}\min_{\|z\|_B\leq\varepsilon} G(z,w),$$

and $(z_\Sigma(\mu), w_1)$ is the solution to both sides. By the stationary condition of the minimax problem,

$$z_\Sigma(\mu) = \operatorname*{argmin}_{\|z\|_B\leq\varepsilon} G(z,w_1).$$

By the definition of dual norm, we also have

$$\min_{\|z\|_B\leq\varepsilon} G(z,w_1) = \min_{\|z\|_B\leq\varepsilon} w_1^T(\mu - z) = w_1^T\mu - \varepsilon\|w_1\|_{B*}.$$

Hence,

$$\|\mu - z_\Sigma(\mu)\|_{\Sigma^{-1}} = G(z_\Sigma(\mu), w_1) = \min_{\|z\|_B\leq\varepsilon} G(z,w_1) = w_1^T\mu - \varepsilon\|w_1\|_{B*}.$$

Thus we completed the proof. □

Now we are ready to prove Theorem 2.2.1.

Proof of Theorem 2.2.1. The proof can be divided into two parts:

1. Show that f_{w_0} has robust risk $R^{B,\varepsilon}_{\mu,\Sigma}(f_{w_0}) = \bar{\Phi}(\|\mu - z_\Sigma(\mu)\|_{\Sigma^{-1}})$.
2. Show that no classifier can achieve robust risk smaller than $\bar{\Phi}(\|\mu - z_\Sigma(\mu)\|_{\Sigma^{-1}})$.

The first part is a consequence of Corollary 2.7.1. In order to see this, we first note that since w_1 is a rescaling of w_0, the induced linear classifiers are the same, hence,

$$R^{B,\varepsilon}_{\mu,\Sigma}(f_{w_0}) = R^{B,\varepsilon}_{\mu,\Sigma}(f_{w_1}).$$

By Lemma 2.6.2, the robust risk of f_{w_1} is

$$R^{B,\varepsilon}_{\mu,\Sigma}(f_{w_1}) = \bar{\Phi}\left(\frac{w_1^T\mu - \varepsilon\|w_1\|_{B*}}{\|w_1\|_\Sigma}\right) = \bar{\Phi}(w_1^T\mu - \varepsilon\|w_1\|_{B*}).$$

By Corollary 2.7.1,

$$\bar{\Phi}(w_1^T\mu - \varepsilon\|w_1\|_{B*}) = \bar{\Phi}(\|\mu - z_\Sigma(\mu)\|_{\Sigma^{-1}}).$$

Therefore, we have proved the first part.

For the second part, we invoke Lemma 2.6.4. By setting $\mu' = \mu - z_\Sigma(\mu)$ in Lemma 2.6.4, we have that for any classifier f,

$$R^{B,\varepsilon}_{\mu,\Sigma}(f) \geq R^{\text{std}}_{\mu - z_\Sigma(\mu),\Sigma}(f).$$

We also know that no classifier can achieve standard risk smaller than the Bayes Risk in $P_{\mu-z_\Sigma(\mu),\Sigma}$. Recall that for a conditional Gaussian kmodel $P_{\mu',\Sigma}$, the standard Bayes Risk is $\bar{\Phi}(\|\mu'\|_{\Sigma^{-1}})$. In other words, for any classifier f, we have

$$R^{\mathrm{std}}_{\mu-z_\Sigma(\mu),\Sigma}(f) \geq \bar{\Phi}(\|\mu - z_\Sigma(\mu)\|_{\Sigma^{-1}}).$$

Combining the two inequalities, we conclude that

$$R^{B,\varepsilon}_{\mu,\Sigma}(f) \geq \bar{\Phi}(\|\mu - z_\Sigma(\mu)\|_{\Sigma^{-1}}) \tag{2.14}$$

holds for all classifiers f. Therefore, we prove the second part and thus complete the proof. □

2.8 Proof of Proposition 2.5.1

Proof of Proposition 2.5.1. Recall that the setting of interest here is $\Sigma = I$ and $\|\cdot\|_B$ corresponds to the ℓ_2 norm. In this setting, we show that $z_\Sigma(\mu)$ has a simplified form. In fact, directly invoking

$$z_\Sigma(\mu) = \operatorname*{argmin}_{\|z\|_B \leq \varepsilon} \|\mu - z\|^2_{\Sigma^{-1}} = \operatorname*{argmin}_{\|z\|_2 \leq \varepsilon} \|\mu - z\|^2_2,$$

gives $z_\Sigma(\mu) = \min(\varepsilon, \|\mu\|_2)\frac{\mu}{\|\mu\|_2}$, and

$$\mu - z_\Sigma(\mu) = \max(0, \frac{\|\mu\|_2 - \varepsilon}{\|\mu\|_2})\mu.$$

From this expression, we can see that when $\varepsilon > \|\mu\|_2$, the Adversarial Signal-to-Noise Ratio of $P_{\mu,\Sigma}$ is $2\|\mu - z_\Sigma(\mu)\|_2 = 0$. Hence, no classifier can achieve accuracy better than $\frac{1}{2}$. Below we only consider the case when $\varepsilon < \|\mu\|_2$.

Recall that we want to compare the minimax rate in adversarial and standard setting. As we showed earlier, the minimax rates are $O(\exp(-\frac{1}{2}\|\mu - z_\Sigma(\mu)\|_2^2)\frac{d}{n})$ and $O(\exp(-\frac{1}{2}\|\mu\|_2^2)\frac{d}{n})$ respectively. The ratio between the two quantities equals to:

$$\frac{\exp(-\frac{1}{2}\|\mu - z_\Sigma(\mu)\|_2^2)\frac{d}{n}}{\exp(-\frac{1}{2}\|\mu - z_\Sigma(\mu)\|_2^2)\frac{d}{n}} = \exp(\frac{1}{2}((\|\mu\|_2 - \varepsilon)^2 - \|\mu\|_2^2)) = \exp(\varepsilon\|\mu\|_2 - \frac{1}{2}\varepsilon^2). \tag{2.15}$$

Since $0 \leq \varepsilon < \|\mu\|_2$, we have

$$\varepsilon\|\mu\|_2 - \frac{1}{2}\varepsilon^2 = \varepsilon(\|\mu\|_2 - \frac{1}{2}\varepsilon) \in \left[\frac{1}{2}\varepsilon\|\mu\|_2, \varepsilon\|\mu\|_2\right].$$

Equipped with the above relation, we are in the position of establishing Proposition 2.5.1.

- When $\varepsilon \leq O(\frac{1}{\|\mu\|_2})$, one has

  $$\varepsilon\|\mu\|_2 - \frac{1}{2}\varepsilon^2 \leq \varepsilon\|\mu\|_2 \leq O(1),$$

  thereby, the adversarial rate is at most $\exp(O(1)) = O(1)$ times slower than the standard rate.

- When $\|\mu\|_2 \geq \Omega(\log d)$ and $\varepsilon \geq \Omega(\frac{\log d}{\|\mu\|_2})$, we conclude

$$\varepsilon\|\mu\|_2 - \frac{1}{2}\varepsilon^2 \geq \frac{1}{2}\varepsilon\|\mu\|_2 \geq \Omega(\log d),$$

the adversarial rate can be slower than the standard rate by an $\Omega(\exp(\log d)) = \Omega(poly(d))$ factor.

- When $\|\mu\|_2 \geq \Omega(\sqrt{d})$ and $\varepsilon \geq \Omega(\frac{d}{\|\mu\|_2})$, it is guaranteed that

$$\varepsilon\|\mu\|_2 - \frac{1}{2}\varepsilon^2 \geq \frac{1}{2}\varepsilon\|\mu\|_2 \geq \Omega(d),$$

therefore, the adversarial rate can be slower than the standard rate by an $\Omega(\exp(d))$ factor.

□

2.9 Improved analysis when Σ is known

Meticulous readers may find a tiny gap between our bounds: the upper bound in Theorem 2.3.1 is $O_P\left(e^{-\frac{1}{8}r^2} \cdot r \cdot \frac{d}{n}\right)$, while the lower bound above gives $\Omega_P\left(e^{-\frac{1}{8}r^2} \cdot \frac{1}{r} \cdot \frac{d}{n}\right)$. Since the dominant factor is $e^{-\frac{1}{8}r^2}$ and $r = \Omega(1)$, this difference is only in a lower order term. This gap is due to the fact that [92] assumed the covariance matrix Σ is known to the learner. In this section, we will prove that under the same assumption, there is a modified version of Algorithm 1 that achieves the truly optimal rate which matches the lower bound even with lower order term in r.

The only modification we made in Algorithm 1 is to replace the sample covariance matrix by the true covariance Σ. The modified algorithm is presented below in Algorithm 3.

Algorithm 3 An improved estimator for w_0 when Σ is known

Input: Data pairs $\{(x_i, y_i)\}_{i=1}^n$.
Output: $\widehat{w}$.
Step 1: Define $\widehat{\mu}$ and $\widehat{\Sigma}$ as

$$\widehat{\mu} := \frac{1}{n}\sum_{i=1}^n y_i x_i, \qquad \widehat{\Sigma} := \Sigma.$$

Step 2: Solve for $\widehat{z}$ in the following

$$\widehat{z} := z_{\widehat{\Sigma}}(\widehat{\mu}) = \underset{\|z\|_B \leq \varepsilon}{\operatorname{argmin}} \|\widehat{\mu} - z\|_{\widehat{\Sigma}^{-1}}^2.$$

Step 3: Define $\widehat{w} := \widehat{\Sigma}^{-1}(\widehat{\mu} - \widehat{z})$.

Theorem 2.9.1. *For the $(\|\cdot\|_B, \varepsilon)$ adversary, suppose the adversarial signal-to-noise ratio $\mathrm{AdvSNR}_{B,\varepsilon}(\mu, \Sigma) = r$, then the excess risk of $f_{\widehat{w}}$ defined in Algorithm 3 is upper bounded by*

$$R^{B,\varepsilon}_{\mu,\Sigma}(f_{\widehat{w}}) - R^{B,\varepsilon}_{\mu,\Sigma}* \leq O_P\left(e^{-\frac{1}{8}r^2} \cdot \frac{1}{r} \cdot \frac{d}{n}\right).$$

This improved rate can be proved by some simple modification to the proof of Theorem 2.3.1.

Proof. We demonstrate that in this setting, there is a stronger upper bound $\delta_n = O_P\left(\frac{1}{r} \cdot \frac{d}{n}\right)$ and the rest of proof follows the same as that of Theorem 2.3.1. To this end, let us recall that by Lemma 2.6.3 and one has the decomposition,

$$\|\widehat{w}\|_\Sigma \delta_n = \underbrace{-\frac{1}{2}\left(\|w_0\|_\Sigma - \|\widehat{w}\|_\Sigma\right)^2}_{T_1} \underbrace{+w_0^T(\widehat{z} - z_\Sigma(\mu))}_{T_2} \underbrace{-\frac{1}{2}\|\widehat{z} - z_\Sigma(\mu)\|^2_{\Sigma^{-1}}}_{T_3} \underbrace{+\frac{1}{2}\|(\Sigma - \widehat{\Sigma})\widehat{w} + (\widehat{\mu} - \mu)\|^2_{\Sigma^{-1}}}_{T_4}.$$

Similar to the proof of Theorem 2.3.1, we shall establish that

$$T_1 \leq 0,\ T_2 \leq 0,\ T_3 \leq 0,\ T_4 \leq O_P\left(\frac{d}{n}\right).$$

Note that the only difference here is that we can now give a tighter upper bound for T_4: $O_P\left(\frac{d}{n}\right)$ instead of $O_P\left(r^2\frac{d}{n}\right)$.

Since $\Sigma = \hat{\Sigma}$, by Lemma 2.6.1, we have

$$T_4 = \frac{1}{2}\|(\Sigma - \widehat{\Sigma})\widehat{w} + (\widehat{\mu} - \mu)\|^2_{\Sigma^{-1}} = \frac{1}{2}\|(\widehat{\mu} - \mu)\|^2_{\Sigma^{-1}} = O_P\left(\frac{d}{n}\right). \tag{2.16}$$

Hence, we have proved that $T_4 = O_P\left(\frac{d}{n}\right)$, and

$$\delta_n = O_P\left(\frac{1}{r} \cdot \frac{d}{n}\right).$$

Therefore we have completed the proof. □

2.10 Proof of Lemma 2.6.3

Proof of Lemma 2.6.3. Recall that our goal is to establish

$$\begin{aligned}\|\widehat{w}\|_\Sigma \delta_n &= \|\widehat{w}\|_\Sigma \|w_0\|_\Sigma - \left(\widehat{w}^T\mu - \varepsilon\|\widehat{w}\|_{B*}\right) \\ &= \underbrace{-\frac{1}{2}\left(\|w_0\|_\Sigma - \|\widehat{w}\|_\Sigma\right)^2}_{T_1} \underbrace{+w_0^T(\widehat{z} - z_\Sigma(\mu))}_{T_2} \underbrace{-\frac{1}{2}\|\widehat{z} - z_\Sigma(\mu)\|^2_{\Sigma^{-1}}}_{T_3} \underbrace{+\frac{1}{2}\|(\Sigma - \widehat{\Sigma})\widehat{w} + (\widehat{\mu} - \mu)\|^2_{\Sigma^{-1}}}_{T_4}.\end{aligned} \tag{2.17}$$

Since $\widehat{w} = \widehat{\Sigma}^{-1}(\widehat{\mu} - z_{\widehat{\Sigma}}(\widehat{\mu}))$, by Theorem 2.2.1, $f_{\widehat{w}}$ is the optimal robust classifier for $P_{\widehat{\mu},\widehat{\Sigma}}$, therefore, one can observe

$$\frac{\widehat{w}^T\widehat{\mu} - \varepsilon\|\widehat{w}\|_{B*}}{\|\widehat{w}\|_{\widehat{\Sigma}}} = \|\widehat{w}\|_{\widehat{\Sigma}}.$$

Hence, direct calculations yield

$$\begin{aligned}
\|\widehat{w}\|_\Sigma \delta_n &= \|w_0\|_\Sigma\|\widehat{w}\|_\Sigma - \|\widehat{w}\|_{\widehat{\Sigma}}^2 - \widehat{w}^T(\mu - \widehat{\mu}) \\
&= \|w_0\|_\Sigma\|\widehat{w}\|_\Sigma - (\widehat{\mu} - \widehat{z})^T\widehat{\Sigma}^{-1}(\widehat{\mu} - \widehat{z}) + (\widehat{\mu} - \widehat{z})^T\widehat{\Sigma}^{-1}(\widehat{\mu} - \mu) \\
&= \|w_0\|_\Sigma\|\widehat{w}\|_\Sigma + \widehat{w}^T(\widehat{z} - \mu).
\end{aligned}$$

Now by use of the relation $\mu = \Sigma w_0 + z_\Sigma(\mu)$, we can further obtain

$$\begin{aligned}
\|\widehat{w}\|_\Sigma \delta_n &= \|w_0\|_\Sigma\|\widehat{w}\|_\Sigma + \widehat{w}^T(\widehat{z} - \Sigma w_0 - z_\Sigma(\mu)) \\
&= \|w_0\|_\Sigma\|\widehat{w}\|_\Sigma - \widehat{w}^T\Sigma w_0 + \widehat{w}^T(\widehat{z} - z_\Sigma(\mu)) \\
&= -\frac{1}{2}\left(\|w_0\|_\Sigma - \|\widehat{w}\|_\Sigma\right)^2 + \frac{1}{2}\|w_0\|_\Sigma^2 + \frac{1}{2}\|\widehat{w}\|_\Sigma^2 - \widehat{w}^T\Sigma w_0 + \widehat{w}^T(\widehat{z} - z_\Sigma(\mu)) \\
&= T_1 + \frac{1}{2}(\widehat{w} - w_0)^T\Sigma(\widehat{w} - w_0) + w_0^T(\widehat{z} - z_\Sigma(\mu)) + (\widehat{w} - w_0)^T(\widehat{z} - z_\Sigma(\mu)) \\
&= T_1 + \frac{1}{2}(\widehat{w} - w_0)^T\Sigma(\widehat{w} - w_0) + T_2 + (\widehat{w} - w_0)^T(\widehat{z} - z_\Sigma(\mu)),
\end{aligned}$$

where the last equality invokes the definitions in expression (2.17). To finish the proof, we make the observation about $\Sigma(\widehat{w} - w_0)$ in the following

$$\begin{aligned}
\Sigma(\widehat{w} - w_0) &= (\Sigma - \widehat{\Sigma})\widehat{w} + (\widehat{\Sigma}\widehat{w} - \Sigma w_0) \\
&= \underbrace{(\Sigma - \widehat{\Sigma})\widehat{w}}_{U_1} + \underbrace{(\widehat{\mu} - \mu)}_{U_2} - \underbrace{(\widehat{z} - z_\Sigma(\mu))}_{U_3} := U_1 + U_2 - U_3.
\end{aligned}$$

Therefore, putting everything together and rearranging terms, it is guaranteed that

$$\begin{aligned}
\|\widehat{w}\|_\Sigma \delta_n &= T_1 + T_2 + \frac{1}{2}(\widehat{w} - w_0)^T\Sigma(\widehat{w} - w_0) + (\widehat{w} - w_0)^T(\widehat{z} - z_\Sigma(\mu)) \\
&= T_1 + T_2 + \frac{1}{2}(\Sigma(\widehat{w} - w_0))^T\Sigma^{-1}(\Sigma(\widehat{w} - w_0)) + (\Sigma(\widehat{w} - w_0))^T\Sigma^{-1}(\widehat{z} - z_\Sigma(\mu)) \\
&= T_1 + T_2 + \frac{1}{2}(U_1 + U_2 - U_3)^T\Sigma^{-1}(U_1 + U_2 - U_3) + (U_1 + U_2 - U_3)\Sigma^{-1}U_3 \\
&= T_1 + T_2 + \frac{1}{2}(U_1 + U_2 - U_3)^T\Sigma^{-1}(U_1 + U_2 + U_3) \\
&= T_1 + T_2 - \frac{1}{2}U_3^T\Sigma^{-1}U_3 + \frac{1}{2}(U_1 + U_2)^T\Sigma^{-1}(U_1 + U_2) \\
&= T_1 + T_2 + T_3 + T_4.
\end{aligned}$$

Thus we have finished the proof. □

Chapter 3

Distributional and Outlier Robust Optimization

3.1 Introduction

Many machine learning tasks require models to perform well under distributional shift, where the training and the testing data distributions are different. One type of distributional shift that arouses great research interest is *subpopulation shift*, where the testing distribution is a specific or the worst-case subpopulation of the training distribution. A wide range of tasks can be modeled as subpopulation shift problems, such as learning for algorithmic fairness [12, 48] where we want to test model's performance on key demographic subpopulations, and learning with class imbalance [52, 68] where we train a classifier on an imbalanced dataset with some minority classes having much fewer samples than the others, and we want to maximize the classifier's accuracy on the minority classes instead of its overall average accuracy.

Distributionally robust optimization (DRO) [47, 103] refers to a family of learning algorithms that minimize the model's loss over the worst-case distribution in a neighborhood of the observed training distribution. Generally speaking, DRO trains the model on the worst-off subpopulation, and when the subpopulation membership is unknown, it focuses on the worst-off training instances, that is, the tail performance of the model. Previous work has shown effectiveness of DRO in subpopulation shift settings, such as algorithmic fairness [58] and class imbalance [155].

However, in our empirical investigations, when we apply DRO to real tasks on modern datasets, we observe that DRO suffers from poor performance and severe instability during training. The issue that DRO is sensitive to outliers has been raised by several previous papers [58, 63, 169] . In this paper, we study the cause of these problems with DRO, and develop approaches to address them.

In particular, we identify and study one key factor that we find directly leads to DRO's sub-optimal behavior: DRO's sensitivity to outliers that widely exist in modern datasets. In general, DRO maximizes a model's tail performance by putting more weights on the "harder" instances, i.e. those which incur higher losses during training. On the one hand, this allows DRO to focus its attention on worst-off sub-populations. But on the other hand, since outliers are intuitively "hard" instances that incur higher losses than inliers, DRO is prone to assign large weights to

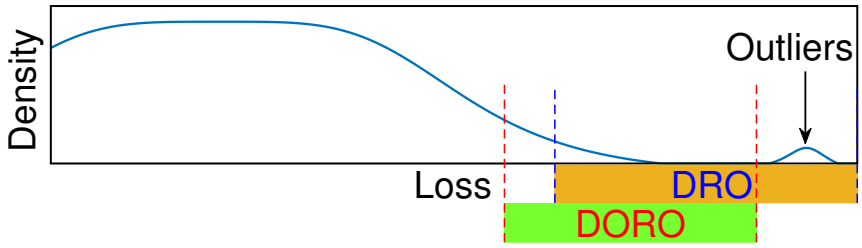


Figure 3.1: DORO avoids overfitting to outliers.

outliers, resulting in both a drop in performance, and training instability. To provide empirical insights into how outliers affect DRO, in Section 3.3 we conducted experiments examining how the performance of DRO changes as we removed or added outliers to the dataset. The results of these experiments indicate that outliers bring about the observed bad performance of DRO. Thus, it is crucial to first enhance the robustness of DRO to outliers before applying it to real-world applications.

To this end, we propose DORO, an outlier robust refinement of DRO which takes inspiration from robust statistics. At the core of this approach is a refined risk function which prevents DRO from overfitting to potential outliers. Intuitively speaking, the new risk function adaptively filters out a small fraction of data with high risk during training, which is potentially caused by outliers. Figure 3.1 illustrates the difference between DRO and DORO. In Section 3.4 we implement DORO for the Cressie-Read family of Rényi divergence, and for our theoretical and empirical study we primarily focus on CVaR-DORO and χ^2-DORO. In Section 3.5 we provide theoretical results guaranteeing that DORO can effectively handle subpopulation shift in the presence of outliers. Then, in Section 3.6 we empirically demonstrate that DORO improves the performance and stability of DRO. We conduct large-scale experiments on three datasets: the tabular dataset COMPAS, the vision dataset CelebA, and the language dataset CivilComments-Wilds.

Contributions Our contributions are summarized below:

- We demonstrate that the sensitivity of DRO to outliers is a direct cause of the irregular behavior of DRO with some intriguing experimental results in Section 3.3.
- We propose and implement DORO as an outlier robust refinement of DRO in Section 3.4. Then, in Section 3.5 we provide theoretical guarantees for DORO.
- We conduct large-scale experiments in Section 3.6 and empirically show that DORO improves the performance and stability of DRO. We also analyze the effect of hyperparameters on DRO and DORO.

Related Work Distributional shift naturally arises in many machine learning applications and has been widely studied in statistics, applied probability and optimization [16, 64, 113, 123]. One common type of distributional shift is *domain generalization* where the training and testing distributions consist of distinct domains, and relevant topics include domain adaptation [108, 145] and transfer learning [106, 129]. Another common type of distributional shift studied in this paper is subpopulation shift, where the two distributions consist of the same group of domains.

Subpopulation shift is closely related to algorithmic fairness and class imbalance. For algorithmic fairness, a number of fairness notions have been proposed, such as individual fairness [48, 160], group fairness [57, 159], counterfactual fairness [83] and Rawlsian Max-Min fairness [58, 116]. The setting of subpopulation shift is most closely related to the Rawlsian Max-Min fairness notion. Several recent papers [58, 104, 155] proposed using DRO to deal with subpopulation shift, but it was also observed that DRO was prone to overfit in practice [117, 119]. [58] raised the open question whether it is possible to design algorithms both fair to unknown latent subpopulations and robust to outliers, and this work answers this question positively.

Outlier robust estimation is a classic problem in statistics starting with the pioneering works of [65, 136]. Recent works in statistics and machine learning [42, 43, 85, 110] provided efficiently computable outlier-robust estimators for high-dimensional mean estimation with corresponding error guarantees. Outliers have a greater effect on the performance of DRO than ERM [63], due to its focus on the tail performance, so removing this negative impact of outliers is crucial for the success of DRO in its real-world applications. One closely related recent work is [88], and DORO can be viewed as a combination of risk-averse and risk-seeking methods discussed in this paper.

3.2 Background

This section provides the necessary background of subpopulation shift and DRO.

3.2.1 Subpopulation Shift

A machine learning task with subpopulation shift requires a model that performs well on the data distribution of each subpopulation. Let the input space be $\mathcal{X}$ and the label space be $\mathcal{Y}$. We are given a training set containing m samples i.i.d. sampled from some data distribution P over $\mathcal{X} \times \mathcal{Y}$. There are K predefined domains (subpopulations) $\mathcal{D}_1, \cdots, \mathcal{D}_K$, each of which is a subset of $\mathcal{X} \times \mathcal{Y}$. For example, in an algorithmic fairness task, domains are demographic groups defined by a number of *protected features* such as race and sex. Let $P_k(z) = P(z|z \in \mathcal{D}_k)$ be the conditional training distribution over $\mathcal{D}_k$, where $z = (x, y)$. The goal is to train a model $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by $\theta \in \Theta$ that performs well over every P_k. Denote the expected risk over P by $\mathcal{R}(\theta; P) = \mathbb{E}_{Z\sim P}[\ell(\theta; Z)]$ where $\ell(\theta; z)$ is a measurable loss function. Then the expected risk over P_k is $\mathcal{R}_k(\theta; P) = \mathbb{E}_{Z\sim P_k}[\ell(\theta; Z)]$. The objective is to minimize the *worst-case risk* defined as

$$\mathcal{R}_{\max}(\theta; P) = \max_{k=1,\cdots,K} \mathcal{R}_k(\theta; P) \tag{3.1}$$

Several different settings were studied by previous work:

Overlapping vs Non-overlapping The *overlapping* setting allows the domains to overlap with each other while *non-overlapping* does not. For example, suppose we have two protected features: race (White and Others) and sex (Male and Female). Under either setting we will have four domains. Under the overlapping setting we will have *White*, *Others*, *Male* and *Female*, while under the non-overlapping setting we will have *White Male*, *White Female*, *Others Male* and

Others Female. All the experiments in this work are conducted under the overlapping setting. Each instance may belong to zero, one or more domains.

Domain-Aware vs Domain-Oblivious Some previous work has assumed that domain memberships of instances are known at least during training. This is called the *domain-aware* setting. However, [58] argue that in many real applications, domain memberships are unknown during training, either because it is hard to extract the domain information from the input, or because it is hard to identify all protected features. Thus, a line of recent work [58, 84] studies the *domain-oblivious* setting, in which the training algorithm does not know the domain membership of any instance (even the number of domains K is unknown). In this work, we focus on the domain-oblivious setting.

3.2.2 Distributionally Robust Optimization (DRO)

Under the domain-oblivious setting, we cannot compute the worst-case risk since we have no access to $\mathcal{D}_1, \cdots, \mathcal{D}_K$. In this case, the framework of DRO instead maximizes the performance over the worst-off subpopulation in general. Specifically, given some divergence D between distributions, DRO aims to minimize the expected risk over the worst-case distribution Q (that is absolutely continuous with respect to training distribution P, so that $Q \ll P$) in a ball w.r.t. divergence D around the training distribution P.

Thus, while empirical risk minimization (ERM) algorithm minimizes the expected risk $\mathcal{R}(\theta; P)$, DRO minimizes the *expected DRO risk* defined as:

$$\mathcal{R}_{D,\rho}(\theta; P) = \sup_{Q \ll P} \{\mathbb{E}_Q[\ell(\theta; Z)] : D(Q \parallel P) \leq \rho\} \tag{3.2}$$

for some $\rho > 0$. Different divergence functions D derive different DRO risks. In this work, we focus on the Cressie-Read family of Rényi divergence [32] formulated as:

$$D_\beta(Q \parallel P) = \int f_\beta(\frac{dQ}{dP}) dP \tag{3.3}$$

where $\beta > 1$, and $f_\beta(t)$ is defined as:

$$f_\beta(t) = \frac{1}{\beta(\beta - 1)} \left(t^\beta - \beta t + \beta - 1\right) \tag{3.4}$$

An advantage of the Cressie-Read family is that it has the following convenient dual characterization (see Lemma 1 of [47] for the proof):

$$\mathcal{R}_{D_\beta,\rho}(\theta; P) = \inf_{\eta \in \mathbb{R}} \left\{ c_\beta(\rho) \mathbb{E}_P[(l(\theta; Z) - \eta)_+^{\beta_*}]^{\frac{1}{\beta_*}} + \eta \right\} \tag{3.5}$$

where $\beta_* = \frac{\beta}{\beta - 1}$, and $c_\beta(\rho) = (1 + \beta(\beta - 1)\rho)^{\frac{1}{\beta}}$.

The following proposition shows that DRO can handle subpopulation shift under the domain-oblivious setting. The only information DRO needs during training is α, the ratio between the size of the smallest domain and the size of the population. See the proof in Appendix 3.8.1.

Proposition 1. *Let* $\alpha = \min_{k=1,\cdots,K} P(\mathcal{D}_k) \leq \exp(-1) \approx 36.8\%$ *be the minimal group size, and define* $\rho = f_\beta(\frac{1}{\alpha})$*, then we have*

$$\mathcal{R}_{\max}(\theta; P) \leq \mathcal{R}_{D_\beta,\rho}(\theta; P). \tag{3.6}$$

While the Cressie-Read formulation only defines the f-divergence for finite $\beta \in (1, +\infty)$, it can be shown that the dual characterization is valid for $\beta = \infty$ as well, for which the DORO risk becomes the well-known *conditional value-at-risk* (CVaR) (See e.g. [47], Example 3). In our theoretical analysis and experiments, we delve into two most widely-used sepecial cases of the Cressie-Read family: (i) $\beta = \infty$, which corresponds to CVaR; (ii) $\beta = 2$, which corresponds to χ^2*-DRO risk* used in [58]. Table 3.1 summarizes the relevant quatities in these two special cases.

Table 3.1: CVaR and χ^2-DRO. α is the ratio between the size of the smallest domain and the size of the population.

	CVaR
β	∞
β_*	1
ρ	$-\log(\alpha)$
$c_\beta(\rho)$	α^{-1}
$D_\beta(Q \parallel P)$	$\sup \log \frac{dQ}{dP}$
DRO Risk	$/home/chen/Dropbox/App/Overleaf/ICML'21 : DRORobusttoOutliers/math_commands.tex_\alpha($

For example, the dual form of CVaR is

$$\text{CVaR}_\alpha(\theta; P) = \inf_{\eta \in \mathbb{R}} \{\alpha^{-1} \mathbb{E}_P[(\ell(\theta; Z) - \eta)_+] + \eta\} \tag{3.7}$$

It is easy to see that the optimal η of (3.7) is the α-quantile of $l(\theta; Z)$ defined as

$$q_\theta(\alpha) = \inf_q \{P_{Z\sim P}(\ell(\theta; Z) > q) \leq \alpha\} \tag{3.8}$$

The dual form (3.7) shows that CVaR in effect minimizes the expected risk on the worst α portion of the training data.

The following corollary of Proposition 1 shows that both $\text{CVaR}_\alpha(\theta; P)$ and $\mathcal{R}_{D_{\chi^2},\rho}(\theta; P)$ are upper bounds of $\mathcal{R}_{\max}(\theta; P)$, so that minimizing either of them guarantees a small worst-case risk (see the proof in Appendix 3.8.2):

Corollary 2. *Let* $\alpha = \min_{k=1,\cdots,K} P(\mathcal{D}_k)$ *be the minimal group size, and* $\rho = \frac{1}{2}(\frac{1}{\alpha} - 1)^2$*. Then*

$$\mathcal{R}_{\max}(\theta; P) \leq \text{CVaR}_\alpha(\theta; P) \leq \mathcal{R}_{D_{\chi^2},\rho}(\theta; P) \tag{3.9}$$

3.3 DRO is Sensitive to Outliers

Although the construction of DRO aims to be effective against subpopulation shift as detailed in the previous section, when applied to real tasks DRO is found to have poor and unstable

performance. After some examination, we pinpoint one direct cause of this phenomenon: the vulnerablity of DRO to outliers that widely exist in modern datasets. In this section, we will provide some intriguing experimental results to show that:

1. DRO methods have poor and unstable performances.
2. Sensitivity to outliers is a direct cause of DRO's poor performance. To support this argument, we show that DRO becomes good and stable on a "clean" dataset constructed by removing the outliers from the original dataset, and new outliers added to this "clean" dataset compromise DRO's performance and stability.

We conduct experiments on COMPAS [86], a recidivism prediction dataset with 5049 training instances (after preprocessing and train-test splitting). We select two features as protected features: race and sex. The two protected features define four overlapping demographic groups: *White*, *Others*, *Male* and *Female*. A two-layer feed-forward neural network with ReLU activations is used as the classification model. We train three models on this dataset with ERM, CVaR and χ^2-DRO. Then we remove the outliers from the training set using the following procedure: We first train a model with ERM, and then remove 200 training instances that incur the highest loss on this model, as outliers are likely to have poorer fit. Then we reinitialize the model, train it on the new training set with ERM, and remove 200 more instances with the highest loss from the new training set. This process is repeated 5 times, so that 1000 training instances are removed and we get a new training set with 4049 instances. Note that this procedure is not guaranteed to remove all outliers and retain all inliers, but is sufficient for the purposes of our demonstration. We then run the three algorithms again on this *same* "clean" training set.

We plot the test accuracies (average and worst across four demographic groups) of the models achieved by the three methods in Figure 3.2. The first row shows the results on the original dataset, and the second row shows the results on the "clean" dataset with the outliers removed. We can see that in the first row, for both average and worst-case test accuracies, the DRO curves are below the ERM curves and jumping up and down, which implies that DRO has lower performance than ERM and is very unstable on the original dataset. However, the third row shows that DRO becomes good and stable after the outliers are removed. For comparison, in the second row we plot the train/test loss on the original dataset of the three methods (for ERM we plot the ERM loss, and for DRO we plot the corresponding DRO loss). The train and test losses of DRO descend steadily while the average and worst-case accuracies jump up and down, which indicates that the instability is not an optimization issue, but rather stems from the existence of outliers. It should also be emphasized that these outliers naturally exist in the original dataset since no outliers have been manually added yet.

To further substantiate our conclusion, we consider another common source of outliers: incorrect labels. We randomly flip 20% of the labels of the "clean" COMPAS dataset with the outliers removed, and run the three training methods again. The results are plotted in the fourth row of Figure 3.2, which shows that while the label noise just slightly influences ERM, it significantly downgrades the performance and stability of the two DRO methods.

Likewise, [63] also found in their experiments that DRO had even lower performance than ERM (see their Table 1). Essentially, DRO methods minimize the expected risk on the worst portion of the training data, which contains a higher density of outliers than the whole population. Training on these instances naturally result in the observed bad performance of DRO.

In the next section we will propose DORO as a solution to the problem revealed by the experiments in this section. We plot the performances of the two DORO algorithms we implement in the last row of Figure 3.2, which compared to the first row shows that DORO improves the performance and stability of DRO on the original dataset.

3.4 DORO

Problem Setting The goal is to train a model on a dataset with outliers to achieve high tail performance on the clean underlying data distribution P. Denote the observed contaminated training distribution by p_{train}. We formulate p_{train} with Huber's ϵ-contamination model [65], in which the training instances are i.i.d. sampled from

$$p_{\text{train}} = (1-\epsilon)P + \epsilon\tilde{P} \tag{3.10}$$

where $\tilde{P}$ is an *arbitrary* outlier distribution, and $0 < \epsilon < \frac{1}{2}$ is the noise level. The objective is to minimize $\mathcal{R}_{\max}(\theta; P)$, the worst-case risk over the clean distribution P.

DORO Risk We propose to minimize the following *expected ϵ-DORO risk*:

$$\begin{aligned}&\mathcal{R}_{D,\rho,\epsilon}(\theta; p_{\text{train}}) = \\ &\inf_{P'}\{\mathcal{R}_{D,\rho}(\theta; P') : \exists \tilde{P}' \text{ s.t. } p_{\text{train}} = (1-\epsilon)P' + \epsilon\tilde{P}'\}\end{aligned} \tag{3.11}$$

The DORO risk is motivated by the following intuition: we would like the algorithm to avoid the "hardest" instances that are likely to be outliers, and the optimal P' of (3.11) consists of the "easiest" $(1-\epsilon)$-portion of the training set given the current model parameters θ. The ϵ in DORO is a hyperparameter selected by the user since the real noise level of the dataset is unknown. Let the real noise level of p_{train} be ϵ_0. For any $\epsilon \geq \epsilon_0$, there exist $\tilde{P}_0$ and $\tilde{P}$ such that $p_{\text{train}} = (1-\epsilon_0)P + \epsilon_0\tilde{P}_0 = (1-\epsilon)P + \epsilon\tilde{P}$, so we only need to make sure that ϵ is not less than the real noise level.

The following proposition provides the formula for computing the DORO risk for the Cressie-Read family (See the proof in Appendix 3.8.3):

Proposition 3. *Let ℓ be a continuous non-negative loss function, and suppose p_{train} is a continuous distribution. Then the formula for computing the DORO risk with D_β is*

$$\begin{aligned}&\mathcal{R}_{D_\beta,\rho,\epsilon}(\theta; p_{\text{train}}) = \\ &\quad\inf_{\eta}\{c_\beta(\rho)\mathbb{E}_{Z\sim p_{\text{train}}}[(\ell(\theta;Z)-\eta)_+^{\beta_*} \mid \\ &\quad P_{Z'\sim p_{\text{train}}}(\ell(\theta;Z') > \ell(\theta;Z)) \geq \epsilon]^{\frac{1}{\beta_*}} + \eta\}\end{aligned} \tag{3.12}$$

Remark In Proposition 3, we assume the continuity of p_{train} to keep the formula simple. For an arbitrary distribution p_{train}, we can obtain a similar formula, but the formula is much more complex than (3.12). The general formula can be found in Appendix 3.8.3.

With this formula, we develop Algorithm 4. In the algorithm, we first order the batch samples according to their training losses, then find the optimal η^* using some numerical method (we use

Algorithm 4 DORO with D_β Divergence

Input: Batch size n, outlier fraction ϵ, minimal group size α
for each iteration **do**
  Sample a batch $z_1, \cdots, z_n \sim p_{\text{train}}$
  Compute losses: $\ell_i = \ell(\theta, z_i)$ for $i = 1, \cdots, n$
  Sort the losses: $\ell_{i_1} \geq \cdots \geq \ell_{i_n}$
  Find $\eta^* = \text{argmin}_\eta F(\theta, \eta)$ where $F(\theta, \eta) = c_\beta(\rho) \cdot [\frac{1}{n - \lfloor \epsilon n \rfloor} \sum_{j=\lfloor \epsilon n \rfloor + 1}^{n} (\ell(\theta; z_{i_j}) - \eta)_+^{\beta_*}]^{1/\beta_*} + \eta$
  Update θ by one step to minimize $\ell(\theta) = F(\theta, \eta^*)$ with some gradient method
end for

Brent's method [21] in our implementation), and finally update θ with some gradient method. Note that generally it is difficult to find the minimizer of the DORO risk for neural networks, and our algorithm is inspired by the ITLM algorithm [122], in which they proved that the optimization converges to ground truth for a few simple problems. Particularly, using the quantities listed in Table 3.1, we can implement CVaR-DORO and χ^2-DORO. In the sections that follow, we will focus on the performances of CVaR-DORO and χ^2-DORO in particular. We denote the CVaR-DORO risk by $\text{CVaR}_{\alpha,\epsilon}(\theta; p_{\text{train}})$, and the χ^2-DORO risk by $\mathcal{R}_{D_{\chi^2},\rho,\epsilon}(\theta; p_{\text{train}})$.

3.5 Theoretical Analysis

Having the DORO algorithms implemented, in this section we prove that DORO can effectively handle subpopulation shift in the presence of outliers. The proofs to the results in this section can be found in Appendix 3.8.4. We summarize our theoretical results as follows:

1. The minimizer of DORO over the contaminated distribution p_{train} achieves a DRO risk close to the minimum over the clean distribution P (Theorem 5). We complement our analysis with information-theoretical lower bounds (Theorem 6) implying that the optimality gaps given by Theorem 5 are optimal.
2. The worst-case risk $\mathcal{R}_{\max}$ over P is upper bounded by the DORO risk over p_{train} times a constant factor (Theorem 7). This result parallels Corollary 2 in the uncontaminated setting and guarantees that minimizing the DORO risk over p_{train} effectively minimizes $\mathcal{R}_{\max}$ over P.

Our results are based on the following lemma which lower bounds the DORO risk over p_{train} by the infimum of the original DRO risk in a TV-ball centered at P:

Lemma 4. *Let* $\mathsf{TV}(P, Q) = \frac{1}{2} \int_{\mathcal{X} \times \mathcal{Y}} |P(z) - Q(z)| dz$ *be the total variation, and* p_{train} *be defined by (3.10). Then the DORO risk can be lower bounded by:*

$$\begin{aligned} &\mathcal{R}_{D,\rho,\epsilon}(\theta; p_{\text{train}}) \geq \\ &\quad \inf_{P''} \{\mathcal{R}_{D,\rho}(\theta; P'') : \mathsf{TV}(P, P'') \leq \frac{\epsilon}{1 - \epsilon}\} \end{aligned} \tag{3.13}$$

The main results we are about to present only require very mild assumptions. For the first result, we assume that ℓ has a bounded $(2k)$-th moment on P, a standard assumption in the robust

statistics literature:

Theorem 5. *Let p_{train} be defined by (3.10). Denote the minimizer of the DORO risk by $\hat{\theta}$. If ℓ is non-negative, and $\ell(\hat{\theta}; Z)$ has a bounded $(2k)$-th moment: $\mathbb{E}_{Z\sim P}[l(\hat{\theta}; Z)^{2k}] = \sigma_{2k}^{2k} < +\infty$, then we have:*

$$\text{CVaR}_\alpha(\hat{\theta}; P) - \inf_\theta \text{CVaR}_\alpha(\theta; P) \le O_{\alpha,k}(1)\sigma_{2k}\epsilon^{1-\frac{1}{2k}} \tag{3.14}$$

and if $k > 1$, then we have:

$$\mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta}; P) - \inf_\theta \mathcal{R}_{D_{\chi^2},\rho}(\theta; P) \le O_{\rho,k}(1)\sigma_{2k}\epsilon^{\left(\frac{1}{2}-\frac{1}{2k}\right)} \tag{3.15}$$

Furthermore, the above optimality gaps are optimal:

Theorem 6. *There exists a pair of (P, p_{train}) where $p_{\text{train}} = (1-\epsilon)P + \epsilon P'$ and P has uniformly bounded $2k$-th moment: $\forall\theta \in \Theta$, $\mathbb{E}_P[l(\theta, Z)^{2k}] \le \sigma_{2k}^{2k}$ such that for any learner with only access to p_{train}, the best achievable error in DRO over P is lower bounded by*

$$\text{CVaR}_\alpha(\hat{\theta}; P) - \inf_{\theta\in\Theta} \text{CVaR}_\alpha(\theta; P) \ge \Omega_{\alpha,k}(1)\sigma_{2k}\epsilon^{1-\frac{1}{2k}} \tag{3.16}$$

$$\mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta}; P) - \inf_{\theta\in\Theta} \mathcal{R}_{D_{\chi^2},\rho}(\theta; P) \ge \Omega_{\rho,k}(1)\sigma_{2k}\epsilon^{\left(\frac{1}{2}-\frac{1}{2k}\right)} \tag{3.17}$$

We make a few remarks on these theoretical results. The $O(\epsilon^{1-\frac{1}{2k}})$ and $O(\epsilon^{\frac{1}{2}-\frac{1}{2k}})$ rates resemble the existing works on robust mean/moment estimation, see e.g. [79, 109]. The robust mean estimation problem can be seen as a special case of CVaR when $\alpha = 1$, where CVaR of any θ is just the mean of $l(\theta, Z)$. On the other hand, the connection between CVaR and robust moment estimation can be built with the dual characterization (3.5): for any *fixed* dual variable η, evaluating the dual is nothing but a robust (β_*-th) moment estimation of the random variable $(l(\theta, Z) - \eta)_+$. However, the problem we are trying to tackle in the above theorems is more challenging, in the sense that (1) DRO risk involves taking infimum over all $\eta \in \mathbb{R}$, but the moments of $(l(\theta, Z) - \eta)_+$ are not *uniformly* bounded for all possible η's; and (2) the optimal dual variable η^* can be very different even for distributions extremely close in total-variation distance. In Appendix 3.8.4 we discuss how to overcome these difficulties in detail.

Our second result is a robust analogue to Corollary 2: we show that the worst-case risk $\mathcal{R}_{\max}$ can be upper bounded by a constant factor times the DORO risk $\text{CVaR}_{\alpha,\epsilon}$, under the very mild assumption that ℓ has a uniformly bounded second moment on P and $\mathcal{R}_{\max}$ is not exceedingly small:

Theorem 7. *Let p_{train} be defined by (3.10). Let $\alpha = \min_{k=1,\cdots,K} P(\mathcal{D}_k)$, and $\rho = \frac{1}{2}(\frac{1}{\alpha} - 1)^2$. If $\ell(\theta; Z)$ is a non-negative loss function with a uniformly bounded second moment: $\mathbb{E}_{Z\sim P}[\ell(\theta; Z)^2] \le \sigma^2$ for all θ, then we have:*

$$\begin{aligned}\mathcal{R}_{\max}(\theta; P) &\le \max\{3\text{CVaR}_{\alpha,\epsilon}(\theta; p_{\text{train}}), 3\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}}\} \\ &\le \max\{3D_{\chi^2,\rho,\epsilon}(\theta; p_{\text{train}}), 3\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}}\}\end{aligned} \tag{3.18}$$

Note that a similar result can be derived under the bounded $2k$-th moment condition with different constants.

3.6 Experiments

In this section, we conduct large-scale experiments on modern datasets. Our results show that DORO improves the performance and stability of DRO. We also analyze the effect of hyperparameters on DRO and DORO.

3.6.1 Setup

Datasets Our goal is to apply DRO to real tasks with subpopulation shift on modern datasets. While many previous work used small tabular datasets such as COMPAS, these datasets are insufficient for our purpose. Therefore, apart from COMPAS, we use two large datasets: CelebA [94] and CivilComments-Wilds [20, 77]. CelebA is a widely used vision dataset with 162,770 training instances, and CivilComments-Wilds is a recently released language dataset with 269,038 training instances. Both datasets are captured in the wild and labeled by potentially biased humans, so they can reveal many challenges we need to face in practice.

We summarize the datasets we use as follows: (i) COMPAS: recidivism prediction, where the target is whether the person will reoffend in two years; (ii) CelebA: human face recognition, where the target is whether the person has blond hair; (iii) CivilComments-Wilds: toxicity identification, where the target is whether the user comment contains toxic contents. All targets are binary. For COMPAS, we randomly sample 70% of the instances to be the training data (with a fixed random seed) and the rest is the validation/testing data. Both CelebA and CivilComments-Wilds have official train-validation-test splits, so we use them directly.

Domain Definition On COMPAS we define 4 domains (subpopulations), and on CelebA and CivilComments-Wilds we define 16 domains for each. Our domain definitions cover several types of subpopulation shift, such as different demographic groups, class imbalance, labeling biases, confounding variables, etc. See Appendix 3.9.1 for details.

Training We use a two-layer feed-forward neural network activated by ReLU on COMPAS, a ResNet18 [61] on CelebA, and a BERT-base-uncased model [41] on CivilComments-Wilds. On each dataset, we run ERM, CVaR, χ^2-DRO, CVaR-DORO and χ^2-DORO. Each algorithm is run 300 epochs on COMPAS, 30 epochs on CelebA and 5 epochs on CivilComments-Wilds. For each method we collect the model achieved at the end of every epoch, and select the best model through validation. (On CivilComments-Wilds we collect 5 models each epoch, one for every $\sim$20% of the training instances.)

Model Selection To select the best model, we assume that the domain membership of each instance is available in the validation set, and select the model with the highest worst-case validation accuracy. This is an oracle strategy since it requires a domain-aware validation set. Over the course of our experiments, we have realized that model selection with no group labels during validation is a very hard problem. On the other hand, model selection has a huge impact on the performance of the final model. We include some preliminary discussions on this issue in

Table 3.2: The average and worst-case test accuracies of the best models achieved by different methods. (%)

Dataset	Method	Average Accuracy	Worst-case Accuracy
COMPAS	ERM	69.31 ± 0.19	68.83 ± 0.18
	CVaR	68.52 ± 0.31	68.22 ± 0.30
	CVaR-DORO	69.38 ± 0.10	69.11 ± 0.05
	χ^2-DRO	67.93 ± 0.40	67.32 ± 0.60
	χ^2-DORO	69.62 ± 0.16	69.22 ± 0.11
CelebA	ERM	95.01 ± 0.38	53.94 ± 2.02
	CVaR	82.83 ± 1.33	66.44 ± 2.34
	CVaR-DORO	92.91 ± 0.48	72.17 ± 3.14
	χ^2-DRO	83.85 ± 1.42	67.76 ± 3.22
	χ^2-DORO	82.18 ± 1.17	68.33 ± 1.79
CivilComments-Wilds	ERM	92.04 ± 0.24	64.62 ± 2.48
	CVaR	89.11 ± 0.76	63.90 ± 4.42
	CVaR-DORO	90.45 ± 0.70	68.00 ± 2.10
	χ^2-DRO	90.08 ± 0.92	65.55 ± 1.51
	χ^2-DORO	90.11 ± 1.09	67.19 ± 2.51

Table 3.3: Standard deviations of average/worst-case test accuracies during training on CelebA. ($\alpha = 0.1$ for CVaR/CVaR-DORO; $\alpha = 0.3$ for χ^2-DRO/χ^2-DORO. $\epsilon = 0.01$) (%)

Method	Average	Worst-case
ERM	0.73 ± 0.06	8.59 ± 0.90
CVaR	11.53 ± 1.72	21.47 ± 0.71
CVaR-DORO	4.03 ± 1.57	16.84 ± 0.91
χ^2-DRO	8.88 ± 2.98	19.06 ± 1.18
χ^2-DORO	1.60 ± 0.34	13.01 ± 1.40

Appendix 3.9.2. Since model selection is not the main focus of this paper, we pose it as an open question.

3.6.2 Results

The 95% confidence intervals of the mean test accuracies on each dataset are reported in Table 3.2. For every DRO and DORO method, we do a grid search to pick the best α and ϵ that achieve the best worst-case accuracy (see the optimal hyperparameters in Appendix 3.9.3). Each experiment is repeated 10 times on COMPAS and CelebA, and 5 times on CivilComments-Wilds with different random seeds. Table 3.2 clearly shows that on all datasets, DORO consistently improves the average and worst-case accuracies of DRO.

Next, we analyze the stability of the algorithms on the CelebA dataset. We use the α that achieves the optimal DRO performance for each of CVaR and χ^2-DRO, and compare them to DORO with the same value of α and $\epsilon = 0.01$. χ^2-DRO achieves its optimal performance with a

bigger α than CVaR because it is less stable. To quantitatively compare the stability, we compute the standard deviations of the test accuracies across epochs and report the results in Table 3.3. To further visualize the training dynamics, we run all algorithms with one fixed random seed, and plot the test accuracies during training in Figures 3.3 and 3.4. Table 3.3 shows that the standard deviation of the test accuracy of DORO is smaller and in Figures 3.3a and 3.4a the DORO curves are flatter than the DRO curves, which implies that DORO improves the stability of DRO. Although it is hard to tell whether DORO has a more stable worst-case accuracy from the figures, our quantitative results in Table 3.3 confirm that DORO has more stable worst-case test accuracies.

3.6.3 Effect of Hyperparameters

In this part, we study how α and ϵ affect the test accuracies of DORO with two experiments on CelebA, providing insight into how to select the optimal hyperparameters.

In the first experiment, we fix $\alpha = 0.2$, and run the two DORO algorithms with different values of ϵ. The results are plotted in Figure 3.5. We can see that for both methods, as ϵ increases, the average accuracy slightly decreases, while the worst-case accuracy first rises and then drops. Both average and worst-case accuracies will drop if ϵ is too big. Moreover, both methods achieve the optimal worst-case accuracy at $\epsilon = 0.005$. We conjecture that the real noise level of the CelebA dataset is around 0.005, and that the optimal ϵ should be close to the real noise level.

In the second experiment, we run DRO and DORO ($\epsilon = 0.01$) with different values of α. The results are plotted in Figure 3.6. First, we observe that for all methods, the optimal α is much bigger than the real α of the dataset. The real α of the CelebA dataset is around 0.008 (see Appendix 3.9.1, Table 3.4), much smaller than those achieving the highest worst-case accuracies in the figures. Second, in all four figures the overall trend of the average accuracy is that it grows with α. Third, both CVaR-DORO and χ^2-DORO achieve the optimal worst-case accuracy at $\alpha = 0.25$, but the worst-case accuracy drops as α goes to 0.3.

3.7 Discussion

In this work we pinpointed one direct cause of the performance drop and instability of DRO: the sensitivity of DRO to outliers in the dataset. We proposed DORO as an outlier robust refinement of DRO, and implemented DORO for the Cressie-Read family of Rényi divergence. We made a positive response to the open question raised by [58] by demonstrating the effectiveness of DORO both theoretically and empirically.

One alternative approach to making DRO robust to outliers is removing the outliers from the dataset via preprocessing. In Section 3.3 we used a simple version of iterative trimming [122] to remove outliers from the training set. Compared to iterative trimming, DORO does not require retraining the model and does not throw away any data. In addition, preprocessing methods such as iterative trimming cannot cope with online data (where new instances are received sequentially), but DORO is still feasible.

The high-level idea of DORO can be extended to other algorithms that deal with subpopulation shift, such as static reweighting [123], adversarial reweighting [63, 84] and group DRO [117].

The implementations might be different, but the basic ideas are the same: to prevent the algorithm from overfitting to potential outliers. We leave the design of such algorithms to future work.

There is one large open question from this work. In our experiments, we found that model selection without domain information in the validation set is very hard. In Appendix 3.9.2 we study several strategies, such as selecting the model with the lowest CVaR risk or the lowest CVaR-DORO risk, but none of them is satisfactory. A recent paper [99] proposed two selection methods Minmax and Greedy-Minmax, but their performances are still much lower than the oracle's (see their Table 2a). [55] also pointed out the difficulty of model selection in domain-oblivious distributional shift tasks. Thus, we believe this question to be fairly non-trivial.

3.8 Proofs

3.8.1 Proof of Proposition 1

We have the following observation: when $\alpha \leq \exp(-1)$, we have:

$$\forall t \in [0, \frac{1}{\alpha}], f_\beta(t) \leq f_\beta(\frac{1}{\alpha}). \tag{3.19}$$

Notice that

$$f'_\beta(t) = \frac{1}{\beta - 1}(t^{\beta-1} - 1) \tag{3.20}$$

Hence, $f'_\beta(t)$ is decreasing when $t \in [0, 1]$ and increasing when $t \in [1, \frac{1}{\alpha}]$. therefore, $f_\beta(t) \leq \max(f_\beta(0), f_\beta(\frac{1}{\alpha}))$. We can further verify that

$$f_\beta(\frac{1}{\alpha}) - f_\beta(0) = \frac{1}{\beta(\beta-1)}\left(\frac{1}{\alpha^\beta} - \frac{\beta}{\alpha}\right) \tag{3.21}$$

which is nonnegaative whenever $\alpha \leq \beta^{-\frac{1}{\beta-1}}$. Since when $\beta > 1$, one always have $\exp(-1) \leq \beta^{-\frac{1}{\beta-1}}$ and we have proved equation 3.19.

Since P_k is a mixture component of P with probability mass at least α, we can see that

$$\frac{dP_k}{dP} \leq \frac{1}{\alpha} \tag{3.22}$$

thus, by equation 3.19,

$$D_\beta(P_k||P) = \int f_\beta(\frac{dP_k}{dP})dP \tag{3.23}$$

$$\leq \int f_\beta(\frac{1}{\alpha})dP \tag{3.24}$$

$$= f_\beta(\frac{1}{\alpha}) \tag{3.25}$$

by the definition of β-DRO risk, we have completed the proof. □

3.8.2 Proof of Corollary 2

For any k, let $p_k = P(\mathcal{D}_k)$, then $P(z) = p_k P(z|\mathcal{D}_k) + (1-p_k)P(z|\overline{\mathcal{D}_k})$ holds for all x. Let $Q = P_k$ and $Q'(z) = \frac{p_k - \alpha}{1-\alpha} P(z|\mathcal{D}_k) + \frac{1-p_k}{1-\alpha} P(z|\overline{\mathcal{D}_k})$. Then $P = \alpha Q + (1-\alpha)Q'$, which implies that $\mathbb{E}_{P_k}[\ell(\theta; Z)] \le \text{CVaR}_\alpha(\theta; P)$. Thus, $\mathcal{R}_{\max}(\theta; P) \le \text{CVaR}_\alpha(\theta; P)$. On the other hand, for any Q such that there exists Q' satisfying $P = \alpha Q + (1-\alpha)Q'$, we have $\frac{dQ}{dP}(z) \le \frac{1}{\alpha}$ a.e., so that $D_{\chi^2}(Q \parallel P) \le \frac{1}{2}(\frac{1}{\alpha} - 1)^2 = \rho$. Thus, $\text{CVaR}_\alpha(\theta; P) \le \mathcal{R}_{D_{\chi^2},\rho}(\theta; P)$. □

3.8.3 Proposition 3

Proof of Proposition 3

By (3.6) and (3.11) we have

$$\begin{aligned}
\mathcal{R}_{D_\beta,\rho,\epsilon}(\theta; p_{\text{train}}) &= \inf_{P'} \left\{ \mathcal{R}_{D_\beta,\rho}(\theta; P') : \exists \tilde{P}' \text{ s.t. } p_{\text{train}} = (1-\epsilon)P' + \epsilon \tilde{P}' \right\} \\
&= \inf_{P',\eta} \left\{ c_\beta(\rho) \mathbb{E}_{P'}[(\ell(\theta; Z) - \eta)_+^{\beta_*}]^{\frac{1}{\beta_*}} + \eta \right\} \\
&= \inf_{\eta} \left\{ c_\beta(\rho) \inf_{P'} \{ [\int_{\mathbb{R}_+} P'((\ell(\theta; Z) - \eta)_+^{\beta_*} > u) du]^{\frac{1}{\beta_*}} \} + \eta \right\}
\end{aligned} \tag{3.26}$$

By $p_{\text{train}} = (1-\epsilon)P' + \epsilon \tilde{P}'$ we have for all ℓ_0,

$$P'(\ell(\theta; Z) \le \ell_0) \le \min \left\{ 1, \frac{1}{1-\epsilon} p_{\text{train}}(\ell(\theta; Z) \le \ell_0) \right\} \tag{3.27}$$

and we can also show that there exists a $P^* = P'$ such that the equality is achieved in (3.27) for all ℓ_0: Since both ℓ and p_{train} are continuous, $p_{\text{train}}(\ell(\theta; z))$ is a continuous function of z for any fixed θ, so there exists an ℓ^* such that $p_{\text{train}}(\ell(\theta; Z) > \ell^*) = \epsilon$. Define

$$P^*(z) = \begin{cases} \frac{1}{1-\epsilon} p_{\text{train}}(z) & , \ell(\theta; z) \le \ell^* \\ 0 & , \ell(\theta; z) > \ell^* \end{cases} \tag{3.28}$$

For (3.28), we have $\int_{\mathcal{X}\times\mathcal{Y}} P^*(z) dz = \frac{1}{1-\epsilon} \int_{\ell(\theta;z)<\ell^*} p_{\text{train}}(z) dz = \frac{1}{1-\epsilon} p_{\text{train}}(\ell(\theta; Z) < \ell^*) = 1$ because $p_{\text{train}}(\ell(\theta; Z) = \ell^*) = 0$, so (3.28) is a distribution function.

Let $v = u^{\frac{1}{\beta_*}}$. Plugging $P^*(\ell(\theta; Z) \le \ell_0) = \min\{1, \frac{1}{1-\epsilon} p_{\text{train}}(\ell(\theta; Z) \le \ell_0)\}$ into (3.26) produces

$$\begin{aligned}
\mathcal{R}_{D_\beta,\rho,\epsilon}(\theta; p_{\text{train}}) &= \inf_{\eta} \left\{ c_\beta(\rho) \left[\int_{\mathbb{R}_+} [1 - P^*((\ell(\theta; Z) - \eta)_+^{\beta_*} \le v^{\beta_*})] dv^{\beta_*} \right]^{\frac{1}{\beta_*}} + \eta \right\} \\
&= \inf_{\eta} \left\{ c_\beta(\rho) \left[\int_{\mathbb{R}_+} [1 - \frac{1}{1-\epsilon} p_{\text{train}}(\ell(\theta; Z) \le \eta + v)]_+ dv^{\beta_*} \right]^{\frac{1}{\beta_*}} + \eta \right\} \\
&= \inf_{\eta} \left\{ c_\beta(\rho) \left[\int_0^{(\ell^* - \eta)_+} \frac{1}{1-\epsilon} [(1-\epsilon) - p_{\text{train}}(\ell(\theta; Z) \le \eta + v)]_+ dv^{\beta_*} \right]^{\frac{1}{\beta_*}} + \eta \right\}
\end{aligned} \tag{3.29}$$

On the other hand, we have

$$\begin{aligned}
&\mathbb{E}_{p_{\text{train}}}[(\ell-\eta)_+^{\beta_*} \mid P_{Z'\sim p_{\text{train}}}(\ell(\theta;Z')>\ell(\theta;Z))\geq\epsilon] \\
=&\frac{1}{1-\epsilon}\int_0^{\ell^*}(u-\eta)_+^{\beta_*}d(p_{\text{train}}(\ell\leq u)) \\
=&\frac{1}{1-\epsilon}\left\{\left[(u-\eta)_+^{\beta_*}p_{\text{train}}(\ell\leq u)\right]_0^{\ell^*}-\int_0^{\ell^*}p_{\text{train}}(\ell\leq u)d((u-\eta)_+^{\beta_*})\right\} \\
=&\frac{1}{1-\epsilon}\left\{(\ell^*-\eta)_+^{\beta_*}(1-\epsilon)-\int_0^{\ell^*}p_{\text{train}}(\ell\leq u)d((u-\eta)_+^{\beta_*})\right\} \\
=&\frac{1}{1-\epsilon}\left\{\int_0^{(\ell^*-\eta)_+}(1-\epsilon)dv^{\beta_*}-\int_0^{(\ell^*-\eta)_+}p_{\text{train}}(\ell\leq\eta+w)dw^{\beta_*}\right\}
\end{aligned} \tag{3.30}$$

where $w=(u-\eta)_+$. Thus, (3.29) is equal to the right-hand side of (3.12). □

Extension to Arbitrary p_{train}

For any distribution p_{train}, we can obtain a similar but more complex formula (3.33). For any p_{train}, there exists an ℓ^* such that $p_{\text{train}}(\ell(\theta;Z)>\ell^*)\leq\epsilon$ and $p_{\text{train}}(\ell(\theta;Z)<\ell^*)\leq 1-\epsilon$. If $p_{\text{train}}(\ell(\theta;Z)=\ell^*)=0$, then the proof above is still correct, so the formula is still (3.12).

Now assume that $p_{\text{train}}(\ell(\theta;Z)=\ell^*)>0$. Similar to (3.28), define

$$P^*(z)=\begin{cases}\frac{1}{1-\epsilon}p_{\text{train}}(z) & ,\ell(\theta;z)<\ell^* \\ \left[1-\frac{1}{1-\epsilon}p_{\text{train}}(\ell(\theta;Z)<\ell^*)\right]/p_{\text{train}}(\ell(\theta;Z)=\ell^*) & ,\ell(\theta;z)=\ell^* \\ 0 & ,\ell(\theta;z)>\ell^*\end{cases} \tag{3.31}$$

Then we still have $P^*(\ell(\theta;Z)\leq\ell_0)=\min\left\{1,\frac{1}{1-\epsilon}p_{\text{train}}(\ell(\theta;Z)\leq\ell_0)\right\}$, so (3.29) still holds. On the other hand, we have

$$\begin{aligned}
&E_{p_{\text{train}}}[(\ell-\eta)_+^{\beta_*} \mid P_{Z'\sim p_{\text{train}}}(\ell(\theta;Z')>\ell(\theta;Z))>\epsilon] \\
=&\frac{1}{p_{\text{train}}(\ell(\theta;Z)<\ell^*)}\left\{\int_0^{(\ell^*-\eta)_+}(1-\epsilon)dv^{\beta_*}-\int_0^{(\ell^*-\eta)_+}p_{\text{train}}(\ell\leq\eta+w)dw^{\beta_*}\right\}
\end{aligned} \tag{3.32}$$

Thus, the formula becomes

$$\begin{aligned}
\mathcal{R}_{D_\beta,\rho,\epsilon}(\theta;p_{\text{train}})=&\inf_\eta\{c_\beta(\rho)(\frac{p_{\text{train}}(\ell<\ell^*)}{1-\epsilon}\mathbb{E}_Z[(\ell(\theta;Z)-\eta)_+^{\beta_*} \mid P_{Z'}(\ell(\theta;Z')>\ell(\theta;Z))>\epsilon] \\
&+\frac{1-p_{\text{train}}(\ell<\ell^*)}{1-\epsilon}(\ell^*-\eta)_+^{\beta_*})^{\frac{1}{\beta_*}}+\eta\}
\end{aligned} \tag{3.33}$$

3.8.4 Proofs of Results in Section 3.5

A Key Technical Lemma

The following lemma will be useful in the analysis of CVaR-DORO and χ^2-DORO: it controls the difference of dual objective in two distributions P, P' by their total variation distance, with the assumption that loss function l has bounded $2k$-th moment under P.

Lemma 8. *For any distributions P, P', non-negative loss function $l(\cdot, Z)$ and $1 \leq \beta_* < 2k$, such that $\mathbb{E}_P[l(\theta, Z)^{2k}] < \infty$, we have*

$$\mathbb{E}_P[(\ell-\eta)_+^{\beta_*}]^{\frac{1}{\beta_*}} \leq \mathbb{E}_{P'}[(\ell-\eta)_+^{\beta_*}]^{\frac{1}{\beta_*}} + \mathbb{E}_P[(l(\theta,Z)-\eta)_+^{2k}]^{\frac{1}{2k}} \mathsf{TV}(P,P')^{\left(\frac{1}{\beta_*}-\frac{1}{2k}\right)} \beta_*^{-\frac{1}{2k}} \cdot \left(\frac{2k}{2k-\beta_*}\right)^{\frac{1}{\beta_*}} \tag{3.34}$$

Proof. By the definition of total variation distance, we have

$$P(\ell(\theta;Z) > u) - P'(\ell(\theta;Z') > u) \leq \mathsf{TV}(P,P') \tag{3.35}$$

holds for any $u \geq 0$.

By Markov's Inequality and the non-negativity of ℓ, we have for any $\eta \geq 0$,

$$P(\ell - \eta > u) \leq \frac{\mathbb{E}[(\ell-\eta)_+^{2k}]}{u^{2k}} := \left(\frac{s_{2k}}{u}\right)^{2k} \tag{3.36}$$

where we introduced the shorthand $s_{2k} := \mathbb{E}[(\ell-\eta)_+^{2k}]^{\frac{1}{2k}}$ Using integration by parts, we can see that:

$$\mathbb{E}_P[(\ell-\eta)_+^{\beta_*}] = \int_\eta^\infty \beta_*(t-\eta)^{(\beta_*-1)} P(\ell \geq t)dt \tag{3.37}$$

$$= \int_0^\infty \beta_* u^{(\beta_*-1)} P(\ell - \eta \geq u)du \tag{3.38}$$

Thus,

$$\begin{aligned}\mathbb{E}_P[(\ell-\eta)_+^{\beta_*}] - \mathbb{E}_{P'}[(\ell-\eta)_+^{\beta_*}] &= \int_0^\infty \beta_* u^{(\beta_*-1)} \left(P(\ell-\eta\geq u) - P'(\ell-\eta\geq u)\right) du \\ &= \left(\int_0^M + \int_M^\infty\right)\left(\beta_* u^{(\beta_*-1)} \left(P(\ell-\eta\geq u) - P'(\ell-\eta\geq u)\right) du\right)\end{aligned} \tag{3.39}$$

Here, M is a positive parameter whose value will be determined later. Next, we will upper bound each of the two integrals separately. By equation 3.39,

$$\int_0^M \beta_* u^{(\beta_*-1)} \left(P(\ell-\eta\geq u) - P'(\ell-\eta\geq u)\right) du \leq \int_0^M \beta_* u^{(\beta_*-1)} \mathsf{TV}(P,P')du \tag{3.40}$$

$$= M^{\beta_*}\mathsf{TV}(P,P'), \tag{3.41}$$

which gives an upper bound for the first integral. For the second integral, notice that $P'(\ell-\eta \geq u)$ is non-negative and use equation 3.36, we have:

$$\int_M^\infty \beta_* u^{(\beta_*-1)} \left(P(\ell-\eta \geq u) - P'(\ell-\eta \geq u)\right) du \leq \int_M^\infty \beta_* u^{(\beta_*-1)} P(\ell-\eta \geq u) du \tag{3.42}$$

$$\leq \int_M^\infty \beta_* u^{(\beta_*-1)} \left(\frac{s_{2k}}{u}\right)^{2k} \tag{3.43}$$

$$= \frac{s_{2k}^{2k}}{2k-\beta_*} \cdot \frac{1}{M^{2k-\beta_*}} \tag{3.44}$$

Therefore, by setting $M = s_{2k}(\mathsf{TV}(P,P')\beta_*)^{-1/2k}$ which minimizes the sum of two terms, we have

$$\mathbb{E}_P[(\ell-\eta)_+^{\beta_*}] - \mathbb{E}_{P'}[(\ell-\eta)_+^{\beta_*}] \leq \inf_{M>0} \left(M^{\beta_*}\mathsf{TV}(P,P') + \frac{s_{2k}^{2k}}{2k-\beta_*} \cdot \frac{1}{M^{2k-\beta_*}}\right) = s_{2k}^{\beta_*}\mathsf{TV}(P,P')^{1-\frac{\beta_*}{2k}}\beta_*^{-\frac{\beta_*}{2k}} \cdot \frac{2k}{2k-\beta_*} \tag{3.45}$$

Using the inequality $(A+B)^{\frac{1}{\beta_*}} \leq A^{\frac{1}{\beta_*}} + B^{\frac{1}{\beta_*}}$ when $\beta_* \geq 1$, we have:

$$\mathbb{E}_P[(\ell-\eta)_+^{\beta_*}]^{\frac{1}{\beta_*}} \leq \mathbb{E}_{P'}[(\ell-\eta)_+^{\beta_*}]^{\frac{1}{\beta_*}} + s_{2k}\mathsf{TV}(P,P')^{\left(\frac{1}{\beta_*}-\frac{1}{2k}\right)}\beta_*^{-\frac{1}{2k}} \cdot \left(\frac{2k}{2k-\beta_*}\right)^{\frac{1}{\beta_*}} \tag{3.46}$$

□

Proof of Lemma 4

For any P' such that $p_{\text{train}} = (1-\epsilon)P' + \epsilon\tilde{P}'$ for some $\tilde{P}'$, let $U = P \wedge P'$, i.e. $U(z) = \min\{P(z), P'(z)\}$ for any $z \in \mathcal{X}\times\mathcal{Y}$. We have

$$(1-\epsilon)U(z) + \epsilon\tilde{P}(z) + \epsilon\tilde{P}'(z) \geq p_{\text{train}}(z) \quad \text{for any } z \in \mathcal{X}\times\mathcal{Y} \tag{3.47}$$

because both $\tilde{P}(z)$ and $\tilde{P}'(z)$ are non-negative. Integrating both sides produces

$$\int_{\mathcal{X}\times\mathcal{Y}} U(z)dz \geq \frac{1-2\epsilon}{1-\epsilon} \tag{3.48}$$

which implies $\mathsf{TV}(P,P') \leq \frac{\epsilon}{1-\epsilon}$. Thus,

$$\mathcal{R}_{D,\rho}(\theta;P') \geq \inf_{P''}\{\mathcal{R}_{D,\rho}(\theta,P'') : \mathsf{TV}(P,P'') \leq \frac{\epsilon}{1-\epsilon}\} \tag{3.49}$$

which together with (3.11) proves (3.13). □

Proof of Theorem 5, Analysis of CVaR-DORO

Proof of Theorem 5, CVaR-DORO. For any θ, by Lemma 4 we have

$$\text{CVaR}_{\alpha,\epsilon}(\theta;p_{\text{train}}) \geq \text{CVaR}_{\alpha,\epsilon}(\hat{\theta};p_{\text{train}}) \geq \inf_{P'}\{\text{CVaR}_\alpha(\hat{\theta};P') : \mathsf{TV}(P,P') \leq \frac{\epsilon}{1-\epsilon}\} \tag{3.50}$$

By Lemma 8, we have for any $\eta \geq 0$ and $\mathsf{TV}(P, P') \leq \frac{\epsilon}{1-\epsilon}$,

$$\begin{aligned}\mathbb{E}_P[(\ell-\eta)_+] - \mathbb{E}_{P'}[(\ell-\eta)_+] &\leq \left(1+\frac{1}{2k-1}\right)\mathbb{E}_P[(\ell-\eta)_+^{2k}]^{\frac{1}{2k}}\mathsf{TV}(P,P')^{1-\frac{1}{2k}} \\ &\leq \left(1+\frac{1}{2k-1}\right)\sigma_{2k}\mathsf{TV}(P,P')^{1-\frac{1}{2k}}\end{aligned} \tag{3.51}$$

Here, we used the fact that $0 \leq (\ell-\eta)_+^{2k} \leq \ell^{2k}$ Moreover, for any $\eta < 0$, $\mathbb{E}_P[(\ell-\eta)_+] - \mathbb{E}_{P'}[(\ell-\eta)_+] = \mathbb{E}_P[(\ell-0)_+] - \mathbb{E}_{P'}[(\ell-0)_+]$ because ℓ is non-negative. So (3.51) holds for all $\eta \in \mathbb{R}$. Thus, by (3.7) we have for any $\eta \in \mathbb{R}$,

$$\mathrm{CVaR}_\alpha(\hat{\theta}; P) \leq \alpha^{-1}\mathbb{E}_P[(\ell-\eta)_+]+\eta \leq \alpha^{-1}\left\{\mathbb{E}_{P'}[(\ell-\eta)_+] + \left(1+\frac{1}{2k-1}\right)\left(\frac{\epsilon}{1-\epsilon}\right)^{1-\frac{1}{2k}}\right\}+\eta \tag{3.52}$$

and taking the infimum over η, we have the following inequality holds for any θ:

$$\mathrm{CVaR}_\alpha(\hat{\theta}; P) \leq \mathrm{CVaR}_\alpha(\hat{\theta}; P') + \left(1+\frac{1}{2k-1}\right)\alpha^{-1}\sigma_{2k}\left(\frac{\epsilon}{1-\epsilon}\right)^{1-\frac{1}{2k}} \tag{3.53}$$

By (3.11) we have $\mathrm{CVaR}_{\alpha,\epsilon}(\theta; p_{\mathrm{train}}) \leq \mathrm{CVaR}_\alpha(\theta; P)$. Thus, by (5.69), taking the infimum over P' yields

$$\mathrm{CVaR}_\alpha(\hat{\theta}; P) \leq \mathrm{CVaR}_{\alpha,\epsilon}(\theta; p_{\mathrm{train}}) + \left(1+\frac{1}{2k-1}\right)\alpha^{-1}\sigma_{2k}\left(\frac{\epsilon}{1-\epsilon}\right)^{1-\frac{1}{2k}} \tag{3.54}$$

$$\leq \mathrm{CVaR}_\alpha(\theta; P) + \left(1+\frac{1}{2k-1}\right)\alpha^{-1}\sigma_{2k}\left(\frac{\epsilon}{1-\epsilon}\right)^{1-\frac{1}{2k}} \tag{3.55}$$

Taking the infimum over θ completes the proof. □

Proof of Theorem 5, Analysis of χ^2-DORO

We begin with a structral lemma about the optimal dual variable η in the dual formulation equation 3.5. Recall that $\beta = \beta_* = 2$ for χ^2 divergence.

Lemma 9. *Let $\eta^*(P)$ be the minimizer of equation 3.5. We have the following characterization about $\eta^*(P)$:*

1. *When $\rho \leq \frac{\mathrm{Var}_P[l(\theta,Z)]}{2\mathbb{E}[l(\theta,Z)]^2}$, we have $\eta^* \leq 0$;*
   Furthermore, the DRO risk and optimal dual variable η^* can be formulated as:

$$\mathcal{R}_{D_{\chi^2},\rho}(\theta; P) = \mathbb{E}_P[l(\theta,Z)] + \sqrt{2\rho\mathrm{Var}_P[l(\theta,Z)]} \tag{3.56}$$

$$\eta^* = \mathbb{E}_P[l(\theta,Z)] - \sqrt{\frac{\mathrm{Var}_P[l(\theta,Z)]}{2\rho}} \tag{3.57}$$

2. *When $\rho \geq \frac{\mathrm{Var}_P[l(\theta,Z)]}{2\mathbb{E}[l(\theta,Z)]^2}$, we have $\eta^* \geq 0$.*

Proof. (1) We will prove that for any $\rho > 0$,

$$\mathcal{R}_{D_{\chi^2},\rho}(\theta; P) \leq \mathbb{E}_P[l(\theta, Z)] + \sqrt{2\rho \mathrm{Var}_P[l(\theta, Z)]} \tag{3.58}$$

and the equality is achievable when $\rho \leq \frac{\mathrm{Var}_P[l(\theta,Z)]}{2\mathbb{E}[l(\theta,Z)]^2}$.

By the definition of χ^2-DRO risk,

$$\mathcal{R}_{D_{\chi^2},\rho}(\theta; P) = \sup_{Q: D_{\chi^2}(Q||P) \leq \rho} E_Q[l(\theta, Z)] \tag{3.59}$$

Let $\mu := \mathbb{E}_P[l(\theta, Z)]$, notice that

$$\begin{aligned}
\mathbb{E}_Q[l(\theta, Z)] &= \mathbb{E}_P[l(\theta, Z)\frac{dQ}{dP}] && (3.60)\\
&= \mathbb{E}_P[l(\theta, Z)] + \mathbb{E}_P[l(\theta, Z)\left(\frac{dQ}{dP} - 1\right)] && (3.61)\\
&= \mu + \mathbb{E}_P[(l(\theta, Z) - \mu)\left(\frac{dQ}{dP} - 1\right)] && (3.62)
\end{aligned}$$

where in the last step we used the fact that $E_P \frac{dQ}{dP} = 1$.

By the definition of χ^2 divergence,

$$\mathbb{E}_P[\left(\frac{dQ}{dP} - 1\right)^2] = 2D_{\chi^2}(Q||P) \leq 2\rho, \tag{3.63}$$

Therefore, by Cauchy-Schwarz inequality,

$$\begin{aligned}
\mathbb{E}_P[(l(\theta, Z) - \mu)\left(\frac{dQ}{dP} - 1\right)] &\leq (\mathbb{E}_P[(l(\theta, Z) - \mu)])^{1/2} \left(\mathbb{E}_P[\left(\frac{dQ}{dP} - 1\right)^2]\right)^{1/2} && (3.64)\\
&\leq \sqrt{\mathrm{Var}_P[l(\theta, Z)] \cdot 2\rho} && (3.65)
\end{aligned}$$

Plug in this upper bound to equation 3.60 completes the proof of equation 3.58.

To see that the equality can be achieved when $\rho \leq \frac{\mathrm{Var}_P[l(\theta,Z)]}{\mathbb{E}[l(\theta,Z)]^2}$, we only need to verify that $\eta = \eta^*$ gives the same dual objective $\mathbb{E}_P[l(\theta, Z)] + \sqrt{2\rho \mathrm{Var}_P[l(\theta, Z)]}$. Since $\eta^* < 0$, we have

$$\begin{aligned}
&\mathbb{E}_P[(l(\theta, Z) - \eta^*)_+^2] && (3.66)\\
=&\mathbb{E}_P[(l(\theta, Z) - \eta^*)^2] && (3.67)\\
=&\mathbb{E}_P[(l(\theta, Z) - \mathbb{E}_P[l(\theta, Z)] + \sqrt{\frac{1}{2\rho}\mathrm{Var}_P[l(\theta, Z)]})^2] && (3.68)\\
=&\mathbb{E}_P[(l(\theta, Z) - \mathbb{E}_P[l(\theta, Z)])^2] + 2\sqrt{\frac{1}{2\rho}\mathrm{Var}_P[l(\theta, Z)]}\mathbb{E}[(l(\theta, Z) - \mathbb{E}_P[l(\theta, Z)])] + \frac{1}{2\rho}\mathrm{Var}_P[l(\theta, Z)] && (3.69)\\
=&\mathrm{Var}_P[l(\theta, Z)] + 0 + + \frac{1}{2\rho}\mathrm{Var}_P[l(\theta, Z)] = \frac{1 + 2\rho}{2\rho}\mathrm{Var}_P[l(\theta, Z)] && (3.70)
\end{aligned}$$

Therefore,

$$\sqrt{1+2\rho}\left(\mathbb{E}_P[(l(\theta,Z)-\eta^*)_+^2]\right)^{1/2}+\eta^* = \frac{1+2\rho}{\sqrt{2\rho}}\sqrt{\mathrm{Var}_P[l(\theta,Z)]}+\mathbb{E}_P[l(\theta,Z)]-\frac{1}{\sqrt{2\rho}}\sqrt{\mathrm{Var}_P[l(\theta,Z)]} \tag{3.71}$$

$$= \mathbb{E}_P[l(\theta,Z)]+\sqrt{2\rho\mathrm{Var}_P[l(\theta,Z)]} \tag{3.72}$$

and we have completed the proof.

(2) Let $g(\eta,P) = \sqrt{1+2\rho}\left(\mathbb{E}_P[(l(\theta,Z)-\eta)_+^2]\right)^{\frac{1}{2}}+\eta$ and recall that $\mathcal{R}_{D_{\chi^2},\rho}(\theta;P) = \inf_{\eta\in\mathbb{R}} g(\eta,P)$. To show that $\eta^*\le 0$, we only need to prove that $g(\eta)\le g(0)$ whenever $\eta<0$, which is equivalent to:

$$\sqrt{1+2\rho}\left(\mathbb{E}_P[(l(\theta,Z)-\eta)^2]\right)^{\frac{1}{2}} \ge \sqrt{1+2\rho}\left(\mathbb{E}_P[l(\theta,Z)^2]\right)^{\frac{1}{2}}-\eta \tag{3.73}$$

Since both sides are non-negative, this inequality is equivalent to:

$$(1+2\rho)\mathbb{E}_P[(l(\theta,Z)-\eta)^2 \ge (1+2\rho)\mathbb{E}_P[l(\theta,Z)^2+\eta^2-2\eta\sqrt{1+2\rho}\left(\mathbb{E}_P[l(\theta,Z)^2]\right)^{\frac{1}{2}} \tag{3.74}$$

After re-organizing terms, it remains to prove

$$2\rho\eta^2-2(1+2\rho)\eta E_P[l(\theta,Z)]+2\eta\sqrt{1+2\rho}\left(\mathbb{E}_P[l(\theta,Z)^2]\right)^{\frac{1}{2}}\ge 0 \tag{3.75}$$

Since $\rho\ge\frac{\mathrm{Var}_P[l(\theta,Z)]}{2\mathbb{E}[l(\theta,Z)]^2}$, we have $(1+2\rho)\ge\frac{\mathbb{E}[l(\theta,Z)^2]}{\mathbb{E}[l(\theta,Z)]^2}$. Therefore,

$$LHS \ge 2\eta\sqrt{1+2\rho}\left(\mathbb{E}_P[l(\theta,Z)^2]\right)^{\frac{1}{2}}-2(1+2\rho)\eta E_P[l(\theta,Z)] \tag{3.76}$$

$$= 2\eta\sqrt{1+2\rho}\left(\left(\mathbb{E}_P[l(\theta,Z)^2]\right)^{\frac{1}{2}}-\sqrt{1+2\rho}E_P[l(\theta,Z)]\right) \tag{3.77}$$

$$\ge 0 \tag{3.78}$$

where in the last step we used the assumption that $\eta\le 0$. Therefore we have completed the proof. □

Having prepared with Lemma 9, we are now ready to prove the χ^2-DORO part of Theorem 5.

Proof of Theorem 5, χ^2-DORO. We will first show that

$$\mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta};P)\le\mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta};P')+\sqrt{1+2\rho}(1+C_\rho)\sigma_{2k}\mathsf{TV}(P,P')^{\left(\frac{1}{2}-\frac{1}{2k}\right)}2^{-\frac{1}{2k}}\cdot\left(\frac{k}{k-1}\right)^{\frac{1}{2}} \tag{3.79}$$

This inequality will be proved by combining two different strategies: when $\eta^*(P')$ is relatively large, we will use an argument based on Lemma 8, similar to what we did in the analysis of CVaR-DORO. Otherwise, when $\eta^*(P')$ is small, we need a different proof which builds upon the structral result Lemma 9.

Define $C_\rho=\frac{\sqrt{1+2\rho}}{2\rho}$. Below we discuss two cases: $\eta^*(P')<-C_\rho\sigma_{2k}$ and $\eta^*(P')\ge-C_\rho\sigma_{2k}$.

Case 1: $\eta^*(P') < -C_\rho \sigma_{2k}$**.** When $\eta^*(P') < -C_\rho\sigma_{2k}$, by Lemma 9, we have

$$\mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta}; P') = \mathbb{E}_{P'}[l(\hat{\theta}, Z)] + \sqrt{2\rho \mathrm{Var}_{P'}[l(\hat{\theta}, Z)]} \tag{3.80}$$

$$\eta^*(P') = \mathbb{E}_{P'}[l(\hat{\theta}, Z)] - \sqrt{\frac{\mathrm{Var}_{P'}[l(\hat{\theta}, Z)]}{2\rho}} < -C_\rho\sigma_{2k} \tag{3.81}$$

Therefore, we can lower bound $\sqrt{\mathrm{Var}_{P'}[l(\hat{\theta}, Z)]}$ as:

$$\sqrt{\mathrm{Var}_{P'}[l(\hat{\theta}, Z)]} \geq \sqrt{2\rho}\mathbb{E}_{P'}[l(\hat{\theta}, Z)] + \sqrt{2\rho}C_\rho\sigma_{2k} \geq \sqrt{2\rho}C_\rho\sigma_{2k}, \tag{3.82}$$

and consequently, we have a lower bound for $\mathcal{R}_{D_{\chi^2},\rho}(\theta; P')$:

$$\mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta}; P') = \mathbb{E}_{P'}[l(\hat{\theta}, Z)] + \sqrt{2\rho \mathrm{Var}_{P'}[l(\hat{\theta}, Z)]} \tag{3.83}$$

$$\geq \sqrt{2\rho \mathrm{Var}_{P'}[l(\hat{\theta}, Z)]} \geq 2\rho C_\rho\sigma_{2k} = \sqrt{1+2\rho}\sigma_{2k} \tag{3.84}$$

On the other hand, by setting the dual variable $\eta = 0$, we have a simple upper bound for $\mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta}; P)$:

$$\mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta}; P) \leq \sqrt{1+2\rho}\mathbb{E}_P[l(\hat{\theta}, Z)^2]^{1/2} \leq \sqrt{1+2\rho}\sigma_{2k} \tag{3.85}$$

Combining equation 3.83 and equation 3.85, we conclude that $\mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta}; P') \geq \mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta}; P)$ and the inequality is trivially true.

Case 2: $\eta^*(P') \geq -C_\rho\sigma_{2k}$ By Lemma 8, we have

$$\mathbb{E}_P[(\ell-\eta)_+^2]^{\frac{1}{2}} \leq \mathbb{E}_{P'}[(\ell-\eta)_+^2]^{\frac{1}{2}} + \mathbb{E}_Z[(l(\theta,Z)-\eta)_+^{2k}]^{\frac{1}{2k}} \mathsf{TV}(P,P')^{\left(\frac{1}{2}-\frac{1}{2k}\right)} 2^{-\frac{1}{2k}} \cdot \left(\frac{k}{k-1}\right)^{\frac{1}{2}} \tag{3.86}$$

holds for any $\eta \in \mathbb{R}$. Since $\eta^*(P') \geq -C_\rho\sigma_{2k}$, we can upper bound the $2k$-th moment $E_Z[(l(\theta, Z) - \eta^*(P'))_+^{2k}]^{\frac{1}{2k}}$ as:

$$E_Z[(l(\theta, Z) - \eta^*(P'))_+^{2k}]^{\frac{1}{2k}} \leq E_Z[(l(\theta, Z) + C_\rho\sigma_{2k})_+^{2k}]^{\frac{1}{2k}} \tag{3.87}$$

$$\leq E_Z[(l(\theta, Z)]^{\frac{1}{2k}} + C_\rho\sigma_{2k} = (1 + C_\rho)\sigma_{2k} \tag{3.88}$$

Hence,

$$\mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta}; P) \leq \sqrt{1+2\rho}\mathbb{E}_P[(\ell - \eta^*(P'))_+^2]^{\frac{1}{2}} + \eta^*(P') \tag{3.89}$$

$$\leq \sqrt{1+2\rho}\mathbb{E}_{P'}[(\ell - \eta^*(P'))_+^2]^{\frac{1}{2}} + \eta^*(P') + \sqrt{1+2\rho}(1 + C_\rho)\sigma_{2k}\mathsf{TV}(P,P')^{\left(\frac{1}{2}-\frac{1}{2k}\right)} 2^{-\frac{1}{2k}} \cdot \left(\frac{k}{k-1}\right. \tag{3.90}$$

$$= \mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta}; P') + \sqrt{1+2\rho}(1 + C_\rho)\sigma_{2k}\mathsf{TV}(P,P')^{\left(\frac{1}{2}-\frac{1}{2k}\right)} 2^{-\frac{1}{2k}} \cdot \left(\frac{k}{k-1}\right)^{\frac{1}{2}} \tag{3.91}$$

Hence, we have proved the inequality equation 3.79. The rest of proof mimics CVaR-DORO. For any θ, by Lemma 4 we have

$$\mathcal{R}_{D_{\chi^2},\rho,\varepsilon}(\theta; p_{\text{train}}) \geq \mathcal{R}_{D_{\chi^2},\rho,\varepsilon} \geq \inf_{P'}\{\mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta}; P') : \mathsf{TV}(P, P') \leq \frac{\epsilon}{1-\epsilon}\} \tag{3.92}$$

By (3.11) we have $\mathcal{R}_{D_{\chi^2},\rho,\varepsilon}(\theta; p_{\text{train}}) \leq \mathcal{R}_{D_{\chi^2},\rho}(\theta; P)$. Thus, by (5.69), taking the infimum over P' yields

$$\mathcal{R}_{D_{\chi^2},\rho}(\hat{\theta}; P) \leq \mathcal{R}_{D_{\chi^2},\rho,\epsilon}(\theta; p_{\text{train}}) + \sqrt{1+2\rho}(1+C_\rho)\sigma_{2k}\left(\frac{\epsilon}{1-\epsilon}\right)^{\left(\frac{1}{2}-\frac{1}{2k}\right)} 2^{-\frac{1}{2k}} \cdot \left(\frac{k}{k-1}\right)^{\frac{1}{2}} \tag{3.93}$$

$$\leq \mathcal{R}_{D_\beta,\rho}(\theta; P) + \sqrt{1+2\rho}(1+C_\rho)\sigma_{2k}\left(\frac{\epsilon}{1-\epsilon}\right)^{\left(\frac{1}{2}-\frac{1}{2k}\right)} 2^{-\frac{1}{2k}} \cdot \left(\frac{k}{k-1}\right)^{\frac{1}{2}} \tag{3.94}$$

Taking the infimum over θ completes the proof. □

Proof of Theorem 6

We consider an optimization problem with the parameter space restricted to only two possible values $\Theta = \{\theta_0, \theta_1\}$. Our proof is constructive, which relies on the following distribution $P_{M,\Delta,\varepsilon}$:

$$l(\theta_0, Z) = 0, \quad l(\theta_1, Z) = \Delta \quad w.p. \quad (1-\varepsilon) \tag{3.95}$$

$$l(\theta_0, Z) = M, \quad l(\theta_1, Z) = \Delta \quad w.p. \quad \varepsilon \tag{3.96}$$

here M, Δ are some non-negative parameters whose value to be determined later and the probability is taken over the randomness of Z.

We have the following characterization of CVaR and χ^2-DRO risk:

Lemma 10 (DRO Risk of $P_{M,\Delta,\varepsilon}$)**.** *Assume that $\alpha \geq \varepsilon$ and $1 + 2\rho \leq \frac{1}{\varepsilon}$, we have the following closed-form expressions for CVaR and χ^2-DRO risk:*

$$\text{CVaR}_\alpha(\theta_0; P_{M,\Delta,\varepsilon}) = \frac{M\varepsilon}{\alpha} \tag{3.97}$$

$$\text{CVaR}_\alpha(\theta_1; P_{M,\Delta,\varepsilon}) = \Delta \tag{3.98}$$

and

$$\mathcal{R}_{D_{\chi^2},\rho}(\theta_0; P_{M,\Delta,\varepsilon}) = M\varepsilon + M\sqrt{2\rho\varepsilon(1-\varepsilon)} \tag{3.99}$$

$$\mathcal{R}_{D_{\chi^2},\rho}(\theta_1; P_{M,\Delta,\varepsilon}) = \Delta \tag{3.100}$$

Proof. Since $l(\theta_1, Z)$ is always a constant Δ, it's immediate to see $\text{CVaR}_\alpha(\theta_1; P_{M,\Delta,\varepsilon}) = \mathcal{R}_{D_{\chi^2},\rho}(\theta_1; P_{M,\Delta,\varepsilon}) = \Delta$. Hence we only need to focus on θ_0.

By the dual formulation of DRO risk, we have $\text{CVaR}_\alpha(\theta_0; P_{M,\Delta,\varepsilon}) = \inf_{\eta\in\mathbb{R}} h(\eta)$ and $\mathcal{R}_{D_{\chi^2},\rho}(\theta_0; P_{M,\Delta,\varepsilon}) = \inf_{\eta\in\mathbb{R}} g(\eta)$, where we use the shorthand $g(\eta)$ and $h(\eta)$ for

$$g(\eta) := \sqrt{1+2\rho}\left(\mathbb{E}_P[(l(\theta,Z)-\eta)_+^2]\right)^{\frac{1}{2}} + \eta \tag{3.101}$$

$$h(\eta) = \frac{1}{\alpha}\mathbb{E}_P[(l(\theta,Z)-\eta)_+] + \eta \tag{3.102}$$

Direct calculation gives:

$$g(\eta) = \begin{cases} \sqrt{1+2\rho}\sqrt{(\eta-\epsilon M)^2 + \varepsilon(1-\varepsilon)M^2} + \eta, & \text{for } \eta < 0 \\ \sqrt{\varepsilon(1+2\rho)}(M-\eta) + \eta, & \text{for } 0 \le \eta \le M \\ \eta, & \text{for } \eta > M \end{cases} \tag{3.103}$$

and

$$h(\eta) = \begin{cases} \frac{M\varepsilon-\eta}{\alpha} + \eta, & \text{for } \eta < 0 \\ \frac{\varepsilon(M-\eta)}{\alpha} + \eta, & \text{for } 0 \le \eta \le M \\ \eta, & \text{for } \eta > M \end{cases} \tag{3.104}$$

Therefore, when $\alpha \ge \varepsilon$ and $1+2\rho \le \frac{1}{\varepsilon}$, we have

$$\text{CVaR}_\alpha(\theta_0; P_{M,\Delta,\varepsilon}) = \inf_{\eta\in\mathbb{R}} h(\eta) \qquad = h(0) = \frac{M\varepsilon}{\alpha} \tag{3.105}$$

$$\mathcal{R}_{D_{\chi^2},\rho}(\theta_0; P_{M,\Delta,\varepsilon}) = \inf_{\eta\in\mathbb{R}} g(\eta) \quad = g(\varepsilon M - \frac{M\sqrt{\varepsilon(1-\varepsilon)}}{\sqrt{2\rho}}) = M\varepsilon + M\sqrt{2\rho\varepsilon(1-\varepsilon)} \tag{3.106}$$

and we have completed the proof. □

Equipped with Lemma 10, we are now ready to prove the main lower bound Theorem 6.

Proof of Theorem 6. Consider $p_{\text{train}} = P_{M,\Delta,\varepsilon}$. We have two different ways to decompose p_{train} into mixture of two distributions:

$$p_{\text{train}} = P_{M,\Delta,\varepsilon} = (1-\varepsilon)P_{M,\Delta,\varepsilon} + \varepsilon P_{M,\Delta,\varepsilon} = (1-\varepsilon)P_{0,\Delta,0} + \varepsilon P_{M,\Delta,0} \tag{3.107}$$

In other words, with only access to $p_{\text{train}} = P_{M,\Delta,\varepsilon}$, the learner cannot distinguish the following two possibilities:

- (a) The clean distribution is $P = P_{M,\Delta,\varepsilon}$, and the outlier distribution is $P' = P_{M,\Delta,\varepsilon}$.
- (b) The clean distribution is $Q = P_{0,\Delta,1}$, and the outlier distribution is $Q' = P_{M,\Delta,1}$.

Furthermore, as long as $M \le \sigma_{2k}\varepsilon^{-\frac{1}{2k}}$ and $\Delta \le \sigma_{2k}$, both P and Q satisfy the bounded $2k$-th moment condition $\mathbb{E}[l(\theta,Z)^{2k}] \le \sigma_{2k}^{2k}$. With our construction below, we can ensure that θ_1 is $\Theta(\Delta)$-suboptimal under P, while θ_0 is $\Theta(\Delta)$-suboptimal under Q. Therefore, in the worst case scenario, it's impossible for the learner to find a solution with $O(\Delta)$ sub-optimality gap under both P and Q.

CVaR lower bound Assume that $\alpha \geq \frac{1}{2}\varepsilon^{1-\frac{1}{2k}}$. Let $M = \sigma_{2k}\varepsilon^{-\frac{1}{2k}}, \Delta = \sigma_{2k}\frac{\varepsilon^{1-\frac{1}{2k}}}{2\alpha} \leq \sigma_{2k}$. Recall that $P = P_{M,\Delta,\varepsilon}$, by Lemma 10, we have:

$$\text{CVaR}_\alpha(\theta_0; P) = \frac{M\varepsilon}{\alpha} = \frac{\sigma_{2k}}{\alpha}\varepsilon^{1-\frac{1}{2k}} = 2\Delta \tag{3.108}$$

$$\text{CVaR}_\alpha(\theta_1; P) = \Delta \tag{3.109}$$

Therefore,

$$\text{CVaR}_\alpha(\theta_0; P) - \inf_{\theta\in\Theta} \text{CVaR}_\alpha(\theta; P) = \Delta = \Omega(\frac{1}{\alpha}\sigma_{2k}\varepsilon^{1-\frac{1}{2k}}) \tag{3.110}$$

For $Q = P_{0,\Delta,1}$, both $l(\theta_0, Z)$ and $l(\theta_1, Z)$ are constants, and hence

$$\text{CVaR}_\alpha(\theta_0; Q) = 0 \tag{3.111}$$

$$\text{CVaR}_\alpha(\theta_1; Q) = \Delta \tag{3.112}$$

and

$$\text{CVaR}_\alpha(\theta_1; Q) - \inf_{\theta\in\Theta} \text{CVaR}_\alpha(\theta; Q) = \Delta = \Omega(\frac{1}{\alpha}\sigma_{2k}\varepsilon^{1-\frac{1}{2k}}) \tag{3.113}$$

Combining equation 3.110 and equation 3.113 completes the proof.

χ^2-DRO lower bound Assume that $\rho = O(\varepsilon^{\frac{1}{k}-1})$. Let $M = \sigma_{2k}\varepsilon^{-\frac{1}{2k}}, \Delta = \frac{M}{2}\left(\varepsilon + \sqrt{2\rho\varepsilon(1-\varepsilon)}\right) \leq \sigma_{2k}$. Recall that $P = P_{M,\Delta,\varepsilon}$, by Lemma 10, we have:

$$\mathcal{R}_{D_{\chi^2},\rho}(\theta_0; P) = M\varepsilon + M\sqrt{2\rho\varepsilon(1-\varepsilon)} = 2\Delta \tag{3.114}$$

$$\mathcal{R}_{D_{\chi^2},\rho}(\theta_1; P) = \Delta \tag{3.115}$$

Therefore,

$$\mathcal{R}_{D_{\chi^2},\rho}(\theta_0; P) - \inf_{\theta\in\Theta} \mathcal{R}_{D_{\chi^2},\rho}(\theta; P) = \Delta = \Omega(\sigma_{2k}\sqrt{\rho}\varepsilon^{\frac{1}{2}-\frac{1}{2k}}) \tag{3.116}$$

For $Q = P_{0,\Delta,1}$, both $l(\theta_0, Z)$ and $l(\theta_1, Z)$ are constants, and hence

$$\text{CVaR}_\alpha(\theta_0; Q) = 0 \tag{3.117}$$

$$\text{CVaR}_\alpha(\theta_1; Q) = \Delta \tag{3.118}$$

and

$$\text{CVaR}_\alpha(\theta_0; Q) - \inf_{\theta\in\Theta} \text{CVaR}_\alpha(\theta; Q) = \Delta = \Omega(\sigma_{2k}\sqrt{\rho}\varepsilon^{\frac{1}{2}-\frac{1}{2k}}) \tag{3.119}$$

Combining equation 3.116 and equation 3.119 completes the proof.

□

Proof of Theorem 7

By Lemma 8, for any P' such that $\mathsf{TV}(P, P') \leq \frac{\epsilon}{1-\epsilon}$,

$$\mathrm{CVaR}_\alpha(\theta; P) - \mathrm{CVaR}_\alpha(\theta; P') \leq 2\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}} \tag{3.120}$$

By Proposition 2, if $\mathcal{R}_{\max}(\theta; P) > 3\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}}$, then $\mathrm{CVaR}_\alpha(\theta; P) > 3\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}}$, which implies that

$$\frac{\mathrm{CVaR}_\alpha(\theta; P')}{\mathcal{R}_{\max}(\theta; P)} \geq \frac{\mathrm{CVaR}_\alpha(\theta; P')}{\mathrm{CVaR}_\alpha(\theta; P)} = 1 - \frac{\delta}{\mathrm{CVaR}_\alpha(\theta; P)} \geq 1 - \frac{2\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}}}{3\alpha^{-1}\sigma\sqrt{\frac{\epsilon}{1-\epsilon}}} = \frac{1}{3} \tag{3.121}$$

holds for any P' such that $\mathsf{TV}(P, P') \leq \frac{\epsilon}{1-\epsilon}$. By Lemma 4, taking the infimum over P' yields the first inequality of (3.18). Moreover, by Proposition 2, for any θ and P', $D_{\chi^2,\rho}(\theta; P') \geq \mathrm{CVaR}_\alpha(\theta; P')$, which together with (3.121) yields the second inequality of (3.18). □

3.9 Experiment Details

3.9.1 Domain Definition

One important decision we need to make when we design a task with subpopulation shift is how to define the domains (subpopulations). We refer our readers to the Wilds paper [77], which discusses in detail the desiderata and considerations of domain definition, and defines 16 domains on the CivilComments-Wilds dataset which we use directly. The authors selected 8 features such as race, sex and religion, and crossed them with the two classes to define the 16 domains. Such a definition naturally covers class imbalance. There is no official domain definition on CelebA, so we define the domains on our own. Following their approach, on CelebA we also select 8 features and cross them with the two classes to compose the 16 domains. Our definition is inspired by [117], but we cover more types of subpopulation shift apart from demographic differences.

We select 8 features on CelebA: *Male*, *Female*, *Young*, *Old*, *Attractive*, *Not-attractive*, *Straight-hair* and *Wavy-hair*. We explain why we select these features as follows:

- The first four features cover sex and age, two protected features widely used in algorithmic fairness papers.
- We select the next two features in order to cover labeling biases, biases induced by the labelers into the dataset. Among the 40 features provided by CelebA, the *Attractive* feature is the most subjective one. Table 3.4 shows that among the people with blond hair, more than half are labeled *Attractive*; while among the other people, more than half are labeled *Not-attractive*. It might be that the labelers consider blond more attractive than other hair colors, or it might be that the labelers consider females more attractive than males, and it turns out that more females have blond hair than males in this dataset. Although the reason behind is unknown, we believe that these two features well represent the labeling biases in this dataset, and should be taken into consideration.

Table 3.4: Number of training instances in each domain of CelebA and CivilComments-Wilds.

CelebA	Blond	Others
Male	1387	66874
Female	22880	71629
Young	20230	106558
Old	4037	31945
Attractive	17008	66595
Not-attractive	7259	71908
Straight-hair	5178	28769
Wavy-hair	11342	40640
Total	162770	

CivilComments-Wilds	Toxic	Non-toxic
Male	4437	25373
Female	4962	31282
LGBTQ	2265	6155
Christian	2446	24292
Muslim	3125	10829
Other Religions	1003	5541
Black	3111	6785
White	4682	12016
Total	269038	

- We select the last two features in order to cover confounding variables, features the model uses to do classification that should have no correlation with the target by prior knowledge. Since the target is the hair color, a convolutional network trained on this dataset would focus on the hair of the person, so we conjecture that the output of the convolutional network is highly correlated with the hair style. In our experiments, we find that models trained with ERM misclassify about 20% of the test instances with blond straight hair, much more than the other three combinations.

Table 3.4 lists the number of training instances in each domain of each dataset. Each instance may belong to zero, one or more domains. In CivilComments-Wilds, the aggregated group size of the 16 groups is less than the total number 269,038, because most online comments do not contain sensitive words.

3.9.2 Model Selection

In Section 3.6 we assume access to a domain-aware validation set, which is not available in real domain-oblivious tasks. In this part we study several domain-oblivious model selection strategies, and discuss why model selection is hard.

We study the following model selection strategies:

- Max Average Accuracy: The model with the highest average accuracy in validation.
- Min CVaR: The model with the lowest CVaR risk ($\alpha = 0.2$) over the validation set.
- Min CVaR-DORO: The model with the lowest CVaR-DORO risk ($\alpha = 0.2, \epsilon = 0.005$) over the validation set.

Note that selecting the model that achieves the highest average accuracy over the worst α portion of the data is almost equivalent to the Max Average Accuracy strategy because the model with the highest average accuracy over the population also achieves the highest accuracy on the worst α portion (see e.g. [63], Theorem 1).

We conduct experiments on CelebA and report the results in Table 3.5. From the table we draw the following conclusions:

Table 3.5: The average and worst-case test accuracies of the best models selected by different strategies. (%)

Training Algorithm	Model Selection	Average Accuracy	Worst-case Accuracy
ERM	Oracle	95.01 ± 0.38	53.94 ± 2.02
	Max Avg Acc	95.65 ± 0.05	45.83 ± 1.87
	Min CVaR	95.68 ± 0.04	44.83 ± 2.74
	Min CVaR-DORO	95.69 ± 0.04	44.50 ± 2.72
CVaR ($\alpha = 0.2$)	Oracle	95.52 ± 0.08	49.94 ± 3.36
	Max Avg Acc	95.74 ± 0.06	39.28 ± 3.58
	Min CVaR	95.79 ± 0.05	38.67 ± 2.06
	Min CVaR-DORO	95.81 ± 0.05	38.83 ± 2.05
CVaR-DORO ($\alpha = 0.2, \epsilon = 0.005$)	Oracle	92.91 ± 0.48	72.17 ± 3.14
	Max Avg Acc	95.60 ± 0.05	45.39 ± 3.22
	Min CVaR	95.58 ± 0.06	39.83 ± 2.37
	Min CVaR-DORO	95.56 ± 0.07	41.28 ± 3.26
χ^2-DRO ($\alpha = 0.2$)	Oracle	82.44 ± 1.22	63.36 ± 2.51
	Max Avg Acc	90.70 ± 0.26	20.67 ± 3.86
	Min CVaR	87.28 ± 2.05	21.44 ± 11.13
	Min CVaR-DORO	89.16 ± 1.41	25.50 ± 9.14
χ^2-DORO ($\alpha = 0.2, \epsilon = 0.005$)	Oracle	80.73 ± 1.41	65.36 ± 1.02
	Max Avg Acc	90.06 ± 0.57	22.06 ± 5.82
	Min CVaR	84.37 ± 4.08	29.83 ± 12.10
	Min CVaR-DORO	88.76 ± 0.81	23.61 ± 7.45

1. For every training algorithm, the oracle strategy achieves a much higher worst-case test accuracy than the other three strategies, and the gap between the oracle and the non-oracle strategies for DRO and DORO is larger than ERM. While it is expected that the oracle achieves a higher worst-case accuracy, the large gap indicates that there is still huge room for improvement.
2. For χ^2-DRO/DORO, Min CVaR and Min CVaR-DORO work better than Max Average Accuracy. However, for the other three algorithms, Max Average Accuracy is better. This shows that model selection based on CVaR and selection based on CVaR-DORO are not good strategies.
3. With the three non-oracle strategies, ERM achieves the highest worst-case test accuracy. This does not mean that DRO and DORO are not as good as ERM, but suggests that we need other model selection strategies that work better with DRO and DORO.

The reason why Min CVaR is not a good strategy is that CVaR does not decrease monotonically with $\mathcal{R}_{\max}$. Corollary 2 only guarantees that CVaR is an upper bound of $\mathcal{R}_{\max}$, but the θ that achieves the minimum CVaR does not necessarily have the smallest $\mathcal{R}_{\max}$. For the same reason,

Min CVaR-DORO is not a good strategy either.

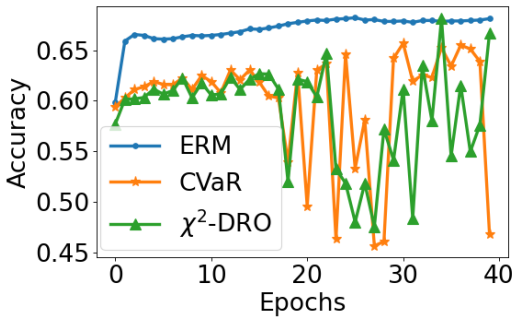
Model selection under the domain oblivious setting is a very difficult task. In fact, Theorem 1 of [63] implies that no strategy can be provably better than Max Average Accuracy under the domain-oblivious setting, i.e. for any model selection strategy, there always exist $\mathcal{D}_1, \cdots, \mathcal{D}_K$ such that the model it selects is not better than the model selected by the Max Average Accuracy strategy. Thus, to design a provably model selection strategy, prior knowledge or reasonable assumptions on the domains are necessary.

3.9.3 Training Hyperparameters

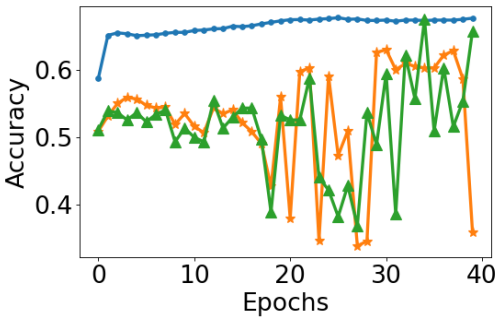
On the COMPAS dataset, we use a two-layer feed-forward neural network activated by ReLU as the classification model. For optimization we use ASGD with learning rate 0.01. The batch size is 128. The hyperparameters we used in Table 3.2 were: $\alpha = 0.5$ for CVaR; $\alpha = 0.5$, $\epsilon = 0.2$ for CVaR-DORO; $\alpha = 0.5$ for χ^2-DRO; $\alpha = 0.5$, $\epsilon = 0.2$ for χ^2-DORO.

On the CelebA dataset, we use a standard ResNet18 as the classification model. For optimization we use momentum SGD with learning rate 0.001, momentum 0.9 and weight decay 0.001. The batch size is 400. The hyperparameters we used in Table 3.2 were: $\alpha = 0.1$ for CVaR; $\alpha = 0.2$, $\epsilon = 0.005$ for CVaR-DORO; $\alpha = 0.25$ for χ^2-DRO; $\alpha = 0.25$, $\epsilon = 0.01$ for χ^2-DORO.

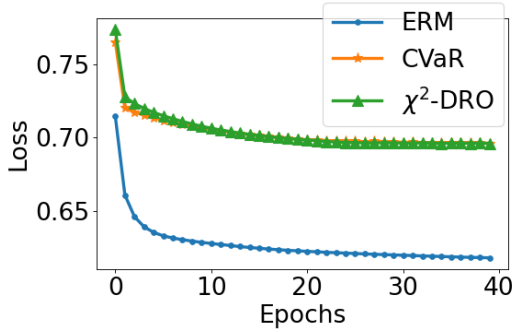
On the CivilComments-Wilds dataset, we use a pretrained BERT-base-uncased model as the classification model. For optimization, we use AdamW with learning rate 0.00001 and weight decay 0.01. The batch size is 128. The hyperparameters we used in Table 3.2 were: $\alpha = 0.1$ for CVaR; $\alpha = 0.1$, $\epsilon = 0.01$ for CVaR-DORO; $\alpha = 0.2$ for χ^2-DRO; $\alpha = 0.2$, $\epsilon = 0.01$ for χ^2-DORO.



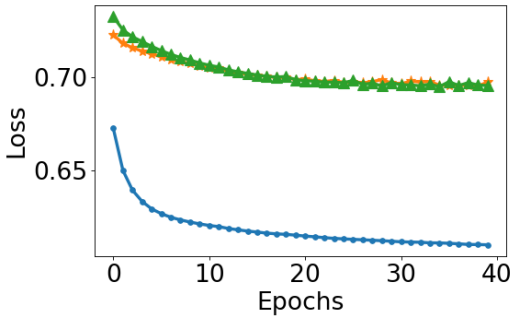
(a) Average (Original)



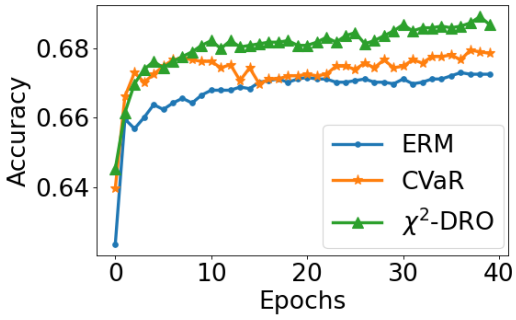
(b) Worst (Original)



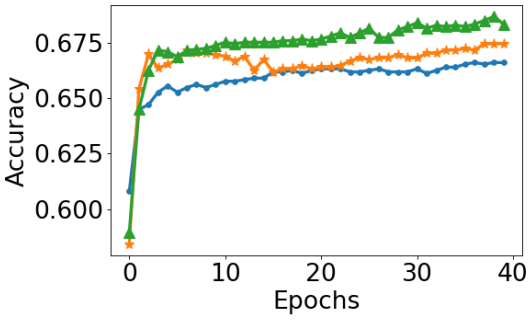
(c) Train Loss (Original)



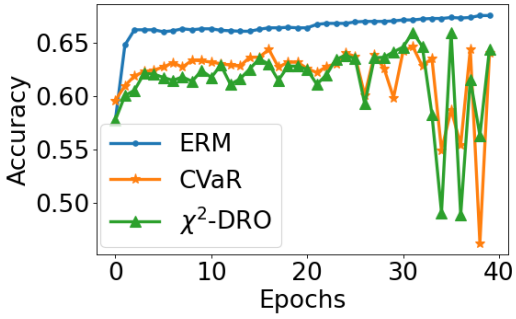
(d) Test Loss (Original)



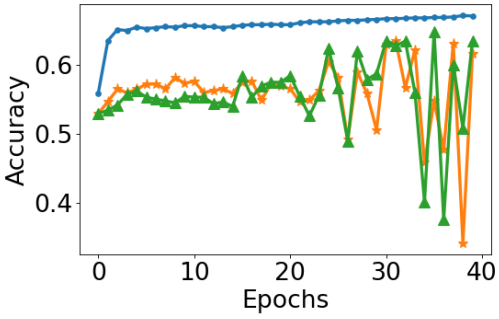
(e) Average (Outliers removed)



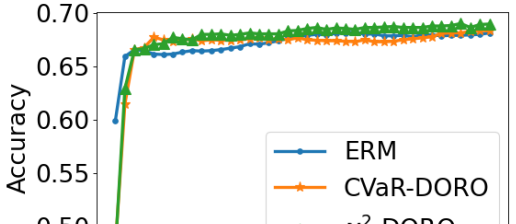
(f) Worst (Outliers removed)

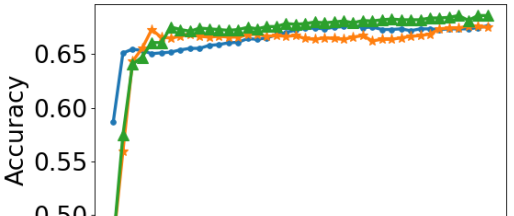


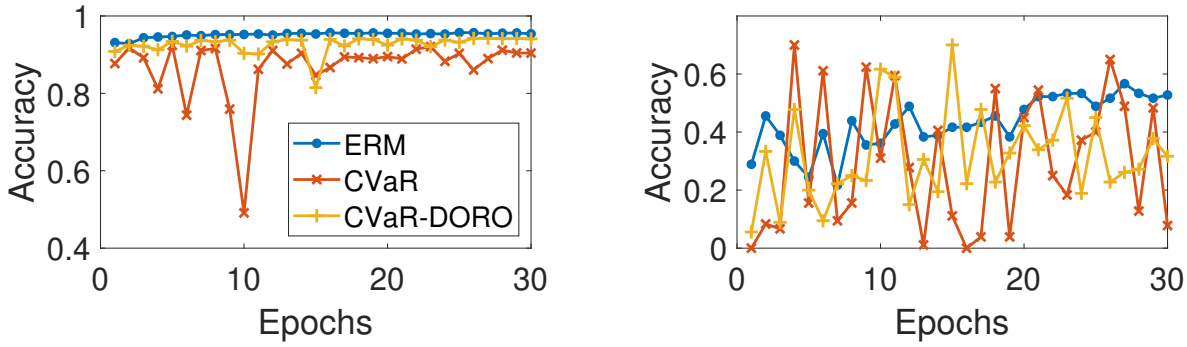
(g) Average (Labels flipped)



(h) Worst (Labels flipped)

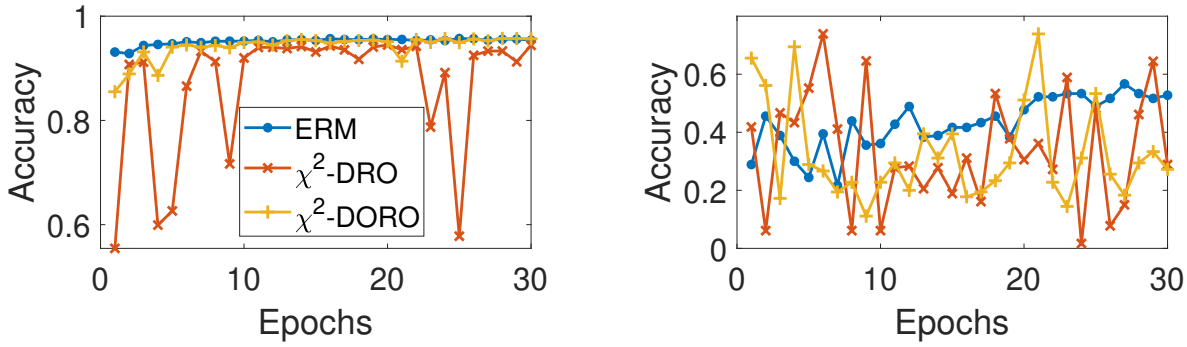






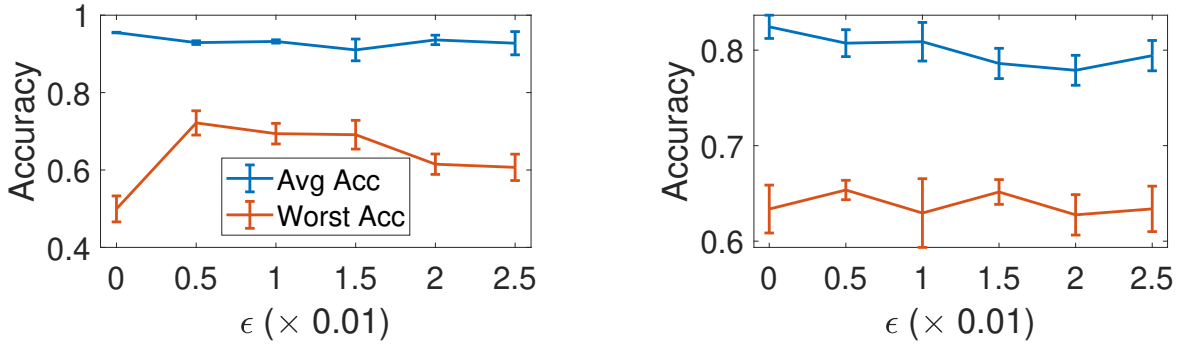
(a) Average Accuracy (b) Worst-case Accuracy

Figure 3.3: Test accuracies of CVaR and CVaR-DORO on CelebA ($\alpha = 0.1$, $\epsilon = 0.01$).



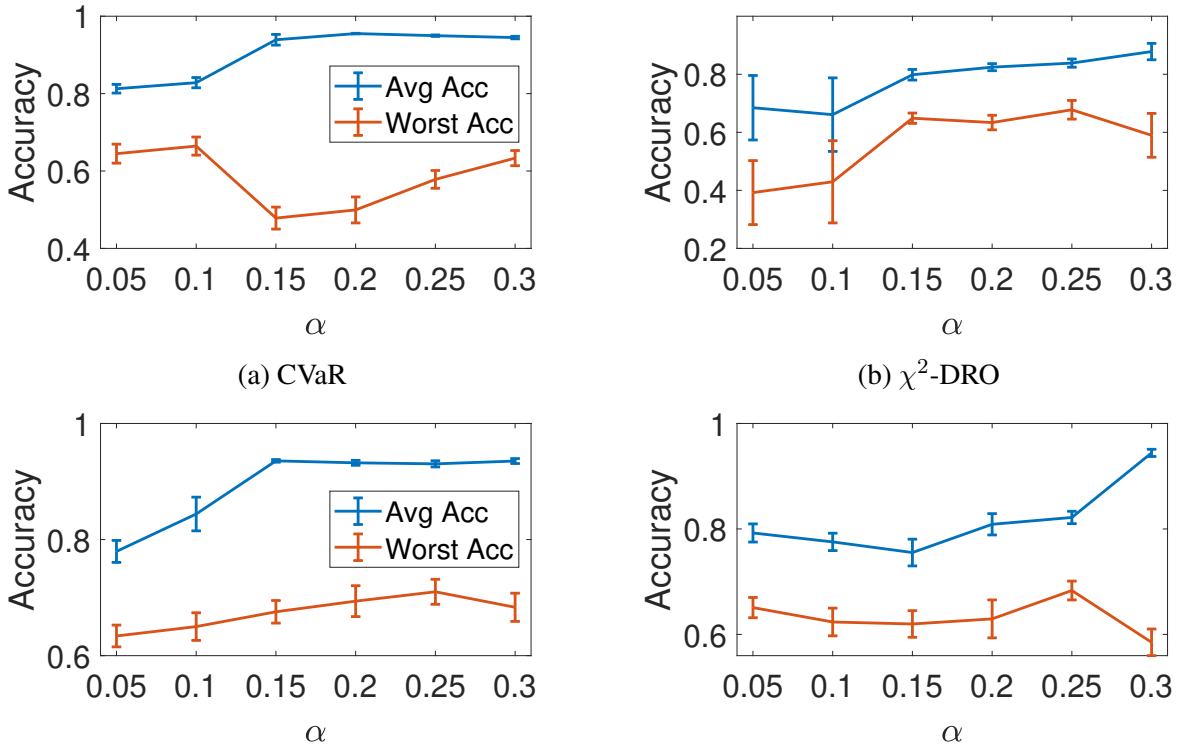
(a) Average Accuracy (b) Worst-case Accuracy

Figure 3.4: Test accuracies of χ^2-DRO and χ^2-DORO on CelebA ($\alpha = 0.3$, $\epsilon = 0.01$).



(a) CVaR-DORO (b) χ^2-DORO

Figure 3.5: Effect of ϵ on the test accuracies of CVaR/χ^2-DORO on CelebA ($\alpha = 0.2$). DORO with $\epsilon = 0$ is equivalent to DRO.



(a) CVaR

(b) χ^2-DRO

(c) CVaR-DORO

(d) χ^2-DORO

Figure 3.6: Effect of α on the test accuracies of DRO and DORO on CelebA ($\epsilon = 0.01$).

Chapter 4

High Dimensional Imbalanced Classification

4.1 Introduction

Datasets with class imbalance — that is, the number of samples of one class far exceeds the number of data of another class — are prevalent in cutting-edge data science applications [30, 60]. Take COVID-19 testing data for example: a dominant fraction of data samples often come from the negative class (i.e., non-targeted people who have not contracted the virus). The evaluation criterion in reality, however, might place equal, or even higher, emphasis on the minority class (e.g., the infected people in the COVID-19 case). The ability to generalize favorably in both majority and minority classes plays a pivotal role in critical scientific and societal issues (e.g., fairness/equity in machine learning, discovery of rare disease, transferability of knowledge to sample-starved tasks). It has been widely recognized, however, that the imbalanced availability of data can cause severe issues to modern data-limited learning algorithms including neural networks (e.g., [24, 60, 138]), particularly when reasoning about the underrepresented class.

For concreteness, let us discuss a puzzling phenomenon that arises in a classical binary classification problem. Imagine there are two classes: the majority class has n_0 samples and the minority class has n_1 samples, where $n_1 \ll n_0$. We take the generalization error to be the misclassification error when averaged over two classes (with equal weight). Prior statistical theory typically suggested a generalization bound that scales as $O(n_0^{-1/2} + n_1^{-1/2})$ as long as the sample sizes for both classes tend to infinity [27, 155]; if this were true, then one would predict that the sample size of the minority class plays a dominant role in imbalanced classification, while adding more data to either class helps improve generalization. However, an intriguing empirical phenomenon seems to contradicts this theory: in data-hungry settings, adding more data in the majority class might sometimes even *hurt* generalization [140, 156].

In this paper, we aim to take steps towards *theoretically* understanding the detrimental role of majority data for M-estimators, including the popular learning algorithms like Fisher linear discriminant analysis (LDA), logistic regression and SVM among others. We pursue a comprehensive understanding of these classifiers in the proportional regime — where the number of parameters d scales linearly with the number of linear equations n, with their ratio d/n held fixed to be a constant $\delta \in (0, \infty)$. It is demonstrated that more data from majority class can *provably* hurt the performance of these algorithms, using recent techniques from high dimensional statistics like random matrix theory or convex Gaussian minimax theorem (CGMT). Finally, we also develop effective schemes to correct the biases brought by having imbalanced classes.

4.1.1 Binary classification

Consider classifying a mixture of two d-dimensional Gaussian components, with n_0 (resp. n_1) samples for the majority (resp. minority) class. In each sample (x_i, y_i), the binary variable $y_i \in \{\pm 1\}$ denotes the class label, while $x_i = [1, \widetilde{x}_i] = [1, \widetilde{x}_{i,1}, \cdots, \widetilde{x}_{i,d}]^\top \in \mathbb{R}^{d+1}$ comprises the (augmented) feature variables, with the first coordinate encoding the intercept. The problem is this: based on n independent observations, can we hope to identify a classifier such that the classification error on a new sample x_{new} is as small as possible?

Formally, suppose we have acquired n independent observations $\{(x_i, y_i)\}_{i=1}^n$, with n_0 (resp. n_1) samples drawn from the majority class with $y_i = -1$ (resp. minority class with

$y_i = 1$). The Gaussian mixture model assumes that

$$\begin{aligned} \widetilde{X} \mid Y = +1 \sim \mathbb{P}_1 := \mathcal{N}(\mu_1, \Sigma) \\ \widetilde{X} \mid Y = -1 \sim \mathbb{P}_0 := \mathcal{N}(\mu_0, \Sigma) \end{aligned} \tag{4.1}$$

for different mean vectors $\mu_1, \mu_0 \in \mathbb{R}^d$ and shared covariance structure Σ.

Despite the imbalanced availability of training data, we would like to treat both classes equally in the generalization error. Specifically, given any classifier f that maps a feature vector $x \in \mathbb{R}^{d+1}$ to $\{+1, -1\}$, the generalization error is defined w.r.t. a *balanced* mixture of $\mathbb{P}_0$ and $\mathbb{P}_1$ as follows

$$\mathsf{Risk}(f) := \frac{1}{2}\Big\{\mathbb{E}_{x\sim\mathbb{P}_1}[\mathbb{1}\{f(x) \neq 1\}] + \mathbb{E}_{x\sim\mathbb{P}_0}[\mathbb{1}\{f(x) \neq -1\}]\Big\}. \tag{4.2}$$

In other words, $R(f)$ characterizes the out-of-same test error of classifier f at a new sample drawn from $P_{\mathsf{test}} := \frac{1}{2}\mathbb{P}_0 + \frac{1}{2}\mathbb{P}_1$. The focus is the challenging proportional growth regime

$$n_1/d = \alpha_1, \quad n_0/d = \alpha_0, \quad (n_0, n_1, d) \to \infty, \tag{4.3}$$

where both α_0 and α_1 are fixed constants.

4.1.2 Surprises in high-dimensional imbalanced classification

An illustrative example: Fisher's LDA in high dimensions. In order to determine whether an observed vector x has been drawn from distribution $\mathbb{P}_1$ or distribution $\mathbb{P}_0$, arguably the most widely used procedure is Fisher's linear discriminant analysis (LDA). In its simplest form, when the covariance matrix is known *a priori* as identity, Fisher's LDA is given by

$$\widehat{f}(x) = \mathrm{sign}\left((\widehat{\mu}_1 - \widehat{\mu}_0)^T x - \frac{1}{2}(\|\widehat{\mu}_1\|_2^2 - \|\widehat{\mu}_0\|_2^2)\right), \tag{4.4}$$

where $\widehat{\mu}_1 := \frac{1}{n_1}\sum_{i:y_i=1} x_i$ and $\widehat{\mu}_0 := \frac{1}{n_0}\sum_{i:y_i=-1} x_i$, corresponding to the sample average of each class. Most classical analyses on LDA have focused on the asymptotic regime where the number of samples n largely overwhelms the feature dimension d.In this regime, LDA is known to achieve classification error approaching Bayes risk $\Phi(-r/2)$ as n grows to infinity[1], where $r := \|\mu_1 - \mu_0\|_2$ denotes the ℓ_2-norm separation between the means. In addition, LDA is also known to be minimax-optimal among certain natural family of distributions.

However, in the regime where n and d grows proportionally to infinity, the performance of LDA is no longer characterized by the classical analyses (see [38] and a nice survey paper by [115] and references therein). As is also explained in [141, Chapter 1.2.1], assuming $n_1/d = n_0/d \to \alpha$, [38] showed that the classification error of $\widehat{f}$ converges in probability to a fixed number — in particular,

$$\mathsf{Risk}(\widehat{f}) \to \Phi\Big(-\frac{r}{2\sqrt{1 + \frac{2}{r^2\alpha}}}\Big) \neq \Phi(-\frac{r}{2}), \tag{4.5}$$

[1]Throughout, the cumulative distribution function (CDF) of the standard normal distribution is denoted as $\Phi(\cdot)$.

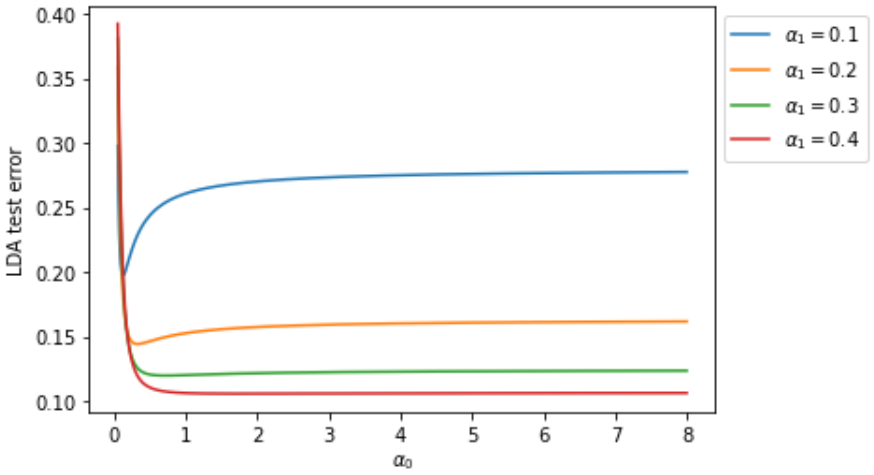


Figure 4.1: Asymptotic risk versus the sample ratio α_0 of the majority class for different choices of α_1. The ℓ_2 separation in population means is set to be $r = \|\mu_1 - \mu_0\|_2 = 6$.

which is strictly larger than the Bayes risk for non-zero α. In addition, [38] also considered an *imbalanced* variant of this result, which is of particular interest: under the imbalanced proportional regime equation 4.3 the classification risk of $\widehat{f}$ converges to:

$$\mathsf{Risk}(\widehat{f}) \to \frac{1}{2}\Phi\Big(-\frac{r^2 + \alpha_0^{-1} - \alpha_1^{-1}}{\sqrt{r^2 + \alpha_0^{-1} + \alpha_1^{-1}}}\Big) + \frac{1}{2}\Phi\Big(-\frac{r^2 - \alpha_0^{-1} + \alpha_1^{-1}}{\sqrt{r^2 + \alpha_0^{-1} + \alpha_1^{-1}}}\Big). \tag{4.6}$$

Risk (non)-monotonicity. To give the readers a sense of how $\mathsf{Risk}(\widehat{f})$ behaves, we present here an illustrative example. Fixing the value of α_1 (the proportion of samples from the minority class), let α_0 vary from 0 to ∞ and we can plot the risk curve $\mathsf{Risk}(\widehat{f})$ against different values of α_0. The asymptotics risk provided in expression equation 4.6 is shown in Figure 4.1.

When $\alpha_1 = 0.1$ or 0.2, a surprising non-monotonic risk behavior arises (shown in the blue/orange curve): as one increases the number of samples from the majority class, the test error actually *increases*. In other words,

More data from the majority class can hurt the generalization of Fisher's LDA.

It contradicts the common intuition that more data always help. This non-monotonic or U-shape behavior, however, does not always appear in this plot. In fact, it disappears after α_1, the number of samples from minority class, grows above a certain threshold; for example, the red curve — corresponding to $r = 6$ and $\alpha_1 = 0.4$ — is monotonically decreasing.

It turns out that the phenomenon aforementioned is not merely restrictive to Fisher's LDA. This paper also considers a spectrum of other standard classifiers such as support vector machine (SVM), logistic regression with or without regularization and other type of M-estimators, all of which suffer from the same sample inefficiency regarding the majority class. See also an illustrative plot in Figure 4.2.

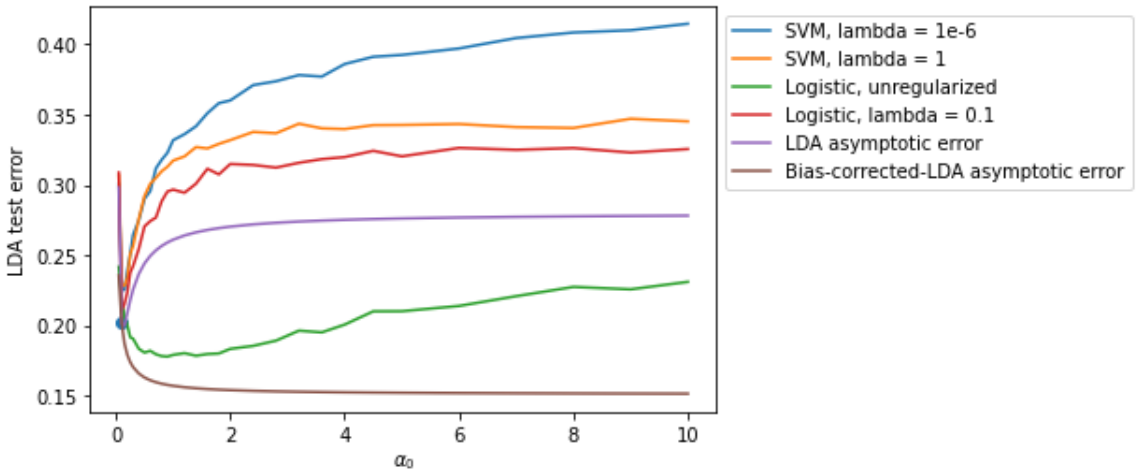


Figure 4.2: Non-monotonicity of the asymptotic risk behavior: Test errors of logistic regression and SVM are shown for $\alpha_1 = 0.1, r = 6, \Sigma = I$. We generate the data and fit the models 20 times and average over their generalization error (on a new data sample).

4.1.3 Other related works

Classifying Gaussian mixtures. Classifying mixture of Gaussians has been one of the most widely studied prototypes for binary classification due to its simplicity and applicability. While its study can be dated back to Fisher in 1930s, this stylized model has been used as a starting point for analysis of various new challenges in machine learning, such as adversarial robustness [37, 120], self-training algorithms [28, 82], non-convex optimization [11, 150], and domain adaptation [82]. Another evolving line of recent work studies this model in the high dimensional proportional regime where $d/n = \Theta(1)$, see for example [44, 49, 69, 142]. Some earlier works also considered sparse variant of LDA.

Imbalanced classification. Class-imbalanced classification has received a great amount of attention recently due to its pervasiveness in modern data applications (see surveys [30, 60]). The standard approach to correct the bias that introduced by class imbalance is by direct class re-weighting [111, 112]. It also motivates the study of designing proper loss functions to accommodate class variability, see for example [27, 45, 75]. Another line of research considers another kind of re-weighting framework by sub-sampling the major class (see [50, 121, 143] and references therein). [105] provided interesting asymptotic characterization of logistic regression for infinitely imbalanced datasets.

Exact high-dimensional asymptotics. The framework that adopted in the first part of this paper is the exact high-dimensional asymptotics of convex optimization based estimators. In order to provide sharp characterization of these estimators in the proportional regime, this framework has recently been considered in various problems in statistics and machine learning such as the analysis of LASSO estimators [29**?**], logistic regression [40, 127], double-descent phenomena [13, 97] and adversarial training [69], just to name a few. Most related to our work are the recent

work [40, 75] which study the binary and multi-class classification error respectively, without explicitly modeling the role of class imbalance. The classifiers analyzed in this paper are also closely related to the M-estimators used prominently for high dimensional regression. The efforts of characterizing the risk and distribution of robust regression estimators are initiated in [**? ? ?**] and also considered in [44, 59, 132**?**].

Fairness/Distributionally Robust Generalization The algorithmic bias of machine predictions in the face of biased data collection has been a pressing issue that raises serious concerns across various communities [66]. Our results concerning data-imbalanced classification and bias correction should be integrated broadly with the growing efforts in equitable learning and fairness. See [51, 57, 149, 159] and references therein.

4.2 Analysis of re-weighted M-estimators

4.2.1 A warm-up example: Diagnosis of LDA

Here we provide a brief analysis of why Fisher LDA is sub-optimal in the high dimensional imbalanced setting. For simiplicity, in this section we assume that $\mu_1 = \mu, \mu_0 = -\mu$ be symmetric around the origin.

In this setting, the Bayes classifier is $f_{Bayes} = \text{sign}(\mu^T x)$. Notice that the constant term is 0 in the Bayes classifier.

However, if we look at the LDA estimator closely - the constant term is $\frac{1}{2}\|\hat{\mu}_0\|_2^2 - \frac{1}{2}\|\hat{\mu}_1\|_2^2$. Using basic properties of Gaussian distribution, we can show that

$$\mathbb{E}[\|\hat{\mu}_0\|_2^2] = \|\mu\|_2^2 + \frac{d}{n_0}$$
$$\mathbb{E}[\|\hat{\mu}_1\|_2^2] = \|\mu\|_2^2 + \frac{d}{n_1}$$

In the classical regime where $n_0, n_1 >> d$, the $O(1/n)$ terms are negligible and the constant term in LDA does converge to 0 as n grows to infinity. Or alternatively, in the high dimensional *balanced* regime where $n_0 = n_1$, the $O(1/n)$ terms cancel out with each other, and the constant term is unbiased too. But when the dataset is *both* imbalanced and high-dimensional, this estimation causes trouble: the difference becomes $\frac{1}{2\alpha_0} - \frac{1}{2\alpha_1}$, i.e. the bias is non-zero!

In light of this observation, [38] proposed a simple improved version of LDA in this regime: He uses a *shifted* linear predictor, which corrects bias in the intercept term:

$$\hat{f}_{shifted}(x) = \text{sign}\left((\hat{\mu}_1 - \hat{\mu}_0)^T x - \frac{1}{2}(\|\hat{\mu}_1\|_2^2 - \|\hat{\mu}_0\|_2^2) + \frac{1}{2}(\frac{d}{n_0} - \frac{d}{n_1})\right) \tag{4.7}$$

Deev showed that the risk of this bias-corrected version of LDA converges to $\Phi\left(-\frac{r^2}{2\sqrt{r+\alpha_0^{-1}+\alpha_1^{-1}}}\right)$, which not only always outperform LDA, but also always monotone in both α_0 and α_1, as shown in the figure 4.3. An interesting interpretation of Deev's result is that the "effective" sample size seem to be twice of the harmonic mean of n_0 and n_1, i.e. $\frac{4n_0n_1}{n_0+n_1}$. Since harmonic mean is always upper bounded the arithmetic mean (with the equality when $n_0 = n_1$), this indeed suggest that imbalanced classification is always harder than balanced case.

4.2.2 Main results

Without loss of generality, we assume $\alpha_1 < \alpha_0$, i.e. class 1 is the minority class. We also shuffle the training data so that the first n_0 samples are from class 0. In other words, the samples from class 0 are $(x_1, y_1 = 0), \cdots, (x_{n_0}, y_{n_0} = 0)$, and samples from class 1 are $(x_{n_0+1}, y_{n_0+1} = 1), \cdots, (x_{n_0+n_1}, y_{n_0+n_1} = 1)$.

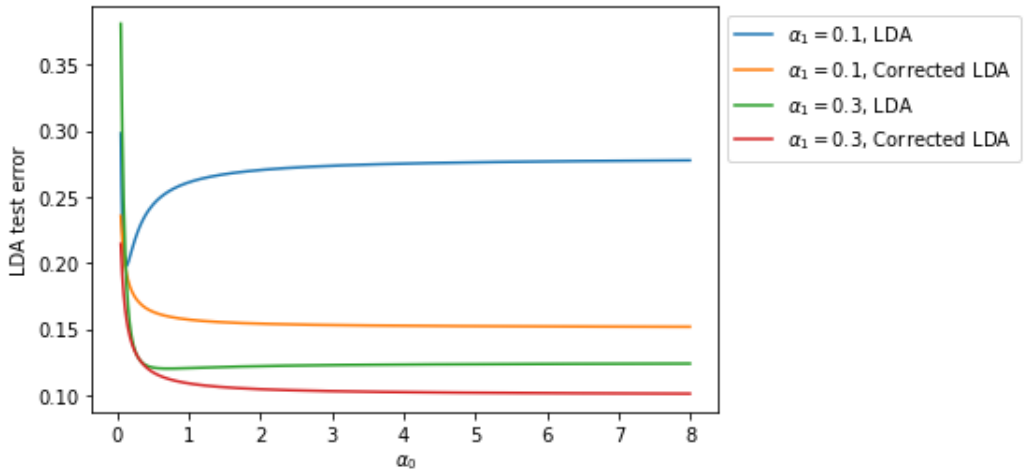


Figure 4.3: Comparing LDA and Bias-corrected LDA.

Class-weighted M-estimator: Throughout, we consider a class of classifiers that referred to as *class-weighted* M-estimators. Specifically, let $l(\cdot) : \mathbb{R} \to [0, \infty)$ be a convex classification loss function. For every linear classifier $f(x; \beta) = \text{sign}(\beta^T x)$, define its risk function as

$$\hat{R}_{\text{emp}}(\beta; l, \lambda) = \frac{1}{2n_0} \sum_{i=1}^{n_0} l(y_i \beta^T x_i) + \frac{1}{2n_1} \sum_{i=n_0+1}^{n_1} l(y_i \beta^T x_i) + \lambda \|\beta\|_2^2. \tag{4.8}$$

Here the loss from class 0 (resp. class 1) is scaled by $\frac{1}{n_0}$ (resp. $\frac{1}{n_1}$) and the corresponding M-estimator is given by

$$\hat{f}_\lambda(x) := \text{sign}(\hat{\beta}^T x) \qquad \hat{\beta} := \underset{\beta \in \mathbb{R}^{d+1}}{\text{argmin}} \ \hat{R}_{\text{emp}}(\beta; l, \lambda). \tag{4.9}$$

Our goal is to understand the performance of $\hat{f}_\lambda(x)$ in terms of population risk equation 4.2. When f is a linear classifier $f(x; \beta)$, we write $R(\beta)$ as a shorthand for $R(f(\cdot; \beta))$ when the meaning is clear from context.

Here $l(t)$ denotes any convex classification loss function. For example, $l(t) := \log(1 + e^{-t})$ corresponds to the logistic loss, whereas $l(t) := \max\{1 - t, 0\}$ is the hinge loss. For notational convenience, we adopt the conventional Moreau envelope definition as

$$e_l(x; \tau) := \min_u \{\frac{1}{2\tau}(x - u)^2 + l(u)\}. \tag{4.10}$$

In this section, we study the high dimensional asymptotic behavior of the convexified class weighted ERM estimator. The main technical tool used in the analysis is Convex Gaussian Minimax Theorem (CGMT), stated below.

We consider two settings, where the linear predictor is either homogeneous (i.e. $f(x) = \text{sign}(\beta^T x)$) or non-homogeneous (i.e. $f(x) = \text{sign}(\beta^T x + c)$). The analysis for both settings are similar and we will only provide the proof for non-homogeneous setting since it's more general.

Theorem 4.2.1 (Asymptotic Classification Error of Homogeneous Linear Classifiers)**.** *Consider the class-weighted convexified ERM estimator $\widehat{\beta}$ which minimizes the normalized logistic loss:*

$$\hat{R}_{emp}(\beta) = \frac{1}{2n_0}\sum_{i=1}^{n_0} l(y_i\beta^T x_i) + \frac{1}{2n_1}\sum_{i=n_0+1}^{n_0+n_1} l(y_i\beta^T x_i) + \lambda\|\beta\|_2^2. \tag{4.11}$$

Assuming the Gaussian Mixture model, $\frac{n_i}{d} \to \alpha_i$ for $i \in \{0,1\}$. Then, the asymptotic classification error can be characterized as:

$$R(\widehat{\beta}) \to_P \Phi\left(-\frac{r_1^*}{R^*}\|\mu\|_2\right), \tag{4.12}$$

where e_l is the Moreau envelope of $l(\cdot)$ defined in equation 4.10, and (r_1^*, R^*, θ^*) is the optimal solution of the following convex-concave scalar optimization problem:

$$\min_{r_1,R:|r_1|\le R}\max_{\theta}\frac{1}{2}\mathbb{E}_{Z\sim N(0,1)}\left[e_l\left(RZ + r_1\|\mu\|_2; \frac{\alpha_0+\alpha_1}{2\alpha_0\theta}\right) + e_l\left(RZ + r_1\|\mu\|_2; \frac{\alpha_0+\alpha_1}{2\alpha_1\theta}\right)\right] - \frac{\theta(R^2-r_1^2)}{2(\alpha_0+\alpha_1)} + \lambda R^2 \tag{4.13}$$

Theorem 4.2.2 (Asymptotic Classification Error of Non-Homogeneous Linear Classifiers)**.** *Consider the class-weighted convexified ERM estimator β which minimizes the weighted logistic loss:*

$$\hat{R}_{emp}(\beta; q_0, q_1) = \frac{q_0}{2n_0}\sum_{i=1}^{n_0} l(y_i(\beta^T x_i + c)) + \frac{q_1}{2n_1}\sum_{i=n_0+1}^{n_0+n_1} l(y_i(\beta^T x_i + c)) + \lambda\|\beta\|_2^2. \tag{4.14}$$

Assuming the Gaussian Mixture model, $\frac{n_i}{d} \to \alpha_i$ for $i \in \{0,1\}$. Then, the asymptotic classification error can be characterized as:

$$R(\beta) \to_P \frac{1}{2}\Phi\left(-\frac{r_1^*\|\mu\|_2 + c^*}{R^*}\right) + \frac{1}{2}\Phi\left(-\frac{r_1^*\|\mu\|_2 - c^*}{R^*}\right), \tag{4.15}$$

where e_l is the Moreau envelope of $l(\cdot)$ defined in equation 4.10, and (r_1^*, R^*, θ^*) is the optimal solution of the following convex-concave scalar optimization problem:

$$\min_{r_1,R,c:|r_1|\le R}\max_{\theta}$$
$$\frac{1}{2}\mathbb{E}_{Z\sim N(0,1)}\left[q_0 e_l\left(RZ + r_1\|\mu\|_2 - c; \frac{(\alpha_0+\alpha_1)q_0}{2\alpha_0\theta}\right) + q_1 e_l\left(RZ + r_1\|\mu\|_2 + c; \frac{(\alpha_0+\alpha_1)q_1}{2\alpha_1\theta}\right)\right] - \frac{\theta(R^2-r_1^2)}{2(\alpha_0+\alpha_1)} +$$

Proof techniques. This theorem is established by exploiting tools from modern high-dimensional probability, particularly the convex Gaussian minimax theorem (CGMT) [54] that has recently proven to be effective in enabling fine-grained characterization for both high-dimensional asymptotics and over-parameterized problems [6, 69, 133**? ?**]. As it turns out, this theorem allows one to pin down the asymptotic class-balanced classification error for a broad family of $\ell(\cdot)$. The theoretical prediction can be numerically computed, matching the empirical behavior of, say, the

Figure 4.4: Comparing empirical and theoretical test error of logistic regression . In this figure, we set $\lambda = 0.2$ and $\frac{n_1}{d} = \alpha_1 = 0.1$, and $d = 5000$, $T = 3$ times average of indpendent generation of dataset for the empirical curve.

LDA classifier. Let $G \in \mathbb{R}^{n\times d}, g \in \mathbb{R}^n, h \in \mathbb{R}^d$ have i.i.d. Gaussian entries, $S_w \subset \mathbb{R}^d, S_u \subset \mathbb{R}^n$ be bounded, compact, convex sets. Define two Gaussian processes:

$$X_{w,u} = u^T G w + \psi(w,u)$$
$$Y_{w,u} = \|w\|_2 g^T u + \|u\|_2 h^T w + \psi(w,u)$$

Define two optimization problems, primary optimization (PO) and auxilary optimization (AO):

$$PO(G) = \min_{w\in S_w} \max_{u\in S_u} X_{w,u} \tag{4.16}$$
$$AO(g,h) = \min_{w\in S_w} \max_{u\in S_u} Y_{w,u} \tag{4.17}$$

Assuming $\psi(\cdot,\cdot)$ is convex-concave on $S_w \times S_u$, then for any $\nu \in \mathbb{R}, t > 0$, we have:

$$\Pr[|PO(G) - \nu| > t] \leq 2\Pr[|AO(g,h) - \nu| > t] \tag{4.18}$$

where the randomness is from G, g, h. In other words,

$$\text{Concentration of } OPT(AO) \Rightarrow \text{Concentration of } OPT(PO)$$

The main benefit of CGMT is that it reduces the analysis of PO to that of AO, which is typically easier since it does not involve the Gaussian random matrix G and only depends on two Gaussian vectors g, h. Furthermore, it's often possible to simplify AO, which is a $O(n)$-dimensional problem, to an equivalent convex-concave program which involves only a small (constant) number of scalar variables, where the scalar variables also encode the quantities of our interest, e.g. the distance to the population optimal solution, the limit correlation with ground truth parameter, to name a few.

4.2.3 Bias correction for M-estimators

In Section 4.1.2, we showed that applying bias-correction on Fisher's LDA leads to a better test accuracy and monotonic risk. A natural question is whether it's possible to apply a similar bias-correction procedure to other classifiers, like logistic regression or SVMs. In this section, we answer this question in affirmative and provide two different ways of de-biasing high dimensional logistic regression in imbalanced dataset.

Figure 4.5: Comparing test error of logistic regression before and after bias-correction. In this figure, we set $\lambda = 0.2$ and $\frac{n_1}{d} = \alpha_1 = 0.1$

Method 1: De-biasing with high dimensional asymptotics

Recall that in section 4.2, we provided sharp high dimensional asymptotics for weighted logistic regression. In particular, we showed that the solution (β, c) satisfies

$$\beta^T(\mu_1 - \mu_0) \to r_1^* \|\mu_1 - \mu_0\|_2, \|\beta\|_2 \to R^*, -\frac{1}{2}\beta^T(\mu_1 + \mu_0) + c \to c^*, \tag{4.19}$$

where (r_1^*, R^*, c^*) is the solution to a convex-concave program. By inspecting the asymptotic behavior, we have the following observation:

- The correlation between β and β^* is increasing with α_0, the sample size of majority class.
- The bias of c^* is also growing with α_0, and eventually becomes the dominating factor in classification error.

Motivated by these observations, we propose the following de-biasing procedure.

First, let $c_0 = \frac{1}{2}(\beta^*)^T(\mu_1 + \mu_0)$ be the constant term of Bayes-Optimal classifier. We use the weighted logistic loss to learn a linear classifier $\beta^T x + c$, then replace c with $c_{\text{corrected}} = c - c^*$. This step makes c' asymptotically unbiased. This procedure can be viewed as a generalization to Deev's bias correction for LDA, for which he also derived the asymptotic formula of the bias term and substracted it from the estimator.

As shown in figure **??**, this correction improves the classification accuracy of logistic regression and eliminates the non-monotonicity of risk, although still being worse than the bias-corrected LDA.

Method 2: De-biasing with validation Set

While Method 1 is effective under the high-dimensional asymptotic setting we considered, it has a few crucial drawbacks. First of all, the validity of this procedure depends heavily on the Gaussian distributional assumption. Without such assumptions, it is very difficult to derive the asymptotic formula for the bias. Furthermore, even if we know the data is indeed Gaussian, this asymptotic bias varies from different Signal-to-noise ratios (i.e. the separation between the mixture components), which is unlikely to be known apriori. These drawbacks motivate us to explore alternative, practical way of de-biasing the estimator.

Another popular way of de-biasing is using a hold-out validation set, and choose the best parameter based on the accuracy therein. This method is often not statistically efficient, especially in the proportional regime $d/n = \Theta(1)$. This is because it often requires a constant fraction of data for validation, which changes the ratio of d/n and leads to an inferior statistical accuracy. However, a key observation here is that we only need to tune a *single thereshold parameter*, for which we only use a *sublinear* number of samples for validation.

More specifically, suppose we use $O(\frac{n}{\log n})$ samples from each class for validation. We first use a weighted logistic loss to learn a linear classifier $\beta^T x + c$. Then, we replace c with $c_{\text{validation}}$ which minimizes the validation error of the shifted linear classifier.

Since tuning c is equivalent to (agnostic) learning a threshold function over real line, by a VC dimension argument, we can show that:

$$R(\beta, c_{\text{validation}}) \geq \inf_c R(\beta, c) + \tilde{O}(\frac{1}{\sqrt{n}}), \tag{4.20}$$

because the generalization error scales like $\tilde{O}(\frac{1}{\sqrt{n_{\text{validation}}}})$, and $n_{\text{validation}} = O(\frac{n}{\log n}) = \tilde{O}(n)$. Note that this inequality holds without any distributional assumptions. In fact, we (should be) able to prove that this gap can be improved to $O(\frac{1}{n})$ for Gaussian mixtures.

Since all of our previous analysis only requires $n_0/d \to \alpha_0, n_1/d \to \alpha_1$ and removing a sublinear number of samples does not change the limiting behavior, the homogeneous part of the linear predictor inherits all of the theoretical properties we derived in the previous section. Since $1/\sqrt{n} \to 0$ in the asymptotic regime, we can achieve the bias-corrected asymptotic error with this validation-based approach.

4.3 Sharp non-asymptotic analysis of Deev's estimator

In this section, we provide a sharp, non-asymptotic analysis to Deev's Estimator:

$$\hat{f}_{Deev}(x) = \text{sign}\left((\hat{\mu}_1 - \hat{\mu}_0)^T x - \frac{1}{2}(\|\hat{\mu}_1\|_2^2 - \|\hat{\mu}_0\|_2^2) + \frac{1}{2}(\frac{d}{n_0} - \frac{d}{n_1})\right) \tag{4.21}$$

Theorem 4.3.1. *(Informal) Assuming $\Delta = \Omega(1), n_0 \geq n_1 = \Omega(d)$, we have:*

$$R(\hat{f}_{Deev}) = \Phi\left(-\frac{\Delta^2}{2\sqrt{\Delta^2 + \frac{d}{n_0} + \frac{d}{n_1}}}\right) + \tilde{O}\left(\frac{\sqrt{d}}{n_1}\right) \tag{4.22}$$

In contrast, as long as $\frac{d}{n_1} - \frac{d}{n_0} = \Omega(1)$, we have:

$$R(\hat{f}_{LDA}) = \Phi\left(-\frac{\Delta^2}{2\sqrt{\Delta^2 + \frac{d}{n_0} + \frac{d}{n_1}}}\right) + \tilde{\Omega}\left(\frac{d}{n_1}\right) \tag{4.23}$$

We first equivalently re-formulate Deev's Estimator in the following form:

$$\hat{f}_{Deev}(x) = \text{sign}\left((\hat{\mu}_1 - \hat{\mu}_0)^T(x - \frac{\mu_1 + \mu_0}{2}) - \frac{1}{2}(\hat{\mu}_1 - \hat{\mu}_0)^T(\hat{\mu}_1 - \mu_1 + \hat{\mu}_0 - \mu_0) + \frac{1}{2}(\frac{d}{n_0} - \frac{d}{n_1})\right) \tag{4.24}$$

Let $\mu_1 - \mu_0 = 2v$, $\hat{\mu}_1 - \mu_1 = z_1 \sim N(0, \frac{1}{n_1} I_d)$, $\hat{\mu}_0 - \mu_0 = -z_0 \sim N(0, \frac{1}{n_0} I_d)$, we can further simplify the above equation to:

$$\hat{f}_{Deev}(x) = \text{sign}\left(w^T(x - \frac{\mu_1 + \mu_0}{2}) + b)\right) \tag{4.25}$$

where:

$$w = 2v + z_1 + z_0 \tag{4.26}$$

$$b = -v^T(z_1 - z_0) + \frac{1}{2}(\|z_1\|^2 - \frac{d}{n_1}) - \frac{1}{2}(\|z_0\|^2 - \frac{d}{n_0}) \tag{4.27}$$

Recall that the test error can be written as:

$$R(f) = \frac{1}{2}\Phi\left(\frac{-w^T v - b}{\|w\|_2}\right) + \frac{1}{2}\Phi\left(\frac{-w^T v + b}{\|w\|_2}\right) \tag{4.28}$$

We have the following simple lemma:

Lemma 4.3.1.

$$\frac{1}{2}\Phi\left(-s+t\right) + \frac{1}{2}\Phi\left(-s-t\right) = \Phi(s) + \frac{1}{2\sqrt{2\pi}}e^{-\frac{s^2}{2}}st^2 + O(t^4) \tag{4.29}$$

Therefore,

$$R(f) = \frac{1}{2}\Phi\left(\frac{-w^T v}{\|w\|_2}\right) + O\left(\frac{b^2}{\|w\|_2^2}\right). \tag{4.30}$$

Next, we will analyze the two terms separately.

Lemma 4.3.2. *With probability* $1 - \frac{\delta}{2}$,

$$\frac{w^T v}{\|w\|_2} \geq ... \tag{4.31}$$

Lemma 4.3.3. *With probability* $1 - 6\delta_2 - \exp(-0.04 \cdot d)$,

$$\frac{b^2}{\|w\|_2^2} \leq 5\left(\frac{1}{n_0} + \frac{1}{n_1}\right)\log(\frac{1}{\delta_2}) + \frac{20}{\|v\|_2^2}\left(d\log(\frac{1}{\delta_2}) + \log(\frac{1}{\delta_2})^2\right)\left(\frac{1}{n_0} + \frac{1}{n_1}\right)^2 \tag{4.32}$$

Proof of Lemma 4.3.2. First, we notice that $w \sim N(2v, (\frac{1}{n_0} + \frac{1}{n_1})I_d)$ can be decomposed as

$$w = 2v + \sqrt{\frac{1}{n_0} + \frac{1}{n_1}}(z_v \cdot \frac{v}{\|v\|_2} + z_\perp), \tag{4.33}$$

where $z_v \sim N(0,1)$ and $z_\perp \sim N(0, I - \frac{1}{\|v\|_2^2}vv^T)$ are independent Gaussian variables. Note that $v^T z_\perp = 0$, hence we have

$$\|w\|_2^2 = \left(2\|v\|_2 + \sqrt{\frac{1}{n_0} + \frac{1}{n_1}}z_v\right)^2 + \left(\frac{1}{n_0} + \frac{1}{n_1}\right)\|z_\perp\|_2^2 \tag{4.34}$$

and

$$w^T v = 2\|v\|_2^2 + \sqrt{\frac{1}{n_0} + \frac{1}{n_1}}z_v\|v\|_2 \tag{4.35}$$

therefore,

$$\frac{w^T v}{\|w\|_2} = \frac{2\|v\|_2^2 + \sqrt{\frac{1}{n_0}+\frac{1}{n_1}} z_v \|v\|_2}{\left(\left(2\|v\|_2 + \sqrt{\frac{1}{n_0}+\frac{1}{n_1}} z_v\right)^2 + \left(\frac{1}{n_0}+\frac{1}{n_1}\right)\|z_\perp\|_2^2\right)^{1/2}} \tag{4.36}$$

By the Gaussian/chi-square tail bounds ([87], Lemma 1), we have with probability $1-2\delta_4$,

$$z_v \geq -\sqrt{\log(\frac{1}{\delta_4})}, \tag{4.37}$$

and

$$\|z_\perp\|_2^2 \leq (d-1) + 2\sqrt{(d-1)\log(\frac{1}{\delta_4})} + 2\log(\frac{1}{\delta_4}) \tag{4.38}$$

Let

$$T_4 = \sqrt{\left(\frac{1}{n_0}+\frac{1}{n_1}\right)\log(\frac{1}{\delta_4})} \tag{4.39}$$

$$T_5 = \left(2\sqrt{(d-1)\log(\frac{1}{\delta_4})} + 2\log(\frac{1}{\delta_4})\right)\left(\frac{1}{n_0}+\frac{1}{n_1}\right) \tag{4.40}$$

Assuming $T_4 \leq \|v\|_2$ (which is equivalent to $\delta_4 \geq \exp\left(-\frac{1}{\|v\|_2}\frac{n_0 n_1}{n_0+n_1}\right) = \exp\left(-\Theta(n_1)\right)$), we have:

$$\frac{w^T v}{\|w\|_2} \geq \|v\|_2 \frac{2\|v\|_2 - T_4}{\left((2\|v\|_2 - T_4)^2 + \frac{d-1}{n_0} + \frac{d-1}{n_1} + T_5\right)^{1/2}} \tag{4.41}$$

$$= \frac{2\|v\|_2^2}{\left(4\|v\|_2^2 + \frac{d-1}{n_0} + \frac{d-1}{n_1}\right)^{\frac{1}{2}}} - \frac{\|v\|_2\left(\left(\frac{d-1}{n_0}+\frac{d-1}{n_1}\right)T_4 + \|v\|_2 T_5\right)}{\left(4\|v\|_2^2 + \frac{d-1}{n_0} + \frac{d-1}{n_1}\right)^{\frac{3}{2}}} + O(T_4^2 + T_5^2) \tag{4.42}$$

Plugin □

Proof of Lemma 4.3.3. b consists of three terms:

$$b = \underbrace{-v^T(z_1 - z_0)}_{T_1} + \underbrace{\frac{1}{2}(\|z_1\|^2 - \frac{d}{n_1})}_{T_2} \underbrace{-\frac{1}{2}(\|z_0\|^2 - \frac{d}{n_0})}_{T_3} \tag{4.43}$$

We will bound each term separately.

Upper Bounding T_1 Notice that $z_1 - z_0 \sim N(0, (\frac{1}{n_0}+\frac{1}{n_1})I_d)$, therefore, $T_1 \sim N(0, (\frac{1}{n_0}+\frac{1}{n_1})\|v\|_2^2)$. Consequently, by the normal tail bound $Pr[N(0,1) \geq t] \leq \exp(-\frac{t^2}{2})$, we have

$$\Pr\left[|T_1| \leq \sqrt{\left(\frac{1}{n_0}+\frac{1}{n_1}\right)\log(\frac{2}{\delta_1})}\|v\|_2\right] \geq 1-\delta_1 \tag{4.44}$$

Upper Bounding T_2 and T_3 Notice that $n_1\|z_1\|^2 \sim \chi^2(d)$, by Lemma 1 of [87],

$$\Pr\left[n_1\|z_1\|_2^2 - d \geq 2\sqrt{dt} + 2t\right] \leq \exp(-t) \tag{4.45}$$

$$\Pr\left[n_1\|z_1\|_2^2 - d \leq -2\sqrt{dt}\right] \leq \exp(-t) \tag{4.46}$$

Therefore, w.p $(1-2\delta_2)$,

$$-\frac{\sqrt{d\log(\frac{1}{\delta_2})}}{n_1} \leq T_2 \leq \frac{\sqrt{d\log(\frac{1}{\delta_2})}}{n_1} + \frac{\log(\frac{1}{\delta_2})}{n_1} \tag{4.47}$$

Similarly, w.p $(1-2\delta_2)$,

$$-\frac{\sqrt{d\log(\frac{1}{\delta_2})}}{n_0} \leq T_3 \leq \frac{\sqrt{d\log(\frac{1}{\delta_2})}}{n_0} + \frac{\log(\frac{1}{\delta_2})}{n_0} \tag{4.48}$$

Therefore, w.p $(1-4\delta_2)$, we have

$$|T_2 + T_3| \leq \sqrt{d\log(\frac{1}{\delta_2})}\left(\frac{1}{n_0} + \frac{1}{n_1}\right) + \frac{\log(\frac{1}{\delta_2})}{\min(n_0, n_1)} \tag{4.49}$$

Putting things together, we have

$$|b| \leq \sqrt{\left(\frac{1}{n_0} + \frac{1}{n_1}\right)\log(\frac{2}{\delta_1})}\|v\|_2 + \left(\sqrt{d\log(\frac{1}{\delta_2})} + \log(\frac{1}{\delta_2})\right)\left(\frac{1}{n_0} + \frac{1}{n_1}\right) \tag{4.50}$$

with probability $1 - \delta_1 - 4\delta_2$. Thus, by Cauchy-Schwarz,

$$|b|^2 \leq 4\|v\|_2^2\left(\frac{1}{n_0} + \frac{1}{n_1}\right)\log(\frac{2}{\delta_1}) + 16\left(d\log(\frac{1}{\delta_2}) + \log(\frac{1}{\delta_2})^2\right)\left(\frac{1}{n_0} + \frac{1}{n_1}\right)^2 \tag{4.51}$$

Roughly speaking, this implies that $b = \tilde{O}(\frac{d}{n_*^2} + \frac{1}{n_*})$, where $n_* = \frac{n_0 n_1}{n_0 + n_1}$

Lower Bounding $\|w\|_2$ Recall that $w = 2v + z_1 + z_0 \sim N(2v, (\frac{1}{n_0} + \frac{1}{n_1})I_d)$, we have $(\frac{1}{n_0} + \frac{1}{n_1})^{-1}\|w\|_2^2$ follows non-central χ^2 distribution $\chi_d^2(\lambda)$, where $\lambda = 4\|v\|_2^2(\frac{1}{n_0} + \frac{1}{n_1})^{-1}$. By [17], we have:

$$\Pr\left[\chi_d^2(\lambda) \leq d + \lambda - 2\sqrt{(d+2\lambda)t}\right] \leq \exp(-t) \tag{4.52}$$

as a result, as long as $t \leq \frac{d+2\lambda}{25}$, we have

$$d + \lambda - 2\sqrt{(d+2\lambda)t} \geq \frac{3d+\lambda}{5}, \tag{4.53}$$

and

$$\Pr\left[(\frac{1}{n_0} + \frac{1}{n_1})^{-1}\|w\|_2^2 \leq \frac{3d+\lambda}{5}\right] \leq \exp(-t), \tag{4.54}$$

thus, we have with probability $1 - \exp(-\Omega(d))$,

$$\|w\|_2^2 \geq \frac{4}{5}\|v\|_2^2 + \frac{3}{5}(\frac{d}{n_0} + \frac{d}{n_1}) \geq \frac{4}{5}\|v\|_2^2 \tag{4.55}$$

Finally, choosing $\delta_1 = 2\delta_2$ and apply union bound, we have with probability at least $1 - 6\delta_2 - \exp(-0.04 \cdot d)$,

$$\frac{b^2}{\|w\|_2^2} \leq 5\left(\frac{1}{n_0} + \frac{1}{n_1}\right)\log(\frac{1}{\delta_2}) + \frac{20}{\|v\|_2^2}\left(d\log(\frac{1}{\delta_2}) + \log(\frac{1}{\delta_2})^2\right)\left(\frac{1}{n_0} + \frac{1}{n_1}\right)^2 \tag{4.56}$$

□

4.4 Minimax lower bounds

For any classifier $\hat{f}(x_{n+1}; x_1, \cdots, x_n)$,$n = n_0 + n_1$, the expected classification error is defined as:

$$R(\hat{f}) = \Pr[\hat{f}(x_{n+1}; x_1, \cdots, x_n) \neq y_{n+1}] \tag{4.57}$$

where the probability is taken over both training data $\{x_1, \cdots, x_n\}$ and test data (x_{n+1}, y_{n+1}) drawn from the following distribution:

$$x_1, \cdots, x_{n_0} \sim N(\mu_0, I) \tag{4.58}$$
$$x_{n_0+1}, \cdots, x_{n_0+n_1} \sim N(\mu_1, I) \tag{4.59}$$
$$y_{n+1} \sim \text{Uniform}\{+1, -1\} \tag{4.60}$$
$$x|y = 1 \sim N(\mu_1, I) \tag{4.61}$$
$$x|y = -1 \sim N(\mu_0, I) \tag{4.62}$$

We focus on the parameter space where (μ_0, μ_1) are Δ-separated:

$$\mathcal{P}_d(\Delta) = \{\mu_0, \mu_1 \in \mathbb{R}^d, \|\mu_0 - \mu_1\|_2 \geq \Delta\} \tag{4.63}$$

We will sometimes use $\mathcal{P}$ as a shorthand for $\mathcal{P}_d(\Delta)$ when the meaning is clear from context. Now, let us define the minimax classification error:

$$M(n_0, n_1, \mathcal{P}) = \inf_{\hat{f}(x_{n+1}; x_1, \cdots, x_n)} \sup_{(\mu_0, \mu_1) \in \mathcal{P}} R(\hat{f}) \tag{4.64}$$

And the quantity of interest to us is the asymptotic minimax error when $n_0/d = \alpha_0, n_1/d = \alpha_1, \Delta$ are fixed constants and $n_0, n_1, d \to \infty$. To be precise,

$$R^*_{\text{asymp}}(\Delta, \alpha_0, \alpha_1) = \limsup_{n_0/d = \alpha_0, n_1/d = \alpha_1, d \to \infty} M(n_0, n_1, \mathcal{P}_d(\Delta)) \tag{4.65}$$

4.4.1 The Bayesian connection to minimax risk

This part is analogous to Larry's Lecture Notes, chapter 36.8.

Let $Q(\mu_0, \mu_1)$ be a prior distribution over $(\mu_0, \mu_1) \in \mathcal{P}$. The Q-risk of $\hat{f}$ is defined as:

$$B_Q(\hat{f}) = \mathbb{E}_{(\mu_0,\mu_1)\sim Q} R(\hat{f}) \tag{4.66}$$

The classifier that minimizes the Q-risk is the MAP classifier:

$$f_Q(x; x_1, \cdots, x_n) = \operatorname*{argmax}_{y\in\{+1,-1\}} \Pr_Q[Y = y | X = x, x_1, \cdots, x_n], \tag{4.67}$$

for which we denote its Q-risk as $B(n_0, n_1, Q)$. Clearly, the minimax risk is always lower bounded by the Q-risk:

$$M(n_0, n_1, \mathcal{P}) \geq B(n_0, n_1, Q). \tag{4.68}$$

Utilizing this Bayesian connection, [92] provided a minimax lower bound on classification error when the classes are balanced ($n_0 = n_1$). In particular, they chose the prior distribution as uniform over a d-dimensional sphere: $\mu_1 = -\mu_0 = \mu \sim \frac{1}{2}\Delta \cdot \text{Uniform}(\mathbb{S}^{d-1})$, and used a delicate argument to show that the MAP classifier is a linear classifier. However, we weren't able to generalize their proof to imbalanced setting, mostly due to the fact that posterior distribution $p(y_{n+1}|x_{n+1}, x_1, \cdots, x_n)$ is nearly intractable without the balancedness assumption. Motivated by this, we introduce a notion of "improper prior", where the prior distribution $Q(\mu)$ may have small probability mass outside of the parameter space $\mathcal{P}$. This will enable more flexible choice of prior distributions, e.g. Gaussian priors, which in turn help us circumvent the intractable posterior issue.

Definition 4.4.1 (Improper Prior)**.** *A prior distribution $Q(\mu)$ is a δ-improper prior over $\mathcal{P}$, if*

$$\Pr_Q[\mu \in \mathcal{P}] = 1 - \delta \tag{4.69}$$

The following lemma shows that we can also link the minimax risk with the bayes risk of an improper prior:

Lemma 4.4.1. *Let $Q(\mu)$ be a δ-improper prior over $\mathcal{P}$. Then, the minimax risk over $\mathcal{P}$ is lower bounded by:*

$$M(n_0, n_1, \mathcal{P}) \geq B(n_0, n_1, Q) - \delta. \tag{4.70}$$

Proof. By definition of minimax risk, for any $\varepsilon > 0$, there exists an estimator $\hat{g}$, such that

$$\sup_{(\mu_0,\mu_1)\in\mathcal{P}} R(\hat{g}) \leq M(n_0, n_1, \mathcal{P}) + \varepsilon \tag{4.71}$$

By definition of Q-risk, we have

$$B_Q(\hat{g}) \geq B(n_0, n_1, Q) \tag{4.72}$$

The rest of proof is reminiscent of Markov's inequality. Notice that the 0-1 loss function is $R(\hat{g})$ bounded in $[0,1]$, we can upper bound $B_Q(\hat{g})$ as folows:

$$B_Q(\hat{g}) = \mathbb{E}_{(\mu_0,\mu_1)\sim Q} R(\hat{g}) \tag{4.73}$$

$$= \mathbb{E}\left[R(\hat{g})|(\mu_0,\mu_1)\in\mathcal{P}\right]\Pr_Q\left[(\mu_0,\mu_1)\in\mathcal{P}\right] + \mathbb{E}\left[R(\hat{g})|(\mu_0,\mu_1)\notin\mathcal{P}\right]\Pr_Q\left[(\mu_0,\mu_1)\notin\mathcal{P}\right] \tag{4.74}$$

$$\leq (M(n_0,n_1,\mathcal{P})+\varepsilon)(1-\delta) + 1\cdot\delta \tag{4.75}$$

$$< M(n_0,n_1,\mathcal{P}) + \varepsilon + \delta. \tag{4.76}$$

Therefore,

$$B(n_0,n_1,Q) < M(n_0,n_1,\mathcal{P}) + \varepsilon + \delta, \tag{4.77}$$

and set $\varepsilon \to 0^+$ completes the proof. □

4.4.2 Imbalanced lower bounds

In this section, we try to establish a tight lower bound to the high dimensional imbalanced classification problem. Below we state the main theorem:

Theorem 4.4.1. *Assuming $n_0 = \alpha_0 d$ and $n_1 = \alpha_1 d$, the minimax risk $M(n_0,n_1,\mathcal{P}_d(\Delta))$ is lower bounded by:*

$$M(n_0,n_1,\mathcal{P}_d(\Delta)) \geq \Phi\left(-\frac{\Delta^2}{2\sqrt{\Delta^2+\alpha_0^{-1}+\alpha_1^{-1}}}\right) - O\left(\sqrt{\frac{\log d}{d}}\right) \tag{4.78}$$

As an immediate corollary of this theorem, we can see that Deev's Bias correction achieves the information-theoretically optimal risk under the asymptotic regime $\frac{n_0}{d} \to \alpha_0, \frac{n_1}{d} \to \alpha_1; n_0, n_1, d \to \infty$. Furthermore, our lower bound is non-asymptotic, i.e. it holds for any finite (n_0, n_1, d), which generalizes the exising results even in the balanced setting.

Our proof relies on a careful analysis of Bayesian Q-risk, just like a few recent works which studied various Bayesian learning settings in high dimensional Gaussian classification [100], [134]. However, since we are interested in the minimax risk, an additional step required here is to choose an (approximately) least favorable prior over parameters (μ_0, μ_1). This step is not required in the Bayesian learning setting where the prior distribution is fixed, and it is usually a very difficult task (maybe add some reference here). Nevertheless, our analysis below indicates that it is approximately tractable when the dimensionality d is very large.

Our choice of prior distribution $Q(\mu_0,\mu_1)$ is a generalization of those appeared in recent works [100] and [134]. In [100], the prior distribution is chosen as symmetric Gaussian Prior $\mu_1 \sim N(0, \frac{\Delta^2}{4d}I), \mu_0 = -\mu_1$. In [134], the prior is chosen as independent Gaussian Prior $\mu_1, \mu_0 \sim N(0, \frac{\Delta^2}{2d}I)$. While neither of these priors leads to a tight minimax lower bound, a key observation is that both of these priors can be written in the following form:

$$\begin{pmatrix}\mu_0\\ \mu_1\end{pmatrix} \sim N\left(0, \begin{bmatrix}(R^2+r^2)I_d & (R^2-r^2)I_d\\ (R^2-r^2)I_d & (R^2+r^2)I_d\end{bmatrix}\right), \tag{4.79}$$

Another way of interepreting equation 4.79 is as follows: let u, v be independent random variables each with $N(0, I_d)$ distribution, then equation 4.79 is equivalent to:

$$\mu_0 = Ru - rv \tag{4.80}$$
$$\mu_1 = Ru + rv. \tag{4.81}$$

Indeed, the prior considered in [100] corresponds to $R = 0, r = \frac{\Delta}{2\sqrt{d}}$, and the one in [134] corresponds to $R = r = \frac{\Delta}{2\sqrt{d}}$. As we will see in the analysis, our choice of R is much larger than both of them: we require $R = \Omega(\sqrt{d})$ for our desired lower bound. From a high level, this choice of prior is inspired by a classical result in determining the exact minimax risk of Normal mean estimation, where the prior distribution was chosen as $N(0, \gamma^2 I_d)$ and let the variance $\gamma^2 \to \infty$ (See e.g. Example 36.67 of Larry's Lecture Notes). In our case, the difference of means $\mu_1 - \mu_0 \sim N(0, 4r^2 I_d)$, which implies $\|\mu_1 - \mu_d\|_2 \approx 2r\sqrt{d} \approx \Delta$, while $\mu_1 + \mu_0 \sim N(0, 4R^2 I_d)$, which has a very large variance similar to the text-book proof for normal mean estimation.

The main technical lemma is a careful non-asymptotic analysis of the Bayes risk under the prior distribution mentioned above, stated below:

Lemma 4.4.2. *Under the prior distribution Q defined in equation 4.79, with the choice of parameters $R = \Omega(\sqrt{d})$ and $r = \frac{\tilde{\Delta}}{2\sqrt{d}}$, the Q-risk is lower bounded by:*

$$B(n_0, n_1, Q) \geq \Phi\left(-\frac{\tilde{\Delta}^2}{2\sqrt{\tilde{\Delta}^2 + \alpha_0^{-1} + \alpha_1^{-1}}}\right) - O\left(\sqrt{\frac{\log d}{d}}\right) \tag{4.82}$$

Theorem 4.4.1 is a consequence of Lemma 4.4.1 and Lemma 4.4.2.

Proof. We choose $\tilde{\Delta} = \Delta(1 + C\sqrt{\frac{\log d}{d}})$. By our construction, $\frac{1}{2r}(\mu_1 - \mu_0) \sim N(0, I_d)$. Therefore, $\frac{1}{4r^2}\|\mu_1 - \mu_0\|_2^2 \sim \chi^2(d)$. Using the tail bound for chi-square variables [87], we have

$$\Pr\left[\frac{1}{4r^2}\|\mu_1 - \mu_0\|_2^2 \leq d - C\sqrt{d \log d}\right] \leq \frac{1}{\sqrt{d}} \tag{4.83}$$

In other words, with probability $1 - d^{-1/2}$, we have

$$\|\mu_1 - \mu_0\|_2^2 \geq 4r^2\left(d - C\sqrt{d \log d}\right) = \tilde{\Delta}^2\left(1 - C\sqrt{\frac{\log d}{d}}\right) \geq \Delta^2 \tag{4.84}$$

Thus, Q is a δ-improper prior with parameter $\delta \leq d^{-1/2}$. Therefore, by Lemma 4.4.1,

$$M(n_0, n_1, \mathcal{P}_d(\Delta)) \geq B(n_0, n_1, Q) - \delta \tag{4.85}$$
$$\geq \Phi\left(-\frac{\tilde{\Delta}^2}{2\sqrt{\tilde{\Delta}^2 + \alpha_0^{-1} + \alpha_1^{-1}}}\right) - O\left(\sqrt{\frac{\log d}{d}}\right) \tag{4.86}$$

Finally, it remains to show that

$$\Phi\left(-\frac{\tilde{\Delta}^2}{2\sqrt{\tilde{\Delta}^2+\alpha_0^{-1}+\alpha_1^{-1}}}\right) \geq \Phi\left(-\frac{\Delta^2}{2\sqrt{\Delta^2+\alpha_0^{-1}+\alpha_1^{-1}}}\right) - O\left(\sqrt{\frac{\log d}{d}}\right) \tag{4.87}$$

This can be proven via a simple Lipschitzness argument. Notice that $\Phi(\cdot)$ is $\frac{1}{\sqrt{2\pi}}$-Lipschitz, and by Lemma 4.4.3, $h(t) = -\frac{t^2}{2\sqrt{t^2+c}}$ is $\sqrt{\frac{32}{27}}$-Lipschitz for any $c > 0$. Therefore, $\Phi(h(t))$ is $\frac{1}{\sqrt{2\pi}} \cdot \sqrt{\frac{32}{27}} = \frac{4}{3\sqrt{3\pi}}$-Lipschitz, and we complete the proof by using the fact that $|\tilde{\Delta} - \Delta| = O\left(\sqrt{\frac{\log d}{d}}\right)$. □

The remaining work is to prove the main technical lemma 4.4.2. Before proceeding, it'll be useful to establish a closed form of posterior probability $p(y_{n+1}|x_{n+1}, x_1^n, y_1^n)$, as summarized in the following claim (whose proof is included in the appendix):

Claim 4.4.2. *(Posterior probability)*

$$\Pr[Y_{new} = y_{n+1}|X_{new} = x_{n+1}; \{(x_i, y_i)\}_{i=1}^n] \propto \exp\left(\frac{T}{2Z} - \frac{1}{2}d\log Z - \frac{1}{2}\sum_{i=1}^{n+1}\|x_i\|^2\right) \tag{4.88}$$

where

$$Z = 4R^2r^2|C_0||C_1| + (R^2+r^2)(|C_0|+|C_1|) + 1 \tag{4.89}$$

$$T = (4R^2r^2|C_1| + R^2 + r^2)\|\sum_{i\in C_0} x_i\|_2^2 + (4R^2r^2|C_0| + R^2 + r^2)\|\sum_{i\in C_1} x_i\|_2^2 + 2(R^2 - r^2)(\sum_{i\in C_0} x_i)^T(\sum_{i\in C_1} x_i) \tag{4.90}$$

and

$$C_0 = \{1, \cdots, n_0, n+1\}, C_1 = \{n_0+1, \cdots, n\} \text{ if } y_{new} = 0 \tag{4.91}$$

$$C_0 = \{1, \cdots, n_0\}, C_1 = \{n_0+1, \cdots, n, n+1\} \text{ if } y_{new} = 1 \tag{4.92}$$

Equipped with Claim 4.4.2, we are now ready to prove Lemma 4.4.2.

Proof of Lemma 4.4.2. To avoid notational clutter, we will use Δ instead of $\tilde{\Delta}$ in the proof below. By definition of MAP classifier,

$$f_Q(x; x_1, \cdots, x_n) = \operatorname*{argmax}_{y\in\{+1,-1\}} \Pr[Y_{new} = y_{n+1}|X_{new} = x_{n+1}; \{(x_i, y_i)\}_{i=1}^n] \tag{4.93}$$

Denote

$$l(y) = Pr[Y_{new} = y|X_{new} = x_{n+1}; \{(x_i, y_i)\}_{i=1}^n] \tag{4.94}$$

The expected classification error of the MAP classifier can be formulated as:

$$B_Q(f_Q) = \Pr[f_Q(x_{n+1}) \neq y_{n+1}] \tag{4.95}$$

$$= \frac{1}{2}\Pr[l(1) < l(0)|y_{n+1} = 1] + \frac{1}{2}\Pr[l(0) < l(1)|y_{n+1} = 0] \tag{4.96}$$

where the probability is taken over the prior distribution $(\mu_0, \mu_1) \sim Q$, as well as $\{(x_i, y_i)\}_{i\in[n+1]}$ generated according to equation 4.58. By symmetry, we will focus our attention on the first term, i.e. the probability of $l(1) < l(0)$ when $x_{n+1} \sim N(\mu_1, I_d)$.

Next, we will compare $l(1)$ and $l(0)$. Define S_0, S_1 as the summation of all training data from two classes:

$$S_0 = \sum_{i=1}^{n_0} x_i \tag{4.97}$$

$$S_1 = \sum_{i=n_0+1}^{n} x_i \tag{4.98}$$

Notice that in Claim 4.4.2, when $y_{n+1} = 1$, we have $|C_0| = n_0, |C_1| = n_1+1$, while when $y_{n+1} = 0$, instead we have $|C_0| = n_0 + 1, |C_1| = n_1$. For notational simplicity, denote

$$Z_0 := 4R^2r^2(n_0 + 1)n_1 + (R^2 + r^2)(n_0 + n_1 + 1) + 1 \tag{4.99}$$

$$Z_1 := 4R^2r^2(n_1 + 1)n_0 + (R^2 + r^2)(n_0 + n_1 + 1) + 1 \tag{4.100}$$

$$T_0 := (4R^2r^2n_1 + R^2 + r^2)\|S_0 + x_{n+1}\|_2^2 + (4R^2r^2(n_0 + 1) + R^2 + r^2)\|S_1\|_2^2 + 2(R^2 - r^2)(S_0 + x_{n+1})^T S_1 \tag{4.101}$$

$$T_1 := (4R^2r^2(n_1 + 1) + R^2 + r^2)\|S_0\|_2^2 + (4R^2r^2n_0 + R^2 + r^2)\|S_1 + x_{n+1}\|_2^2 + 2(R^2 - r^2)S_0^T(S_1 + x_{n+1}) \tag{4.102}$$

Then,

$$\log\left(\frac{l(1)}{l(0)}\right) = \frac{T_1}{2Z_1} - \frac{T_0}{2Z_0} + \frac{1}{2}d\log\left(\frac{Z_0}{Z_1}\right) \tag{4.103}$$

$$= \frac{T_1Z_0 - T_0Z_1}{2Z_0Z_1} + \frac{1}{2}d\log\left(\frac{Z_0}{Z_1}\right) \tag{4.104}$$

In order to analyze equation equation 4.104, we will take a closer look at each term appeared in equation 4.104, namely, $T_1Z_0 - T_0Z_1$, $2Z_0Z_1$, and $\frac{1}{2}d\log\left(\frac{Z_0}{Z_1}\right)$.

It is easy to see that when $r = \frac{\Delta}{2\sqrt{d}}$, $n_0 = \alpha_0 d$ and $n_1 = \alpha_1 d$, we have

$$Z_0 = R^2d\left(\Delta^2\alpha_0\alpha_1 + \alpha_0 + \alpha_1\right)\left(1 + O(R^{-1} + d^{-1})\right) \tag{4.105}$$

$$Z_1 = R^2d\left(\Delta^2\alpha_0\alpha_1 + \alpha_0 + \alpha_1\right)\left(1 + O(R^{-1} + d^{-1})\right) \tag{4.106}$$

and

$$\frac{Z_0}{Z_1} = 1 + d^{-1} \cdot \frac{\Delta^2(\alpha_1 - \alpha_0)}{\Delta^2\alpha_0\alpha_1 + \alpha_0 + \alpha_1} + O(d^{-2}) \tag{4.107}$$

Therefore,

$$\frac{1}{2}d\log\left(\frac{Z_0}{Z_1}\right) = \frac{\Delta^2(\alpha_1 - \alpha_0)}{2\left(\Delta^2\alpha_0\alpha_1 + \alpha_0 + \alpha_1\right)} + O(d^{-1}) \tag{4.108}$$

and

$$2Z_0Z_1 = 2R^4\left(\Delta^2\alpha_0\alpha_1 + \alpha_0 + \alpha_1\right)^2\left(1 + O(R^{-1} + d^{-1})\right) \tag{4.109}$$

The remaining work, also the most challenging part, is to analyze the term $T_1Z_0 - T_0Z_1$.

Since $S_0 = \sum_{i=1}^{n_0} x_i$ where $x_i \sim_{i.i.d.} N(\mu_0, I_d)$, we have the decomposition

$$S_0 = n_0\mu_0 + \sqrt{n_0}w_0 = \alpha_0 d(Ru - rv) + \sqrt{\alpha_0 d}w_0, w_0 \sim N(0, I_d). \tag{4.110}$$

Similarly,

$$S_1 = n_1\mu_1 + \sqrt{n_1}w_1 = \alpha_1 d(Ru + rv) + \sqrt{\alpha_1 d}w_1, w_1 \sim N(0, I_d). \tag{4.111}$$

We also denote that

$$x_{n+1} = \mu_1 + w_{n+1} = (Ru + rv) + w_{n+1}, w_{n+1} \sim N(0, I_d). \tag{4.112}$$

Putting equation 4.110, equation 4.111, equation 4.112 and $r = \frac{\Delta}{2\sqrt{d}}$ into $T_1Z_0 - T_0Z_1$ and with the help of symbolic computation, we have the following decomposition:

$$T_1Z_0 - T_0Z_1 = \sum_{i=0}^{4}\sum_{j=i-4}^{i-1} C(i,j)R^i d^{j/2} \tag{4.113}$$

Here, $C(i,j)$, are bi-linear forms of u, v, w_{n+1}, w_0, w_1, whose coefficients only depend on $\alpha_0, \alpha_1, \Delta$ but independent with d and R.

Recall that u, v, w_{n+1}, w_0, w_1 are all d-dimensional normal variables $\sim N(0, I_d)$. For the convenience of our analysis, we denote the following event as e_{good}:

The event e_{good}: For all $(\gamma, \nu) \in \{u, v, w_{n+1}, w_0, w_1\} \times \{u, v, w_{n+1}, w_0, w_1\}$:

$$\|\gamma\|_2^2 = d + O(\sqrt{d\log d}) \tag{4.114}$$

$$\gamma^T\nu = O(\sqrt{d\log d}) \tag{4.115}$$

Since $\|\gamma\|_2^2 \sim \chi^2(d)$, $\gamma^T\nu = \frac{1}{2}\left(\frac{1}{2}\|\gamma+\nu\|_2^2 - \frac{1}{2}\|\gamma-\nu\|_2^2\right)$ and $\|\gamma+\nu\|_2^2, \|\gamma-\nu\|_2^2$ are also χ^2 distributed, by standard concentration inequalities for chi-squared variables and union bounds over all (γ, ν) pairs, we have e_{good} happens with probability at least $1 - d^{-2021}$.

Hence, condition on the event e_{good}, all of quadratic forms $C(i,j)$ are bounded by $O(d)$. Since when $i \le 3$, we have $j \le 2$ and $C(i,j)R^i d^{j/2}$ at most $O(R^3d^2)$. Using the same argument, we can see that $C(i,j)R^i d^{j/2}$ is at most $O(R^4 d^{\frac{3}{2}})$ when $i = 4$, $j = 0, 1$. As we will show below, these terms are all negligible and we can focus on the $C(4,3)$ and $C(4,2)$ terms.

The precise expressions for $C(4,3), C(4,2)$ are as follows:

$$C(4,3) = 2\Delta^2 w_{n+1}^T\left(\Delta^2\alpha_0\alpha_1 + \alpha_0 + \alpha_1\right)\left(\Delta\alpha_0\alpha_1 v - \sqrt{\alpha_0}\alpha_1 w_0 + \alpha_0\sqrt{\alpha_1}w_1\right), \tag{4.116}$$

and

$$C(4,2) = \Delta^2\Bigg(\Delta^2\alpha_0^2\alpha_1\left(\Delta^2\alpha_1+2\right)\|v\|_2^2$$
$$-\alpha_0\alpha_1\left(\Delta^2\alpha_0+2\right)\|w_1\|_2^2+\left(\Delta^2\alpha_0^2\alpha_1-\Delta^2\alpha_0\alpha_1^2+\alpha_0^2-\alpha_1^2\right)\|w_{n+1}\|_2^2+\alpha_0\alpha_1\left(\Delta^2\alpha_1+2\right)\|w_0\|_2^2$$
$$-2\Delta v^T\left(\alpha_0^{\frac{3}{2}}\alpha_1\left(\Delta^2\alpha_1+2\right)w_0-\alpha_0\sqrt{\alpha_1}\left(\alpha_0-\alpha_1\right)w_1\right)-2\sqrt{\alpha_0}\sqrt{\alpha_1}\left(\alpha_0-\alpha_1\right)w_0^Tw_1\Bigg). \tag{4.117}$$

Condition on the event e_{good}, we know that $v^Tw_0, v^Tw_1, w_0^Tw_1 = O(\sqrt{d\log d})$, and $\|v\|_2, \|w_1\|_2^2, \|w_0\|_2^2, \|w_{n+1}\|_2^2 = d + O(\sqrt{d\log d})$. Therefore,

$$\begin{aligned}
C(4,2) =& d\Delta^2\left(\Delta^2\alpha_0^2\alpha_1\left(\Delta^2\alpha_1+2\right)-\alpha_0\alpha_1\left(\Delta^2\alpha_0+2\right)+\left(\Delta^2\alpha_0^2\alpha_1-\Delta^2\alpha_0\alpha_1^2+\alpha_0^2-\alpha_1^2\right)+\alpha_0\alpha_1\left(\Delta^2\alpha_1+2\right)\right. \\
&+O(\sqrt{d\log d}) && (4.118)\\
=&\Delta^2\left(\Delta^4\alpha_0^2\alpha_1^2+2\Delta^2\alpha_0^2\alpha_1+\alpha_0^2-\alpha_1^2\right)d+O(\sqrt{d\log d}) && (4.119)\\
=&\Delta^2\left(\Delta^2\alpha_0\alpha_1+\alpha_0-\alpha_1\right)\left(\Delta^2\alpha_0\alpha_1+\alpha_0+\alpha_1\right)d+O(\sqrt{d\log d}) && (4.120)
\end{aligned}$$

Regarding $C(4,3)$, notice that $\tilde{w} := \left(\Delta^2\alpha_0\alpha_1 d+\alpha_0+\alpha_1\right)^{-1/2}\left(\Delta\alpha_0\alpha_1 v-\sqrt{\alpha_0}\alpha_1 w_0+\alpha_0\sqrt{\alpha_1}w_1\right) \sim N(0,I_d)$. By central limit theorem, $z_d := \frac{1}{\sqrt{d}}w_{n+1}^T\tilde{w} \to_P N(0,1)$ when $d\to\infty$. Futhermore, by Berry-Esseen theorem, there exists an universal constant C such that

$$\left|\Pr\{z_d \geq t\} - \Phi(t)\right| \leq \frac{C}{\sqrt{d}}, \tag{4.121}$$

and $C(4,3)$ can be equivalently re-formulated as:

$$\begin{aligned}
C(4,3) &= 2\Delta^2\left(\Delta^2\alpha_0\alpha_1+\alpha_0+\alpha_1\right)w_{n+1}^T\left(\Delta\alpha_0\alpha_1 v-\sqrt{\alpha_0}\alpha_1 w_0+\alpha_0\sqrt{\alpha_1}w_1\right) && (4.122)\\
&= 2\sqrt{d\alpha_0\alpha_1}\Delta^2\left(\Delta^2\alpha_0\alpha_1+\alpha_0+\alpha_1\right)^{\frac{3}{2}}z_d && (4.123)
\end{aligned}$$

Substituting equation 4.118 and equation 4.122 into equation 4.113, we get

$$\begin{aligned}
&T_1Z_0 - T_0Z_1 && (4.124)\\
=&R^4d^2\Delta^2\left(2\sqrt{\alpha_0\alpha_1}\left(\Delta^2\alpha_0\alpha_1+\alpha_0+\alpha_1\right)^{\frac{3}{2}}z_d+\left(\Delta^2\alpha_0\alpha_1+\alpha_0-\alpha_1\right)\left(\Delta^2\alpha_0\alpha_1+\alpha_0+\alpha_1\right)\right) && (4.125)\\
&+O(R^4d^{\frac{3}{2}}+R^3d^2) && (4.126)
\end{aligned}$$

Substituting equation 4.109, equation 4.108 and equation 4.124 into equation 4.104, we have

$$\log\left(\frac{l(1)}{l(0)}\right) = \frac{T_1 Z_0 - T_0 Z_1}{2 Z_0 Z_1} + \frac{1}{2} d \log\left(\frac{Z_0}{Z_1}\right) \tag{4.127}$$

$$= \frac{\Delta^2}{2\left(\Delta^2 \alpha_0 \alpha_1 + \alpha_0 + \alpha_1\right)} \left(2\sqrt{\alpha_0 \alpha_1}\left(\Delta^2 \alpha_0 \alpha_1 + \alpha_0 + \alpha_1\right)^{\frac{1}{2}} z_d + \left(\Delta^2 \alpha_0 \alpha_1 + \alpha_0 - \alpha_1\right)\right) \tag{4.128}$$

$$+ O\left(\sqrt{\frac{\log d}{d}}\right) + \frac{\Delta^2(\alpha_1 - \alpha_0)}{2\left(\Delta^2 \alpha_0 \alpha_1 + \alpha_0 + \alpha_1\right)} + O(d^{-1}) \tag{4.129}$$

$$= \frac{\Delta^2}{2\left(\Delta^2 \alpha_0 \alpha_1 + \alpha_0 + \alpha_1\right)} \left(2\sqrt{\alpha_0 \alpha_1}\left(\Delta^2 \alpha_0 \alpha_1 + \alpha_0 + \alpha_1\right)^{\frac{1}{2}} z_d + \Delta^2 \alpha_0 \alpha_1 + O\left(\sqrt{\frac{\log d}{d}}\right)\right) \tag{4.130}$$

$$= \frac{\Delta^2 \sqrt{\alpha_0 \alpha_1}}{\left(\Delta^2 \alpha_0 \alpha_1 + \alpha_0 + \alpha_1\right)^{1/2}} \left(z_d + \frac{\Delta^2}{2\left(\Delta^2 + \alpha_0^{-1} + \alpha_1^{-1}\right)^{\frac{1}{2}}} + O\left(\sqrt{\frac{\log d}{d}}\right)\right) \tag{4.131}$$

Hence, condition on the event e_{good}, $l(1) < l(0)$ happens if and only if

$$z_d \leq -\frac{\Delta^2}{2\left(\Delta^2 + \alpha_0^{-1} + \alpha_1^{-1}\right)^{\frac{1}{2}}} - O\left(\sqrt{\frac{\log d}{d}}\right) \tag{4.132}$$

By Berry-Esseen and union bound, the probability is lower bounded by:

$$\Pr[l(1) < l(0) | y_{n+1} = 1] \geq \Phi\left(-\frac{\Delta^2}{2\left(\Delta^2 + \alpha_0^{-1} + \alpha_1^{-1}\right)^{\frac{1}{2}}} - O\left(\sqrt{\frac{\log d}{d}}\right)\right) - O\left(\frac{1}{\sqrt{d}}\right) - \Pr(\mathrm{e}_{\text{good}}) \tag{4.133}$$

$$\geq \Phi\left(-\frac{\Delta^2}{2\left(\Delta^2 + \alpha_0^{-1} + \alpha_1^{-1}\right)^{\frac{1}{2}}}\right) - O\left(\sqrt{\frac{\log d}{d}}\right), \tag{4.134}$$

where the last step is due to the $\frac{1}{\sqrt{2\pi}}$-Lipschitzness of $\Phi(t)$.

By symmetricity, when $y_{n+1} = 0$, we also have:

$$\Pr[l(0) < l(1) | y_{n+1} = 0] \geq \Phi\left(-\frac{\Delta^2}{2\left(\Delta^2 + \alpha_0^{-1} + \alpha_1^{-1}\right)^{\frac{1}{2}}}\right) - O\left(\sqrt{\frac{\log d}{d}}\right). \tag{4.135}$$

Therefore we have completed the proof.

□

Proof of Claim 4.4.2

Proof of Claim 4.4.2: By law of total probability and Bayes's theorem,

$$\Pr[Y_{new} = y_{n+1}|X_{new} = x_{n+1}; \{(x_i, y_i)\}_{i=1}^n] \tag{4.136}$$
$$\propto p[Y_{new} = y_{n+1}, X_{new} = x_{n+1}; \{(x_i, y_i)\}_{i=1}^n] \tag{4.137}$$
$$= \mathbb{E}_{(\mu_0,\mu_1)\sim Q} p(x_1^{n+1}|\mu_0, \mu_1, y_1^{n+1}) p(y_1^{n+1}|\mu_0, \mu_1) \tag{4.138}$$
$$\propto \mathbb{E}_{(\mu_0,\mu_1)\sim Q} \exp(-\frac{1}{2}\sum_{i\in C_0}\|x_i - \mu_0\|_2^2 - \frac{1}{2}\sum_{i\in C_1}\|x_i - \mu_1\|_2^2) \tag{4.139}$$
$$= \int \exp(-\frac{1}{2}\sum_{i\in C_0}\|x_i - \mu_0\|_2^2 - \frac{1}{2}\sum_{i\in C_1}\|x_i - \mu_1\|_2^2) dQ(\mu_0, \mu_1) \tag{4.140}$$

$$\int \exp(-\frac{1}{2}\sum_{i\in C_0}\|x_i - \mu_0\|_2^2 - \frac{1}{2}\sum_{i\in C_1}\|x_i - \mu_1\|_2^2) dQ(\mu_0, \mu_1) \tag{4.141}$$
$$= \exp(-\frac{1}{2}\sum_{i\in[n+1]}\|x_i\|_2^2) \int \exp(\mu_0^T \sum_{i\in C_0} x_i + \mu_1^T \sum_{i\in C_1} x_i - \frac{|C_0|}{2}\|\mu_0\|_2^2 - \frac{|C_1|}{2}\|\mu_1\|_2^2) dQ(\mu_0, \mu_1) \tag{4.142}$$

Substitute $\mu_0 = Ru - rv$ and $\mu_1 = Ru + rv$, we have

$$\int \exp(\mu_0^T \sum_{i\in C_0} x_i + \mu_1^T \sum_{i\in C_1} x_i - \frac{|C_0|}{2}\|\mu_0\|_2^2 - \frac{|C_1|}{2}\|\mu_1\|_2^2) dQ(\mu_0, \mu_1) \tag{4.143}$$
$$\propto \int \exp\left((Ru - rv)^T \sum_{i\in C_0} x_i + (Ru + rv)^T \sum_{i\in C_1} x_i - \frac{|C_0|}{2}\|Ru - rv\|_2^2 - \frac{|C_1|}{2}\|Ru + rv\|_2^2 - \frac{1}{2}\|u\|_2^2 - \frac{1}{2}\|v\|_2^2\right) d$$
(4.144)
$$= \int \exp\left(R(\sum_{i\in C_0} x_i + \sum_{i\in C_1} x_i)^T u + r(\sum_{i\in C_1} x_i - \sum_{i\in C_0} x_i)^T v\right) \cdot \tag{4.145}$$
$$\exp\left(-\frac{R^2(|C_0|+|C_1|)+1}{2}\|u\|_2^2 - Rr(|C_1|-|C_0|))u^T v - \frac{r^2(|C_0|+|C_1|)+1}{2}\|v\|_2^2\right) dudv \tag{4.146}$$
$$:= \int \exp(\theta^T b - \frac{1}{2}\theta^T A\theta) d\theta \tag{4.147}$$
$$= \sqrt{\frac{(2\pi)^{2d}}{\det(A)}} \exp(b^T A^{-1} b) \tag{4.148}$$

Here,

$$\theta := \begin{pmatrix} u \\ v \end{pmatrix} \in \mathbb{R}^{2d} \tag{4.149}$$

$$b := \begin{pmatrix} R(\sum_{i\in C_0} x_i + \sum_{i\in C_1} x_i) \\ r(\sum_{i\in C_1} x_i - \sum_{i\in C_0} x_i) \end{pmatrix} \tag{4.150}$$

$$A := \begin{pmatrix} (R^2(|C_0|+|C_1|)+1)I_d & Rr(|C_1|-|C_0|)I_d \\ Rr(|C_1|-|C_0|)I_d & (r^2(|C_0|+|C_1|)+1)I_d \end{pmatrix} \tag{4.151}$$

Direct calculation gives:

$$Z := 4R^2r^2|C_0||C_1|+(R^2+r^2)(|C_0|+|C_1|)+1 \tag{4.152}$$

$$\det A = Z^d \tag{4.153}$$

$$A^{-1} = \frac{1}{Z}\begin{pmatrix} (r^2(|C_0|+|C_1|)+1)I_d & -Rr(|C_1|-|C_0|)I_d \\ -Rr(|C_1|-|C_0|)I_d & (R^2(|C_0|+|C_1|)+1)I_d \end{pmatrix} \tag{4.154}$$

$$\frac{1}{2}b^TA^{-1}b = \frac{1}{2Z}\Big(R^2(r^2|C_0|+r^2|C_1|+1)\|\sum_{i\in C_0} x_i + \sum_{i\in C_1} x_i\|_2^2 \tag{4.155}$$

$$+ r^2(R^2|C_0|+R^2|C_1|+1)\|\sum_{i\in C_0} x_i - \sum_{i\in C_1} x_i\|_2^2 \tag{4.156}$$

$$- 2R^2r^2(|C_1|-|C_0|)(\|\sum_{i\in C_1} x_i\|_2^2 - \|\sum_{i\in C_0} x_i\|_2^2)\Big) \tag{4.157}$$

$$= \frac{1}{2Z}\Big((4R^2r^2|C_1|+R^2+r^2)\|\sum_{i\in C_0} x_i\|_2^2 + (4R^2r^2|C_0|+R^2+r^2)\|\sum_{i\in C_1} x_i\|_2^2 \tag{4.158}$$

$$+ 2(R^2-r^2)(\sum_{i\in C_0} x_i)^T(\sum_{i\in C_1} x_i)\Big) \tag{4.159}$$

$$:= \frac{T}{2Z}. \tag{4.160}$$

and we have proved the claim. □

Full expansion of Equation equation 4.113 In this section, we provide a full expansion of equation equation 4.113 in power series of R and $\sqrt{d}$, which is given by the following equation

$$T_1Z_0 - T_0Z_1 = \sum_{i=0}^{4}\sum_{j=i-4}^{i-1} C(i,j)R^i d^{j/2} \tag{4.161}$$

and the coefficients $C(i,j)$ are given by:

$$C(4,3) = 2\Delta^2\left(\Delta^2\alpha_0\alpha_1+\alpha_0+\alpha_1\right)w_{n+1}^T\left(\Delta\alpha_0\alpha_1 v - \sqrt{\alpha_0}\alpha_1 w_0 + \alpha_0\sqrt{\alpha_1}w_1\right)$$

$$C(4,2) = \|v\|_2^2\left(\Delta^6\alpha_0^2\alpha_1^2 + 2\Delta^4\alpha_0^2\alpha_1\right)$$

$$+ v^T\left(w_0\left(-2\Delta^5\alpha_0^{\frac{3}{2}}\alpha_1^2 - 4\Delta^3\alpha_0^{\frac{3}{2}}\alpha_1\right) + w_1\left(2\Delta^3\alpha_0^2\sqrt{\alpha_1} - 2\Delta^3\alpha_0\alpha_1^{\frac{3}{2}}\right)\right)$$

$$+\|w_0\|_2^2\left(\Delta^4\alpha_0\alpha_1^2+2\Delta^2\alpha_0\alpha_1\right)+w_0^Tw_1\left(-2\Delta^2\alpha_0^{\frac{3}{2}}\sqrt{\alpha_1}+2\Delta^2\sqrt{\alpha_0}\alpha_1^{\frac{3}{2}}\right)+\|w_1\|_2^2\left(-\Delta^4\alpha_0^2\alpha_1-2\Delta^2\alpha_0\alpha_1\right)$$

$$+\|w_{n+1}\|_2^2\left(\Delta^4\alpha_0^2\alpha_1 - \Delta^4\alpha_0\alpha_1^2 + \Delta^2\alpha_0^2 - \Delta^2\alpha_1^2\right)$$

$$C(4,1) = w_{n+1}^T \Big(v \left(2\Delta^5 \alpha_0^2 \alpha_1 + 2\Delta^3 \alpha_0^2\right) + w_0 \left(-2\Delta^4 \alpha_0^{\frac{3}{2}} \alpha_1 - 2\Delta^2 \alpha_0^{\frac{3}{2}}\right) + w_1 \left(2\Delta^4 \alpha_0 \alpha_1^{\frac{3}{2}} + 2\Delta^2 \alpha_1^{\frac{3}{2}}\right)\Big)$$

$$C(4,0) = \|v\|_2^2 \left(\Delta^6 \alpha_0^2 \alpha_1 + \Delta^4 \alpha_0^2\right) + v^T w_0 \left(-2\Delta^5 \alpha_0^{\frac{3}{2}} \alpha_1 - 2\Delta^3 \alpha_0^{\frac{3}{2}}\right) + \|w_0\|_2^2 \left(\Delta^4 \alpha_0 \alpha_1 + \Delta^2 \alpha_0\right) + \|w_1\|_2^2 \left(-\Delta^4 \alpha_0 \alpha_1 - \Delta^2 \alpha_1\right)$$

$$C(3,2) = w_{n+1}^T \left(-\Delta^4 \alpha_0^2 \alpha_1 u + \Delta^4 \alpha_0 \alpha_1^2 u - \Delta^2 \alpha_0^2 u + \Delta^2 \alpha_1^2 u\right)$$

$$C(3,1) = v^T \left(\Delta^5 \alpha_0 \alpha_1^2 u - \Delta^3 \alpha_0^2 u + 3\Delta^3 \alpha_0 \alpha_1 u\right) + w_0^T \left(-\Delta^4 \sqrt{\alpha_0} \alpha_1^2 u + \Delta^2 \alpha_0^{\frac{3}{2}} u - 3\Delta^2 \sqrt{\alpha_0} \alpha_1 u\right) + w_1^T \left(\Delta^4 \alpha_0^2 \sqrt{\alpha_1} u + 3\Delta^2 \alpha_0 \sqrt{\alpha_1} u - \Delta^2 \alpha_1^{\frac{3}{2}} u\right)$$

$$C(3,0) = w_{n+1}^T \left(\Delta^2 \alpha_0 u - \Delta^2 \alpha_1 u\right)$$

$$C(3,-1) = v^T \left(\Delta^5 \alpha_0 \alpha_1 u + \Delta^3 \alpha_0 u\right) + w_0^T \left(-\Delta^4 \sqrt{\alpha_0} \alpha_1 u - \Delta^2 \sqrt{\alpha_0} u\right) + w_1^T \left(\Delta^4 \alpha_0 \sqrt{\alpha_1} u + \Delta^2 \sqrt{\alpha_1} u\right)$$

$$C(2,1) = w_{n+1}^T \Bigg(v \left(\Delta^5 \alpha_0^2 \alpha_1 + \Delta^5 \alpha_0 \alpha_1^2 + \frac{\Delta^3 \alpha_0^2}{2} + 3\Delta^3 \alpha_0 \alpha_1 + \frac{\Delta^3 \alpha_1^2}{2}\right) + w_0 \left(-\frac{3\Delta^4 \alpha_0^{\frac{3}{2}} \alpha_1}{2} - \frac{\Delta^4 \sqrt{\alpha_0} \alpha_1^2}{2} - \Delta^2 \alpha_0^{\frac{3}{2}} - 3\Delta^2 \sqrt{\alpha_0} \alpha_1\right) + w_1 \left(\frac{\Delta^4 \alpha_0^2 \sqrt{\alpha_1}}{2} + \frac{3\Delta^4 \alpha_0 \alpha_1^{\frac{3}{2}}}{2} + 3\Delta^2 \alpha_0 \sqrt{\alpha_1} + \Delta^2 \alpha_1^{\frac{3}{2}}\right)\Bigg)$$

$$C(2,0) = -\frac{\Delta^4 \alpha_0^2 \|u\|_2^2}{4} + \frac{\Delta^4 \alpha_1^2 \|u\|_2^2}{4} - \Delta^2 \alpha_0 \|u\|_2^2 + \Delta^2 \alpha_1 \|u\|_2^2 + \|v\|_2^2 \left(\frac{\Delta^6 \alpha_0^2 \alpha_1}{2} + \frac{\Delta^6 \alpha_0 \alpha_1^2}{2} + \frac{\Delta^4 \alpha_0^2}{2} + \frac{3\Delta^4 \alpha_0 \alpha_1}{2}\right) + v^T \left(w_0 \left(-\Delta^5 \alpha_0^{\frac{3}{2}} \alpha_1 - \frac{\Delta^5 \sqrt{\alpha_0} \alpha_1^2}{2} - \frac{3\Delta^3 \alpha_0^{\frac{3}{2}}}{2} - \frac{3\Delta^3 \sqrt{\alpha_0} \alpha_1}{2}\right) + w_1 \left(\frac{\Delta^5 \alpha_0 \alpha_1^{\frac{3}{2}}}{2} + \frac{3\Delta^3 \alpha_0 \sqrt{\alpha_1}}{2} - \frac{\Delta^3 \alpha_1^{\frac{3}{2}}}{2}\right)\right) + \|w_0\|_2^2 \left(\frac{\Delta^4 \alpha_0 \alpha_1}{2} + \Delta^2 \alpha_0\right) + w_0^T w_1 \left(\frac{\Delta^4 \alpha_0^{\frac{3}{2}} \sqrt{\alpha_1}}{2} - \frac{\Delta^4 \sqrt{\alpha_0} \alpha_1^{\frac{3}{2}}}{2}\right) + \|w_1\|_2^2 \left(-\frac{\Delta^4 \alpha_0 \alpha_1}{2} - \Delta^2 \alpha_1\right) + \|w_{n+1}\|_2^2 \left(\frac{\Delta^4 \alpha_0^2}{4} - \frac{\Delta^4 \alpha_1^2}{4} + \Delta^2 \alpha_0 - \Delta^2 \alpha_1\right)$$

$$C(2,-1) = w_{n+1}^T \Bigg(v \left(\frac{\Delta^5 \alpha_0^2}{2} + \Delta^5 \alpha_0 \alpha_1 + \frac{3\Delta^3 \alpha_0}{2} - \frac{\Delta^3 \alpha_1}{2}\right) + w_0 \left(-\frac{\Delta^4 \alpha_0^{\frac{3}{2}}}{2} - \Delta^4 \sqrt{\alpha_0} \alpha_1 - \Delta^2 \sqrt{\alpha_0}\right) + w_1 \left(\Delta^4 \alpha_0 \sqrt{\alpha_1} + \frac{\Delta^4 \alpha_1^{\frac{3}{2}}}{2} + \Delta^2 \sqrt{\alpha_1}\right)\Bigg)$$

$$
\begin{aligned}
C(2,-2) &= -\frac{\Delta^4\alpha_0\|u\|_2^2}{4} + \frac{\Delta^4\alpha_0\|w_0\|_2^2}{4} + \frac{\Delta^4\alpha_1\|u\|_2^2}{4} \\
&\quad - \frac{\Delta^4\alpha_1\|w_1\|_2^2}{4} + \|v\|_2^2\left(\frac{\Delta^6\alpha_0^2}{4} + \frac{\Delta^6\alpha_0\alpha_1}{2} + \frac{\Delta^4\alpha_0}{2}\right) \\
&\quad + v^T\left(w_0\left(-\frac{\Delta^5\alpha_0^{\frac{3}{2}}}{2} - \frac{\Delta^5\sqrt{\alpha_0}\alpha_1}{2} - \frac{\Delta^3\sqrt{\alpha_0}}{2}\right) + w_1\left(\frac{\Delta^5\alpha_0\sqrt{\alpha_1}}{2} + \frac{\Delta^3\sqrt{\alpha_1}}{2}\right)\right) \\
C(1,0) &= w_{n+1}^T\left(-\frac{\Delta^4\alpha_0^2 u}{4} + \frac{\Delta^4\alpha_1^2 u}{4} - \Delta^2\alpha_0 u + \Delta^2\alpha_1 u\right) \\
C(1,-1) &= v^T\left(\frac{\Delta^5\alpha_0\alpha_1 u}{4} + \frac{\Delta^5\alpha_1^2 u}{4} + \Delta^3\alpha_1 u\right) + w_0^T\left(-\frac{\Delta^4\alpha_0^{\frac{3}{2}} u}{4} - \frac{\Delta^4\sqrt{\alpha_0}\alpha_1 u}{4} - \Delta^2\sqrt{\alpha_0}u\right) \\
&\quad + w_1^T\left(\frac{\Delta^4\alpha_0\sqrt{\alpha_1}u}{4} + \frac{\Delta^4\alpha_1^{\frac{3}{2}}u}{4} + \Delta^2\sqrt{\alpha_1}u\right) \\
C(1,-2) &= w_{n+1}^T\left(-\frac{\Delta^4\alpha_0 u}{4} + \frac{\Delta^4\alpha_1 u}{4}\right) \\
C(1,-3) &= \frac{\Delta^5\alpha_1 u^T v}{4} - \frac{\Delta^4\sqrt{\alpha_0}u^T w_0}{4} + \frac{\Delta^4\sqrt{\alpha_1}u^T w_1}{4} \\
C(0,-1) &= w_{n+1}^T\left(v\left(\frac{\Delta^5\alpha_0^2}{8} + \frac{\Delta^5\alpha_0\alpha_1}{4} + \frac{\Delta^5\alpha_1^2}{8} + \frac{\Delta^3\alpha_0}{2} + \frac{\Delta^3\alpha_1}{2}\right)\right. \\
&\quad \left. + w_0\left(-\frac{\Delta^4\alpha_0^{\frac{3}{2}}}{4} - \frac{\Delta^4\sqrt{\alpha_0}\alpha_1}{4} - \Delta^2\sqrt{\alpha_0}\right) + w_1\left(\frac{\Delta^4\alpha_0\sqrt{\alpha_1}}{4} + \frac{\Delta^4\alpha_1^{\frac{3}{2}}}{4} + \Delta^2\sqrt{\alpha_1}\right)\right) \\
&C(0,-2) \\
&= \|v\|_2^2\left(\frac{\Delta^6\alpha_0^2}{16} + \frac{\Delta^6\alpha_0\alpha_1}{8} + \frac{\Delta^6\alpha_1^2}{16} + \frac{\Delta^4\alpha_0}{4} + \frac{\Delta^4\alpha_1}{4}\right) \\
&\quad + v^T\left(w_0\left(-\frac{\Delta^5\alpha_0^{\frac{3}{2}}}{8} - \frac{\Delta^5\sqrt{\alpha_0}\alpha_1}{8} - \frac{\Delta^3\sqrt{\alpha_0}}{2}\right) + w_1\left(\frac{\Delta^5\alpha_0\sqrt{\alpha_1}}{8} + \frac{\Delta^5\alpha_1^{\frac{3}{2}}}{8} + \frac{\Delta^3\sqrt{\alpha_1}}{2}\right)\right) \\
C(0,-3) &= w_{n+1}^T\left(-\frac{\Delta^4\sqrt{\alpha_0}w_0}{4} + \frac{\Delta^4\sqrt{\alpha_1}w_1}{4} + v\left(\frac{\Delta^5\alpha_0}{8} + \frac{\Delta^5\alpha_1}{8}\right)\right) \\
C(0,-4) &= \|v\|_2^2\left(\frac{\Delta^6\alpha_0}{16} + \frac{\Delta^6\alpha_1}{16}\right) + v^T\left(-\frac{\Delta^5\sqrt{\alpha_0}w_0}{8} + \frac{\Delta^5\sqrt{\alpha_1}w_1}{8}\right)
\end{aligned}
$$

Computer-aided verification of equation 4.161 We use the python symbolic computation package sympy for generating and verifying the expansion equation 4.161. Notice that we only need to verify the simpler case where u, v, w_0, w_1, w_{n+1} are all scalars, since the identity is separable in different coordinates and we can repeat the same process for all d dimensions of these vectors.

The following python script was used to verify equation 4.161 and generate the LaTeX equation. We only did some minor edits in the auto-generated LaTeX equation, such as replacing scalar uv with inner product $u^T v$ and adding linebreaks.

4.4.3 The importance of a carefully selected prior

We briefly discuss the importance of prior distribution chosen here. In particular, it can be shown that the prior distributions considered in [100] and [134] lead to strictly weaker (and therefore sub-optimal) lower bounds than ours in imbalanced classification. The analysis for both cases are very similar to Theorem 4.4.2, with the only difference being that the dominating terms in 4.113 are no longer $C(4,3)R^4d^{3/2}$ and $C(4,2)R^4d$. We will provide the results for both cases but only summarize the main difference in the proofs and omit the details.

The case of $R = 0$. When $R = 0$ as considered in [100], all of the terms in 4.161 disappeared except those with $i = 0$, and among them the dominating terms are $C(0,-1)d^{-1/2}$ and $C(0,-2)d^{-1}$. Using a similar argument as in Lemma 4.4.2, we can show that the Bayes Q-risk in this setting is

$$\Phi\left(-\frac{\Delta^2}{2(\Delta^2+4(\alpha_0+\alpha_1)^{-1})^{1/2}}\right) - O\left(\sqrt{\frac{\log d}{d}}\right) \tag{4.162}$$

This is a weaker bound than Lemma 4.4.2, since $\Phi(t)$ is monotone and by Cauchy-Schwarz,

$$\frac{1}{\alpha_0}+\frac{1}{\alpha_1} \geq \frac{4}{\alpha_0+\alpha_1} \tag{4.163}$$

The equality is achieved only when $\alpha_0 = \alpha_1$, which means this lower bound is only optimal when the classes are balanced.

The case of $R = \frac{\Delta}{2\sqrt{d}}$. When $R = r = \frac{\Delta}{2\sqrt{d}}$ as considered in [134], the dominating terms become linear combinations of the coefficient $C(i,j)$'s:

$$d^{-1/2}\sum_{i=0}^{4} C(i,i-1)(\frac{\Delta}{2})^i \text{ and } d^{-1}\sum_{i=0}^{4} C(i,i-2)(\frac{\Delta}{2})^i \tag{4.164}$$

Using a similar argument as in Lemma 4.4.2, we can show that the Bayes Q-risk in this setting is

$$\Phi\left(-\frac{\Delta^2}{2(\Delta^2+\alpha_0^{-1}+\alpha_1^{-1})^{1/2}}\cdot\frac{(\Delta^2+\alpha_0^{-1}+\alpha_1^{-1})}{(\Delta^2+2\alpha_0^{-1})^{1/2}(\Delta^2+2\alpha_1^{-1})^{1/2}}\right) - O\left(\sqrt{\frac{\log d}{d}}\right) \tag{4.165}$$

This is, again, a weaker bound than Lemma 4.4.2, since $\Phi(t)$ is monotone and by AM-GM inequality,

$$\Delta^2+\alpha_0^{-1}+\alpha_1^{-1} = \frac{1}{2}\left((\Delta^2+2\alpha_0^{-1})+(\Delta^2+2\alpha_1^{-1})\right) \geq (\Delta^2+2\alpha_0^{-1})^{1/2}(\Delta^2+2\alpha_1^{-1})^{1/2}. \tag{4.166}$$

The equality is achieved only when $\alpha_0 = \alpha_1$, which means this lower bound is only optimal when the classes are balanced.

4.4.4 Additional lemmas

Lemma 4.4.3. *Let $h(t) = \frac{t^2}{\sqrt{t^2+c}}$ where c is any positive constant. Then, $h(t)$ is $\sqrt{\frac{32}{27}}$-Lipschitz.*

Proof. Direct computation of the derivative $h'(t)$ yields

$$h'(t) = \frac{t\left(2c + t^2\right)}{\left(c + t^2\right)^{\frac{3}{2}}} \tag{4.167}$$

By AM-GM inequality,

$$\begin{aligned}
|h'(t)|^2 &= \frac{t^2\left(2c + t^2\right)^2}{\left(c + t^2\right)^3} && (4.168)\\
&= \frac{1}{2}\frac{\left(2t^2\right) \cdot \left(2c + t^2\right) \cdot \left(2c + t^2\right)}{\left(c + t^2\right)^3} && (4.169)\\
&\leq \frac{1}{2}\frac{\left(\frac{1}{3}\left(2t^2 + 2c + t^2 + 2c + t^2\right)\right)^3}{\left(c + t^2\right)^3} && (4.170)\\
&= \frac{32}{27} && (4.171)
\end{aligned}$$

Therefore, $|h'(t)| \leq \sqrt{\frac{32}{27}}$ and we have completed the proof. □

4.5 Numerical experiments

In the previous section, we revisited the example of [38], showed the non-monotonicity of LDA classification error and how to fix it. One may ask if this U-shape behavior is limited to LDA, or maybe it's true with much more generality? In this section, we first show empirically that this is indeed the case for widely-used algorithm like logistic regression or SVM, even in simple settings like classifying two Gaussians. Furthermore, we conjecture that the reason for the non-monotonicity is similar to LDA: the estimation of constant terms are biased in the regime of our interest, as illustrated in the experiments below.

Non-monotonicity of Test Error In Figure 4.6, we plot the test error of Logistic Regression and SVM when $\alpha_1 = 0.1, r = 6$. We choose the regularization parameter λ to be 0 and 0.1 for logistic regression / 10^{-6} and 1 for SVM. The extremely small regularization is with the purpose of simulating the behavior of "interpolating" classifiers, which caught a lot of attention these years. As shown in the figure, all of the four classifiers have the U-shape curve just like LDA. Note that the "interpolating" logistic classifier seems to be the least affected model among the four models presented here.

Effect of Minority Sample Size We repeat the above experiment for larger $\alpha_1 = 0.4$, where the U-shape curve is known as disappearing for LDA. In this scenario, the behavior of four models are more similar to what conventional wisdom suggests: unregularized models performs poorly; the U-shape curve disappeared for appropriately regularized models (both SVM and Logistic).

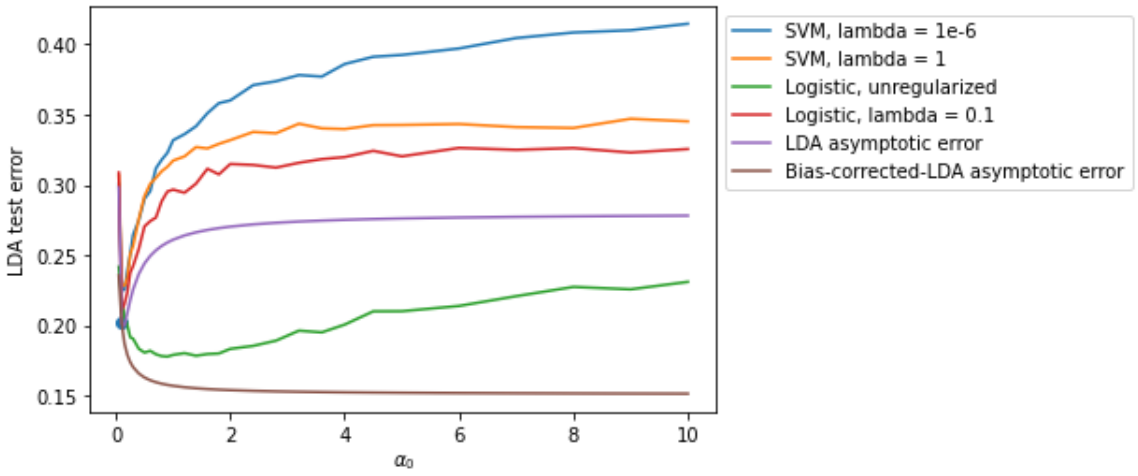


Figure 4.6: Non-monotonicity of Test Error: Test Error of Logistic Regression and SVM when $\alpha_1 = 0.1, r = 6$. The error is estimated with 20-time average of random samples.

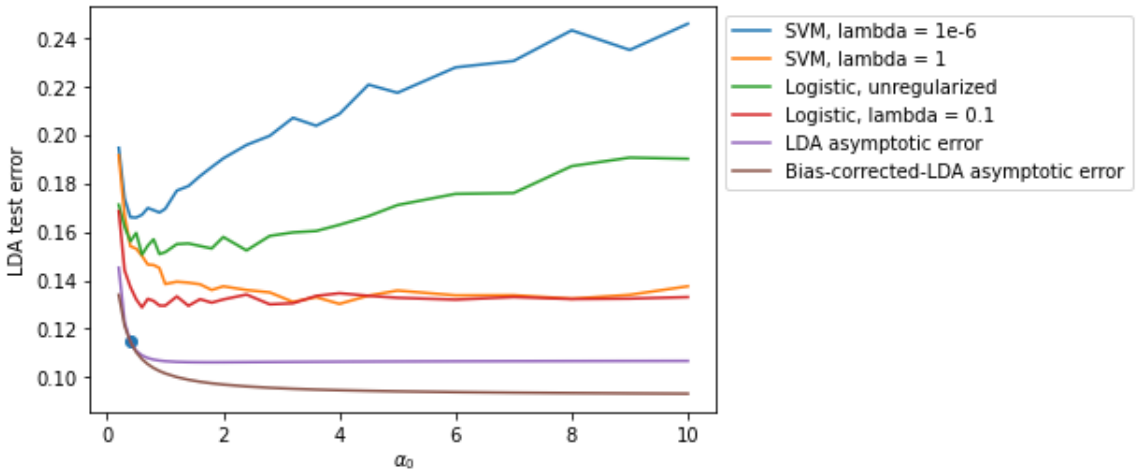


Figure 4.7: Effect of Minority Sample Size: Test Error of Logistic Regression and SVM when $\alpha_1 = 0.4, r = 6$. The error is estimated with 5-time average of random samples.

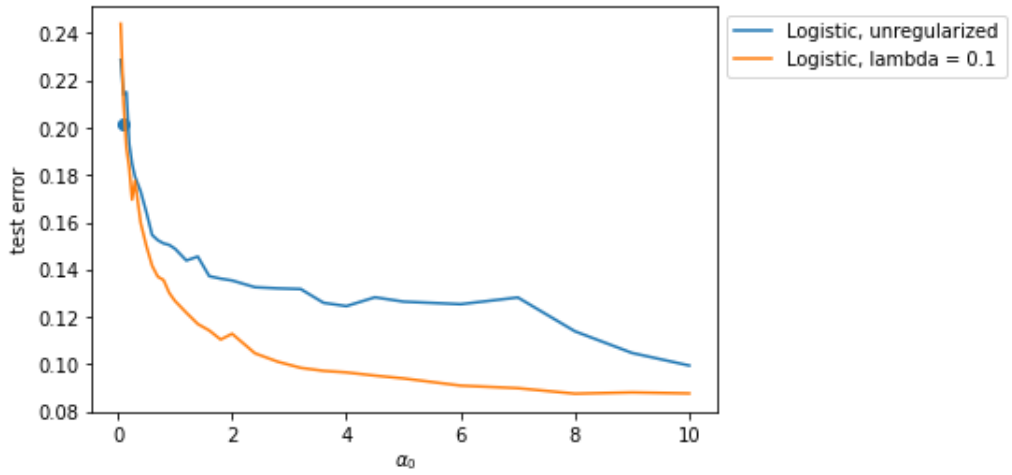


Figure 4.8: Effect of Bias-correction: Test Error of Logistic Regression when the constant terms are fixed to be 0. The error is estimated with 20-time average of random samples.

Effect of Correcting the Bias In this experiment, we fix the two gaussians to be origin-symmetric, where the population-level optimal constant is known to be zero. The only modification here is that we *do NOT* include the constant term in the linear model. The data-generation is completely the same as Figure 4.6, where the U-shape curve appears.

As shown in Figure 4.8, the U-shape curve disappears for both regularized and unregularized logistic regression. This suggests that the bias in estimating constant term may cause the irregular behavior in imbalanced classification, at least for logistic regression.

Note that the sharp transition in the unregularized model around $\alpha_0 = 7$ is likely due to the interpolation threshold: the data is linearly separable when $\alpha_0 < 7$ and the unregularized logistic regression behaves like SVM in this scenario.

Effect of Regularization In this experiment, we examine the effect of regularization by varying the level of regularization for SVM. As shown in Figure 4.9, very strong ($\lambda = 10$) regularization does somewhat mitigate the majority data effect. However, the U-shape curve seem to be only flattened down instead of disappearing. It's also worth noting that adding even stronger regularization $\lambda = 30$ completely breaks the model (sometimes test error goes to 50%), so $\lambda = 10$ is nearly an unreasonable choice of regularization.

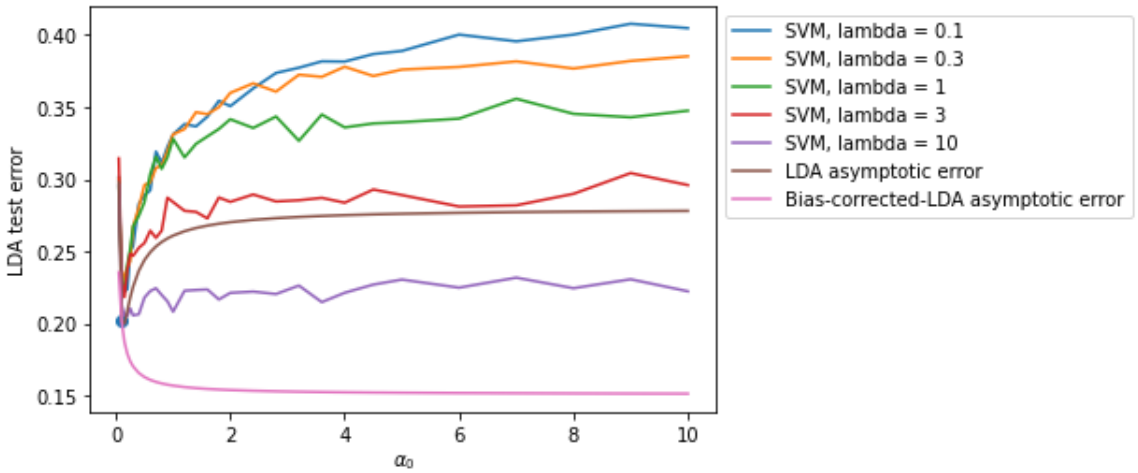


Figure 4.9: Effect of Regularization: Test Error of SVM with different level of regularization. The error is estimated with 5-time average of random samples.

4.6 Proof of Theorem 4.2.2

Proof. The convexified class-weighted ERM objective is equivalent to:

$$\min_{\beta,u} \sum_{i=1}^{n} \pi_i l(u_i) + \lambda \|\beta\|_2^2$$
$$\text{subject to:} u_i = y_i(\beta^T x_i + c)$$

where $\pi_i = \frac{q_0}{2n_0}$ if $i \in \{1, \cdots, n_0\}$, and $\pi_i = \frac{q_1}{2n_1}$ if $i \in \{n_0 + 1, \cdots, n_0 + n_1\}$.

We can re-formulate this program in Lagrangian form:

$$\min_{\beta,u,c} \max_{v} \sum_{i=1}^{n} \left(\pi_i l(u_i) + v_i y_i \beta^T x_i + v_i y_i c - v_i u_i\right) + \lambda \|\beta\|_2^2$$

Step 1: Writing the optimization problem in the form of CGMT. Let $x_i = y_i(\mu + z_i), z_i \sim N(0, I_d)$. Objective becomes

$$\min_{\beta,u,c} \max_{v} \sum_{i=1}^{n} \left(\pi_i l(u_i) + v_i y_i^2 \beta^T (\mu + z_i) + v_i y_i c - v_i u_i\right) + \lambda \|\beta\|_2^2$$

since $y_i = +1/-1$, this program is equivalent to:

$$\min_{\beta,u,c} \max_{v} \sum_{i=1}^{n} \left(\pi_i l(u_i) + v_i \beta^T \mu + v_i \beta^T z_i + v_i y_i c - v_i u_i\right) + \lambda \|\beta\|_2^2$$

Let $Z = [z_1, \cdots, z_n] \in \mathbb{R}^{d \times n}$ which has i.i.d. $N(0, 1)$ entries. Now this program can be written in matrix form:

$$\min_{\beta,u,c} \max_{v} \sum_{i=1}^{n} \pi_i l(u_i) + \beta^T Z v + (1^T v)(\beta^T \mu) - v^T u + c v^T y + \lambda \|\beta\|_2^2$$

which is in the form of PO. Using CGMT, we consider the following AO program:

$$\min_{\beta,u,c} \max_{v} \sum_{i=1}^{n} \pi_i l(u_i) + \|\beta\|_2 g^T v + \|v\|_2 h^T \beta + (1^T v)(\beta^T \mu) - v^T u + c v^T y + \lambda \|\beta\|_2^2$$

Step 2: Simplify AO so that it only involves a constant number of variables. Let $\beta = r_1 \frac{\mu}{\|\mu\|} + r_2 \mu_\perp$, where $\mu_\perp^T \mu = 0, \|\mu_\perp\|_2 = 1, r_2 \geq 0$, then $\|\beta\|_2 = \sqrt{r_1^2 + r_2^2}, \beta^T \mu = r_1 \|\mu\|_2$, AO becomes:

$$\min_{r_1, r_2, \mu_\perp, c, u} \max_{v} \sum_{i=1}^{n} \pi_i l(u_i) + \sqrt{r_1^2 + r_2^2} g^T v + \|v\|_2 h^T (r_1 \frac{\mu}{\|\mu\|} + r_2 \mu_\perp) + r_1 \|\mu\|_2 (1^T v) - v^T u + c v^T y \lambda \|\beta\|_2^2$$

Fix r_1, r_2 and minimize over $\mu_\perp$ gives the optimal $\mu_\perp$:

$$\mu_\perp^* = -\frac{P_\perp h}{\|P_\perp h\|_2}, \tag{4.172}$$

Hence we can simplify AO as:

$$\min_{r_1,r_2\geq 0,c,u} \max_{v} \sum_{i=1}^{n} \pi_i l(u_i) + \sqrt{r_1^2+r_2^2} g^T v + r_1\|v\|_2 h^T \frac{\mu}{\|\mu\|} - r_2\|v\|_2\|P_\perp h\|_2 + r_1\|\mu\|_2(1^T v) - v^T u + cv^T y + \lambda($$

Re-organizing terms in v:

$$\min_{r_1,r_2\geq 0,c,u} \max_{v} \sum_{i=1}^{n} \pi_i l(u_i) + \left(\sqrt{r_1^2+r_2^2} g + r_1\|\mu\|_2 1 + cy - u\right)^T v + r_1\|v\|_2 h^T \frac{\mu}{\|\mu\|} - r_2\|v\|_2\|P_\perp h\|_2 + \lambda(r_1^2 + $$

Fix $\|v\|_2 = r_3$ and maximize over the direction ov v, we have the objective is maximized when v is parallel to $\sqrt{r_1^2+r_2^2} g + r_1\|\mu\|_2 1 + cy - u$, and the objective becomes

$$\min_{r_1\geq 0,r_2\geq 0,c,u\in\mathbb{R}^n} \max_{r_3} \sum_{i=1}^{n} \pi_i l(u_i) + r_3\left\|\sqrt{r_1^2+r_2^2} g + r_1\|\mu\|_2 1 + cy - u\right\|_2 + r_1 r_3 h^T \frac{\mu}{\|\mu\|} - r_2 r_3\|P_\perp h\|_2 + \lambda(r_1^2 + r$$

The $\|\sqrt{r_1^2+r_2^2} g + r_1\|\mu\|_2 1 + cy - u\|_2$ term is not separable w.r.t. u. We will use the following variational representation as a workaround: $\|x\|_2 = \inf_{t>0} \frac{t}{2} + \frac{1}{2t}\|x\|_2^2$. After switching the ordering of max-min, we get:

$$\min_{r_1\geq 0,r_2\geq 0,t>0,c,u\in\mathbb{R}^n} \max_{r_3} \sum_{i=1}^{n} \pi_i l(u_i) + \frac{r_3 t}{2} + \frac{r_3}{2t}\left\|\sqrt{r_1^2+r_2^2} g + r_1\|\mu\|_2 1 + cy - u\right\|_2^2 + r_1 r_3 h^T \frac{\mu}{\|\mu\|} - r_2 r_3\|P_\perp h\|_2$$

Recall the Moreau envolope function (for $l(\cdot)$):

$$e_l(x;\tau) = \min_u \frac{1}{2\tau}(x-u)^2 + l(u) \tag{4.173}$$

Then,

$$\min_u \sum_{i=1}^{n} \pi_i l(u_i) + \frac{r_3}{2t}\left\|\sqrt{r_1^2+r_2^2} g + r_1\|\mu\|_2 1 + cy - u\right\|_2^2 = \sum_{i=1}^{n} \pi_i e_l\left(\sqrt{r_1^2+r_2^2} g_i + r_1\|\mu\|_2 + cy_i; \frac{\pi_i t}{r_3}\right) \tag{4.174}$$

Therefore, we have completed the scalarization of (AO):

$$\min_{r_1,r_2\geq 0,c,t>0} \max_{r_3} \sum_{i=1}^{n} \pi_i e_l\left(\sqrt{r_1^2+r_2^2} g_i + r_1\|\mu\|_2 + cy_i; \frac{\pi_i t}{r_3}\right) + \frac{r_3 t}{2} + r_1 r_3 h^T \frac{\mu}{\|\mu\|} - r_2 r_3\|P_\perp h\|_2 + \lambda(r_1^2 + r_2^2), \tag{4.175}$$

which is a convex-concave(?) optimization over four scalar variables. In fact, we can simplify this a little further by reducing the number of scalar variables from 4 to 3.

Let $\theta = \frac{nr_3}{t}$, the optimization problem can be re-written as:

$$\min_{r_1\geq 0, r_2\geq 0, c, t>0} \max_{\theta} \sum_{i=1}^{n} \pi_i e_l \left(\sqrt{r_1^2 + r_2^2} g_i + r_1 \|\mu\|_2 + cy_i; \frac{n\pi_i}{\theta} \right) + \frac{\theta t^2}{2n} + \frac{r_1 \theta t}{n} h^T \frac{\mu}{\|\mu\|} - \frac{r_2 \theta t}{n} \|P_\perp h\|_2 + \lambda(r_1^2 + r_2^2) \tag{4.176}$$

Now the objective is a convex quadratic form in t, for which we can find the closed-form minimizer: $t^* = r_1 h^T \frac{\mu}{\|\mu\|} + r_2 \|P_\perp h\|_2$. After introducing another change of variable $R = \sqrt{r_1^2 + r_2^2}$, we finally reached the fully simplified "scalarized" version of AO:

$$\min_{r_1, R, c: |r_1| \leq R} \max_{\theta} \sum_{i=1}^{n} \pi_i e_l \left(R g_i + r_1 \|\mu\|_2 + cy_i; \frac{n\pi_i}{\theta} \right) - \frac{\theta}{2n} \left(r_1 h^T \mu - \sqrt{R^2 - r_1^2} \|P_\perp h\|_2 \right)^2 + \lambda R^2 \tag{4.177}$$

Step 3: Analyze the high dimensional asymptotics of scalarized AO. Next, we will study the high dimensional asymptotics of the objective [2] .

We start with the second term: note that $h^T \mu \sim N(0, \|\mu\|_2^2)$, so $\frac{1}{\sqrt{n}} h^T \mu \to_{a.s.} 0$. Similarly, recall that h is d-dimensional standard Gaussian, we have $P_\perp h \sim N(0, P_\perp P_\perp^T)$ (which is informally, a $d-1$ dimensional standard Gaussian), hence $\frac{1}{\sqrt{d-1}} \|P_\perp h\|_2 \to_{a.s.} 1$, and $\frac{1}{\sqrt{n}} \|P_\perp h\|_2 \to (\alpha_0 + \alpha_1)^{-1/2}$. Therefore, the second term becomes

$$\frac{\theta(R^2 - r_1^2)}{2(\alpha_0 + \alpha_1)} \tag{4.178}$$

Now we focus on the first term, which can be decomposed according to the two class of samples:

$$\frac{q_0}{2n_0} \sum_{i=1}^{n_0} e_l \left(R g_i + r_1 \|\mu\|_2 - c; \frac{nq_0}{2n_0 \theta} \right) + \frac{q_1}{2n_1} \sum_{i=n_0+1}^{n_0+n_1} e_l \left(R g_i + r_1 \|\mu\|_2 + c; \frac{nq_1}{2n_1 \theta} \right) \tag{4.179}$$

Notice that $\frac{n}{n_0} \to \frac{\alpha_0+\alpha_1}{\alpha_0}$, $\frac{n}{n_1} \to \frac{\alpha_0+\alpha_1}{\alpha_1}$, and g_i are independent (scalar) standard normal variables, it's not difficult to see the above summation converges to a Gaussian expectation:

$$\frac{q_0}{2} \mathbb{E}_{Z\sim N(0,1)} \left[e_l \left(RZ + r_1 \|\mu\|_2 - c; \frac{(\alpha_0 + \alpha_1) q_0}{2\alpha_0 \theta} \right) \right] + \frac{q_1}{2} \mathbb{E}_{Z\sim N(0,1)} \left[e_l \left(RZ + r_1 \|\mu\|_2 + c; \frac{(\alpha_0 + \alpha_1) q_1}{2\alpha_1 \theta} \right) \right] \tag{4.180}$$

To summarize, the high-dimensional asymptotics of the scalarized AO can be formulated as follows:

$$\min_{r_1, R: 0 \leq r_1 \leq \frac{R}{\|\mu\|}} \max_{\theta} \frac{1}{2} \mathbb{E}_{Z\sim N(0,1)} \left[q_0 e_l \left(RZ + r_1 \|\mu\|_2 - c; \frac{(\alpha_0 + \alpha_1) q_0}{2\alpha_0 \theta} \right) + q_1 e_l \left(RZ + r_1 \|\mu\|_2 + c; \frac{(\alpha_0 + \alpha_1) q_1}{2\alpha_1 \theta} \right) \right] \tag{4.181}$$

[2] Here, for simplicity we are talking about point-wise covergence, i.e. the limit for a fixed choice of (r_1, R, θ) when $d, n_0, n_1 \to \infty$. To justify that this point-wise limit indeed characterizes the limiting behavior of the optimization problem, we need to check several technical conditions provided in [40].

Step 4: Characterizing the Asymptotic Classification Error We pause here and provide some interpretation of the asymptotic scalarized AO equation 4.181. Recall that in Step 2, we introduced change of variables:

$$\beta = r_1 \frac{\mu}{\|\mu\|_2} + r_2 \mu_\perp \tag{4.182}$$

$$R = \sqrt{r_1^2 + r_2^2} \tag{4.183}$$

Hence, $R = \|\beta\|_2$ and $r_1 = \frac{\beta^T \mu}{\|\mu\|_2}$. It was shown in [132] that, under mild technical conditions, $\|\beta\|_2 \to_P R^*$ and $\frac{\beta^T \mu}{\|\mu\|_2} \to_P r_1^*$. Since the classification error can be formulated as

$$R(\beta) = \frac{1}{2}\Phi\left(-\frac{\beta^T\mu + c}{\|\beta\|_2}\right) + \frac{1}{2}\Phi\left(-\frac{\beta^T\mu - c}{\|\beta\|_2}\right), \tag{4.184}$$

Therefore,

$$R(\beta) \to_P \Phi\frac{1}{2}\left(-\frac{r_1^*\|\mu\|_2 + c^*}{R^*}\right) + \frac{1}{2}\left(-\frac{r_1^*\|\mu\|_2 - c^*}{R^*}\right), \tag{4.185}$$

and we have completed the proof. □

4.7 Comparison with Related Works

4.7.1 Comparison with [69]

Sanity Check: Consider the balanced setting as a special case, where we have $\alpha_0 = \alpha_1$. Let $\alpha = \alpha_0 + \alpha_1$, the optimization problem above is equivalent to:

$$\min_{r_1, R: |r_1| \le R} \max_{\theta} \mathbb{E}_{Z\sim N(0,1)}\left[e_l\left(RZ + r_1\|\mu\|_2; \frac{1}{\theta}\right)\right] - \frac{\theta(R^2 - r_1^2)}{2\alpha} + \lambda R^2 \tag{4.186}$$

Let $r_2 = \sqrt{R^2 - r_1^2}$, $\lambda = 0$

$$\min_{r_1, r_2} \max_{\theta} \mathbb{E}_{Z\sim N(0,1)}\left[e_l\left(\sqrt{r_1^2 + r_2^2}Z + r_1\|\mu\|_2; \frac{1}{\theta}\right)\right] - \frac{\theta r_2^2}{2\alpha} \tag{4.187}$$

Let $\theta = \theta' \frac{\sqrt{\alpha}}{r_2}$:

$$\min_{r_1, r_2} \max_{\theta'} \mathbb{E}_{Z\sim N(0,1)}\left[e_l\left(\sqrt{r_1^2 + r_2^2}Z + r_1\|\mu\|_2; \frac{r_2}{\theta'\sqrt{\alpha}}\right)\right] - \frac{\theta' r_2}{2\sqrt{\alpha}} \tag{4.188}$$

This is equivalent to a special case $\epsilon_0 = 0$ in [69], corollary 5.1.c: after changing notation (ours $\to$ theirs) as $r_1 \to \theta, r_2 \to \alpha, \theta' \to \beta, \alpha \to \delta, Z \to g, \|\mu\|_2 \to \sigma_{M,2}$, the optimization problem above becomes

$$\min_{\theta, \alpha \ge 0} \max_{\theta'} \mathbb{E}_{g\sim N(0,1)}\left[e_l\left(\sqrt{\alpha^2 + \theta^2}g + \theta\|\mu\|_2; \frac{\alpha}{\beta\sqrt{\delta}}\right)\right] - \frac{\alpha\beta}{2\sqrt{\delta}} \tag{4.189}$$

which is the same as equation (5.8) in [69].

4.8 Technical Lemmas

4.8.1 Moreau Envelope

Recall that for convex loss function $l(\cdot)$, the Monreau envelope is defined as:

$$e_l(x;\tau) = \min_u \frac{1}{2\tau}(x-u)^2 + l(u), \tag{4.190}$$

Define

$$l_{\text{square}}(u) = \frac{1}{2}(1-u)^2 \tag{4.191}$$

$$l_{\text{hinge}}(u) = \max\{1-u, 0\} \tag{4.192}$$

$$l_{\text{logistic}}(u) = \ln(1+e^{-u}), \tag{4.193}$$

We can compute the Moreau envelope as follows.

(1) **Square loss:**

$$e_{l_{\text{square}}}(x;\tau) = \frac{1}{2(1+\tau)}(1-x)^2 \tag{4.194}$$

Proof:

$$e_{l_{\text{square}}}(x;\tau) = \min_u \frac{1}{2\tau}(x-u)^2 + \frac{1}{2}(1-u)^2 \tag{4.195}$$

Let $\frac{\partial}{\partial u} = 0$, we have

$$\frac{1}{\tau}(u-x) + u - 1 = 0 \Rightarrow u = \frac{x+\tau}{1+\tau} \tag{4.196}$$

Therefore,

$$e_{l_{\text{square}}}(x;\tau) = \min_u \frac{1}{2\tau}(x-u)^2 + \frac{1}{2}(1-u)^2 \tag{4.197}$$

$$= \frac{1}{2\tau} \cdot \frac{\tau^2}{(1+\tau)^2}(x-1)^2 + \frac{1}{2}\frac{1}{(1+\tau)^2}(x-1)^2 \tag{4.198}$$

$$= \frac{1}{2(1+\tau)}(x-1)^2 \tag{4.199}$$

(2) **Hinge Loss:**

$$e_{l_{\text{hinge}}}(x) = \begin{cases} 1-x-\frac{\tau}{2} & \text{if } x < 1-\tau \\ \frac{1}{2\tau}(1-x)^2 & \text{if } 1-\tau \le x \le 1 \\ 0 & \text{if } x > 1 \end{cases} \tag{4.200}$$

Proof:

$$e_{l_{\text{hinge}}}(x;\tau) = \min_u \frac{1}{2\tau}(x-u)^2 + \max(1-u, 0) \tag{4.201}$$

Let $\frac{\partial}{\partial u} = 0$, we have

$$\frac{\partial}{\partial u} = \frac{1}{\tau}(u-x) - I[u<1] \tag{4.202}$$

when $x < 1-\tau$, $u = x+\tau$; when $x \geq 1$, $u = x$; otherwise, $u = 1$.

(3) **Logistic Loss:**

$e_{l_{\text{logistic}}}(x;\tau)$ does not have a closed form solution, but it can be very efficiently approximated by newton iterations [39].

Define

$$g(u;\tau) = \frac{1}{2\tau}(x-u)^2 + \ln(1+e^{-u}) \tag{4.203}$$

Then,

$$g'(u;\tau) = \frac{1}{\tau}(u-x) - \frac{1}{1+e^u}$$
$$g''(u;\tau) = \frac{1}{\tau} + \frac{e^u}{(1+e^u)^2}$$

The Newton update can be formulated as:

$$u_{k+1} = u_k - \frac{g'(u_k;\tau)}{g''(u_k;\tau)} \tag{4.204}$$

According to [39], it usually takes at most 3 iterations.

4.8.2 Solving the asymptotic version of AO numerically

We need a few useful properties of the Moreau envelope, in particular, the ones about its derivatives:

Lemma 4.8.1 (From Lemma D.1 of [132])**.** *Define*

$$\frac{\partial}{\partial x} e_l(x,\tau) = \frac{1}{\tau}(x - \textit{prox}_l(x,\tau)) \tag{4.205}$$
$$\frac{\partial}{\partial \tau} e_l(x,\tau) = -\frac{1}{2\tau^2}(x - \textit{prox}_l(x,\tau))^2 \tag{4.206}$$

Below we derive the fixed point iterations for solving asymptotic version of AO.

We will use the following equivalent form of AO:

$$\min_{r_1,r_2\geq 0} \max_{\theta}$$
$$\frac{1}{2}\mathbb{E}_{Z\sim N(0,1)}\left[q_0 e_l\left(\sqrt{r_1^2+r_2^2}Z + r_1\|\mu\|_2 - c; \frac{(\alpha_0+\alpha_1)q_0}{2\alpha_0\theta}\right) + q_1 e_l\left(\sqrt{r_1^2+r_2^2}Z + r_1\|\mu\|_2 + c; \frac{(\alpha_0+\alpha_1)q_1}{2\alpha_1\theta}\right)\right]$$
$$- \frac{\theta r_2^2}{2(\alpha_0+\alpha_1)} + \lambda(r_1^2+r_2^2)$$

By the first order optimality condition,

$$\begin{aligned}\frac{\partial}{\partial r_1} &= 0\\ \frac{\partial}{\partial r_2} &= 0\\ \frac{\partial}{\partial \theta} &= 0\\ \frac{\partial}{\partial c} &= 0\end{aligned}$$

(Here are some details about the calculation)

Introducing a few shorthands: (only works for logistic; otherwise the definition of C_0, C_1 needs to be changed. Also requrires differentiability, otherwise won't be in a nice form like below).

$$\begin{aligned}Z_0 &\sim N(r_1\|\mu\|_2 - c, r_1^2 + r_2^2)\\ Z_1 &\sim N(r_1\|\mu\|_2 + c, r_1^2 + r_2^2)\\ \tau_0 &= \frac{(\alpha_0+\alpha_1)q_0}{2\alpha_0\theta}\\ \tau_1 &= \frac{(\alpha_0+\alpha_1)q_1}{2\alpha_1\theta}\\ A_0 &= \mathbb{E}[l'\left(\text{prox}_l(Z_0,\tau_0)\right)]\\ A_1 &= \mathbb{E}[l'\left(\text{prox}_l(Z_1,\tau_1)\right)]\\ B_0 &= \frac{\tau_0}{\theta}\mathbb{E}[\left(l'\left(\text{prox}_l(Z_0,\tau_0)\right)\right)^2]\\ B_1 &= \frac{\tau_1}{\theta}\mathbb{E}[\left(l'\left(\text{prox}_l(Z_1,\tau_1)\right)\right)^2]\\ C_0 &= \mathbb{E}\left[\frac{l''\left(\text{prox}_l(Z_0,\tau_0)\right)}{1+\tau_0 l''\left(\text{prox}_l(Z_0,\tau_0)\right)}\right]\\ C_1 &= \mathbb{E}\left[\frac{l''\left(\text{prox}_l(Z_1,\tau_1)\right)}{1+\tau_1 l''\left(\text{prox}_l(Z_1,\tau_1)\right)}\right]\\ A &= \frac{q_0}{2}A_0 + \frac{q_1}{2}A_1\\ B &= \frac{q_0}{2}B_0 + \frac{q_1}{2}B_1\\ C &= \frac{q_0}{2}C_0 + \frac{q_1}{2}C_1\end{aligned}$$

Then,

$$\frac{\partial}{\partial r_1} = \|\mu\|_2 A + r_1 C + 2\lambda r_1 \qquad = 0 \tag{4.207}$$

$$\frac{\partial}{\partial r_2} = r_2 C - \frac{\theta r_2}{\alpha_0 + \alpha_1} + 2\lambda r_2 \qquad = 0 \tag{4.208}$$

$$\frac{\partial}{\partial \theta} = \frac{1}{2} B - \frac{r_2^2}{2(\alpha_0 + \alpha_1)} \qquad = 0 \tag{4.209}$$

$$\frac{\partial}{\partial c} = -\frac{q_0}{2} A_0 + \frac{q_1}{2} A_1 \qquad = 0 \tag{4.210}$$

$$\tag{4.211}$$

By equation 4.209,

$$r_2 = \sqrt{(\alpha_0 + \alpha_1) B} \tag{4.212}$$

By equation 4.208,

$$\theta = (\alpha_0 + \alpha_1)(C + 2\lambda) \tag{4.213}$$

Combining with equation 4.207,

$$r_1 = \frac{\|\mu\|_2 A}{C + 2\lambda} = \frac{(\alpha_0 + \alpha_1)\|\mu\|_2 A}{\theta} \tag{4.214}$$

equation 4.210 does not have a closed form solution for c. Instead, we use one Newton iteration to find a approximate solution.

$$c_{t+1} = c_t - \frac{\frac{\partial}{\partial c}(c_t)}{\frac{\partial^2}{\partial c^2}(c_t)} = c_t - \frac{-\frac{q_0}{2} A_0 + \frac{q_1}{2} A_1}{C} \tag{4.215}$$

4.8.3 Lemmas for Lower Bounds

Lemma 4.8.2. *, Let $\theta \sim$ Uniform(S^{d-1}), where S^{d-1} is the unit ball in $\mathbb{R}^d$. Let v be any unit vector in $\mathbb{R}^d$,then*

$$2\theta^T v - 1 \sim B(\frac{d-1}{2}, \frac{d-1}{2}), \tag{4.216}$$

where $B(\alpha, \beta)$ denotes the Beta distribution.

Lemma 4.8.3. *Let $X \sim B(\alpha, \beta)$, then*

$$\mathbb{E}[\exp(tX)] =_1 F_1(\alpha, \alpha + \beta, t), \tag{4.217}$$

where ${}_1F_1(\alpha, \alpha + \beta, t)$ is the confluent hypergeometric function*:*

$${}_1F_1(\alpha, \alpha + \beta, t) = \sum_{k=0}^{\infty} \frac{t^k}{k!} \prod_{m=0}^{k-1} \frac{\alpha + m}{\alpha + \beta + m} \tag{4.218}$$

Lemma 4.8.4. *Let $\theta \sim$ Uniform(S^{d-1}), where S^{d-1} is the unit ball in $\mathbb{R}^d$. Then for any $v \in \mathbb{R}^d$,*

$$\mathbb{E}[\exp(\theta^T v)] = \exp(-\|v\|_2) \sum_{k=0}^{\infty} \frac{\|v\|_2^k}{k!} \prod_{m=0}^{k-1} (1 + \frac{m}{d+1+m}) \tag{4.219}$$

which is an increasing function of $\|v\|_2$

Proof. Monotonicity:

$$\frac{\partial}{\partial \|v\|_2} \mathbb{E}[\exp(\theta^T v)] = \exp(-\|v\|_2) \sum_{k=0}^{\infty} \frac{\|v\|_2^k}{k!} \prod_{m=0}^{k} (1 + \frac{m}{d+1+m}) - \exp(-\|v\|_2) \sum_{k=0}^{\infty} \frac{\|v\|_2^k}{k!} \prod_{m=0}^{k-1} (1 + \frac{m}{d+1+m} \tag{4.220}$$

$$= \exp(-\|v\|_2) \sum_{k=0}^{\infty} \frac{\|v\|_2^k}{k!} \frac{k}{d+1+k} \prod_{m=0}^{k-1} (1 + \frac{m}{d+1+m}) \tag{4.221}$$

$$\geq 0 \tag{4.222}$$

□

The following lemma (analogous to Theorem 36.64 in the reference above) provides a sufficient condition for asymptotic minimax optimality.

Lemma 4.8.5. *Let $\{f_d\}_{d=1}^{\infty}$ be a sequence of classifiers, which satisfies the following conditions:*

1. *For each f_d, there exists a prior distribution Q_d, so that f_d is the MAP classifier under prior Q_d.*
2. *$\lim_{d\to\infty} \sup_{(\mu_0,\mu_1)\in\mathcal{P}_d(\Delta)} R_{n_0,n_1,\mu_0,\mu_1}(f_d) = R_\infty$ exists.*
3. *There exists a sequence of positive real number $\{\gamma_d\}_{d\geq 1}$, where $\gamma_d = o_d(1)$, such that the following inequality holds uniformly for all $(\mu_0, \mu_1) \in \mathcal{P}_d$:*

$$|R_{n_0,n_1,\mu_0,\mu_1}(f_d) - R_\infty| \leq \gamma_d \tag{4.223}$$

Then, we can claim that $\{f_d\}_{d=1}^{\infty}$ is asymptotically minimax optimal.

In other words, if $\{f_d\}_{d=1}^{\infty}$ is a sequence of MAP classifiers with approximately constant risk among $(\mu_0, \mu_1) \in \mathcal{P}_d$, then it's asymptotically minimax optimal.

Proof. By condition (1), since Minimax Risk is lower bounded by Bayes Risk, we have

$$M(n_0, n_1, \mathcal{P}_d) \geq B_{n_0,n_1,Q}(f_d) \tag{4.224}$$

By condition (3), since expectation is lower bounded by the minimum, we have

$$B_{n_0,n_1,Q}(f_d) \geq R_\infty - \gamma_d \tag{4.225}$$

Combining the two steps above and let $d \to \infty$, we have

$$\limsup_{n_0/d=\alpha_0, n_1/d=\alpha_1, d\to\infty} M(n_0, n_1, \mathcal{P}_d) \geq R_\infty \tag{4.226}$$

However, by condition (2), we know that the asymptotic risk of $\{f_d\}_{d=1}^{\infty}$ is R_∞, therefore, it's asymptotically minimax optimal and we have completed the proof. □

4.8.4 Exact Asymptotic Minimax Risk in Balanced Setting

In this section, we discuss the asymptotic optimality of LDA in the regime $n_0 = n_1 = \frac{1}{2}\alpha d, d \to \infty$. We have already showed that the variant of LDA with known covariance has the asymptotic risk of

$$R_{LDA} \to \Phi(-\frac{\Delta^2}{2\sqrt{\Delta^2 + 4\alpha^{-1}}}), \tag{4.227}$$

But is this risk asymptotically optimal? The answer is yes, as shown in the following theorem:

Theorem 4.8.1. *The asymptotic minimax risk is given by*

$$R^*_{asymp}(\Delta, \frac{1}{2}\alpha, \frac{1}{2}\alpha) = \Phi(-\frac{\Delta^2}{2\sqrt{\Delta^2 + 4\alpha^{-1}}}), \tag{4.228}$$

which is achieved by LDA with known covariances.

Proof. We already know that LDA achieves the asymptotic risk above, so it's sufficient to prove $LHS \geq RHS$. In fact, we can show that in a more restricted parameter space

$$\mathcal{P}_d^{sym} = \{(\mu_0, \mu_1) : \mu_0 = -\mu, \mu_1 = \mu, \mu = \frac{\Delta}{2}\} \subseteq \mathcal{P}_d \tag{4.229}$$

the minimax risk remain the same. The strategy is to use Lemma 4.8.5.

Let Q_d be the uniform distribution over the hyper-sphere $\frac{\Delta}{2} \cdot S^{d-1}$, and f_d be the MAP classifier under Q_d. [92] had the following observation:

$$f_d(x; x_1, \cdots, x_n) = \text{sign}(\frac{1}{n}\sum_{i=1}^n y_i x_i)^T x) \tag{4.230}$$

where we recall that $y_i = -1$ when $1 \leq i \leq n_0$, and $y_i = 1$ when $n_0 + 1 \leq i \leq n$. Below we rephrase their proof.

By definition of MAP classifier,

$$f_d(x; x_1, \cdots, x_n) = \underset{y \in \{+1, -1\}}{\text{argmax}} \Pr[Y = y | X = x; x_1, \cdots, x_n] \tag{4.231}$$

Notice that

$$\begin{aligned}
\Pr[Y = y | X = x; x_1, \cdots, x_n] &\propto p(Y = y, X = x, x_1, \cdots, x_n) && (4.232)\\
&= \mathbb{E}_{\mu \sim Q_d} p(Y = y, X = x, x_1, \cdots, x_n | \mu) && (4.233)\\
&\propto \mathbb{E}_{\mu \sim Q_d} \exp(-\frac{1}{2}\|x - y\mu\|_2^2 - \frac{1}{2}\sum_{i=1}^n \|x_i - y_i\mu\|_2^2) && (4.234)\\
&\propto \mathbb{E}_{\mu \sim Q_d}[\exp(yx + \sum_{i=1}^n y_i x_i)] && (4.235)
\end{aligned}$$

By Lemma 4.8.4, this is a monotone function of $\|yx + \sum_{i=1}^n y_i x_i\|_2$. Therefore, $f_d(x) = 1$ if and only if

$$\|x + \sum_{i=1}^n y_i x_i\|_2 \geq \|-x + \sum_{i=1}^n y_i x_i\|_2 \tag{4.236}$$

which is equivalent to

$$f_d(x; x_1, \cdots, x_n) = \text{sign}(\frac{1}{n}\sum_{i=1}^{n} y_i x_i)^T x) \tag{4.237}$$

and we have completed the proof of equation 4.230.

Let $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} y_i x_i$. Since $f_d(x; x_1, \cdots, x_n) = \hat{\mu}^T x$ is a linear classifier, its classification error condition on any fixed μ can be written as

$$\Phi(-\frac{\hat{\mu}^T \mu}{\|\hat{\mu}\|_2}) \tag{4.238}$$

(the calculation below is not very rigorous)

Since $\hat{\mu} \sim N(\mu, \frac{1}{n} I_d)$, we have $\hat{\mu}^T \mu = \|\mu\|_2^2 + o_d(1) = \frac{\Delta^2}{4} + o_d(1)$, and $\|\hat{\mu}\|_2 = \sqrt{\|\mu\|_2^2 + \frac{d}{n}} + o_d(1) = \sqrt{\frac{\Delta^2}{4} + \frac{1}{\alpha}} + o_d(1)$, hence, for any μ,

$$R_\mu(f_d) = \Phi(\frac{\Delta^2}{2\sqrt{\Delta^2 + \frac{4}{\alpha}}}) + o_d(1) \tag{4.239}$$

which proves condition (2,3) of Lemma 4.8.5. Therefore, $\{f_d\}_{d=1}^{\infty}$ is asymptotically minimax-optimal and we have completed the proof. □

Chapter 5

Interpolation in Distributionally Robust Optimization

5.1 Introduction

It has been well established by prior work that overparameterized models, whose number of parameters is much larger than the number of training samples, can empirically achieve high test performance on a variety of tasks, in contrast to the theory that models with too many parameters could have large generalization error.

This high performance however is *on average*; a large body of prior work [18, 62, 130] showed that these models tend to learn *spurious features*, such as learning the background in image classification instead of the object, and learning keywords like "not" in language sentiment analysis instead of really understanding the sentences. Consequently, these models are *unfair*, i.e. they fail on certain minority groups (such as positive sentences containing "not") while still having high average-case performance. To solve this problem, people have proposed various *reweighting algorithms* to improve the model's worst-group performance, such as upweighting the minority groups or using distributionally robust optimization (DRO) based methods [47, 58, 117, 123].

While reweighting algorithms in principle can improve the worst-group performance compared to vanilla empirical risk minimization (ERM), previous work empirically found that when applied to modern overparameterized models, these methods could overfit very easily, so that they have poor test worst-group performance. For example, Sagawa et al. [117] studied a reweighting algorithm called group DRO. They found that compared to ERM, group DRO does improve the worst-group test accuracy by a large margin at the early stage of training. However, if no regularization is applied, then as training goes on, the worst-group test accuracy of group DRO will drop significantly and eventually to a level almost the same as ERM. Some previous work tried to explain why reweighting algorithms can overfit so easily. For instance, Sagawa et al. [119] argued that with these algorithms, an overparameterized model would typically memorize all training samples in the minority groups while still learning the spurious features from the majority groups.

In this work, we aim to understand the overfitting phenomenon in reweighting algorithms by studying their implicit biases. Specifically, we prove for a family of overparameterized neural

networks that for almost all reweighting algorithms, the model always converges to the *same interpolator* that fits all training samples, no matter the reweighting. Since ERM is a special case of such reweighting algorithms (where each sample receives the same weight), this means that the implicit biases of all reweighting algorithms are equivalent to that of ERM. Consequently, the model trained by any reweighting algorithm always overfits to the ERM interpolator, so we cannot hope for its worst-group test performance to be better than ERM. In short, *reweighting algorithms always overfit*.

Given this pessimistic result, we analyze whether regularization can help mitigate overfitting, as proposed by Sagawa et al. [117]. We find that a necessary condition for regularization to work is that it considerably lowers the training performance. Specifically, we prove that if the overparameterized model trained by a reweighting algorithm with regularization can still perform almost perfectly on the training set, then overfitting is still inevitable. This explains why in practice we need very large regularization that prevents the model from achieving nearly zero training error to avoid overfitting.

Our results have two important consequences for practice: (i) We should always use large regularization or early stopping when optimizing for worst-group performance; (ii) We should always try to obtain more training samples, e.g. with strong data augmentation or semi-supervised learning.

5.1.1 Related work

Group fairness. Group fairness in machine learning was first studied in Hardt et al. [57] and Zafar et al. [159], where they required the model to perform equally well over all groups. Later, Hashimoto et al. [58] studied another type of group fairness called Rawlsian max-min fairness [116], which does not require equal performance but rather requires high performance on the worst-off group. The problem we study in this paper is most closely related to Rawlsian max-min fairness. A large body of recent work in machine learning have studied how to improve this worst-group performance [47, 93, 104, 155, 163]. Recent work however observe that these approaches, when used with modern overparameterized models, easily overfit [117, 119]. Apart from group fairness, there are also other notions of fairness, such as individual fairness [48, 160] and counterfactual fairness [83], which we do not study in this work.

Implicit bias under the overparameterized setting. For overparameterized models, there could be many model parameters which all minimize the training loss. In such cases, it is of interest to study the implicit bias of specific optimization algorithms such as gradient descent i.e. to what training loss minimizer the model parameters will converge to [1, 46]. Our results use the NTK formulation of wide neural networks [67], and specifically we use linearized neural networks to approximate such wide neural networks following Lee et al. [89]. There is some criticism of this line of work, e.g. Chizat et al. [31] argued that infinitely wide neural networks fall in the "lazy training" regime and results might not be transferable to general neural networks. Nonetheless such wide neural networks are being widely studied in recent years, since they provide considerable insights into the behavior of more general neural networks, which are typically intractable to analyze otherwise.

5.2 Preliminaries

Consider a data domain $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^d \times \mathbb{R}$ that consists of K groups (subdomains)[1], where each data point belongs to one of the groups[2]. We assume that the input space $\mathcal{X}$ is a subset of the unit ball of $\mathbb{R}^d$, such that any $\boldsymbol{x} \in \mathcal{X}$ satisfies $\|\boldsymbol{x}\|_2 \leq 1$. We are given a training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ *i.i.d.* sampled from some underlying distribution P over $\mathcal{X} \times \mathcal{Y}$. Let the K groups be $\mathcal{D}_1, \cdots, \mathcal{D}_K$ where each $\mathcal{D}_i$ is a subset of $\mathcal{X} \times \mathcal{Y}$. Let $P_k(\boldsymbol{z}) = P(\boldsymbol{z} | \boldsymbol{z} \in \mathcal{D}_k)$ be the conditional data distribution over $\mathcal{D}_k$, where $\boldsymbol{z} = (\boldsymbol{x}, y)$. Denote $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) \in \mathbb{R}^{d \times n}$, and $\boldsymbol{Y} = (y_1, \cdots, y_n) \in \mathbb{R}^n$; for any function $g : \mathcal{X} \mapsto \mathbb{R}$, we overload notation and use $g(\boldsymbol{X}) = (g(\boldsymbol{x}_1), \cdots, g(\boldsymbol{x}_n))$. Let the loss function be $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$. In vanilla training, the goal is to minimize the expected risk denoted by $\mathcal{R}(f; P) = \mathbb{E}_{\boldsymbol{z} \sim P}[\ell(f(\boldsymbol{x}), y)]$, which is done by minimizing the empirical risk $\hat{\mathcal{R}}(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(\boldsymbol{x}_i), y_i)$.

For tasks requiring high worst-group performance, the goal is to train a model $f : \mathcal{X} \to \mathcal{Y}$ that performs well over every P_k, which can be achieved by minimizing the *worst-group risk* defined as

$$\mathcal{R}_{\max}(f; P) = \max_{k=1,\cdots,K} \mathcal{R}(f; P_k) = \max_{k=1,\cdots,K} \mathbb{E}_{\boldsymbol{z} \sim P}[\ell(f(\boldsymbol{x}), y) | z \in \mathcal{D}_k] \tag{5.1}$$

5.2.1 Reweighting Algorithms

Most existing methods that minimize the worst-group risk are *reweighting algorithms* that assign each sample with a weight during training and minimize the weighted average risk. At time t, we assign a weight $q_i^{(t)}$ to sample $\boldsymbol{z}_i$, and minimize the weighted empirical risk:

$$\hat{\mathcal{R}}_{\boldsymbol{q}^{(t)}}(f) = \sum_{i=1}^n q_i^{(t)} \ell(f(\boldsymbol{x}_i), y_i) \tag{5.2}$$

where $\boldsymbol{q}^{(t)} = (q_1^{(t)}, \cdots, q_n^{(t)})$ and $q_1^{(t)} + \cdots + q_n^{(t)} = 1$.

A *static reweighting algorithm* assigns to each $\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)$ a fixed weight q_i that does not change during training, i.e. $q_i^{(t)} \equiv q_i$. A famous example is *Importance Weighting* (IW, Shimodaira [123]), in which if $\boldsymbol{z}_i \in \mathcal{D}_k$ and the size of $\mathcal{D}_k$ is n_k, then $q_i = (K n_k)^{-1}$. Under IW, each group has the same weight, and the reweighted empirical risk is a simple (unweighted) average of the empirical risk over each group, so that each group has an equal contribution to the overall risk objective. Note that ERM is also a special case of static reweighting algorithms: by assigning $q_1 = \cdots = q_n = 1/n$.

On the other hand, in a *dynamic reweighting algorithm*, $\boldsymbol{q}^{(t)}$ changes with t. Specifically, it upweights samples over which the model has a high risk in order to help the model learn "hard" samples. A popular dynamic reweighting algorithm is *Group DRO* [117]. Denote the empirical risk over group k by $\hat{\mathcal{R}}_k(f)$, and the model at time t by $f^{(t)}$. Group DRO sets $q_i^{(t)} = g_k^{(t)} / n_k$ for

[1] We prove our results for $\mathcal{Y} \subseteq \mathbb{R}$, but our results can be easily extended to the multi-class scenario $\mathcal{Y} \subseteq \mathbb{R}^m$.

[2] This is the non-overlapping setting. There is also the overlapping setting where groups can overlap with each other. We focus on the non-overlapping setting in this paper.

all $\boldsymbol{z}_i \in \mathcal{D}_k$ where $g_k^{(t)}$ is the group weight that is updated by

$$g_k^{(t)} \propto g_k^{(t-1)} \exp\left(\nu \hat{\mathcal{R}}_k(f^{(t-1)})\right) \qquad (\forall k = 1, \cdots, K) \tag{5.3}$$

for some $\nu > 0$, and then normalized so that $q_1^{(t)} + \cdots + q_n^{(t)} = 1$. Sagawa et al. [117] proved a convergence rate theorem (their Proposition 2) showing that in the convex setting, the worst-group training risk of Group DRO converges to the global minimum with the rate $O(t^{-1/2})$.

There are many other reweighting algorithms. Particularly, all variants of DRO and DRO-based methods like CVaR and χ^2-DRO are reweighting algorithms. See Appendix 5.6 for more examples.

5.2.2 Reweighting algorithms can easily overfit

In this section, we will empirically demonstrate that while IW and Group DRO can achieve higher worst-group test performances than ERM at the early stage of training, they can easily overfit after a number of training epochs.

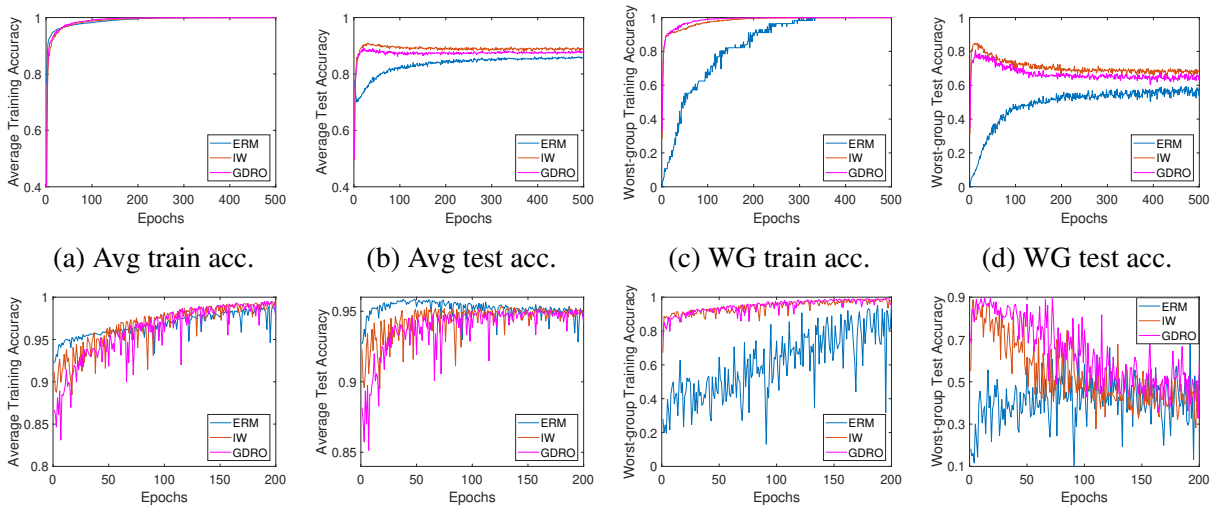
Following Sagawa et al. [117], we conduct the experiment on two datasets: Waterbirds and CelebA. Each dataset contains a binary confounding variable a and a binary target variable y, dividing the dataset into four groups (four combinations of (a, y)). In Waterbirds y is the type of the bird and a is the background; In CelebA y is whether the person has blond hair and a is whether the person is male. On each dataset, a model trained by ERM always exhibits a very strong empirical correlation between y and a, so its performance on one of the groups is extremely poor. The goal is to make the model perform well on every group. See Appendix C.1 of Sagawa et al. [117] for detailed information of these datasets.

On each dataset, we use the ResNet18 model as the classifier and optimize it with momentum SGD. We run each of the three algorithms: ERM, IW and group DRO (GDRO), for 500 epochs on Waterbirds and 200 epochs on CelebA, and plot the average training/test and worst-group (WG) training/test accuracy curves throughout training in Figure 5.1. From the plots we can conclude that:

- All algorithms can achieve and maintain high average training/test accuracy throughout training, i.e. there is almost no overfitting in the average test accuracy.
- Regarding the worst-group test accuracy, while the two reweighting algorithms outperform ERM by a large margin at the early epochs, they overfit very quickly. On CelebA after roughly 100 epochs, the worst-group test accuracies of the two reweighting algorithms become the same as ERM. On Waterbirds, the worst-group test performances of IW and Group DRO drop significantly after around 30 epochs though they are still better than ERM.

5.3 Implicit biases of reweighting algorithms

In the previous section, we empirically demonstrated that the worst-group test performances of reweighting algorithms converge to the same level as ERM. To theoretically understand why this happens in practice, we analyze the implicit biases of reweighting algorithms. Our main theorem



(a) Avg train acc. (b) Avg test acc. (c) WG train acc. (d) WG test acc.

(e) Avg train acc. (f) Avg test acc. (g) WG train acc. (h) WG test acc.

Figure 5.1: Performances of ERM, IW and Group DRO. First row: Waterbirds. Second row: CelebA.

(Theorem 16) states that almost all reweighting algorithms (including ERM) have equivalent implicit biases, in the sense that they converge to the same interpolator. Meanwhile, it is observed in practice that the ERM interpolator has a poor worst-group test performance. This leads to the pessimistic result that *reweighting algorithms always overfit*. All proofs can be found in Appendix 5.7.

5.3.1 Linear models

We first demonstrate this pessimistic result on simple linear models to provide our readers with a key intuition, and later we will apply this same intuition to neural networks. Let the linear model be $f(\boldsymbol{x}) = \langle \theta, \boldsymbol{x} \rangle$, where $\theta \in \mathbb{R}^d$. In the overparameterized setting, we have $d > n$. Consider using the squared loss $\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$, and minimizing the weighted empirical risk with gradient descent:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \sum_{i=1}^{n} q_i^{(t)} \nabla_\theta \ell(f^{(t)}(\boldsymbol{x}_i), y_i) \tag{5.4}$$

where $\eta > 0$ is the learning rate. For a linear model with the squared loss, the update rule is

$$\theta^{(t+1)} = \theta^{(t)} - \eta \sum_{i=1}^{n} q_i^{(t)} \boldsymbol{x}_i (f^{(t)}(\boldsymbol{x}_i) - y_i) \tag{5.5}$$

It is a well known result that under the overparameterization setting where $d > n$, if $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ are linearly independent, then with a sufficiently small η, a linear model trained by ERM can always converge to an *interpolator* which fits all training samples (i.e. $\theta^{(t)} \to \theta^*$ such that $\langle \theta^*, \boldsymbol{x}_i \rangle = y_i$ for all i). Here the linear independence is necessary, because otherwise in the extreme case where $\boldsymbol{x}_1 = \boldsymbol{x}_2$ but $y_1 \neq y_2$, the model cannot fit $(\boldsymbol{x}_1, y_1)$ and $(\boldsymbol{x}_2, y_2)$ simultaneously.

In this section, we aim to extend this ERM convergence analysis to general reweighting algorithms. Our results require the following assumption:

Assumption 1. *There exist constants $q_1, \cdots, q_n$ such that for all i, $q_i^{(t)} \to q_i$ as $t \to \infty$. And* $\min_i q_i = q^* > 0$.

This assumption avoids the scenario where there is some i such that $q_i^{(t)} \approx 0$ for all t, in which case the model could never fit $\boldsymbol{z}_i$. Assumption 1 empirically holds for Group DRO on Waterbirds and CelebA (see Appendix 5.9.2). Under this assumption, we can prove that the model always converges to an interpolator:

Theorem 11. *For any reweighting algorithm satisfying Assumption 1, if $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ are linearly independent, then there exists an $\eta_0 > 0$ such that for any $\eta \le \eta_0$, as $t \to \infty$, $\theta^{(t)}$ converges to some interpolator θ^* such that for all i, $\langle \theta^*, \boldsymbol{x}_i \rangle = y_i$.*

We now make the following key observation regarding the update rule (5.5): $\theta^{(t+1)} - \theta^{(t)}$ is a linear combination of $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ for all t, and thus $\theta^{(t)} - \theta^{(0)}$ always lies in the linear subspace $\text{span}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$. Note that this is an n-dimensional linear subspace if $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ are linearly independent, and by Cramer's rule, there is exactly one $\tilde{\theta}$ in this subspace such that $\langle \tilde{\theta} + \theta^{(0)}, \boldsymbol{x}_i \rangle = y_i$ for all i, which implies that $\theta^* = \tilde{\theta} + \theta^{(0)}$ is unique. Together with Theorem 11, this leads to:

Theorem 12. *If $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ are linearly independent, then there exists $\eta_0 > 0$ such that for any reweighting algorithm satisfying Assumption 1, and any $\eta \le \eta_0$, $\theta^{(t)}$ converges to the same interpolator θ^* that does not depend on $q_i^{(t)}$.*

Note that ERM is also a reweighting algorithm satisfying Assumption 1. Therefore, we have essentially proved the following result: *The implicit bias of any reweighting algorithm satisfying Assumption 1 is equivalent to ERM, so reweighting algorithms always overfit*[3].

The key intuition here is that no matter what reweighting algorithm we use, $\theta^{(t)} - \theta^{(0)}$ always lies in a low-dimensional subspace, in which the interpolator is unique. Therefore, as long as a model trained by the algorithm converges to some interpolator, it must converge to that unique interpolator, which means that the implicit bias of the algorithm is equivalent to ERM.

5.3.2 Linearized neural networks

Now we prove the same result for neural networks. Of course it would be very hard to prove it for all neural networks. However, we can prove the result for a family of overparameterized neural networks that can be approximated by their linearized counterparts [89]. Denote the neural network at time t by $f^{(t)}(\boldsymbol{x}) = f(\boldsymbol{x}; \theta^{(t)})$ which is parameterized by $\theta^{(t)} \in \mathbb{R}^p$ where p is the number of parameters. The *linearized neural network* of $f^{(t)}(\boldsymbol{x})$ is defined as

$$f_{\text{lin}}^{(t)}(\boldsymbol{x}) = f^{(0)}(\boldsymbol{x}) + \langle \theta^{(t)} - \theta^{(0)}, \nabla_\theta f^{(0)}(\boldsymbol{x}) \rangle \tag{5.6}$$

where we use the shorthand $\nabla_\theta f^{(0)}(\boldsymbol{x}) := \nabla_\theta f(\boldsymbol{x}; \theta)|_{\theta=\theta_0}$. Consider training $f_{\text{lin}}^{(t)}(\boldsymbol{x})$ via gradient descent on the reweighted risk (as in (5.4)) using the squared loss. Given a training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, we can construct a new training set $\{\big(\nabla_\theta f^{(0)}(\boldsymbol{x}_i), y_i - f^{(0)}(\boldsymbol{x}_i)\big)\}_{i=1}^n$, so that training a linearized neural network on the original training set is equivalent to training a linear model on the new training set. Based on this observation, we have the following corollary of Theorem 12:

[3] By *overfit*, we are saying that the training error of the model trained by the reweighting algorithm will converge to zero, but the worst-group test performance will converge to the same low level as ERM.

Corollary 13. *If $\nabla_\theta f^{(0)}(\boldsymbol{x}_1), \cdots, \nabla_\theta f^{(0)}(\boldsymbol{x}_n)$ are linearly independent, then there exists $\eta_0 > 0$ such that for any reweighting algorithm satisfying Assumption 1, and any $\eta \leq \eta_0$, $\theta^{(t)}$ converges to the same interpolator θ^* that does not depend on q_i.*

Here we are still using the key intuition: $\theta^{(t)} - \theta^{(0)}$ always lies in the n-dimensional linear subspace span $\left(\nabla_\theta f^{(0)}(\boldsymbol{x}_1), \cdots, \nabla_\theta f^{(0)}(\boldsymbol{x}_n)\right)$. By Cramer's rule, there is a unique interpolator θ^* such that $\theta^* - \theta^{(0)} \in$ span $\left(\nabla_\theta f^{(0)}(\boldsymbol{x}_1), \cdots, \nabla_\theta f^{(0)}(\boldsymbol{x}_n)\right)$, and $\theta^{(t)}$ always converges to that θ^*. Thus, we have essentially proved that for linearized neural networks, reweighting algorithms always overfit.

Now let us delve deeper into the training dynamics of a linearized neural network. Note that $\nabla_\theta f_{\text{lin}}^{(t)}(\boldsymbol{X}) = \nabla_\theta f^{(0)}(\boldsymbol{X}) \in \mathbb{R}^{p \times n}$, so the change in the training function value vector is

$$f_{\text{lin}}^{(t+1)}(\boldsymbol{X}) - f_{\text{lin}}^{(t)}(\boldsymbol{X}) = -\eta \, \nabla_\theta f^{(0)}(\boldsymbol{X})^\top \nabla_\theta f^{(0)}(\boldsymbol{X}) \boldsymbol{Q}^{(t)} \, \nabla_{\hat{y}} \ell(f_{\text{lin}}^{(t)}(\boldsymbol{X}), \boldsymbol{Y}) \tag{5.7}$$

where $\boldsymbol{Q}^{(t)} = \text{diag}(q_1^{(t)}, \cdots, q_n^{(t)})$. The function value vector moves along the kernel gradient with respect to $\Theta^{(0)}_{\boldsymbol{q}^{(t)}} = \nabla_\theta f^{(0)}(\boldsymbol{X})^\top \nabla_\theta f^{(0)}(\boldsymbol{X}) \boldsymbol{Q}^{(t)}$. Meanwhile, the *neural tangent kernel* (NTK, [67]) is $\Theta^{(0)}(\boldsymbol{x}, \boldsymbol{x}') = \nabla_\theta f^{(0)}(\boldsymbol{x})^\top \nabla_\theta f^{(0)}(\boldsymbol{x}')$, and the *Gram matrix* is $\Theta^{(0)} = \Theta^{(0)}(\boldsymbol{X}, \boldsymbol{X})$, so $\Theta^{(0)}_{\boldsymbol{q}^{(t)}} = \Theta^{(0)} \boldsymbol{Q}^{(t)}$. We can thus extend our result for gradient descent on linearized neural networks to a kernel gradient descent algorithm as above.

5.3.3 Wide fully-connected neural networks

Now we prove the result for sufficiently wide fully-connected neural networks, which can be approximated by the linearized neural networks. First we define a fully-connected neural network with L hidden layers (we always assume $L \geq 1$ so there is at least one hidden layer). Let $\boldsymbol{h}^l$ and $\boldsymbol{x}^l$ be the pre- and post-activation outputs of layer l, and d_l be the width of layer l. Let $\boldsymbol{x}^0 = \boldsymbol{x}$ and $d_0 = d$. Define the neural network as

$$\begin{cases} \boldsymbol{h}^{l+1} = \dfrac{W^l}{\sqrt{d_l}} \boldsymbol{x}^l + \beta \boldsymbol{b}^l \\ \boldsymbol{x}^{l+1} = \sigma(\boldsymbol{h}^{l+1}) \end{cases} \qquad (l = 0, \cdots, L) \tag{5.8}$$

where σ is a non-linear activation function, $W^l \in \mathbb{R}^{d_{l+1} \times d_l}$ and $W^L \in \mathbb{R}^{1 \times d_L}$. The parameters θ consist of $W^0, \cdots, W^L$ and $b^0, \cdots, b^L$ (θ is the concatenation of all flattened weights and biases). The final output of the neural network is $f(\boldsymbol{x}) = \boldsymbol{h}^{L+1}$. And let the neural network be initialized as

$$\begin{cases} W_{i,j}^{l(0)} \sim \mathcal{N}(0,1) \\ \boldsymbol{b}_j^{l(0)} \sim \mathcal{N}(0,1) \end{cases} (l = 0, \cdots, L-1) \quad \text{and} \quad \begin{cases} W_{i,j}^{L(0)} = 0 \\ \boldsymbol{b}_j^{L(0)} \sim \mathcal{N}(0,1) \end{cases} \tag{5.9}$$

We also need the following assumption for our approximation theorem:

Assumption 2. *σ is differentiable everywhere, and both σ and $\dot{\sigma}$ are Lipschitz.*[4]

[4] f is *Lipschitz* if there exists a constant $L > 0$ such that for any $\boldsymbol{x}_1, \boldsymbol{x}_2$, $|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)| \leq L \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2$.

Difference from [67]. Our initialization (5.9) is different from the original one in [67] in the last (output) layer. For the output layer, we use the zero initialization $W_{i,j}^{L(0)} = 0$ instead of the Gaussian initialization $W_{i,j}^{L(0)} \sim \mathcal{N}(0,1)$. This modification enables us to accurately approximate the neural network with its linearized counterpart (5.6), as we notice that the proofs in [89] (particularly the proofs of their Theorem 2.1 and their Lemma 1 in Appendix G) are flawed. In Appendix 5.8 we will explain what goes wrong in their proofs and how we manage to fix the proofs with our modification.

For our new initialization, we still have the following NTK theorem:

Theorem 14. *If σ is Lipschitz and $d_l \to \infty$ for $l = 1, \cdots, L$ sequentially, then $\Theta^{(0)}(\boldsymbol{x}, \boldsymbol{x}')$ converges in probability to a non-degenerated[5] deterministic limiting kernel $\Theta(\boldsymbol{x}, \boldsymbol{x}')$.*

The kernel Gram matrix $\Theta = \Theta(\boldsymbol{X}, \boldsymbol{X}) \in \mathbb{R}^{n \times n}$ is a positive semi-definite symmetric matrix. Denote its largest and smallest eigenvalues by $\lambda^{\max}$ and $\lambda^{\min}$. Note that Θ is non-degenerated, so we assume that $\lambda^{\min} > 0$ (which holds almost surely in the overparameterized setting where $d_L \gg n$). Then, we can prove the following approximation theorem:

Theorem 15 (Approximation Theorem)**.** *Let $\eta^* = (\lambda^{\min} + \lambda^{\max})^{-1}$. For a fully-connected neural network $f^{(t)}$ that satisfies Assumption 2 and is trained by any reweighting algorithm satisfying Assumption 1, let $f_{\text{lin}}^{(t)}$ be its linearized neural network which is trained by the same reweighting algorithm (i.e. $\forall i, t$, $q_i^{(t)}$ are the same for both networks). If $d_1 = d_2 = \cdots = d_L = \tilde{d}$ and $\lambda^{\min} > 0$, then for any $\delta > 0$, there exists $\tilde{D} > 0$ and a constant C such that as long as $\eta \le \eta^*$ and $\tilde{d} \ge \tilde{D}$, for any test point $\boldsymbol{x} \in \mathbb{R}^d$ such that $\|\boldsymbol{x}\|_2 \le 1$, with probability at least $1 - \delta$ over random initialization,*

$$\sup_{t \ge 0} \left| f_{\text{lin}}^{(t)}(\boldsymbol{x}) - f^{(t)}(\boldsymbol{x}) \right| \le C \tilde{d}^{-1/4} \tag{5.10}$$

Remark 1. *We can easily extend this theorem to the case where there exists $\alpha_l > 0$ for each of $l = 2, \cdots, L$ such that $d_l / d_1 \to \alpha_l$ and $d_1 \to \infty$.*

Combining all the above results altogether, we achieve our main theorem:

Theorem 16. *Under the conditions of Theorem 15, there exists an $\eta_1 > 0$ such that if $\eta \le \eta_1$ and $\nabla_\theta f^{(0)}(\boldsymbol{x}_1), \cdots, \nabla_\theta f^{(0)}(\boldsymbol{x}_n)$ are linearly independent, then as $\tilde{d} \to \infty$, for any test point $\boldsymbol{x} \in \mathbb{R}^d$ such that $\|\boldsymbol{x}\|_2 \le 1$, with probability close to 1 over random initialization,*

$$\limsup_{t \to \infty} \left| f^{(t)}(\boldsymbol{x}) - f_{\text{ERM}}^{(t)}(\boldsymbol{x}) \right| = O(\tilde{d}^{-1/4}) \to 0 \tag{5.11}$$

where $f^{(t)}$ is trained by the reweighting algorithm and $f_{\text{ERM}}^{(t)}$ is trained by ERM.

The main theorem shows that at any test point $\boldsymbol{x}$, the gap between the function values of the two models converges to an infinitely small term, so the worst-group test performance of the reweighting algorithm will converge to the same level as ERM. Therefore, we have proved that for sufficiently wide fully-connected neural networks, reweighting algorithms always overfit.

Our key intuition tells us that the change in the model parameters always lies in an n-dimensional subspace. Thus, one possible way to improve the worst-group test performance is to enlarge this subspace by adding more training samples, e.g. via data augmentation or semi-supervised learning. However, even if we have more training samples, as long as the model is still

[5] *Non-degenerated* means that $\Theta(\boldsymbol{x}, \boldsymbol{x}')$ depends on $\boldsymbol{x}$ and $\boldsymbol{x}'$ and is not a constant.

overparameterized, and all $\nabla_\theta f^{(0)}(\boldsymbol{x}_i)$ are linearly independent, then our result still says that no reweighting algorithm can do better than ERM in the long run (though the performance of ERM itself might be improved).

Moreover, our theoretical results can explain the surprising empirical observation in [119] that removing some samples from the majority groups to match the group sizes can sometimes achieve even higher worst-group test performance than reweighting even though it wastes lots of data (see their Section 6). When training samples are removed, the model will converge to an interpolator of the smaller training set which is different from the interpolator of the original training set, so there is a chance that the performance of the new interpolator is actually higher.

5.4 Does regularization really help?

In the previous section, we proved the pessimistic result that reweighting algorithms always overfit, i.e. in the long run their worst-group test performances always drop to the same level as ERM. And even if we use strong data augmentation or semi-supervised learning, reweighting algorithms still cannot outperform ERM if the training set is not sufficiently enlarged.

[117] proposed to tackle the overfitting problem of reweighting algorithms via regularization. In particular, they empirically demonstrated with experiments that large regularization is required to prevent reweighting algorithms such as group DRO from overfitting. With a large regularization, the model can maintain a high test worst-group performance, but it cannot obtain perfect training accuracy, in contrast to the case where no regularization is applied.

In this section, we study the necessary conditions for regularization to maintain high worst-group test performance. Specifically, we will show that regularization will not work if it is not large enough to prevent the model from obtaining nearly zero training error. In other words, lowering the training performance is the key to keeping a high worst-group test performance. Note that *the results in this section do not require Assumption 1*, so the results hold for all reweighting algorithms.

5.4.1 Theoretical analysis

Consider a reweighting algorithm with sample weights $q_i^{(t)}$. Following [117], we consider adding L_2 penalty to the weighted empirical risk (6.15):

$$\hat{\mathcal{R}}^{\mu}_{\boldsymbol{q}^{(t)}}(f) = \sum_{i=1}^{n} q_i^{(t)} \ell(f(\boldsymbol{x}_i), y_i) + \frac{\mu}{2} \left\| \theta - \theta^{(0)} \right\|_2^2 \tag{5.12}$$

Given that sufficiently wide neural networks can be approximated by linearized ones, we first focus on linearized neural networks. We will use the subscript "reg" to refer to a regularized model (which is trained trained by minimizing the regularized risk (5.12)). Let $f^{(t)}_{\text{linreg}}$ be a regularized linearized neural network trained by some reweighting algorithm, and $f^{(t)}_{\text{linERM}}$ be an unregularized linearized neural network trained by ERM. As before, we consider training the models with gradient descent under the squared loss $\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$. The following result shows that these two models are very close if $f^{(t)}_{\text{linreg}}$ can achieve low training error:

Theorem 17. *If there is a constant $M_0 > 0$ such that $\left\|\nabla_\theta f^{(0)}(\boldsymbol{x})\right\|_2 \leq M_0$ for all $\|\boldsymbol{x}\|_2 \leq 1$, $\nabla_\theta f^{(0)}(\boldsymbol{x}_1), \cdots, \nabla_\theta f^{(0)}(\boldsymbol{x}_n)$ are linearly independent, and the empirical training risk of $f^{(t)}_{\text{linreg}}$ satisfies*

$$\limsup_{t\to\infty} \hat{\mathcal{R}}(f^{(t)}_{\text{linreg}}) < \epsilon, \tag{5.13}$$

for some $\epsilon > 0$, then for any test point $\boldsymbol{x}$ such that $\|\boldsymbol{x}\|_2 \leq 1$ we have

$$\limsup_{t\to\infty} \left| f^{(t)}_{\text{linreg}}(\boldsymbol{x}) - f^{(t)}_{\text{linERM}}(\boldsymbol{x}) \right| = O(\sqrt{\epsilon}). \tag{5.14}$$

The proof of this theorem also follows the key intuition: we can show that even with the L_2 penalty added, $\theta^{(t)} - \theta^{(0)}$ is still limited in a low-dimensional subspace. And although we cannot prove that $\theta^{(t)}$ always converges to the ERM interpolator, we can prove that it can get very close to that interpolator if its training error is very low, so the resulting model is very close to the ERM model.

Then, we can extend this result to sufficiently wide fully-connected neural networks:

Theorem 18. *If $\lambda^{\min} > 0$ and $\mu > 0$, then let $\eta^* = (\mu + \lambda^{\min} + \lambda^{\max})^{-1}$. For a wide fully-connected neural network $f^{(t)}_{\text{reg}}$ defined by (5.8) and (5.9) and satisfying Assumption 2, and any reweighting algorithm, if $d_1 = d_2 = \cdots = d_L = \tilde{d}$, $\eta \leq \eta^*$, $\nabla_\theta f^{(0)}(\boldsymbol{x}_1), \cdots, \nabla_\theta f^{(0)}(\boldsymbol{x}_n)$ are linearly independent, and the empirical training risk of $f^{(t)}_{\text{reg}}$ satisfies*

$$\limsup_{t\to\infty} \hat{\mathcal{R}}(f^{(t)}_{\text{reg}}) < \epsilon \tag{5.15}$$

for some $\epsilon > 0$, then as $\tilde{d} \to \infty$, with probability close to 1 over random initialization, for any test point $\boldsymbol{x}$ such that $\|\boldsymbol{x}\|_2 \leq 1$ we have

$$\limsup_{t\to\infty} \left| f^{(t)}_{\text{reg}}(\boldsymbol{x}) - f^{(t)}_{\text{ERM}}(\boldsymbol{x}) \right| = O(\tilde{d}^{-1/4} + \sqrt{\epsilon}) \to O(\sqrt{\epsilon}) \tag{5.16}$$

The result shows that a regularized model trained by any reweighting algorithm will get very close to an unregularized ERM model at any test point $\boldsymbol{x}$ if the training error of the former is nearly zero. Thus, regularization only helps when it is large enough to keep the training error of the model away from zero by a margin.

Our results explain the empirical observation of [117] that by using large regularization, the model can maintain a high worst-group test performance, but it cannot achieve perfect training accuracy. If smaller regularization is applied and the model can achieve nearly perfect training accuracy, then its worst-group test performance will still significantly drop.

5.4.2 Empirical study

In this section, we validate our theoretical results above with experiments on Waterbirds and CelebA. We run ERM, IW and group DRO under different levels of weight decay for 500 epochs on Waterbirds and 250 epochs on CelebA. Note that we do not strictly follow our L_2 penalty formulation (5.12), but we study the L_2 weight decay regularization which is most widely used in practice. We repeat each experiment five times with different random seeds and report the

Table 5.1: Mean average training accuracy and worst-group test accuracy (%) of the last 10 training epochs of ERM, IW and Group DRO under different levels of weight decay (WD). Each entry is Average training accuracy / Worst-group test accuracy. Blue entries are mean accuracies of epochs 11-20 with no weight decay. Each experiment is repeated five times with different random seeds.

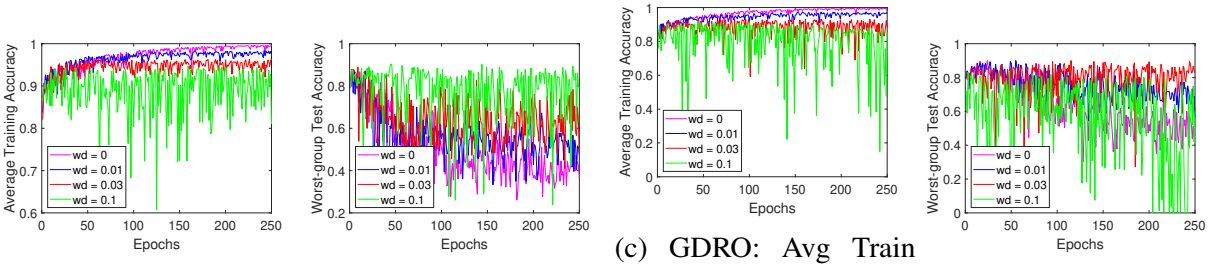
Dataset	WD	ERM	IW	Group DRO
Waterbirds	0	$100.0 \pm 0.0/56.3 \pm 1.8$	$100.0 \pm 0.0/67.6 \pm 1.1$	$100.0 \pm 0.0/64.5 \pm 1.6$
	(11-20)	(Early stopping)	$92.4 \pm 0.4/83.7 \pm 0.6$	$92.9 \pm 0.4/79.9 \pm 2.1$
	0.05		$100.0 \pm 0.0/71.0 \pm 1.9$	$100.0 \pm 0.0/63.5 \pm 2.6$
	0.1		$100.0 \pm 0.0/67.7 \pm 0.7$	$100.0 \pm 0.0/54.7 \pm 2.7$
	0.15		$99.0 \pm 0.7/53.7 \pm 2.7$	$99.4 \pm 0.6/52.5 \pm 2.5$
	0.2		$91.6 \pm 2.0/35.9 \pm 6.9$	$94.8 \pm 0.9/38.0 \pm 7.5$
CelebA	0	$99.0 \pm 0.2/40.2 \pm 5.6$	$99.4 \pm 0.1/42.7 \pm 1.7$	$99.4 \pm 0.1/49.5 \pm 1.9$
	(11-20)	(Early stopping)	$92.1 \pm 0.3/78.2 \pm 3.2$	$90.5 \pm 0.5/85.2 \pm 1.7$
	0.01		$97.9 \pm 0.2/50.0 \pm 2.8$	$96.5 \pm 0.5/67.2 \pm 1.7$
	0.03		$95.0 \pm 0.2/62.8 \pm 2.4$	$88.9 \pm 1.1/83.1 \pm 2.2$
	0.1		$89.4 \pm 2.0/76.0 \pm 2.4$	$75.1 \pm 9.5/50.6 \pm 15.9$

95% confidence interval of the mean average training and worst-group test accuracies of the last 10 training epochs in Table 5.1. To compare with early stopping, we also report the mean accuracies of epochs 11-20 with no regularization in blue. Moreover, we plot the average training and worst-group test accuracy curves throughout training for IW and Group DRO with one of the random seeds in Figure 5.2.

On both datasets, early stopping achieve the best performances. Particularly, on Waterbirds, there is no clear sign that regularization could help prevent overfitting. When the regularization is small, the training accuracy is still 100% and the algorithm continues to overfit. However, when the regularization is large enough to lower the training accuracy, the worst-group test accuracy drops more because the model cannot learn the samples well under such a large regularization. Thus, perhaps not surprisingly, a lower training performance is only a necessary condition but not sufficient.

On CelebA, regularization does help mitigate overfitting, but a useful regularization must be large enough to lower the training accuracy. We observe that Group DRO overfits more slowly than IW, as it still has over 70% worst-group test accuracy after 70 epochs. However, as Figure 5.2d clearly shows, its worst-group test accuracy will still drop to the ERM level at 200 epochs. We also notice that Group DRO requires a smaller regularization than IW: for IW we need the weight decay level to be as large as 0.1 to achieve a similar performance as early stopping, but for Group DRO it only needs to be 0.01, and using 0.1 is actually harmful.

Overall, we find that early stopping achieves a markedly better performance. On the other hand, using large regularization could result in training instability, as well as a loss in overall performance, and there may or may not be a small band for the regularization parameter where the worst-group test performance is better.



(a) IW: Avg Train Acc. (b) IW: WG Test Acc. (c) GDRO: Avg Train Acc. (d) GDRO: WG Test Acc.

Figure 5.2: Average training accuracy and worst-group (WG) test accuracy of IW and Group DRO (GDRO) under different L_2 weight decay levels on CelebA.

5.5 Conclusion

In this work, we theoretically studied why reweighting algorithms overfit in practice by analyzing their implicit biases. Specifically, we proved the pessimistic result that reweighting algorithms always overfit. Our proof was based on the key intuition that the change in model parameters always lies in a low-dimensional subspace, so that even with reweighting, the model still converges to the same unique interpolator. When regularization is applied, we proved that the regularization must be large enough to keep the model from achieving nearly zero training error in order to prevent overfitting. We empirically validated our theoretical results on real datasets, and our results can also explain the empirical observations in previous work. Our results are especially important for large-scale machine learning tasks, where early stopping is not always possible in order to achieve high performances. Practitioners shooting for high worst-group performances in those tasks must be very careful about to what extent overfitting affects reweighting algorithms.

Reproducibility statement

To guarantee the reproducibility of all our empirical results, in all our experiments we use a fixed set of random seeds, and we run some of the experiments twice with the same random seed to make sure that the outputs are the same. See Appendix 5.9.1 for experiment details. After this paper is deanonymized, we will provide a GitHub repository that contains all the codes, datasets, hyperparameters, random seeds, machine speculations and anaconda environment speculations that are sufficient to exactly reproduce our empirical results.

5.6 Other reweighting algorithms

In this section, we will review some other previously proposed reweighting algorithms. First, we will look at DRO-based methods, where DRO stands for Distributionally Robust Optimization.

DRO is designed for tasks with *distributional shift*, where the training distribution and the test distribution are different, and there are some constraints on the distance between these two distributions (typically described by a divergence function D). Since the real test distribution is unknown, DRO minimizes the model's risk over the worst distribution that satisfies the distance constraints, which is an upper bound of the model's real test error. Formally speaking, given a training distribution P, DRO minimizes the expected risk over the worst-case distribution Q in a ball w.r.t. divergence D around the training distribution P. For *group shift* problems which require high worst-group performance, Q also needs to be absolutely continuous with respect to P, i.e. $Q \ll P$. Overall, DRO minimizes the following *expected DRO risk*:

$$\mathcal{R}_{D,\rho}(\theta; P) = \sup_{Q \ll P} \{\mathbb{E}_Q[\ell(\theta; Z)] : D(Q \parallel P) \leq \rho\} \tag{5.17}$$

The expected DRO risk is typically minimized in the following way: for each epoch t, we first find the worst Q that maximizes $\mathbb{E}_Q[\ell(\theta; Z)]$ and satisfies $D(Q \parallel P) \leq \rho, Q \ll P$, and then minimize the model's expected risk over this Q with gradient descent. The rationale behind this algorithm is the famous Danskin's Theorem, which says that if $F(x)$ is the maximum of a family of functions, then its gradient at point x is equal to the gradient of the function that attains the maximum value at x.

Note that in practice we only have a finite set of training samples $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n\}$, so P is always chosen as the empirical distribution, i.e. uniform distribution over $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n$. Then, note that $Q \ll P$, which implies that the support of Q must be a subset of the support of P, which is $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n\}$. This means that Q must be a distribution over $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n$, i.e. it is a reweighting over the training samples. Thus, we have essentially showed that DRO is a reweighting algorithm, and in fact almost all methods based on DRO are reweighting algorithms.

Two widely used variants of DRO are CVaR (Conditional Value at Risk) and χ^2-DRO. In CVaR, for a fixed $\alpha \in (0, 1)$, we let $D(Q \parallel P) = \sup \log \frac{dQ}{dP}$ and $\rho = -\log \alpha$. As a result, suppose that αn is an integer, then CVaR will assign weight $\frac{1}{\alpha n}$ to αn training samples that incur the highest losses, and weight 0 to the rest of the samples, so we can easily see that CVaR is a reweighting algorithm. χ^2-DRO was first used in Hashimoto et al. [58] to deal with fairness tasks where the group labels are unknown, where $D(Q \parallel P) = \frac{1}{2}\int (dQ/dP - 1)^2 dP$ and $\rho = \frac{1}{2}(\frac{1}{\alpha} - 1)^2$. χ^2-DRO is also a reweighting algorithm.

There are many other previously proposed methods of maximizing the worst-group performance that are also reweighting algorithms. For instance, Xu et al. [155] studied the imbalanced class problem where a standard trained model always has high performance over classes with many training samples and low performance over minority classes. They proposed to balance the classes with Label CVaR, which is based on DRO and is a reweighting algorithm.

Liu et al. [93] proposed a two-stage training process called JTT: in the first identification stage they trained a model with ERM to identify training samples that are hard to learn, and in the second upweighting stage they trained a new model with the hard samples upweighted, so that the

model could learn all samples equally well. As the process itself suggests, JTT is a reweighting algorithm.

Finally, Zhai et al. [163] argued that DRO-based methods are very sensitive to outliers in the training set because they upweight training samples with high losses and outliers tend to incur high losses. They proposed the DORO algorithm which at each iteration removes the samples with the highest losses, and then performs DRO on the rest of the samples. DORO is a reweighting algorithm.

5.7 Proofs

Notations. In all of the proofs, for a matrix $\boldsymbol{A}$, we will use $\|\boldsymbol{A}\|_2$ to denote its spectral norm and $\|\boldsymbol{A}\|_F$ to denote its Frobenius norm.

5.7.1 Proof of Theorem 11

To help our readers understand the proof more easily, we will first prove the result for static reweighting algorithms where $q_i^{(t)} = q_i$ for all t, and then we will prove the result for dynamic reweighting algorithms that satisfy $q_i^{(t)} \to q_i$ as $t \to \infty$.

Static reweighting algorithms

We first prove the result for all static reweighting algorithms such that $\min_i q_i = q^* > 0$.

We will use a standard optimization proof technique called *smoothness*. Denote $A = \sum_{i=1}^n \|\boldsymbol{x}_i\|_2^2$. The empirical risk of the linear model $f(\boldsymbol{x}) = \langle \theta, \boldsymbol{x} \rangle$ is

$$F(\theta) = \sum_{i=1}^n q_i (\boldsymbol{x}_i^\top \theta - y_i)^2 \tag{5.18}$$

whose Hessian is

$$\nabla_\theta^2 F(\theta) = 2 \sum_{i=1}^n q_i \boldsymbol{x}_i \boldsymbol{x}_i^\top \tag{5.19}$$

So for any unit vector $\boldsymbol{v} \in \mathbb{R}^d$, we have (since $q_i \in [0,1]$)

$$\boldsymbol{v}^\top \nabla_\theta^2 F(\theta) \boldsymbol{v} = 2 \sum_{i=1}^n q_i (\boldsymbol{x}_i^\top \boldsymbol{v})^2 \le 2 \sum_{i=1}^n q_i \|\boldsymbol{x}_i\|_2^2 \le 2A \tag{5.20}$$

which implies that $F(\theta)$ is $2A$-smooth. Thus, we have the following upper quadratic bound: for any $\theta_1, \theta_2 \in \mathbb{R}^d$,

$$F(\theta_2) \le F(\theta_1) + \langle \nabla_\theta F(\theta_1), \theta_2 - \theta_1 \rangle + A \|\theta_2 - \theta_1\|_2^2 \tag{5.21}$$

Denote $g(\theta^{(t)}) = \sqrt{\boldsymbol{Q}}(\boldsymbol{X}^\top \theta^{(t)} - \boldsymbol{Y}) \in \mathbb{R}^n$ where $\sqrt{\boldsymbol{Q}} = \operatorname{diag}(\sqrt{q_1}, \cdots, \sqrt{q_n})$. We can see that $\left\|g(\theta^{(t)})\right\|_2^2 = F(\theta^{(t)})$, so that $\nabla F(\theta^{(t)}) = 2\boldsymbol{X}\sqrt{\boldsymbol{Q}} g(\theta^{(t)})$. The update rule of a static

reweighting algorithm with gradient descent and the squared loss is:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \sum_{i=1}^{n} q_i \boldsymbol{x}_i (f^{(t)}(\boldsymbol{x}_i) - y_i) = \theta^{(t)} - \eta \boldsymbol{X} \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \tag{5.22}$$

Substituting θ_1 and θ_2 in (5.21) with $\theta^{(t)}$ and $\theta^{(t+1)}$ yields

$$F(\theta^{(t+1)}) \leq F(\theta^{(t)}) - 2\eta g(\theta^{(t)})^\top \sqrt{\boldsymbol{Q}}^\top \boldsymbol{X}^\top \boldsymbol{X} \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) + A \left\| \eta \boldsymbol{X} \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right\|_2^2 \tag{5.23}$$

Since $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ are linearly independent, $\boldsymbol{X}^\top \boldsymbol{X}$ is a positive definite matrix. Denote the smallest eigenvalue of $\boldsymbol{X}^\top \boldsymbol{X}$ by $\lambda^{\min} > 0$. And $\left\| \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right\|_2 \geq \sqrt{q^*} \left\| g(\theta^{(t)}) \right\|_2 = \sqrt{q^* F(\theta^{(t)})}$, so we have $g(\theta^{(t)})^\top \sqrt{\boldsymbol{Q}}^\top \boldsymbol{X}^\top \boldsymbol{X} \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \geq q^* \lambda^{\min} F(\theta^{(t)})$. Thus,

$$\begin{aligned} F(\theta^{(t+1)}) &\leq F(\theta^{(t)}) - 2\eta q^* \lambda^{\min} F(\theta^{(t)}) + A\eta^2 \left\| \boldsymbol{X} \sqrt{\boldsymbol{Q}} \right\|_2^2 \left\| g(\theta^{(t)}) \right\|_2^2 \\ &\leq F(\theta^{(t)}) - 2\eta q^* \lambda^{\min} F(\theta^{(t)}) + A\eta^2 \left\| \boldsymbol{X} \sqrt{\boldsymbol{Q}} \right\|_F^2 F(\theta^{(t)}) \\ &\leq F(\theta^{(t)}) - 2\eta q^* \lambda^{\min} F(\theta^{(t)}) + A\eta^2 \left\| \boldsymbol{X} \right\|_F^2 F(\theta^{(t)}) \\ &= (1 - 2\eta q^* \lambda^{\min} + A^2 \eta^2) F(\theta^{(t)}) \end{aligned} \tag{5.24}$$

Let $\eta_0 = \frac{q^* \lambda^{\min}}{A^2}$. For any $\eta \leq \eta_0$, we have $F(\theta^{(t+1)}) \leq (1 - \eta q^* \lambda^{\min}) F(\theta^{(t)})$ for all t, which implies that $\lim_{t \to \infty} F(\theta^{(t)}) = 0$. Moreover, $\sqrt{F(\theta^{(t+1)})} \leq (1 - \frac{\eta q^* \lambda^{\min}}{2}) \sqrt{F(\theta^{(t)})}$ due to $\sqrt{1 - x} \leq 1 - x/2$.

The convergence in $F(\theta)$ implies the convergence in θ. This is because

$$\begin{aligned} \left\| \theta^{(t+1)} - \theta^{(t)} \right\|_2^2 = \eta^2 \left\| \boldsymbol{X} \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right\|_2^2 &\leq \eta^2 \left\| \boldsymbol{X} \sqrt{\boldsymbol{Q}} \right\|_F^2 \left\| g(\theta^{(t)}) \right\|_2^2 \\ &\leq \eta^2 \left\| \boldsymbol{X} \right\|_F^2 \left\| g(\theta^{(t)}) \right\|_2^2 = A\eta^2 F(\theta^{(t)}) \end{aligned} \tag{5.25}$$

which implies that for any $\eta \leq \eta_0$,

$$\sum_{t=T}^{\infty} \left\| \theta^{(t+1)} - \theta^{(t)} \right\|_2 \leq \sqrt{A\eta^2} \sum_{t=T}^{\infty} \sqrt{F(\theta^{(t)})} \leq \frac{2A}{q^* \lambda^{\min}} \sqrt{F(\theta^{(T)})} \tag{5.26}$$

Therefore, $\lim_{T \to \infty} \sum_{t=T}^{\infty} \left\| \theta^{(t+1)} - \theta^{(t)} \right\|_2 = 0$, which means that $\theta^{(t)}$ converges, and it converges to some interpolator.

Dynamic reweighting algorithms

Now we prove the result for all dynamic reweighting algorithms satisfying Assumption1. By Assumption 1, for any $\epsilon > 0$, there exists t_ϵ such that for all $t \geq t_\epsilon$ and all i,

$$q_i^{(t)} \in (q_i - \epsilon, q_i + \epsilon) \tag{5.27}$$

This is because for all i, there exists t_i such that for all $t \geq t_i$, $q_i^{(t)} \in (q_i - \epsilon, q_i + \epsilon)$. Then, we can define $t_\epsilon = \max\{t_1, \cdots, t_n\}$. Denote the largest and smallest eigenvalues of $\boldsymbol{X}^\top \boldsymbol{X}$ by $\lambda^{\max}$ and $\lambda^{\min}$, and because $\boldsymbol{X}$ is full-rank, we have $\lambda^{\min} > 0$. Select and a fix an ϵ such that $0 < \epsilon < \max\{\frac{q^*}{3}, \frac{(q^*\lambda^{\min})^2}{12\lambda^{\max 2}}\}$, and then t_ϵ is also fixed.

We still denote $\boldsymbol{Q} = \operatorname{diag}(q_1, \cdots, q_n)$. When $t \geq t_\epsilon$, the update rule of a dynamic reweighting algorithm with gradient descent and the squared loss is:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \boldsymbol{X} \boldsymbol{Q}_\epsilon^{(t)} (\boldsymbol{X}^\top \theta^{(t)} - \boldsymbol{Y}) \tag{5.28}$$

where $\boldsymbol{Q}_\epsilon^{(t)} = \boldsymbol{Q}^{(t)}$, and we use the subscript ϵ to indicate that $\left\| \boldsymbol{Q}_\epsilon^{(t)} - \boldsymbol{Q} \right\|_2 < \epsilon$. Then, note that we can rewrite $\boldsymbol{Q}_\epsilon^{(t)}$ as $\boldsymbol{Q}_\epsilon^{(t)} = \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \cdot \sqrt{\boldsymbol{Q}}$ for all $\epsilon < q^*/3$. This is because $q_i + \epsilon < \sqrt{(q_i + 3\epsilon) q_i}$ and $q_i - \epsilon > \sqrt{(q_i - 3\epsilon) q_i}$ for all $\epsilon < q_i/3$, and $q_i \geq q^*$. Thus, we have

$$\theta^{(t+1)} = \theta^{(t)} - \eta \boldsymbol{X} \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} g(\theta^{(t)}) \quad \text{where } \boldsymbol{Q}_\epsilon^{(t)} = \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \cdot \sqrt{\boldsymbol{Q}} \tag{5.29}$$

Again, substituting θ_1 and θ_2 in (5.21) with $\theta^{(t)}$ and $\theta^{(t+1)}$ yields

$$F(\theta^{(t+1)}) \leq F(\theta^{(t)}) - 2\eta g(\theta^{(t)})^\top \sqrt{\boldsymbol{Q}}^\top \boldsymbol{X}^\top \boldsymbol{X} \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} g(\theta^{(t)}) + A \left\| \eta \boldsymbol{X} \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} g(\theta^{(t)}) \right\|_2^2 \tag{5.30}$$

Then, note that

$$\begin{aligned}
&\left| g(\theta^{(t)})^\top \sqrt{\boldsymbol{Q}}^\top \boldsymbol{X}^\top \boldsymbol{X} \left(\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} - \sqrt{\boldsymbol{Q}} \right) g(\theta^{(t)}) \right| \\
\leq &\left\| \sqrt{\boldsymbol{Q}}^\top \boldsymbol{X}^\top \boldsymbol{X} \left(\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} - \sqrt{\boldsymbol{Q}} \right) \right\|_2 \left\| g(\theta^{(t)}) \right\|_2^2 \\
\leq &\left\| \sqrt{\boldsymbol{Q}} \right\|_2 \left\| \boldsymbol{X}^\top \boldsymbol{X} \right\|_2 \left\| \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} - \sqrt{\boldsymbol{Q}} \right\|_2 \left\| g(\theta^{(t)}) \right\|_2^2 \\
\leq &\lambda^{\max} \sqrt{3\epsilon} F(\theta^{(t)})
\end{aligned} \tag{5.31}$$

where the last step comes from the following fact: for all $\epsilon < q_i/3$,

$$\sqrt{q_i + 3\epsilon} - \sqrt{q_i} \leq \sqrt{3\epsilon} \quad \text{and} \quad \sqrt{q_i} - \sqrt{q_i - 3\epsilon} \leq \sqrt{3\epsilon} \tag{5.32}$$

And as proved before, we also have

$$g(\theta^{(t)})^\top \sqrt{\boldsymbol{Q}}^\top \boldsymbol{X}^\top \boldsymbol{X} \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \geq q^* \lambda^{\min} F(\theta^{(t)}) \tag{5.33}$$

Since $\epsilon \leq \frac{(q^*\lambda^{\min})^2}{12\lambda^{\max 2}}$, we have

$$g(\theta^{(t)})^\top \sqrt{\boldsymbol{Q}}^\top \boldsymbol{X}^\top \boldsymbol{X} \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} g(\theta^{(t)}) \geq \left(q^* \lambda^{\min} - \lambda^{\max} \sqrt{3\epsilon} \right) F(\theta^{(t)}) \geq \frac{1}{2} q^* \lambda^{\min} F(\theta^{(t)}) \tag{5.34}$$

Thus,

$$\begin{aligned} F(\theta^{(t+1)}) &\leq F(\theta^{(t)}) - \eta q^* \lambda^{\min} F(\theta^{(t)}) + A\eta^2 \left\| \boldsymbol{X} \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \right\|_2^2 \left\| g(\theta^{(t)}) \right\|_2^2 \\ &\leq (1 - \eta q^* \lambda^{\min} + A^2 \eta^2 (1 + 3\epsilon)) F(\theta^{(t)}) \\ &\leq (1 - \eta q^* \lambda^{\min} + 2A^2 \eta^2) F(\theta^{(t)}) \end{aligned} \tag{5.35}$$

for all $\epsilon < 1/3$. Let $\eta_0 = \frac{q^* \lambda^{\min}}{4A^2}$. For any $\eta \leq \eta_0$, we have $F(\theta^{(t+1)}) \leq (1 - \eta q^* \lambda^{\min}/2) F(\theta^{(t)})$ for all $t \geq t_\epsilon$, which implies that $\lim_{t \to \infty} F(\theta^{(t)}) = 0$. As before, we can prove that the convergence in $F(\theta)$ implies the convergence in θ. Thus, θ converges to some interpolator. □

5.7.2 Proof of Theorem 14

Note that the first l layers (except the output layer) of the original NTK formulation and our new formulation are the same, so we still have the following proposition:

Proposition 19 (Proposition 1 in Jacot et al. [67]). *If σ is Lipschitz and $d_l \to \infty$ for $l = 1, \cdots, L$ sequentially, then for all $l = 1, \cdots, L$, the distribution of a single element of $\boldsymbol{h}^l$ converges in probability to a zero-mean Gaussian process of covariance Σ^l that is defined recursively by:*

$$\begin{aligned} \Sigma^1(\boldsymbol{x}, \boldsymbol{x}') &= \frac{1}{d_0} \boldsymbol{x}^\top \boldsymbol{x}' + \beta^2 \\ \Sigma^l(\boldsymbol{x}, \boldsymbol{x}') &= \mathbb{E}_f[\sigma(f(\boldsymbol{x}))\sigma(f(\boldsymbol{x}'))] + \beta^2 \end{aligned} \tag{5.36}$$

where f is sampled from a zero-mean Gaussian process of covariance $\Sigma^{(l-1)}$.

Now we show that for an infinitely wide neural network with $L \geq 1$ hidden layers, $\Theta^{(0)}$ converges in probability to the following non-degenerated deterministic limiting kernel

$$\Theta = \mathbb{E}_{f \sim \Sigma^L}[\sigma(f(\boldsymbol{x}))\sigma(f(\boldsymbol{x}'))] + \beta^2 \tag{5.37}$$

Consider the output layer $\boldsymbol{h}^{L+1} = \frac{W^L}{\sqrt{\tilde{d}}} \sigma(\boldsymbol{h}^L) + \beta \boldsymbol{b}^L$. We can see that for any parameter θ_i before the output layer,

$$\nabla_{\theta_i} \boldsymbol{h}^{L+1} = \operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^L)) \frac{W^{L\top}}{\sqrt{d_L}} \nabla_{\theta_i} \boldsymbol{h}^L = 0 \tag{5.38}$$

And for W^L and $\boldsymbol{b}^L$, we have

$$\nabla_{W^L} \boldsymbol{h}^{L+1} = \frac{1}{\sqrt{d_L}} \sigma(\boldsymbol{h}^L) \qquad \text{and} \qquad \nabla_{\boldsymbol{b}^L} \boldsymbol{h}^{L+1} = \beta \tag{5.39}$$

Then we can achieve (5.37) by the law of large numbers. □

5.7.3 Proof of Theorem 15

We will use the following short-hand in the proof:

$$\begin{cases} g(\theta^{(t)}) = f^{(t)}(\boldsymbol{X}) - \boldsymbol{Y} \\ J(\theta^{(t)}) = \nabla_\theta f(\boldsymbol{X}; \theta^{(t)}) \in \mathbb{R}^{p \times n} \\ \Theta^{(t)} = J(\theta^{(t)})^\top J(\theta^{(t)}) \end{cases} \tag{5.40}$$

For any $\epsilon > 0$, there exists t_ϵ such that for all $t \geq t_\epsilon$ and all i, $q_i^{(t)} \in (q_i - \epsilon, q_i + \epsilon)$. Like what we have done in (5.29), we can rewrite $\boldsymbol{Q}^{(t)} = \boldsymbol{Q}_\epsilon^{(t)} = \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \cdot \sqrt{\boldsymbol{Q}}$, where $\boldsymbol{Q} = \mathrm{diag}(q_1, \cdots, q_n)$.

The update rule of a reweighting algorithm with gradient descent and the squared loss for the wide neural network is:

$$\theta^{(t+1)} = \theta^{(t)} - \eta J(\theta^{(t)}) \boldsymbol{Q}^{(t)} g(\theta^{(t)}) \tag{5.41}$$

and for $t \geq t_\epsilon$, it can be rewritten as

$$\theta^{(t+1)} = \theta^{(t)} - \eta J(\theta^{(t)}) \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \left[\sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right] \tag{5.42}$$

First, we will prove the following theorem:

Theorem 20. *There exist constants $M > 0$ and $\epsilon_0 > 0$ such that for all $\epsilon \in (0, \epsilon_0]$, $\eta \leq \eta^*$ and any $\delta > 0$, there exist $R_0 > 0$, $\tilde{D} > 0$ and $B > 1$ such that for any $\tilde{d} \geq \tilde{D}$, the following (i) and (ii) hold with probability at least $(1 - \delta)$ over random initialization when applying gradient descent with learning rate η:*

1. *For all $t \leq t_\epsilon$, there is*

$$\left\| g(\theta^{(t)}) \right\|_2 \leq B^t R_0 \tag{5.43}$$

$$\sum_{j=1}^{t} \left\| \theta^{(j)} - \theta^{(j-1)} \right\|_2 \leq \eta M R_0 \sum_{j=1}^{t} B^{j-1} < \frac{M B^{t_\epsilon} R_0}{B - 1} \tag{5.44}$$

2. *For all $t \geq t_\epsilon$, we have*

$$\left\| \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right\|_2 \leq \left(1 - \frac{\eta q^* \lambda^{\min}}{3} \right)^{t - t_\epsilon} B^{t_\epsilon} R_0 \tag{5.45}$$

$$\begin{aligned} \sum_{j=t_\epsilon+1}^{t} \left\| \theta^{(j)} - \theta^{(j-1)} \right\|_2 &\leq \eta \sqrt{1 + 3\epsilon} M B^{t_\epsilon} R_0 \sum_{j=t_\epsilon+1}^{t} \left(1 - \frac{\eta q^* \lambda^{\min}}{3} \right)^{j - t_\epsilon} \\ &< \frac{3\sqrt{1 + 3\epsilon} M B^{t_\epsilon} R_0}{q^* \lambda^{\min}} \end{aligned} \tag{5.46}$$

Proof. The proof is based on the following lemma:

Lemma 21 (Local Lipschitzness of the Jacobian)**.** *Under Assumption 2, there is a constant $M > 0$ such that for any $C_0 > 0$ and any $\delta > 0$, there exists a $\tilde{D}$ such that: If $\tilde{d} \geq \tilde{D}$, then with*

probability at least $(1-\delta)$ *over random initialization, for any* $\boldsymbol{x}$ *such that* $\|\boldsymbol{x}\|_2 \le 1$,

$$\begin{cases} \left\|\nabla_\theta f(\boldsymbol{x};\theta) - \nabla_\theta f(\boldsymbol{x};\tilde{\theta})\right\|_2 \le \dfrac{M}{\sqrt[4]{\tilde{d}}}\left\|\theta-\tilde{\theta}\right\|_2 \\ \|\nabla_\theta f(\boldsymbol{x};\theta)\|_2 \le M \\ \left\|J(\theta) - J(\tilde{\theta})\right\|_F \le \dfrac{M}{\sqrt[4]{\tilde{d}}}\left\|\theta-\tilde{\theta}\right\|_2 \\ \|J(\theta)\|_F \le M \end{cases}, \qquad \forall \theta, \tilde{\theta} \in B(\theta^{(0)}, C_0) \tag{5.47}$$

where $B(\theta^{(0)}, R) = \{\theta : \left\|\theta - \theta^{(0)}\right\|_2 < R\}$.

The proof can be found in Appendix 5.7.4. Note that for any $\boldsymbol{x}$, $f^{(0)}(\boldsymbol{x}) = \beta \boldsymbol{b}^L$ where $\boldsymbol{b}^L$ is sampled from the standard Gaussian distribution. Thus, for any $\delta > 0$, there exists a constant R_0 such that with probability at least $(1-\delta/3)$ over random initialization,

$$\left\|g(\theta^{(0)})\right\|_2 < R_0 \tag{5.48}$$

And by Theorem 14, there exists $D_2 \ge 0$ such that for any $\tilde{d} \ge D_2$, with probability at least $(1-\delta/3)$,

$$\left\|\Theta - \Theta^{(0)}\right\|_F \le \frac{q^* \lambda^{\min}}{3} \tag{5.49}$$

Let M be the constant in Lemma 21. Let $\epsilon_0 = \frac{(q^*\lambda^{\min})^2}{108M^4}$. Let $B = 1 + \eta^* M^2$, and $C_0 = \frac{MB^{t_\epsilon} R_0}{B-1} + \frac{3\sqrt{1+3\epsilon} M B^{t_\epsilon} R_0}{q^* \lambda^{\min}}$. By Lemma 21, there exists $D_1 > 0$ such that with probability at least $(1-\delta/3)$, for any $\tilde{d} \ge D_1$, (5.47) is true for all $\theta, \tilde{\theta} \in B(\theta^{(0)}, C_0)$.

By union bound, with probability at least $(1-\delta)$, (5.47), (5.48) and (5.49) are all true. Now we assume that all of them are true, and prove (5.43) and (5.44) by induction. (5.43) is true for $t=0$ due to (5.48), and (5.44) is always true for $t=0$. Suppose (5.43) and (5.44) are true for t, then for $t+1$ we have

$$\begin{aligned} \left\|\theta^{(t+1)} - \theta^{(t)}\right\|_2 &\le \eta \left\|J(\theta^{(t)})\boldsymbol{Q}^{(t)}\right\|_2 \left\|g(\theta^{(t)})\right\|_2 \le \eta \left\|J(\theta^{(t)})\boldsymbol{Q}^{(t)}\right\|_F \left\|g(\theta^{(t)})\right\|_2 \\ &\le \eta \left\|J(\theta^{(t)})\right\|_F \left\|g(\theta^{(t)})\right\|_2 \le M\eta B^t R_0 \end{aligned} \tag{5.50}$$

So (5.44) is also true for $t+1$. And we also have

$$\begin{aligned} \left\|g(\theta^{(t+1)})\right\|_2 &= \left\|g(\theta^{(t+1)}) - g(\theta^{(t)}) + g(\theta^{(t)})\right\|_2 \\ &= \left\|J(\tilde{\theta}^{(t)})^\top(\theta^{(t+1)} - \theta^{(t)}) + g(\theta^{(t)})\right\|_2 \\ &= \left\|-\eta J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\boldsymbol{Q}^{(t)} g(\theta^{(t)}) + g(\theta^{(t)})\right\|_2 \\ &\le \left\|\boldsymbol{I} - \eta J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\boldsymbol{Q}^{(t)}\right\|_2 \left\|g(\theta^{(t)})\right\|_2 \\ &\le \left(1 + \left\|\eta J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\boldsymbol{Q}^{(t)}\right\|_2\right)\left\|g(\theta^{(t)})\right\|_2 \\ &\le \left(1 + \eta \left\|J(\tilde{\theta}^{(t)})\right\|_F \left\|J(\theta^{(t)})\right\|_F\right)\left\|g(\theta^{(t)})\right\|_2 \\ &\le (1+\eta^* M^2)\left\|g(\theta^{(t)})\right\|_2 \le B^{t+1} R_0 \end{aligned} \tag{5.51}$$

Therefore, (5.43) and (5.44) are true for all $t \le t_\epsilon$, which implies that $\left\|\sqrt{\boldsymbol{Q}} g(\theta^{(t_\epsilon)})\right\|_2 \le \left\|g(\theta^{(t_\epsilon)})\right\|_2 \le B^{t_\epsilon} R_0$, so (5.45) is true for $t = t_\epsilon$. And (5.46) is obviously true for $t = t_\epsilon$. Now, let us prove (ii) by induction. Note that when $t \ge t_\epsilon$, we have the alternative update rule (5.42). If (5.45) and (5.46) are true for t, then for $t+1$, there is

$$\begin{aligned}\left\|\theta^{(t+1)} - \theta^{(t)}\right\|_2 &\le \eta \left\|J(\theta^{(t)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2 \left\|\sqrt{\boldsymbol{Q}} g(\theta^{(t)})\right\|_2 \le \eta \left\|J(\theta^{(t)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_F \left\|\sqrt{\boldsymbol{Q}} g(\theta^{(t)})\right\|_2 \\ &\le \eta\sqrt{1+3\epsilon}\left\|J(\theta^{(t)})\right\|_F \left\|\sqrt{\boldsymbol{Q}} g(\theta^{(t)})\right\|_2 \le M\eta\sqrt{1+3\epsilon}\left(1 - \frac{\eta q^* \lambda^{\min}}{3}\right)^{t-t_\epsilon} B^{t_\epsilon} R_0\end{aligned} \tag{5.52}$$

So (5.46) is true for $t+1$. And we also have

$$\begin{aligned}\left\|\sqrt{\boldsymbol{Q}} g(\theta^{(t+1)})\right\|_2 &= \left\|\sqrt{\boldsymbol{Q}} g(\theta^{(t+1)}) - \sqrt{\boldsymbol{Q}} g(\theta^{(t)}) + \sqrt{\boldsymbol{Q}} g(\theta^{(t)})\right\|_2 \\ &= \left\|\sqrt{\boldsymbol{Q}} J(\tilde{\theta}^{(t)})^\top (\theta^{(t+1)} - \theta^{(t)}) + \sqrt{\boldsymbol{Q}} g(\theta^{(t)})\right\|_2 \\ &= \left\|-\eta\sqrt{\boldsymbol{Q}} J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) \boldsymbol{Q}^{(t)} g(\theta^{(t)}) + \sqrt{\boldsymbol{Q}} g(\theta^{(t)})\right\|_2 \\ &\le \left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}} J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2 \left\|\sqrt{\boldsymbol{Q}} g(\theta^{(t)})\right\|_2 \\ &\le \left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}} J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2 \left(1 - \frac{\eta q^* \lambda^{\min}}{3}\right)^t R_0\end{aligned} \tag{5.53}$$

where $\tilde{\theta}^{(t)}$ is some linear interpolation between $\theta^{(t)}$ and $\theta^{(t+1)}$. Now we prove that

$$\left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}} J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2 \le 1 - \frac{\eta q^* \lambda^{\min}}{3} \tag{5.54}$$

For any unit vector $\boldsymbol{v} \in \mathbb{R}^n$, we have

$$\boldsymbol{v}^\top (\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta\sqrt{\boldsymbol{Q}})\boldsymbol{v} = 1 - \eta \boldsymbol{v}^\top \sqrt{\boldsymbol{Q}}\Theta\sqrt{\boldsymbol{Q}}\boldsymbol{v} \tag{5.55}$$

$\left\|\sqrt{\boldsymbol{Q}}\boldsymbol{v}\right\|_2 \in [\sqrt{q^*}, 1]$, so for any $\eta \le \eta^*$, $\boldsymbol{v}^\top(\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta\sqrt{\boldsymbol{Q}})\boldsymbol{v} \in [0, 1 - \eta\lambda^{\min} q^*]$, which implies that $\left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta\sqrt{\boldsymbol{Q}}\right\|_2 \le 1 - \eta\lambda^{\min} q^*$. Thus,

$$\begin{aligned}&\left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}} J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\sqrt{\boldsymbol{Q}}\right\|_2 \\ \le& \left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta\sqrt{\boldsymbol{Q}}\right\|_2 + \eta\left\|\sqrt{\boldsymbol{Q}}(\Theta - \Theta^{(0)})\sqrt{\boldsymbol{Q}}\right\|_2 + \eta\left\|\sqrt{\boldsymbol{Q}}(J(\theta^{(0)})^\top J(\theta^{(0)}) - J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}))\sqrt{\boldsymbol{Q}}\right\|_2 \\ \le& 1 - \eta\lambda^{\min} q^* + \eta\left\|\sqrt{\boldsymbol{Q}}(\Theta - \Theta^{(0)})\sqrt{\boldsymbol{Q}}\right\|_F + \eta\left\|\sqrt{\boldsymbol{Q}}(J(\theta^{(0)})^\top J(\theta^{(0)}) - J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}))\sqrt{\boldsymbol{Q}}\right\|_F \\ \le& 1 - \eta\lambda^{\min} q^* + \eta\left\|\Theta - \Theta^{(0)}\right\|_F + \eta\left\|J(\theta^{(0)})^\top J(\theta^{(0)}) - J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\right\|_F \\ \le& 1 - \eta\lambda^{\min} q^* + \frac{\eta q^* \lambda^{\min}}{3} + \frac{\eta M^2}{\sqrt[4]{\tilde{d}}}\left(\left\|\theta^{(t)} - \theta^{(0)}\right\|_2 + \left\|\tilde{\theta}^{(t)} - \theta^{(0)}\right\|_2\right) \le 1 - \frac{\eta q^* \lambda^{\min}}{2}\end{aligned} \tag{5.56}$$

for all $\tilde{d} \geq \max\left\{D_1, D_2, \left(\frac{12M^2C_0}{q^*\lambda^{\min}}\right)^4\right\}$, which implies that

$$\begin{aligned}
&\left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2 \\
\leq& 1 - \frac{\eta q^*\lambda^{\min}}{2} + \left\|\eta\sqrt{\boldsymbol{Q}}J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\left(\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} - \sqrt{\boldsymbol{Q}}\right)\right\|_2 \\
\leq& 1 - \frac{\eta q^*\lambda^{\min}}{2} + \eta M^2\sqrt{3\epsilon} \leq 1 - \frac{\eta q^*\lambda^{\min}}{3} \qquad \text{(due to (5.32))}
\end{aligned} \tag{5.57}$$

for all $\epsilon \leq \epsilon_0$. Thus, (5.45) is also true for $t+1$. In conclusion, (5.45) and (5.46) are true with probability at least $(1-\delta)$ for all $\tilde{d} \geq \tilde{D} = \max\left\{D_1, D_2, \left(\frac{12M^2C_0}{q^*\lambda^{\min}}\right)^4\right\}$. □

Returning back to the proof of Theorem 15. Choose and fix an ϵ such that $\epsilon < \min\{\epsilon_0, \frac{1}{3}\left(\frac{q^*\lambda^{\min}}{3\lambda^{\max}+q^*\lambda^{\min}}\right)^2\}$, where ϵ_0 is defined by Theorem 20. Then, t_ϵ is also fixed. There exists $\tilde{D} \geq 0$ such that for any $\tilde{d} \geq \tilde{D}$, with probability at least $(1-\delta)$, Theorem 20 and Lemma 21 are true and

$$\left\|\Theta - \Theta^{(0)}\right\|_F \leq \frac{q^*\lambda^{\min}}{3} \tag{5.58}$$

which immediately implies that

$$\left\|\Theta^{(0)}\right\|_2 \leq \|\Theta\|_2 + \left\|\Theta - \Theta^{(0)}\right\|_F \leq \lambda^{\max} + \frac{q^*\lambda^{\min}}{3} \tag{5.59}$$

We still denote $B = 1 + \eta^* M^2$ and $C_0 = \frac{MB^{t_\epsilon}R_0}{B-1} + \frac{3\sqrt{1+3\epsilon}MB^{t_\epsilon}R_0}{q^*\lambda^{\min}}$. Theorem 20 ensures that for all t, $\theta^{(t)} \in B(\theta^{(0)}, C_0)$. Then we have

$$\begin{aligned}
\left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta^{(0)}\sqrt{\boldsymbol{Q}}\right\|_2 &\leq \left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta\sqrt{\boldsymbol{Q}}\right\|_2 + \eta\left\|\sqrt{\boldsymbol{Q}}(\Theta - \Theta^{(0)})\sqrt{\boldsymbol{Q}}\right\|_2 \\
&\leq 1 - \eta\lambda^{\min}q^* + \frac{\eta q^*\lambda^{\min}}{3} = 1 - \frac{2\eta q^*\lambda^{\min}}{3}
\end{aligned} \tag{5.60}$$

so it follows that

$$\begin{aligned}
\left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta^{(0)}\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2 &\leq \left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta^{(0)}\sqrt{\boldsymbol{Q}}\right\|_2 + \left\|\eta\sqrt{\boldsymbol{Q}}\Theta^{(0)}\left(\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} - \sqrt{\boldsymbol{Q}}\right)\right\|_2 \\
&\leq 1 - \frac{2\eta q^*\lambda^{\min}}{3} + \eta(\lambda^{\max} + \frac{q^*\lambda^{\min}}{3})\sqrt{3\epsilon}
\end{aligned} \tag{5.61}$$

Thus, for all $\epsilon < \frac{1}{3}\left(\frac{q^*\lambda^{\min}}{3\lambda^{\max}+q^*\lambda^{\min}}\right)^2$, there is

$$\left\|\boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta^{(0)}\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2 \leq 1 - \frac{\eta q^*\lambda^{\min}}{3} \tag{5.62}$$

The update rule of the reweighting algorithm for the linearized neural network is:

$$\theta_{\text{lin}}^{(t+1)} = \theta_{\text{lin}}^{(t)} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}g_{\text{lin}}(\theta^{(t)}) \tag{5.63}$$

where we use the subscript "lin" to denote the linearized neural network, and with a slight abuse of notion denote $g_{\text{lin}}(\theta^{(t)}) = g(\theta_{\text{lin}}^{(t)})$.

First, let us consider the training data $\boldsymbol{X}$. Denote $\Delta_t = g_{\text{lin}}(\theta^{(t)}) - g(\theta^{(t)})$. We have

$$\begin{cases} g_{\text{lin}}(\theta^{(t+1)}) - g_{\text{lin}}(\theta^{(t)}) = -\eta J(\theta^{(0)})^\top J(\theta^{(0)}) \boldsymbol{Q}^{(t)} g_{\text{lin}}(\theta^{(t)}) \\ g(\theta^{(t+1)}) - g(\theta^{(t)}) = -\eta J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) \boldsymbol{Q}^{(t)} g(\theta^{(t)}) \end{cases} \tag{5.64}$$

where $\tilde{\theta}^{(t)}$ is some linear interpolation between $\theta^{(t)}$ and $\theta^{(t+1)}$. Thus,

$$\begin{aligned} \Delta_{t+1} - \Delta_t =& \eta \left[J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right] \boldsymbol{Q}^{(t)} g(\theta^{(t)}) \\ & - \eta J(\theta^{(0)})^\top J(\theta^{(0)}) \boldsymbol{Q}^{(t)} \Delta_t \end{aligned} \tag{5.65}$$

By Lemma 21, we have

$$\begin{aligned} & \left\| J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right\|_F \\ \leq & \left\| \left(J(\tilde{\theta}^{(t)}) - J(\theta^{(0)}) \right)^\top J(\theta^{(t)}) \right\|_F + \left\| J(\theta^{(0)})^\top \left(J(\theta^{(t)}) - J(\theta^{(0)}) \right) \right\|_F \\ \leq & 2M^2 C_0 \tilde{d}^{-1/4} \end{aligned} \tag{5.66}$$

which implies that for all $t < t_\epsilon$,

$$\begin{aligned} \|\Delta_{t+1}\|_2 &\leq \left\| \left[\boldsymbol{I} - \eta J(\theta^{(0)})^\top J(\theta^{(0)}) \boldsymbol{Q}^{(t)} \right] \Delta_t \right\|_2 + \left\| \eta \left[J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right] \boldsymbol{Q}^{(t)} g(\theta^{(t)}) \right\|_2 \\ &\leq \left\| \boldsymbol{I} - \eta J(\theta^{(0)})^\top J(\theta^{(0)}) \boldsymbol{Q}^{(t)} \right\|_F \|\Delta_t\|_2 + \eta \left\| J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right\|_F \left\| g(\theta^{(t)}) \right\|_2 \\ &\leq (1 + \eta M^2) \|\Delta_t\|_2 + 2\eta M^2 C_0 B^t R_0 \tilde{d}^{-1/4} \\ &\leq B \|\Delta_t\|_2 + 2\eta M^2 C_0 B^t R_0 \tilde{d}^{-1/4} \end{aligned} \tag{5.67}$$

Therefore, we have

$$B^{-(t+1)} \|\Delta_{t+1}\|_2 \leq B^{-t} \|\Delta_t\|_2 + 2\eta M^2 C_0 B^{-1} R_0 \tilde{d}^{-1/4} \tag{5.68}$$

Since $\Delta_0 = 0$, it follows that for all $t \leq t_\epsilon$,

$$\|\Delta_t\|_2 \leq 2t\eta M^2 C_0 B^{t-1} R_0 \tilde{d}^{-1/4} \tag{5.69}$$

and particularly we have

$$\left\| \sqrt{\boldsymbol{Q}} \Delta_{t_\epsilon} \right\|_2 \leq \|\Delta_{t_\epsilon}\|_2 \leq 2t_\epsilon \eta M^2 C_0 B^{t_\epsilon - 1} R_0 \tilde{d}^{-1/4} \tag{5.70}$$

For $t \geq t_\epsilon$, we have the alternative update rule (5.42). Thus,

$$\begin{aligned} \sqrt{\boldsymbol{Q}} \Delta_{t+1} - \sqrt{\boldsymbol{Q}} \Delta_t =& \eta \sqrt{\boldsymbol{Q}} \left[J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)}) \right] \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \left[\sqrt{\boldsymbol{Q}} g(\theta^{(t)}) \right] \\ & - \eta \sqrt{\boldsymbol{Q}} J(\theta^{(0)})^\top J(\theta^{(0)}) \sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} \left[\sqrt{\boldsymbol{Q}} \Delta_t \right] \end{aligned} \tag{5.71}$$

Let $\boldsymbol{A} = \boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}J(\theta^{(0)})^\top J(\theta^{(0)})\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}} = \boldsymbol{I} - \eta\sqrt{\boldsymbol{Q}}\Theta^{(0)}\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}$. Then, we have

$$\sqrt{\boldsymbol{Q}}\Delta_{t+1} = \boldsymbol{A}\sqrt{\boldsymbol{Q}}\Delta_t + \eta\sqrt{\boldsymbol{Q}}\left[J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)})\right]\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\left(\sqrt{\boldsymbol{Q}}g(\theta^{(t)})\right) \tag{5.72}$$

Let $\gamma = 1 - \frac{\eta q^* \lambda^{\min}}{3} < 1$. Combining with Theorem 20 and (5.62), the above leads to

$$\begin{aligned}
\left\|\sqrt{\boldsymbol{Q}}\Delta_{t+1}\right\|_2 &\le \|\boldsymbol{A}\|_2 \left\|\sqrt{\boldsymbol{Q}}\Delta_t\right\|_2 + \eta\left\|\sqrt{\boldsymbol{Q}}\left[J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)})\right]\sqrt{\boldsymbol{Q}_{3\epsilon}^{(t)}}\right\|_2 \left\|\sqrt{\boldsymbol{Q}}g(\theta^{(t)})\right\|_2 \\
&\le \gamma\left\|\sqrt{\boldsymbol{Q}}\Delta_t\right\|_2 + \eta\left\|J(\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - J(\theta^{(0)})^\top J(\theta^{(0)})\right\|_F \sqrt{1+3\epsilon}\gamma^{t-t_\epsilon}B^{t_\epsilon}R_0 \\
&\le \gamma\left\|\sqrt{\boldsymbol{Q}}\Delta_t\right\|_2 + 2\eta M^2 C_0\sqrt{1+3\epsilon}\gamma^{t-t_\epsilon}B^{t_\epsilon}R_0\tilde{d}^{-1/4}
\end{aligned} \tag{5.73}$$

This implies that

$$\gamma^{-(t+1)}\left\|\sqrt{\boldsymbol{Q}}\Delta_{t+1}\right\|_2 \le \gamma^{-t}\left\|\sqrt{\boldsymbol{Q}}\Delta_t\right\|_2 + 2\eta M^2 C_0\sqrt{1+3\epsilon}\gamma^{-1-t_\epsilon}B^{t_\epsilon}R_0\tilde{d}^{-1/4} \tag{5.74}$$

Combining with (5.70), it implies that for all $t \ge t_\epsilon$,

$$\left\|\sqrt{\boldsymbol{Q}}\Delta_t\right\|_2 \le 2\gamma^{t-t_\epsilon}\eta M^2 C_0 B^{t_\epsilon}R_0\left[t_\epsilon B^{-1} + \sqrt{1+3\epsilon}\gamma^{-1}(t-t_\epsilon)\right]\tilde{d}^{-1/4} \tag{5.75}$$

Next, we consider an arbitrary test point $\boldsymbol{x}$ such that $\|\boldsymbol{x}\|_2 \le 1$. Denote $\delta_t = f_{\text{lin}}^{(t)}(\boldsymbol{x}) - f^{(t)}(\boldsymbol{x})$. Then we have

$$\begin{cases}
f_{\text{lin}}^{(t+1)}(\boldsymbol{x}) - f_{\text{lin}}^{(t)}(\boldsymbol{x}) = -\eta\nabla_\theta f(\boldsymbol{x};\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}g_{\text{lin}}(\theta^{(t)}) \\
f^{(t+1)}(\boldsymbol{x}) - f^{(t)}(\boldsymbol{x}) = -\eta\nabla_\theta f(\boldsymbol{x};\tilde{\theta}^{(t)})^\top J(\theta^{(t)})\boldsymbol{Q}^{(t)}g(\theta^{(t)})
\end{cases} \tag{5.76}$$

which yields

$$\begin{aligned}
\delta_{t+1} - \delta_t =& \eta\left[\nabla_\theta f(\boldsymbol{x};\tilde{\theta}^{(t)})^\top J(\theta^{(t)}) - \nabla_\theta f(\boldsymbol{x};\theta^{(0)})^\top J(\theta^{(0)})\right]\boldsymbol{Q}^{(t)}g(\theta^{(t)}) \\
&- \eta\nabla_\theta f(\boldsymbol{x};\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}\Delta_t
\end{aligned} \tag{5.77}$$

For $t \le t_\epsilon$, we have

$$\begin{aligned}
\|\delta_t\|_2 \le& \eta\sum_{s=0}^{t-1}\left\|\left[\nabla_\theta f(\boldsymbol{x};\tilde{\theta}^{(s)})^\top J(\theta^{(s)}) - \nabla_\theta f(\boldsymbol{x};\theta^{(0)})^\top J(\theta^{(0)})\right]\boldsymbol{Q}^{(s)}\right\|_2 \left\|g(\theta^{(s)})\right\|_2 \\
&+ \eta\sum_{s=0}^{t-1}\left\|\nabla_\theta f(\boldsymbol{x};\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(s)}\right\|_2 \|\Delta_s\|_2 \\
\le& \eta\sum_{s=0}^{t-1}\left\|\nabla_\theta f(\boldsymbol{x};\tilde{\theta}^{(s)})^\top J(\theta^{(s)}) - \nabla_\theta f(\boldsymbol{x};\theta^{(0)})^\top J(\theta^{(0)})\right\|_F \left\|g(\theta^{(s)})\right\|_2 \\
&+ \eta\sum_{s=0}^{t-1}\left\|\nabla_\theta f(\boldsymbol{x};\theta^{(0)})\right\|_2 \left\|J(\theta^{(0)})\right\|_F \|\Delta_s\|_2 \\
\le& 2\eta M^2 C_0\tilde{d}^{-1/4}\sum_{s=0}^{t-1}B^s R_0 + \eta M^2\sum_{s=0}^{t-1}(2s\eta M^2 C_0 B^{s-1}R_0\tilde{d}^{-1/4})
\end{aligned} \tag{5.78}$$

So we can see that there exists a constant C_1 such that $\|\delta_{t_\epsilon}\|_2 \leq C_1 \tilde{d}^{-1/4}$. Then, for $t > t_\epsilon$, we have

$$\begin{aligned}
\|\delta_t\|_2 - \|\delta_{t_\epsilon}\|_2 \leq & \eta \sum_{s=t_\epsilon}^{t-1} \left\| \left[\nabla_\theta f(\boldsymbol{x}; \tilde{\theta}^{(s)})^\top J(\theta^{(s)}) - \nabla_\theta f(\boldsymbol{x}; \theta^{(0)})^\top J(\theta^{(0)}) \right] \sqrt{\boldsymbol{Q}_{3\epsilon}^{(s)}} \right\|_2 \left\| \sqrt{\boldsymbol{Q}} g(\theta^{(s)}) \right\|_2 \\
& + \eta \sum_{s=t_\epsilon}^{t-1} \left\| \nabla_\theta f(\boldsymbol{x}; \theta^{(0)})^\top J(\theta^{(0)}) \sqrt{\boldsymbol{Q}_{3\epsilon}^{(s)}} \right\|_2 \left\| \sqrt{\boldsymbol{Q}} \Delta_s \right\|_2 \\
\leq & 2\eta M^2 C_0 \tilde{d}^{-1/4} \sqrt{1+3\epsilon} \sum_{s=t_\epsilon}^{t-1} \gamma^{s-t_\epsilon} B^{t_\epsilon} R_0 \\
& + \eta M^2 \sqrt{1+3\epsilon} \sum_{s=t_\epsilon}^{t-1} \left(2\gamma^{s-t_\epsilon} \eta M^2 C_0 B^{t_\epsilon} R_0 \left[t_\epsilon B^{-1} + \sqrt{1+3\epsilon} \gamma^{-1} (s - t_\epsilon) \right] \tilde{d}^{-1/4} \right)
\end{aligned} \tag{5.79}$$

Note that $\sum_{t=0}^{\infty} t\gamma^t$ is finite as long as $\gamma \in (0,1)$. Therefore, there is a constant C such that for any t, $\|\delta_t\|_2 \leq C\tilde{d}^{-1/4}$ with probability at least $(1-\delta)$ for any $\tilde{d} \geq \tilde{D}$. □

5.7.4 Proof of Lemma 21

We will use the following theorem regarding the eigenvalues of random Gaussian matrices:

Theorem 22 (Corollary 5.35 in Vershynin [139]). *If $\boldsymbol{A} \in \mathbb{R}^{p \times q}$ is a random matrix whose entries are independent standard normal random variables, then for every $t \geq 0$, with probability at least $1 - 2\exp(-t^2/2)$,*

$$\sqrt{p} - \sqrt{q} - t \leq \lambda^{\min}(\boldsymbol{A}) \leq \lambda^{\max}(\boldsymbol{A}) \leq \sqrt{p} + \sqrt{q} + t \tag{5.80}$$

By this theorem, and also note that W^L is a vector, we can see that for any δ, there exist $\tilde{D} > 0$ and $M_1 > 0$ such that if $\tilde{d} \geq \tilde{D}$, then with probability at least $(1-\delta)$, for all $\theta \in B(\theta^{(0)}, C_0)$, we have

$$\left\|W^l\right\|_2 \leq 3\sqrt{\tilde{d}} \quad (\forall 0 \leq l \leq L-1) \qquad \text{and} \qquad \left\|W^L\right\|_2 \leq C_0 \leq 3\sqrt[4]{\tilde{d}} \tag{5.81}$$

as well as

$$\left\|\beta \boldsymbol{b}^l\right\|_2 \leq M_1 \sqrt{\tilde{d}} \qquad (\forall l = 0, \cdots, L) \tag{5.82}$$

Now we assume that (5.81) and (5.82) are true. Then, for any $\boldsymbol{x}$ such that $\|\boldsymbol{x}\|_2 \leq 1$,

$$\begin{aligned}
\left\|\boldsymbol{h}^1\right\|_2 &= \left\| \frac{1}{\sqrt{d_0}} W^0 \boldsymbol{x} + \beta \boldsymbol{b}^0 \right\|_2 \leq \frac{1}{\sqrt{d_0}} \left\|W^0\right\|_2 \|\boldsymbol{x}\|_2 + \left\|\beta \boldsymbol{b}^0\right\|_2 \leq (\frac{3}{\sqrt{d_0}} + M_1)\sqrt{\tilde{d}} \\
\left\|\boldsymbol{h}^{l+1}\right\|_2 &= \left\| \frac{1}{\sqrt{\tilde{d}}} W^l \boldsymbol{x}^l + \beta \boldsymbol{b}^l \right\|_2 \leq \frac{1}{\sqrt{\tilde{d}}} \left\|W^l\right\|_2 \left\|\boldsymbol{x}^l\right\|_2 + \left\|\beta \boldsymbol{b}^l\right\|_2 \qquad (\forall l \geq 1) \\
\left\|\boldsymbol{x}^l\right\|_2 &= \left\|\sigma(\boldsymbol{h}^l) - \sigma(\boldsymbol{0}^l) + \sigma(\boldsymbol{0}^l)\right\|_2 \leq L_0 \left\|\boldsymbol{h}^l\right\|_2 + \sigma(0)\sqrt{\tilde{d}} \qquad (\forall l \geq 1)
\end{aligned} \tag{5.83}$$

where L_0 is the Lipschitz constant of σ and $\sigma(\boldsymbol{0}^l) = (\sigma(0), \cdots, \sigma(0)) \in \mathbb{R}^{d_l}$. By induction, there exists an $M_2 > 0$ such that $\left\|\boldsymbol{x}^l\right\|_2 \leq M_2\sqrt{\tilde{d}}$ and $\left\|\boldsymbol{h}^l\right\|_2 \leq M_2\sqrt{\tilde{d}}$ for all $l = 1, \cdots, L$.

Denote $\boldsymbol{\alpha}^l = \nabla_{\boldsymbol{h}^l} f(\boldsymbol{x}) = \nabla_{\boldsymbol{h}^l} \boldsymbol{h}^{L+1}$. For all $l = 1, \cdots, L$, we have $\boldsymbol{\alpha}^l = \operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^l)) \frac{W^{l\top}}{\sqrt{\tilde{d}}} \boldsymbol{\alpha}^{l+1}$ where $\dot{\sigma}(x) \leq L_0$ for all $x \in \mathbb{R}$ since σ is L_0-Lipschitz, $\boldsymbol{\alpha}^{L+1} = 1$ and $\left\|\boldsymbol{\alpha}^L\right\|_2 = \left\|\operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^L)) \frac{W^{L\top}}{\sqrt{\tilde{d}}}\right\|_2 \leq \frac{3}{\sqrt[4]{\tilde{d}}} L_0$. Then, we can easily prove by induction that there exists an $M_3 > 1$ such that $\left\|\boldsymbol{\alpha}^l\right\|_2 \leq M_3 / \sqrt[4]{\tilde{d}}$ for all $l = 1, \cdots, L$ (note that this is not true for $L+1$ because $\boldsymbol{\alpha}^{L+1} = 1$).

For $l = 0$, $\nabla_{W^0} f(\boldsymbol{x}) = \frac{1}{\sqrt{d_0}} \boldsymbol{x}^0 \boldsymbol{\alpha}^{1\top}$, so $\left\|\nabla_{W^l} f(\boldsymbol{x})\right\|_2 \leq \frac{1}{\sqrt{d_0}} \left\|\boldsymbol{x}^0\right\|_2 \left\|\boldsymbol{\alpha}^1\right\|_2 \leq \frac{1}{\sqrt{d_0}} M_3 / \sqrt[4]{\tilde{d}}$. And for any $l = 1, \cdots, L$, $\nabla_{W^l} f(\boldsymbol{x}) = \frac{1}{\sqrt{\tilde{d}}} \boldsymbol{x}^l \boldsymbol{\alpha}^{l+1}$, so $\left\|\nabla_{W^l} f(\boldsymbol{x})\right\|_2 \leq \frac{1}{\sqrt{\tilde{d}}} \left\|\boldsymbol{x}^l\right\|_2 \left\|\boldsymbol{\alpha}^{l+1}\right\|_2 \leq M_2 M_3$. (Note that if $M_3 > 1$, then $\left\|\boldsymbol{\alpha}^{L+1}\right\|_2 \leq M_3$; and since $\tilde{d} \geq 1$, there is $\left\|\boldsymbol{\alpha}^l\right\|_2 \leq M_3$ for $l \leq L$.) Moreover, for $l = 0, \cdots, L$, $\nabla_{\boldsymbol{b}^l} f(\boldsymbol{x}) = \beta \boldsymbol{\alpha}^{l+1}$, so $\left\|\nabla_{\boldsymbol{b}^l} f(\boldsymbol{x})\right\|_2 \leq \beta M_3$. Thus, if (5.81) and (5.82) are true, then there exists an $M_4 > 0$, such that $\left\|\nabla_\theta f(\boldsymbol{x})\right\|_2 \leq M_4 / \sqrt{n}$. And since $\left\|\boldsymbol{x}_i\right\|_2 \leq 1$ for all i, so $\left\|J(\theta)\right\|_F \leq M_4$.

Next, we consider the difference in $\nabla_\theta f(\boldsymbol{x})$ between θ and $\tilde{\theta}$. Let $\tilde{f}, \tilde{W}, \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{h}}, \tilde{\boldsymbol{\alpha}}$ be the function and the values corresponding to $\tilde{\theta}$. There is

$$
\begin{aligned}
\left\|\boldsymbol{h}^1 - \tilde{\boldsymbol{h}}^1\right\|_2 &= \left\|\frac{1}{\sqrt{d_0}}(W^0 - \tilde{W}^0)\boldsymbol{x} + \beta(\boldsymbol{b}^0 - \tilde{\boldsymbol{b}}^0)\right\|_2 \\
&\leq \frac{1}{\sqrt{d_0}} \left\|W^0 - \tilde{W}^0\right\|_2 \left\|\boldsymbol{x}\right\|_2 + \beta \left\|\boldsymbol{b}^0 - \tilde{\boldsymbol{b}}^0\right\|_2 \leq \left(\frac{1}{\sqrt{d_0}} + \beta\right) \left\|\theta - \tilde{\theta}\right\|_2 \\
\left\|\boldsymbol{h}^{l+1} - \tilde{\boldsymbol{h}}^{l+1}\right\|_2 &= \left\|\frac{1}{\sqrt{\tilde{d}}} W^l(\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l) + \frac{1}{\sqrt{\tilde{d}}}(W^l - \tilde{W}^l)\tilde{\boldsymbol{x}}^l + \beta(\boldsymbol{b}^l - \tilde{\boldsymbol{b}}^l)\right\|_2 \\
&\leq \frac{1}{\sqrt{\tilde{d}}} \left\|W^l\right\|_2 \left\|\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l\right\|_2 + \frac{1}{\sqrt{\tilde{d}}} \left\|W^l - \tilde{W}^l\right\|_2 \left\|\tilde{\boldsymbol{x}}^l\right\|_2 + \beta \left\|\boldsymbol{b}^l - \tilde{\boldsymbol{b}}^l\right\|_2 \\
&\leq 3 \left\|\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l\right\|_2 + (M_2 + \beta) \left\|\theta - \tilde{\theta}\right\|_2 \qquad (\forall l \geq 1) \\
\left\|\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l\right\|_2 &= \left\|\sigma(\boldsymbol{h}^l) - \sigma(\tilde{\boldsymbol{h}}^l)\right\|_2 \leq L_0 \left\|\boldsymbol{h}^l - \tilde{\boldsymbol{h}}^l\right\|_2 \qquad (\forall l \geq 1)
\end{aligned}
\tag{5.84}
$$

By induction, there exists an $M_5 > 0$ such that $\left\|\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l\right\|_2 \leq M_5 \left\|\theta - \tilde{\theta}\right\|_2$ for all l.

For $\boldsymbol{\alpha}^l$, we have $\boldsymbol{\alpha}^{L+1} = \tilde{\boldsymbol{\alpha}}^{L+1} = 1$, and for all $l \geq 1$,

$$
\begin{aligned}
\left\|\boldsymbol{\alpha}^l - \tilde{\boldsymbol{\alpha}}^l\right\|_2 &= \left\|\operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^l)) \frac{W^{l\top}}{\sqrt{\tilde{d}}} \boldsymbol{\alpha}^{l+1} - \operatorname{diag}(\dot{\sigma}(\tilde{\boldsymbol{h}}^l)) \frac{\tilde{W}^{l\top}}{\sqrt{\tilde{d}}} \tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 \\
&\leq \left\|\operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^l)) \frac{W^{l\top}}{\sqrt{\tilde{d}}} (\boldsymbol{\alpha}^{l+1} - \tilde{\boldsymbol{\alpha}}^{l+1})\right\|_2 + \left\|\operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^l)) \frac{(W^l - \tilde{W}^l)^\top}{\sqrt{\tilde{d}}} \tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 \\
&\qquad + \left\|\operatorname{diag}((\dot{\sigma}(\boldsymbol{h}^l) - \dot{\sigma}(\tilde{\boldsymbol{h}}^l))) \frac{\tilde{W}^{l\top}}{\sqrt{\tilde{d}}} \tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 \\
&\leq 3L_0 \left\|\boldsymbol{\alpha}^{l+1} - \tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 + \left(M_3 L_0 \tilde{d}^{-1/2} + 3M_3 M_5 L_1 \tilde{d}^{-1/4}\right) \left\|\theta - \tilde{\theta}\right\|_2
\end{aligned}
\tag{5.85}
$$

where L_1 is the Lipschitz constant of $\dot{\sigma}$. Particularly, for $l = L$, though $\tilde{\boldsymbol{\alpha}}^{L+1} = 1$, since $\left\|\tilde{W}^L\right\|_2 \leq 3\tilde{d}^{1/4}$, (5.85) is still true. By induction, there exists an $M_6 > 0$ such that $\left\|\boldsymbol{\alpha}^l - \tilde{\boldsymbol{\alpha}}^l\right\|_2 \leq \frac{M_6}{\sqrt[4]{\tilde{d}}}\left\|\theta - \tilde{\theta}\right\|_2$ for all $l \geq 1$ (note that this is also true for $l = L+1$).

Thus, if (5.81) and (5.82) are true, then for all $\theta, \tilde{\theta} \in B(\theta^{(0)}, C_0)$, any $\boldsymbol{x}$ such that $\|\boldsymbol{x}\|_2 \leq 1$, we have

$$\begin{aligned}\left\|\nabla_{W^0} f(\boldsymbol{x}) - \nabla_{\tilde{W}^0}\tilde{f}(\boldsymbol{x})\right\|_2 &= \frac{1}{\sqrt{d_0}}\left\|\boldsymbol{x}\boldsymbol{\alpha}^{1\top} - \boldsymbol{x}\tilde{\boldsymbol{\alpha}}^{1\top}\right\|_2 \\ &\leq \frac{1}{\sqrt{d_0}}\left\|\boldsymbol{\alpha}^1 - \tilde{\boldsymbol{\alpha}}^1\right\|_2 \\ &\leq \frac{1}{\sqrt{d_0}}\frac{M_6}{\sqrt[4]{\tilde{d}}}\left\|\theta - \tilde{\theta}\right\|_2\end{aligned} \tag{5.86}$$

and for $l = 1, \cdots, L$, we have

$$\begin{aligned}\left\|\nabla_{W^l} f(\boldsymbol{x}) - \nabla_{\tilde{W}^l}\tilde{f}(\boldsymbol{x})\right\|_2 &= \frac{1}{\sqrt{\tilde{d}}}\left\|\boldsymbol{x}^l\boldsymbol{\alpha}^{l+1\top} - \tilde{\boldsymbol{x}}^l\tilde{\boldsymbol{\alpha}}^{l+1\top}\right\|_2 \\ &\leq \frac{1}{\sqrt{\tilde{d}}}\left(\left\|\boldsymbol{x}^l\right\|_2\left\|\boldsymbol{\alpha}^{l+1} - \tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 + \left\|\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l\right\|_2\left\|\tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2\right) \\ &\leq \left(\frac{M_2 M_6}{\sqrt[4]{\tilde{d}}} + \frac{M_5 M_3}{\sqrt{\tilde{d}}}\right)\left\|\theta - \tilde{\theta}\right\|_2\end{aligned} \tag{5.87}$$

Moreover, for any $l = 0, \cdots, L$, there is

$$\left\|\nabla_{b^l} f(\boldsymbol{x}) - \nabla_{\tilde{b}^l}\tilde{f}(\boldsymbol{x})\right\|_2 = \beta\left\|\boldsymbol{\alpha}^{l+1} - \tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 \leq \frac{\beta M_6}{\sqrt[4]{\tilde{d}}}\left\|\theta - \tilde{\theta}\right\|_2 \tag{5.88}$$

Overall, we can see that there exists a constant $M_7 > 0$ such that $\left\|\nabla_\theta f(\boldsymbol{x}) - \nabla_{\tilde{\theta}}\tilde{f}(\boldsymbol{x})\right\|_2 \leq \frac{M_7}{\sqrt{n}\cdot\sqrt[4]{\tilde{d}}}\left\|\theta - \tilde{\theta}\right\|_2$, so that $\left\|J(\theta) - J(\tilde{\theta})\right\|_F \leq \frac{M_7}{\sqrt[4]{\tilde{d}}}\left\|\theta - \tilde{\theta}\right\|_2$. □

5.7.5 Proof of Theorem 16

Let $\eta_1 = \min\{\eta_0, \eta^*\}$, where η_0 is defined in Corollary 13 and η^* is defined in Theorem 15. Let $f_{\text{lin}}^{(t)}(\boldsymbol{x})$ and $f_{\text{linERM}}^{(t)}(\boldsymbol{x})$ be the linearized neural networks of $f^{(t)}(\boldsymbol{x})$ and $f_{\text{ERM}}^{(t)}(\boldsymbol{x})$, respectively. By Theorem 15, for any $\delta > 0$, there exists $\tilde{D} > 0$ and a constant C such that

$$\begin{cases}\sup\limits_{t\geq 0}\left|f_{\text{lin}}^{(t)}(\boldsymbol{x}) - f^{(t)}(\boldsymbol{x})\right| \leq C\tilde{d}^{-1/4} \\ \sup\limits_{t\geq 0}\left|f_{\text{linERM}}^{(t)}(\boldsymbol{x}) - f_{\text{ERM}}^{(t)}(\boldsymbol{x})\right| \leq C\tilde{d}^{-1/4}\end{cases} \tag{5.89}$$

By Corollary 13, we have

$$\lim_{t\to\infty}\left|f_{\text{lin}}^{(t)}(\boldsymbol{x}) - f_{\text{linERM}}^{(t)}(\boldsymbol{x})\right| = 0 \tag{5.90}$$

Summing the above yields

$$\limsup_{t\to\infty}\left|f^{(t)}(\boldsymbol{x})-f^{(t)}_{\text{ERM}}(\boldsymbol{x})\right|\le 2C\tilde{d}^{-1/4} \tag{5.91}$$

which is the result we want. □

5.7.6 Proof of Theorem 17

To minimize the regularized risk (5.12) with gradient descent, the update rule is

$$\theta^{(t+1)}=\theta^{(t)}-\eta\sum_{i=1}^{n}q_i^{(t)}\nabla_\theta\ell(f^{(t)}(\boldsymbol{x}_i),y_i)-\eta\mu(\theta^{(t)}-\theta^{(0)}) \tag{5.92}$$

We can see that under the new rule, $\theta^{(t)}-\theta^{(0)}\in\text{span}(\nabla_\theta f^{(0)}(\boldsymbol{x}_1),\cdots,\nabla_\theta f^{(0)}(\boldsymbol{x}_n))$ is still true for all t. Let θ^* be the interpolator in $\text{span}(\nabla_\theta f^{(0)}(\boldsymbol{x}_1),\cdots,\nabla_\theta f^{(0)}(\boldsymbol{x}_n))$, then the empirical risk of θ is $\frac{1}{2n}\sum_{i=1}^{n}\langle\theta-\theta^*,\nabla_\theta f^{(0)}(\boldsymbol{x}_i)\rangle^2=\frac{1}{2n}\left\|\nabla_\theta f^{(0)}(\boldsymbol{X})^\top(\theta-\theta^*)\right\|_2^2$. Thus, there exists $T>0$ such that for any $t\ge T$,

$$\left\|\nabla_\theta f^{(0)}(\boldsymbol{X})^\top(\theta^{(t)}-\theta^*)\right\|_2^2\le 2n\epsilon \tag{5.93}$$

Let the smallest singular value of $\frac{1}{\sqrt{n}}\nabla_\theta f^{(0)}(\boldsymbol{X})$ be $s^{\min}$, and we have $s^{\min}>0$. Note that the column space of $\nabla_\theta f^{(0)}(\boldsymbol{X})$ is exactly $\text{span}(\nabla_\theta f^{(0)}(\boldsymbol{x}_1),\cdots,\nabla_\theta f^{(0)}(\boldsymbol{x}_n))$. Define $\boldsymbol{H}\in\mathbb{R}^{p\times n}$ such that its columns form an orthonormal basis of this subspace, then there exists $\boldsymbol{G}\in R^{n\times n}$ such that $\nabla_\theta f^{(0)}(\boldsymbol{X})=\boldsymbol{H}\boldsymbol{G}$, and the smallest singular value of $\frac{1}{\sqrt{n}}\boldsymbol{G}$ is also $s^{\min}$. Since $\theta^{(t)}-\theta^{(0)}$ is also in this subspace, there exists $\boldsymbol{v}\in\mathbb{R}^n$ such that $\theta^{(t)}-\theta^*=\boldsymbol{H}\boldsymbol{v}$. Then we have $\sqrt{2n\epsilon}\ge\left\|\boldsymbol{G}^\top\boldsymbol{H}^\top\boldsymbol{H}\boldsymbol{v}\right\|_2=\left\|\boldsymbol{G}^\top\boldsymbol{v}\right\|_2$. Thus, $\|\boldsymbol{v}\|_2\le\frac{\sqrt{2\epsilon}}{s^{\min}}$, which implies

$$\left\|\theta^{(t)}-\theta^*\right\|_2\le\frac{\sqrt{2\epsilon}}{s^{\min}} \tag{5.94}$$

By Corollary 13, if we minimize the unregularized risk with ERM, then θ always converges to the interpolator θ^*. So for any $t\ge T$ and any test point $\boldsymbol{x}$ such that $\|\boldsymbol{x}\|_2\le 1$,

$$|f^{(t)}_{\text{linreg}}(\boldsymbol{x})-f^{(t)}_{\text{linERM}}(\boldsymbol{x})|=|\langle\theta^{(t)}-\theta^*,\nabla_\theta f^{(0)}(\boldsymbol{x})\rangle|\le\frac{M_0\sqrt{2\epsilon}}{s^{\min}} \tag{5.95}$$

which implies (5.14). □

5.7.7 Proof of Theorem 18

First of all, with some simple linear algebra analysis, we can prove the following proposition:

Proposition 23. *For any positive definite symmetric matrix $\boldsymbol{H}\in\mathbb{R}^{n\times n}$, denote its largest and smallest eigenvalues by $\lambda^{\max}$ and $\lambda^{\min}$. Then, for any $\boldsymbol{q}\in\mathbb{R}^n_+$ and $\boldsymbol{Q}=\text{diag}(q_1,\cdots,q_n)$, $\boldsymbol{H}\boldsymbol{Q}$ has n positive eigenvalues that are all in $[\min_i q_i\cdot\lambda^{\min},\max_i q_i\cdot\lambda^{\max}]$.*

Proof. $\boldsymbol{H}$ is a positive definite symmetric matrix, so there exists $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ such that $\boldsymbol{H} = \boldsymbol{A}^\top \boldsymbol{A}$, and $\boldsymbol{A}$ is full-rank. First, any eigenvalue of $\boldsymbol{A}\boldsymbol{Q}\boldsymbol{A}^\top$ is also an eigenvalue of $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{Q}$ and vice versa, because for any eigenvalue λ of $\boldsymbol{A}\boldsymbol{Q}\boldsymbol{A}^\top$ we have some $\boldsymbol{v} \neq 0$ such that $\boldsymbol{A}\boldsymbol{Q}\boldsymbol{A}^\top\boldsymbol{v} = \lambda\boldsymbol{v}$. Multiplying both sides by $\boldsymbol{A}^\top$ on the left yields $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{Q}(\boldsymbol{A}^\top\boldsymbol{v}) = \lambda(\boldsymbol{A}^\top\boldsymbol{v})$ which implies that λ is also an eigenvalue of $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{Q}$ because $\boldsymbol{A}^\top\boldsymbol{v} \neq 0$ as $\lambda\boldsymbol{v} \neq 0$. We can prove the other direction similarly.

Second, by condition we know that the eigenvalues of $\boldsymbol{A}^\top\boldsymbol{A}$ are all in $[\lambda^{\min}, \lambda^{\max}]$ where $\lambda^{\min} > 0$, which implies for any unit vector $\boldsymbol{v}$, $\boldsymbol{v}^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{v} \in [\lambda^{\min}, \lambda^{\max}]$, which is equivalent to $\|\boldsymbol{A}\boldsymbol{v}\|_2 \in [\sqrt{\lambda^{\min}}, \sqrt{\lambda^{\max}}]$. Thus, we have $\boldsymbol{v}^\top\boldsymbol{A}^\top\boldsymbol{Q}\boldsymbol{A}\boldsymbol{v} \in [\lambda^{\min}\min_i q_i, \lambda^{\max}\max_i q_i]$, which implies that the eigenvalues of $\boldsymbol{A}^\top\boldsymbol{Q}\boldsymbol{A}$ are all in $[\lambda^{\min}\min_i q_i, \lambda^{\max}\max_i q_i]$.

Thus, the eigenvalues of $\boldsymbol{H}\boldsymbol{Q} = \boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{Q}$ are all in $[\lambda^{\min}\min_i q_i, \lambda^{\max}\max_i q_i]$. □

Now return back to the proof of Theorem 18. We still use the shorthand (5.40). With L_2 penalty, the update rule of the reweighting algorithm for the neural network is:

$$\theta^{(t+1)} = \theta^{(t)} - \eta J(\theta^{(t)})\boldsymbol{Q}^{(t)} g(\theta^{(t)}) - \eta\mu(\theta^{(t)} - \theta^{(0)}) \tag{5.96}$$

And the update rule for the linearized neural network is:

$$\theta_{\text{lin}}^{(t+1)} = \theta_{\text{lin}}^{(t)} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)} g_{\text{lin}}(\theta^{(t)}) - \eta\mu(\theta_{\text{lin}}^{(t)} - \theta^{(0)}) \tag{5.97}$$

First, we need to prove that there exists D_0 such that for all $\tilde{d} \geq D_0$, $\sup_{t\geq 0}\left\|\theta^{(t)} - \theta^{(0)}\right\|_2$ is bounded with high probability. Denote $a_t = \theta^{(t)} - \theta^{(0)}$. By (5.96) we have

$$\begin{aligned} a_{t+1} =& (1-\eta\mu)a_t - \eta[J(\theta^{(t)}) - J(\theta^{(0)})]\boldsymbol{Q}^{(t)} g(\theta^{(t)}) \\ & - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}[g(\theta^{(t)}) - g(\theta^{(0)})] - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)} g(\theta^{(0)}) \end{aligned} \tag{5.98}$$

which implies

$$\begin{aligned} \|a_{t+1}\|_2 \leq & \left\|(1-\eta\mu)\boldsymbol{I} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)} J(\tilde{\theta}^{(t)})^\top\right\|_2 \|a_t\|_2 \\ & + \eta\left\|J(\theta^{(t)}) - J(\theta^{(0)})\right\|_F \left\|g(\theta^{(t)})\right\|_2 + \eta\left\|J(\theta^{(0)})\right\|_F \left\|g(\theta^{(0)})\right\|_2 \end{aligned} \tag{5.99}$$

where $\tilde{\theta}^{(t)}$ is some linear interpolation between $\theta^{(t)}$ and $\theta^{(0)}$. Our choice of η ensures that $\eta\mu < 1$. Similar to (5.48), we can show that for any $\delta > 0$, there exists a constant $R_0 > 0$ such that with probability at least $(1-\delta/3)$, $\left\|g(\theta^{(0)})\right\|_2 < R_0$. Let M be as defined in Lemma 21. Denote $A = \eta M R_0$, and let $C_0 = \frac{4A}{\eta\mu}$ in Lemma 21[6]. By Lemma 21, there exists D_1 such that for all $\tilde{d} \geq D_1$, with probability at least $(1-\delta/3)$, (5.47) is true.

Now we prove by induction that $\|a_t\|_2 < C_0$. It is true for $t = 0$, so we need to prove that if $\|a_t\|_2 < C_0$, then $\|a_{t+1}\|_2 < C_0$.

For the first term on the right-hand side of (5.99), we have

$$\begin{aligned} \left\|(1-\eta\mu)\boldsymbol{I} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)} J(\tilde{\theta}^{(t)})^\top\right\|_2 \leq & (1-\eta\mu)\left\|\boldsymbol{I} - \frac{\eta}{1-\eta\mu} J(\theta^{(0)})\boldsymbol{Q}^{(t)} J(\theta^{(0)})^\top\right\|_2 \\ & + \eta\left\|J(\theta^{(0)})\right\|_F \left\|J(\tilde{\theta}^{(t)}) - J(\theta^{(0)})\right\|_F \end{aligned} \tag{5.100}$$

[6]Note that Lemma 21 only depends on the network structure and does not depend on the update rule, so we can use this lemma here.

Like what we have done before, we can show that all non-zero eigenvalues of $J(\theta^{(0)})\boldsymbol{Q}^{(t)}J(\theta^{(0)})^\top$ are eigenvalues of $J(\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}$. This is because for any $\lambda \neq 0$, if $J(\theta^{(0)})\boldsymbol{Q}^{(t)}J(\theta^{(0)})^\top \boldsymbol{v} = \lambda \boldsymbol{v}$, then $J(\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}(J(\theta^{(0)})^\top \boldsymbol{v}) = \lambda(J(\theta^{(0)})^\top \boldsymbol{v})$, and $J(\theta^{(0)})^\top \boldsymbol{v} \neq 0$ since $\lambda \boldsymbol{v} \neq 0$, so λ is also an eigenvalue of $J(\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}$. On the other hand, by Theorem 14, $J(\theta^{(0)})^\top J(\theta^{(0)})\boldsymbol{Q}^{(t)}$ converges in probability to $\Theta \boldsymbol{Q}^{(t)}$ whose eigenvalues are all in $[0, \lambda^{\max}]$ by Proposition 23. So there exists D_2 such that for all $\tilde{d} \geq D_2$, with probability at least $(1-\delta/3)$, the eigenvalues of $J(\theta^{(0)})\boldsymbol{Q}^{(t)}J(\theta^{(0)})^\top$ are all in $[0, \lambda^{\max} + \lambda^{\min}]$ for all t. Since $\eta/(1-\eta\mu) \leq (\lambda^{\min} + \lambda^{\max})^{-1}$ by our choice of η, we have

$$\left\| \boldsymbol{I} - \frac{\eta}{1-\eta\mu} J(\theta^{(0)})\boldsymbol{Q}^{(t)}J(\theta^{(0)})^\top \right\|_2 \leq 1 \tag{5.101}$$

On the other hand, we can use (5.47) since $\|a_t\|_2 < C_0$, so $\left\|J(\theta^{(0)})\right\|_F \left\|J(\tilde{\theta}^{(t)}) - J(\theta^{(0)})\right\|_F \leq \frac{M^2}{\sqrt[4]{\tilde{d}}} C_0$. Therefore, there exists D_3 such that for all $\tilde{d} \geq D_3$,

$$\left\| (1-\eta\mu)\boldsymbol{I} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}J(\tilde{\theta}^{(t)})^\top \right\|_2 \leq 1 - \frac{\eta\mu}{2} \tag{5.102}$$

For the second term, we have

$$\begin{aligned} \left\|g(\theta^{(t)})\right\|_2 &\leq \left\|g(\theta^{(t)}) - g(\theta^{(0)})\right\|_2 + \left\|g(\theta^{(0)})\right\|_2 \\ &\leq \left\|J(\tilde{\theta}^{(t)})\right\|_2 \left\|\theta^{(t)} - \theta^{(0)}\right\|_2 + R_0 \leq MC_0 + R_0 \end{aligned} \tag{5.103}$$

And for the third term, we have

$$\eta \left\|J(\theta^{(0)})\right\|_F \left\|g(\theta^{(0)})\right\|_2 \leq \eta M R_0 = A \tag{5.104}$$

Thus, we have

$$\|a_{t+1}\|_2 \leq \left(1 - \frac{\eta\mu}{2}\right) \|a_t\|_2 + \frac{\eta M(MC_0 + R_0)}{\sqrt[4]{\tilde{d}}} + A \tag{5.105}$$

So there exists D_4 such that for all $\tilde{d} \geq D_4$, $\|a_{t+1}\|_2 \leq \left(1 - \frac{\eta\mu}{2}\right) \|a_t\|_2 + 2A$. This shows that if $\|a_t\|_2 < C_0$ is true, then $\|a_{t+1}\|_2 < C_0$ will also be true.

In conclusion, by union bound, we have proved that for any $\delta > 0$, with probability at least $(1-\delta)$ for all $\tilde{d} \geq D_0 = \max\{D_1, D_2, D_3, D_4\}$, $\left\|\theta^{(t)} - \theta^{(0)}\right\|_2 < C_0$ is true for all t. This also implies that for $C_1 = MC_0 + R_0$, we have $\left\|g(\theta^{(t)})\right\|_2 \leq C_1$ for all t by (5.103).

Second, let $\Delta_t = \theta_{\text{lin}}^{(t)} - \theta^{(t)}$. Then we have

$$\Delta_{t+1} - \Delta_t = \eta(J(\theta^{(t)})\boldsymbol{Q}^{(t)}g(\theta^{(t)}) - J(\theta^{(0)})\boldsymbol{Q}^{(t)}g_{\text{lin}}(\theta^{(t)}) - \mu\Delta_t) \tag{5.106}$$

which implies

$$\Delta_{t+1} = \left[(1-\eta\mu)\boldsymbol{I} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}J(\tilde{\theta}^{(t)})^\top\right] \Delta_t + \eta(J(\theta^{(t)}) - J(\theta^{(0)}))\boldsymbol{Q}^{(t)}g(\theta^{(t)}) \tag{5.107}$$

By (5.102), with probability at least $(1-\delta)$ for all $\tilde{d} \geq D_0$, we have

$$\begin{aligned}\|\Delta_{t+1}\|_2 &\leq \left\|(1-\eta\mu)\boldsymbol{I} - \eta J(\theta^{(0)})\boldsymbol{Q}^{(t)}J(\tilde{\theta}^{(t)})^\top\right\|_2 \|\Delta_t\|_2 + \eta \left\|J(\theta^{(t)}) - J(\theta^{(0)})\right\|_F \left\|g(\theta^{(t)})\right\|_2 \\ &\leq \left(1 - \frac{\eta\mu}{2}\right)\|\Delta_t\|_2 + \eta\frac{M}{\sqrt[4]{\tilde{d}}}C_0C_1\end{aligned} \tag{5.108}$$

Again, as $\Delta_0 = 0$, we can prove by induction that for all t,

$$\|\Delta_t\|_2 < \frac{2MC_0C_1}{\mu}\tilde{d}^{-1/4} \tag{5.109}$$

For any test point $\boldsymbol{x}$ such that $\|\boldsymbol{x}\|_2 \leq 1$, we have

$$\begin{aligned}\left|f_{\text{reg}}^{(t)}(\boldsymbol{x}) - f_{\text{linreg}}^{(t)}(\boldsymbol{x})\right| &= \left|f(\boldsymbol{x};\theta^{(t)}) - f_{\text{lin}}(\boldsymbol{x};\theta_{\text{lin}}^{(t)})\right| \\ &\leq \left|f(\boldsymbol{x};\theta^{(t)}) - f_{\text{lin}}(\boldsymbol{x};\theta^{(t)})\right| + \left|f_{\text{lin}}(\boldsymbol{x};\theta^{(t)}) - f_{\text{lin}}(\boldsymbol{x};\theta_{\text{lin}}^{(t)})\right| \\ &\leq \left|f(\boldsymbol{x};\theta^{(t)}) - f_{\text{lin}}(\boldsymbol{x};\theta^{(t)})\right| + \left\|\nabla_\theta f(\boldsymbol{x};\theta^{(0)})\right\|_2 \left\|\theta^{(t)} - \theta_{\text{lin}}^{(t)}\right\|_2 \\ &\leq \left|f(\boldsymbol{x};\theta^{(t)}) - f_{\text{lin}}(\boldsymbol{x};\theta^{(t)})\right| + M\|\Delta_t\|_2\end{aligned} \tag{5.110}$$

For the first term, note that

$$\begin{cases} f(\boldsymbol{x};\theta^{(t)}) - f(\boldsymbol{x};\theta^{(0)}) = \nabla_\theta f(\boldsymbol{x};\tilde{\theta}^{(t)})(\theta^{(t)} - \theta^{(0)}) \\ f_{\text{lin}}(\boldsymbol{x};\theta^{(t)}) - f_{\text{lin}}(\boldsymbol{x};\theta^{(0)}) = \nabla_\theta f(\boldsymbol{x};\theta^{(0)})(\theta^{(t)} - \theta^{(0)}) \end{cases} \tag{5.111}$$

where $\tilde{\theta}^{(t)}$ is some linear interpolation between $\theta^{(t)}$ and $\theta^{(0)}$. Since $f(\boldsymbol{x};\theta^{(0)}) = f_{\text{lin}}(\boldsymbol{x};\theta^{(0)})$,

$$\left|f(\boldsymbol{x};\theta^{(t)}) - f_{\text{lin}}(\boldsymbol{x};\theta^{(t)})\right| \leq \left\|\nabla_\theta f(\boldsymbol{x};\tilde{\theta}^{(t)}) - \nabla_\theta f(\boldsymbol{x};\theta^{(0)})\right\|_2 \left\|\theta^{(t)} - \theta^{(0)}\right\|_2 \leq \frac{M}{\sqrt[4]{\tilde{d}}}C_0^2 \tag{5.112}$$

Thus, we have shown that for all $\tilde{d} \geq D_0$, with probability at least $(1-\delta)$ for all t and all $\boldsymbol{x}$,

$$\left|f_{\text{reg}}^{(t)}(\boldsymbol{x}) - f_{\text{linreg}}^{(t)}(\boldsymbol{x})\right| \leq \left(MC_0^2 + \frac{2M^2C_0C_1}{\mu}\right)\tilde{d}^{-1/4} = O(\tilde{d}^{-1/4}) \tag{5.113}$$

Given that $\hat{\mathcal{R}}(f_{\text{linreg}}^{(t)}) < \epsilon$ for sufficiently large t, this also implies that

$$\left|\hat{\mathcal{R}}(f_{\text{linreg}}^{(t)}) - \hat{\mathcal{R}}(f_{\text{reg}}^{(t)})\right| = O(\tilde{d}^{-1/4}\sqrt{\epsilon} + \tilde{d}^{-1/2}) \tag{5.114}$$

So for a fixed ϵ, there exists $D > 0$ such that for all $d \geq D$, for sufficiently large t,

$$\hat{\mathcal{R}}(f_{\text{reg}}^{(t)}) < \epsilon \Rightarrow \hat{\mathcal{R}}(f_{\text{linreg}}^{(t)}) < 2\epsilon \tag{5.115}$$

By Theorem 15, we have

$$\sup_{t\geq 0}\left|f_{\text{linERM}}^{(t)}(\boldsymbol{x}) - f_{\text{ERM}}^{(t)}(\boldsymbol{x})\right| = O(\tilde{d}^{-1/4}) \tag{5.116}$$

Combining Theorem 17 with (5.113) and (5.116) derives

$$\limsup_{t\to\infty}\left|f_{\text{reg}}^{(t)}(\boldsymbol{x}) - f_{\text{ERM}}^{(t)}(\boldsymbol{x})\right| = O(\tilde{d}^{-1/4} + \sqrt{\epsilon}) \tag{5.117}$$

Letting $\tilde{d} \to \infty$ leads to the result we need. □

5.8 A note on the proofs in Lee et al. [89]

We have mentioned that the proofs in Lee et al. [89], particularly the proofs of their Theorem 2.1 and Lemma 1 in their Appendix G, are flawed. In order to fix their proof, we change the network initialization to (5.9). In this section, we will demonstrate what goes wrong in the proofs in Lee et al. [89], and how we manage to fix the proof. For clarity, we are referring to the following version of the paper: https://arxiv.org/pdf/1902.06720v4.pdf.

To avoid confusion, in this section we will still use the notations used in our paper.

5.8.1 Their problems

Lee et al. [89] claimed in their Theorem 2.1 that under the conditions of our Theorem 15, for any $\delta > 0$, there exist $\tilde{D} > 0$ and a constant C such that for any $\tilde{d} \geq \tilde{D}$, with probability at least $(1-\delta)$, the gap between the output of a sufficiently wide fully-connected neural network and the output of its linearized neural network at any test point $\boldsymbol{x}$ can be uniformly bounded by

$$\sup_{t \geq 0} \left| f^{(t)}(\boldsymbol{x}) - f_{\text{lin}}^{(t)}(\boldsymbol{x}) \right| \leq C \tilde{d}^{-1/2} \qquad \text{(claimed)} \tag{5.118}$$

where they used the original NTK formulation and initialization in Jacot et al. [67]:

$$\begin{cases} \boldsymbol{h}^{l+1} = \frac{W^l}{\sqrt{d_l}} \boldsymbol{x}^l + \beta \boldsymbol{b}^l \\ \boldsymbol{x}^{l+1} = \sigma(\boldsymbol{h}^{l+1}) \end{cases} \quad \text{and} \quad \begin{cases} W_{i,j}^{l(0)} \sim \mathcal{N}(0,1) \\ b_i^{l(0)} \sim \mathcal{N}(0,1) \end{cases} \quad (\forall l = 0, \cdots, L) \tag{5.119}$$

where $\boldsymbol{x}_0 = \boldsymbol{x}$ and $f(\boldsymbol{x}) = h^{L+1}$. However, in their proof in their Appendix G, they did not directly prove their result for the NTK formulation, but instead they proved another result for the following formulation which they called the *standard formulation*:

$$\begin{cases} \boldsymbol{h}^{l+1} = W^l \boldsymbol{x}^l + \beta \boldsymbol{b}^l \\ \boldsymbol{x}^{l+1} = \sigma(\boldsymbol{h}^{l+1}) \end{cases} \quad \text{and} \quad \begin{cases} W_{i,j}^{l(0)} \sim \mathcal{N}(0, \frac{1}{d_l}) \\ b_i^{l(0)} \sim \mathcal{N}(0,1) \end{cases} \quad (\forall l = 0, \cdots, L) \tag{5.120}$$

See their Appendix F for the definition of their standard formulation. In the original formulation, they also included two constants σ_w and σ_b for standard deviations, and for simplicity we omit these constants here. Note that the outputs of the NTK formulation and the standard formulation at initialization are actually the same. The only difference is that the norm of the weight W^l and the gradient of the model output with respect to W^l are different for all l.

In their Appendix G, they claimed that if a network with the standard formulation is trained by minimizing the squared loss with gradient descent and learning rate $\eta' = \eta/\tilde{d}$, where η is our learning rate in Theorem 15 and also their learning rate in their Theorem 2.1, then (5.118) is true for this network, so it is also true for a network with the NTK formulation because the two formulations have the same network output. And then they claimed in their equation (S37) that applying learning rate η' to the standard formulation is equivalent to applying the following learning rates

$$\eta_W^l = \frac{d_l}{d_{\max}} \eta \qquad \text{and} \qquad \eta_{\boldsymbol{b}}^l = \frac{1}{d_{\max}} \eta \tag{5.121}$$

to W^l and $\boldsymbol{b}^l$ of the NTK formulation, where $d_{\max} = \max\{d_0, \cdots, d_L\}$.

To avoid confusion, in the following discussions we will still use the NTK formulation and initialization if not stated otherwise.

Problem 1. Claim (5.121) is true, but it leads to two problems. The first problem is that $\eta_{\boldsymbol{b}}^l = O(d_{\max}^{-1})$ since $\eta = O(1)$, while their Theorem 2.1 needs the learning rate to be $O(1)$. Nevertheless, this problem can be simply fixed by modifying their standard formulation as $\boldsymbol{h}^{l+1} = W^l \boldsymbol{x}^l + \beta\sqrt{d_l}\boldsymbol{b}^l$ where $b_i^{l(0)} \sim \mathcal{N}(0, d_l^{-1})$. The real problem that is non-trivial to fix is that by (5.121), there is $\eta_W^0 = \frac{d_0}{d_{\max}}\eta$. However, note that d_0 is a constant since it is the dimension of the input space, while $d_{\max}$ goes to infinity. With that being said, in (5.121) they were essentially using a very small learning rate for the first layer W^0 but a normal learning rate for the rest of the layers, which definitely does not match with their claim in their Theorem 2.1.

Problem 2. Another big problem is that the proof of their Lemma 1 in their Appendix G is erroneous, and consequently their Theorem 2.1 is unsound as it heavily depends on their Lemma 1. In their Lemma 1, they claimed that for some constant $M > 0$, for any two models with the parameters θ and $\tilde{\theta}$ such that $\theta, \tilde{\theta} \in B(\theta^{(0)}, C_0)$ for some constant C_0, there is

$$\left\|J(\theta) - J(\tilde{\theta})\right\|_F \leq \frac{M}{\sqrt{\tilde{d}}}\left\|\theta - \tilde{\theta}\right\|_2 \qquad \text{(claimed)} \tag{5.122}$$

Note that the original claim in their paper was $\left\|J(\theta) - J(\tilde{\theta})\right\|_F \leq M\sqrt{\tilde{d}}\left\|\theta - \tilde{\theta}\right\|_2$. This is because they were proving this result for their standard formulation. Compared to the standard formulation, in the NTK formulation θ is $\sqrt{\tilde{d}}$ times larger, while the Jacobian $J(\theta)$ is $\sqrt{\tilde{d}}$ times smaller. This is also why here we have $\theta, \tilde{\theta} \in B(\theta^{(0)}, C_0)$ instead of $\theta, \tilde{\theta} \in B(\theta^{(0)}, C_0\tilde{d}^{-1/2})$ for the NTK formulation. Therefore, equivalently they were claiming (5.122) for the NTK formulation.

However, their proof of (5.122) in incorrect. Specifically, the right-hand side of their inequality (S86) is incorrect. Using the notations in our Appendix 5.7.4, their (S86) essentially claimed that

$$\left\|\boldsymbol{\alpha}^l - \tilde{\boldsymbol{\alpha}}^l\right\|_2 \leq \frac{M}{\sqrt{\tilde{d}}}\left\|\theta - \tilde{\theta}\right\|_2 \qquad \text{(claimed)} \tag{5.123}$$

for any $\theta, \tilde{\theta} \in B(\theta^{(0)}, C_0)$, where $\boldsymbol{\alpha}^l = \nabla_{\boldsymbol{h}^l}\boldsymbol{h}^{L+1}$ and $\tilde{\boldsymbol{\alpha}}^l$ is the same gradient for the second model. Note that their (S86) does not have the $\sqrt{\tilde{d}}$ in the denominator which appears in (5.123). This is because for their standard formulation, θ is $\sqrt{\tilde{d}}$ times smaller than the original NTK formulation, while $\left\|\boldsymbol{\alpha}^l\right\|_2$ has the same order in the two formulations because all $\boldsymbol{h}^l$ are the same.

However, it is actually impossible to prove (5.123). Consider the following counterexample: Since θ and $\tilde{\theta}$ are arbitrarily chosen, we can choose them such that they only differ in b_1^l for some $1 \leq l < L$. Then, $\left\|\theta - \tilde{\theta}\right\|_2 = \left|b_1^l - \tilde{b}_1^l\right|$. We can see that $\boldsymbol{h}^{l+1}$ and $\tilde{\boldsymbol{h}}^{l+1}$ only differ in the first

element, and $\left|h_1^{l+1} - \tilde{h}_1^{l+1}\right| = \left|\beta(b_1^l - \tilde{b}_1^l)\right|$. Moreover, we have $W^{l+1} = \tilde{W}^{l+1}$, so there is

$$\begin{aligned}
\boldsymbol{\alpha}^{l+1} - \tilde{\boldsymbol{\alpha}}^{l+1} =& \operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^{l+1}))\frac{W^{l+1\top}}{\sqrt{\tilde{d}}}\boldsymbol{\alpha}^{l+2} - \operatorname{diag}(\dot{\sigma}(\tilde{\boldsymbol{h}}^{l+1}))\frac{\tilde{W}^{l+1\top}}{\sqrt{\tilde{d}}}\tilde{\boldsymbol{\alpha}}^{l+2} \\
=& \left[\operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^{l+1})) - \operatorname{diag}(\dot{\sigma}(\tilde{\boldsymbol{h}}^{l+1}))\right]\frac{W^{l+1\top}}{\sqrt{\tilde{d}}}\boldsymbol{\alpha}^{l+2} \\
&+ \operatorname{diag}(\dot{\sigma}(\tilde{\boldsymbol{h}}^{l+1}))\frac{W^{l+1\top}}{\sqrt{\tilde{d}}}(\boldsymbol{\alpha}^{l+2} - \tilde{\boldsymbol{\alpha}}^{l+2})
\end{aligned} \tag{5.124}$$

Then we can lower bound $\left\|\boldsymbol{\alpha}^{l+1} - \tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2$ by

$$\begin{aligned}
\left\|\boldsymbol{\alpha}^{l+1} - \tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 \geq& \left\|\left[\operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^{l+1})) - \operatorname{diag}(\dot{\sigma}(\tilde{\boldsymbol{h}}^{l+1}))\right]\frac{W^{l+1\top}}{\sqrt{\tilde{d}}}\boldsymbol{\alpha}^{l+2}\right\|_2 \\
&- \left\|\operatorname{diag}(\dot{\sigma}(\tilde{\boldsymbol{h}}^{l+1}))\frac{W^{l+1\top}}{\sqrt{\tilde{d}}}(\boldsymbol{\alpha}^{l+2} - \tilde{\boldsymbol{\alpha}}^{l+2})\right\|_2
\end{aligned} \tag{5.125}$$

The first term on the right-hand side is equal to $\left|\left[\dot{\sigma}(h_1^{l+1}) - \dot{\sigma}(\tilde{h}_1^{l+1})\right]\langle W_1^{l+1}/\sqrt{\tilde{d}}, \boldsymbol{\alpha}^{l+2}\rangle\right|$ where W_1^{l+1} is the first row of W^{l+1}. We know that $\left\|W_1^{l+1}\right\|_2 = \Theta\left(\sqrt{\tilde{d}}\right)$ with high probability as its elements are sampled from $\mathcal{N}(0,1)$, and in their (S85) they claimed that $\left\|\boldsymbol{\alpha}^{l+2}\right\|_2 = O(1)$, which is true. In addition, they assumed that $\dot{\sigma}$ is Lipschitz. Hence, we can see that

$$\left\|\left[\operatorname{diag}(\dot{\sigma}(\boldsymbol{h}^{l+1})) - \operatorname{diag}(\dot{\sigma}(\tilde{\boldsymbol{h}}^{l+1}))\right]\frac{W^{l+1\top}}{\sqrt{\tilde{d}}}\boldsymbol{\alpha}^{l+2}\right\|_2 = O\left(\left|h_1^{l+1} - \tilde{h}_1^{l+1}\right|\right) = O\left(\left\|\theta - \tilde{\theta}\right\|_2\right) \tag{5.126}$$

On the other hand, suppose that claim (5.123) is true, then $\left\|\boldsymbol{\alpha}^{l+2} - \tilde{\boldsymbol{\alpha}}^{l+2}\right\|_2 = O\left(\tilde{d}^{-1/2}\left\|\theta - \tilde{\theta}\right\|_2\right)$. Then we can see that the second term on the right-hand side is $O\left(\tilde{d}^{-1/2}\left\|\theta - \tilde{\theta}\right\|_2\right)$ because $\left\|W^{l+1}\right\|_2 = O(\sqrt{\tilde{d}})$ and $\dot{\sigma}(x)$ is bounded by a constant as σ is Lipschitz. Thus, for a very large $\tilde{d}$, the second-term is an infinitely small term compared to the first term, so we can only prove that

$$\left\|\boldsymbol{\alpha}^{l+1} - \tilde{\boldsymbol{\alpha}}^{l+1}\right\|_2 = O\left(\left\|\theta - \tilde{\theta}\right\|_2\right) \tag{5.127}$$

which is different from (5.123) because it lacks a critical $\tilde{d}^{-1/2}$ and thus leads to a contradiction. Hence, we cannot prove (5.123) with the $\tilde{d}^{-1/2}$ factor, and consequently we cannot prove (5.122) with the $\sqrt{\tilde{d}}$ in the denominator on the right-hand side. As a result, their Lemma 1 and Theorem 2.1 cannot be proved without this critical $\tilde{d}^{-1/2}$. Similarly, we can also construct a counterexample where θ and $\tilde{\theta}$ only differ in the first row of some W^l.

5.8.2 Our fixes

Regarding Problem 1, we can still use an $O(1)$ learning rate for the first layer in the NTK formulation given that $\|\boldsymbol{x}\|_2 \leq 1$. This is because for the first layer, we have

$$\nabla_{W^0} f(\boldsymbol{x}) = \frac{1}{\sqrt{d_0}} \boldsymbol{x}^0 \alpha^{1\top} = \frac{1}{\sqrt{d_0}} \boldsymbol{x} \alpha^{1\top} \tag{5.128}$$

For all $l \geq 1$, we have $\left\|\boldsymbol{x}^l\right\|_2 = O(\tilde{d}^{1/2})$. However, for $l = 0$, we instead have $\|\boldsymbol{x}^0\|_2 = O(1)$. Thus, we can prove that the norm of $\nabla_{W^0} f(\boldsymbol{x})$ has the same order as the gradient with respect to any other layer, so there is no need to use a smaller learning rate for the first layer.

Regarding Problem 2, in our formulation (5.8) and initialization (5.9), the initialization of the last layer of the NTK formulation is changed from the Gaussian initialization $W_{i,j}^{L(0)} \sim \mathcal{N}(0,1)$ to the zero initialization $W_{i,j}^{L(0)} = 0$. Now we show how this modification solves Problem 2.

The main consequence of changing the initialization of the last layer is that (5.81) becomes different: instead of $\left\|W^L\right\|_2 \leq 3\sqrt{\tilde{d}}$, we now have $\left\|W^L\right\|_2 \leq C_0 \leq 3\sqrt[4]{\tilde{d}}$. In fact, for any $r \in (0, 1/2)$, we can prove that $\left\|W^L\right\|_2 \leq 3\tilde{d}^r$ for sufficiently large $\tilde{d}$. In our proof we choose $r = 1/4$.

Consequently, instead of $\left\|\boldsymbol{\alpha}^l\right\|_2 \leq M_3$, we can now prove that $\left\|\boldsymbol{\alpha}^l\right\|_2 \leq M_3 \tilde{d}^{r-1/2}$ for all $l \leq L$ by induction. So now we can prove $\left\|\boldsymbol{\alpha}^l - \tilde{\boldsymbol{\alpha}}^l\right\|_2 = O\left(\tilde{d}^{r-1/2} \left\|\theta - \tilde{\theta}\right\|_2\right)$ instead of $O\left(\left\|\theta - \tilde{\theta}\right\|_2\right)$, because

- For $l < L$, we now have $\left\|\boldsymbol{\alpha}^{l+1}\right\|_2 = O(\tilde{d}^{r-1/2})$ instead of $O(1)$, so we can have the additional $\tilde{d}^{r-1/2}$ factor in the bound.
- For $l = L$, although $\left\|\boldsymbol{\alpha}^{L+1}\right\|_2 = 1$, note that $\left\|W^L\right\|_2$ now becomes $O(\tilde{d}^r)$ instead of $O(\tilde{d}^{1/2})$, so again we can decrease the bound by a factor of $\tilde{d}^{r-1/2}$.

Then, with this critical $\tilde{d}^{r-1/2}$, we can prove the approximation theorem with the form

$$\sup_{t \geq 0} \left| f^{(t)}(\boldsymbol{x}) - f_{\text{lin}}^{(t)}(\boldsymbol{x}) \right| \leq C \tilde{d}^{r-1/2} \tag{5.129}$$

for any $r \in (0, 1/2)$, though we cannot really prove the $O(\tilde{d}^{-1/2})$ bound as originally claimed in (5.118). So this is how we solve Problem 2.

One caveat of changing the initialization to zero initialization is whether we can still safely assume that $\lambda^{\min} > 0$ where $\lambda^{\min}$ is the smallest eigenvalue of Θ, the kernel matrix of our new formulation. The answer is yes. In fact, in our Theorem 14 we proved that Θ is non-degenerated (which means that $\Theta(\boldsymbol{x}, \boldsymbol{x}')$ still depends on $\boldsymbol{x}$ and $\boldsymbol{x}'$), and under the overparameterized setting where $d_L \gg n$, chances are high that Θ is full-rank. Hence, we can still assume that $\lambda^{\min} > 0$.

As a final remark, one key reason why we need to initialize W^L as zero is that the dimension of the output space (i.e. the dimension of $\boldsymbol{h}^{L+1}$) is finite, and in our case it is 1. Suppose we allow the dimension of $\boldsymbol{h}^{L+1}$ to be $\tilde{d}$ which goes to infinity, then using the same proof techniques, for the NTK formulation we can prove that $\sup_t \left\|\boldsymbol{h}^{L+1(t)} - \boldsymbol{h}_{\text{lin}}^{L+1(t)}\right\|_2 \leq C$, i.e. the gap between two vectors of infinite dimension is always bounded by a finite constant. This is the approximation

theorem we need for the infinite-dimensional output space. However, when the dimension of the output space is finite, $\sup_t \left\| \boldsymbol{h}^{L+1(t)} - \boldsymbol{h}_{\text{lin}}^{L+1(t)} \right\|_2 \leq C$ no longer suffices, so we need to decrease the order of the norm of W^L in order to obtain a smaller bound.

5.9 Experiment details and additional experiments

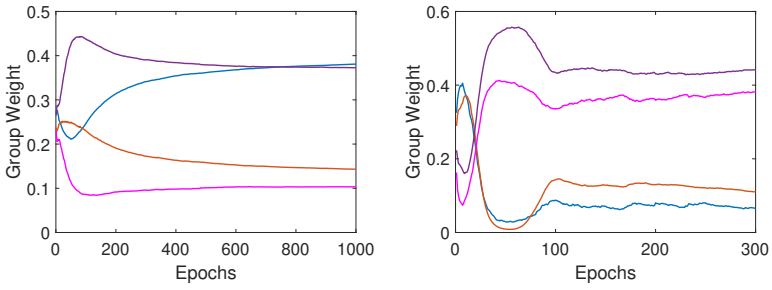
5.9.1 Experiment details

All experiments are conducted on a Ubuntu 18.04.6 machine with NVIDIA Geforce GTX 1080ti GPUs. Each model is trained with one GPU. On each of Waterbirds and CelebA, we use a ResNet18 as the model. The model is trained with SGD with momentum = 0.9. On Waterbirds the learning rate is 10^{-4}, and on CelebA it is 10^{-3}. For Group DRO, ν is selected as 0.01 (see the definition of ν in (5.3)). The batch size used for Waterbirds is 128, and for CelebA it is 400. Data augmentation including random cropping, random horizontal flip and normalization is performed on both datasets.

5.9.2 Sample weights converge in Group DRO

The results in Section 5.3 require Assumption 1 which states that each sample weight $q_i^{(t)}$ converges to some positive value as $t \to \infty$. Our readers might wonder how strong this assumption is, and whether reweighting algorithms satisfy this assumption in practice. In this section we empirically demonstrate that for Group DRO, the dynamic reweighting algorithm we experiment on, this assumption is satisfied on Waterbirds and CelebA.

Recall that in Section 5.2.2 we empirically showed that reweighting algorithms could easily overfit without regularization. Here using the same experimental settings, we keep track of the weight of each group g_k during training, and we plot the group weight curves in Figure 5.3. We also train the models longer (1000 epochs on Waterbirds and 300 epochs on CelebA). Clearly we can see that as the training accuracy converges to 100%, the group weights also converge to an equilibrium. Note that $q_i^{(t)} = g_k^{(t)}/n_k$ for all $z_i \in \mathcal{D}_k$, so the sample weights also converge.



(a) Waterbirds. (b) CelebA.

Figure 5.3: Weights of each group in Group DRO on Waterbirds and CelebA. The four curves correspond to the four groups.

Chapter 6

MSG: Margin Sensitive Group Risk

6.1 Introduction

Modern deep neural networks achieve high performance on a wide range of tasks, but they are also known to be non-robust to *distributional shift* where the training and test distributions can be different. A common instance of such distribution shift is *subpopulation shift*, where the training and test distributions have the same set of sub-populations but with different proportions. For example, many existing datasets are biased, in the sense that some underrepresented demographic groups have sample sizes much smaller than the others, and models trained on such datasets typically achieve high average performance but poor performances on these minority groups, which makes such models unfair. Similarly, many existing datasets have highly imbalanced classes, where some classes have much fewer training samples than others, and models trained on those datasets, while performing well on average, typically fail to learn these minority classes well enough.

To solve this problem, researchers have proposed a variety of methods. The most classical method is importance weighting [123], where the training samples are reweighted so that each group has the same weight in the training objective. Another widely used method is Distributionally Robust Optimization (DRO) [47, 58], which assumes that the test distribution belongs to a family of distributions close to the training distribution called the *uncertainty set*, and trains the model over the worst distribution in that set. Many DRO variants have been proposed recently [63, 155, 163, 165]. One of the most popular variants is Group DRO [117], which defines the uncertainty set as the convex hull of the group-conditionals of the training distribution.

Nevertheless, a line of recent work however has empirically and theoretically shown that these methods above do not necessarily perform better than standard empirical risk minimization (ERM). On the empirical side, [25] observed that the effect of importance weighting diminishes over time on CIFAR-10, and the final performance is close to ERM; [117] found that Group DRO overfits very easily, i.e. its worst-group performance drops to the same low level as ERM as training proceeds; [55, 78] demonstrated that these methods do not perform better than ERM on a variety of realistic tasks. On the theoretical side, based on several previous papers [56, 70, 151], a recent work [164] proved the surprising fact that under certain mild conditions, a broad family of algorithms called generalized reweighting (GRW), which includes all the popular methods mentioned above, has implicit bias equivalent to ERM on both regression and classification tasks,

implying that GRW does not improve distributionally robust generalization (DRG) over ERM. The sobering takeaway from this work is that these popular methods, while they might seem intuitive, do not really help with DRG.

A critical open problem facing the community is thus a principled way to improve DRG. There is some recent pioneering work towards this. [144] proposed to replace the exponentially-tailed logistic loss with a polynomially-tailed loss function, and [118, 148] showed that strong data augmentation, pretraining and semi-supervised learning could help DRG. One line of recent work [27, 76, 90, 98, 157] focuses on the *logit adjustment* technique, which applies a linear transformation to the logits output by a classifier to make it *have larger margins on smaller groups*. This line of work starts from [27] which proposed to add an additive adjustment term to the logits based on the margin theory, but [76] proved that the additive adjustment term only influences optimization and has no effect on the implicit bias, and thus does not improve DRG. Instead, they proposed to combine it with a multiplicative term, which they proved does affect the implicit bias. In their theoretical analysis, they showed that their method leads to robust models on a simple Gaussian Mixture model, but did not consider more general scenarios.

In this work, we propose to improve DRG by minimizing a margin sensitive group risk (MSG-risk) which we derive from the margin theory. Specifically, the margin theory provides a generalization bound for the test worst-group or balanced risk, and the MSG-risk is a surrogate of that generalization bound. Our method has two major improvements over previous work on logit adjustment: (i) Logit adjustment only considers the balanced risk, while our method can also handle the worst-group risk; (ii) Previous methods use fixed margins that are solely determined by the group sizes, and we show that this could be suboptimal. Instead, the MSG-risk takes the margins as *trainable* parameters, and searches for the optimal margins along with the model weights. In this way, our method can make the classifier have larger margins on *more difficult* groups, not just *smaller* groups.

The MSG-risk is a non-convex function *w.r.t.* the margins, and we propose two ways to minimize it: First, we can minimize it with alternating minimization, which can be used in a method called *post-hoc weight normalization* under the *domain-incomplete* setting; Second, we can directly minimize it with stochastic gradient descent (SGD), which is an increasingly standard approach to non-convex optimization. In our experiments, we show that our method achieves state-of-the-art (SOTA) robust test performance on real datasets, and fixes the overfitting problem which exists in many previous methods. We believe our approach therefore presents a substantial advance in both our theoretical and practical understanding of how to tune modern classifiers to improve DRG.

6.2 Preliminaries

6.2.1 Problem Formulation

Consider a classification task where the input space is $\mathcal{X} \subseteq \mathbb{R}^{d_0}$ and the output space is $\mathcal{Y} = \{1, -1\}$ (binary) or $\mathcal{Y} = \{1, \cdots, C\}$ (multi-class). We are given a training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ *i.i.d.* sampled from an underlying training distribution P over $\mathcal{X} \times \mathcal{Y}$. Moreover, the input domain $\mathcal{X}$ contains K groups (demographic groups, classes, etc.) where each sample can belong to zero,

one or more groups. Denote the conditional distribution of P over group k by P_k.

Balanced risk vs Worst-group risk. In most subpopulation shift problems, the goal is to minimize either the balanced risk $\mathcal{R}_{\text{bal}}$ or the worst-group risk $\mathcal{R}_{\max}$ defined as:

$$\mathcal{R}_{\text{bal}}(f; P) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{R}(f; P_k) \qquad \text{and} \qquad \mathcal{R}_{\max}(f; P) = \max_{k=1,\cdots,K}\mathcal{R}(f; P_k) \tag{6.1}$$

where $\mathcal{R}(f; P)$ is the expected risk of f over P. In practice the expected risk $\mathcal{R}(f; P)$ is replaced by the empirical risk $\hat{\mathcal{R}}(f; P)$, and the corresponding empirical balanced and worst-group risks are denoted by $\hat{\mathcal{R}}_{\text{bal}}(f; P)$ and $\hat{\mathcal{R}}_{\max}(f; P)$.

Domain-aware vs Domain-oblivious. If the group labels (i.e. which groups each sample belongs to) are known during training, then it is called the *domain-aware* setting, which is most widely studied. However, in many real applications this might not be the case, either because collecting group labels is expensive, or because we cannot identify all the groups at train time. For instance, we train a face recognition model that is fair *w.r.t.* sensitive features like gender, skin color, age and so on, but after using it for some time we observe that the model performs much worse for people wearing glasses than people who don't, so "wearing glasses" is a group that we fail to identify during training. Thus, a line of recent work [58, 93, 163] considers the *domain-oblivious* setting, where the group labels are unknown during training. The problem of this setting is that it is too pessimistic, and methods based on this setting typically have low performances in practice.

6.2.2 Domain-incomplete Setting and Post-hoc Weight Normalization

We saw that the domain-aware setting is not realistic in many real applications, while the domain-oblivious setting is too pessimistic. Thus, in this work, we study a third setting called the *domain-incomplete* setting which is very common and lies in between domain-aware and domain-oblivious.

Consider the following scenario: We have a trained model, say a face recognition model, that is trained by ERM or perhaps some robust training algorithm with some pre-defined groups. Then, at deployment stage we find that this model performs poorly on a certain group, say the group of people wearing glasses, and we need a "hot-fix" to our model. In this situation, we have no or incomplete group labels during training, but are provided with the complete group information at test time (without additional training samples), and the goal is to make the trained model robust as efficiently as possible. This is the domain-incomplete setting. Of course, we can still view this situation as a domain-aware problem and retrain or fine-tune the model with the new group added to the set of watched groups, but this could be very inefficient.

To deal with the domain-incomplete setting, we will use a method which is called *post-hoc weight normalization* in [76]. Suppose our model has the form $\boldsymbol{w} \circ \Phi(\boldsymbol{x})$, where Φ is a feature encoder and $\boldsymbol{w}$ is a linear classifier. In post-hoc weight normalization, we keep Φ fixed and find a new $\boldsymbol{w}$. This method can also be applied to representation learning, where Φ is learnt from

some self-supervised task, and we just need to find a $\boldsymbol{w}$. This method works as long as Φ encodes features that are robust to distributional shift. Since previous work [72] showed that ERM can learn sufficiently robust features, we can use the encoder trained by ERM as Φ, which is very effective in our experiments.

6.2.3 Generalized Logit Adjustment (GLA)

Now we introduce *logit adjustment* (LA) which motivates this work. The core idea of logit adjustment is to use a new loss function which makes a classifier have larger margins on smaller groups. Intuitively, the statistics of smaller groups are harder to estimate, so a model is prone to higher test error on these groups. Thus, we want there to be a larger margin between these groups and the decision boundary, which acts as a buffer that provides the model with better generalization. A couple of different losses have been proposed, and here we use a general formulation called *generalized logit adjustment loss* (GLA-loss) to cover all of them. Formally speaking, in GLA we apply a linear transformation to the logits output by the classifier before feeding them to the (weighted) original loss function. For example, for the logistic loss used in binary classification, its GLA-loss is defined as

$$\ell_{\text{GLA}}(f; \boldsymbol{x}, y) = q(\boldsymbol{x}, y) \log(1 + \exp\left[-y \cdot (\delta(\boldsymbol{x}, y) f(\boldsymbol{x}) + \tau(\boldsymbol{x}, y))\right]) \tag{6.2}$$

where $\delta(\boldsymbol{x}, y)$ is the multiplicative adjustment term, $\tau(\boldsymbol{x}, y)$ is the additive adjustment term, and $q(\boldsymbol{x}, y)$ is the sample weight. This general formulation covers all existing losses, including the LDAM-loss [27], the LA-loss [98], the CDT-loss [157] and the VS-loss [76].

For linear models, we can show that the only term in the GLA-loss that affects DRG is $\delta(\boldsymbol{x}, y)$:

Theorem 24. *Suppose $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$ is a linear model, the data is linearly independent, and $q(\boldsymbol{x}_i, y_i)$ and $\delta(\boldsymbol{x}_i, y_i)$ are positive for all i. If the model is trained by minimizing the average training GLA-loss under gradient descent with a sufficiently small learning rate, then we have $\|\boldsymbol{w}^{(t)}\|_2 \to \infty$ and*

$$\frac{\boldsymbol{w}^{(t)}}{\|\boldsymbol{w}^{(t)}\|_2} \to \hat{\boldsymbol{w}}_\delta = \underset{\|\boldsymbol{w}\|_2 = 1}{\operatorname{argmax}} \left\{ \min_{1 \le i \le n} \delta(\boldsymbol{x}_i, y_i) \cdot y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle \right\} \qquad \text{as } t \to \infty \tag{6.3}$$

See the proof in Appendix 6.7. This is an extension of Theorem 1 in [76]. Alternatively, we can write $\hat{\boldsymbol{w}}_\delta$ as the direction of the solution to the following cost-sensitive SVM [95]:

$$\begin{aligned} \underset{\boldsymbol{w}}{\text{minimize}} \quad & \|\boldsymbol{w}\|_2 \\ \text{s.t.} \quad & y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle \ge 1/\delta(\boldsymbol{x}_i, y_i), \quad \forall i \in [n] \end{aligned} \tag{6.4}$$

This result shows that losses that only include the additive term and the sample weight, such as LDAM-loss and LA-loss, do not really improve DRG. However, [76] showed that they do improve optimization, so they could lead to better models in practice with early stopping.

While GLA seems intuitive, there are still two remaining questions. First, the GLA-loss is used to minimize the balanced risk, so what about the worst-group risk? Second, Theorem 24 only shows that δ can *affect* DRG, so how to select δ so as to *improve* DRG?

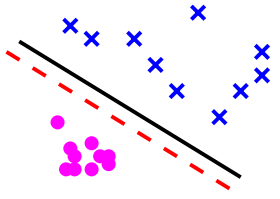


Figure 6.1: Sample task.

In all the losses mentioned above, δ and τ are fixed and only depend on the group sizes. This can be problematic. For example, in Figure 6.1 we have a sample binary classification task where the two classes have the same size. Suppose the groups are equivalent to the classes (as in class imbalance tasks), then all $\delta(\boldsymbol{x}_i, y_i)$ will be the same. By Theorem 24, the resulting linear classifier would be the SVM (black solid line). However, ideally we would like the decision boundary to be farther away from the blue crosses (red dashed line) because this group has a larger variance than the pink dots, which means that the classifier is more difficult to generalize on this group and needs a larger margin. Thus, margins that only depend on the group sizes could be suboptimal, and we need a way to find the optimal margins for each group. In other words, we not only need the classifier to have larger margins on *smaller* groups, but also on *more difficult* groups.

6.3 MSG: Margin Sensitive Group Risk

In this section, we propose a new objective function called the margin sensitive group risk (MSG-risk), which is motivated by the GLA-loss and is based on margin theory, and thus is a principled way to improve DRG. First, we derive the MSG-risk from the margin theory. Then we show two ways to minimize this non-convex objective: (i) Alternating minimization, which we will apply to post-hoc weight normalization under the domain-incomplete setting; (ii) Direct SGD, which we will use to train a neural network end-to-end under the domain-aware setting.

6.3.1 Derivation of the MSG-Risk

First, we derive the MSG-risk from the margin theory. For binary classification where the prediction is given by the sign of the model output $\hat{y}$, define the *ρ-margin loss* as

$$\ell_\rho(\hat{y}, y) = \phi_\rho(\hat{y}y) := \begin{cases} 1, & \hat{y}y < 0 \\ 1 - \frac{\hat{y}y}{\rho}, & 0 \le \hat{y}y \le \rho \\ 0, & \hat{y}y > \rho \end{cases} \tag{6.5}$$

A key advantage of this loss is that it is margin sensitive, as well as $1/\rho$-Lipschitz, which thus allows one to derive generalization bounds *w.r.t.* the margin. Denote the empirical ρ-margin risk of hypothesis h by $\hat{\mathcal{R}}_\rho(h)$. Denote the expected zero-one loss of h over the underlying distribution P by $\mathcal{R}^{0/1}(h)$. Then we have the following generalization bound:

Theorem 25 (Theorem 5.9 and 5.10 in [101])**.** *Let $\Phi : \mathcal{X} \to \mathbb{R}^d$ be a feature mapping. Let the hypothesis set be $\mathcal{H} = \{\boldsymbol{x} \mapsto \langle \boldsymbol{w}, \Phi(\boldsymbol{x})\rangle : \|\boldsymbol{w}\|_2 \le \Lambda\}$ and the input space be $\mathcal{X} \subseteq \{\boldsymbol{x} : \|\Phi(\boldsymbol{x})\|_2 \le r\}$. Let $M > 0$ be fixed. Then for any $\delta > 0$, with probability at least $1 - \delta$ over the random sampling of the training set, the following holds for all $h \in \mathcal{H}$ and all $\rho \in (0, M]$:*

$$\mathcal{R}^{0/1}(h) \le \hat{\mathcal{R}}_\rho(h) + 3\sqrt{\frac{\log \frac{4}{\delta}}{2n}} + \frac{4}{\rho}\sqrt{\frac{r^2\Lambda^2}{n}} + \sqrt{\frac{\log \log_2 \frac{2M}{\rho}}{n}} \tag{6.6}$$

Denote the expected zero-one loss of h over group k by $\mathcal{R}_k^{0/1}(h)$, and the empirical ρ-margin risk over group k by $\hat{\mathcal{R}}_{\rho,k}(h)$. Suppose group k has n_k training samples. Then by union bound we have:

Corollary 26. *Let $\Phi : \mathcal{X} \to \mathbb{R}^d$ be a feature mapping. Let the hypothesis set be $\mathcal{H} = \{\boldsymbol{x} \mapsto \langle \boldsymbol{w}, \Phi(\boldsymbol{x})\rangle : \|\boldsymbol{w}\|_2 \leq \Lambda\}$ and the input space be $\mathcal{X} \subseteq \{\boldsymbol{x} : \|\Phi(\boldsymbol{x})\|_2 \leq r\}$. Let $M > 0$ be fixed. Then for any $\delta > 0$, with probability at least $1 - \delta$ over the random sampling of the training set, the following holds for all $h \in \mathcal{H}$ and all $\rho_1, \cdots, \rho_K \in (0, M]$:*

$$\mathcal{R}_k^{0/1}(h) \leq \hat{\mathcal{R}}_{\rho_k,k}(h) + 3\sqrt{\frac{\log \frac{4K}{\delta}}{2n_k}} + \frac{4}{\rho_k}\sqrt{\frac{r^2\Lambda^2}{n_k}} + \sqrt{\frac{\log\log_2 \frac{2M}{\rho_k}}{n_k}}, \quad \forall k \in [K] \tag{6.7}$$

Suppose ρ_k is bounded below away from zero. Then, the last term on the right hand side of (6.7) becomes much smaller than the other terms and can be ignored. Thus, we define the MSG-risk as the following surrogate of the generalization bound:

$$\hat{\mathcal{R}}_{\alpha,\beta}(h; \rho_1, \cdots, \rho_K) = \max_{k\in[K]} \left[\frac{1}{n_k}\sum_{i=1}^{n_k} \tilde{\ell}_{\rho_k}(h(\boldsymbol{x}_{k,i}), y_{k,i}) + \frac{1}{\sqrt{n_k}}\left(\alpha + \beta\frac{\|\boldsymbol{w}\|_2}{\rho_k}\right)\right] \tag{6.8}$$

where α, β are some non-negative constants given by (6.7) (which we view as hyperparameters), and $\tilde{\ell}_{\rho_k}$ is a surrogate loss function of ℓ_{ρ_k} such that $\ell_{\rho_k}(\hat{y}, y) \leq \tilde{\ell}_{\rho_k}(\hat{y}, y)$ (we need this because ℓ_{ρ_k} is not convex). Eqn. (6.8) is a surrogate of the worst-group risk, and if we want to minimize the balanced risk, we only need to replace $\max_{k\in[K]}$ with $\sum_{k\in[K]}$ in (6.8) (in which case α makes no difference and can be set to 0). Note that in previous GLA-losses, the margins $\rho_1, \cdots, \rho_K$ are fixed, but the MSG-risk takes them as trainable parameters and optimizes them too.

For multi-class classification where the output of $h(\boldsymbol{x})$ is a logit vector in $\mathbb{R}^C$, we have similar results. Define the ρ-margin loss as

$$\ell_\rho(h(\boldsymbol{x}), y) = \phi_\rho(h_y(\boldsymbol{x}) - \max_{y'\neq y} h_{y'}(\boldsymbol{x})) \tag{6.9}$$

where ϕ_ρ is given by (6.5). Then we have the following generalization bound:

Theorem 27 (Corollary 9.4 in [101])**.** *Let $\Phi : \mathcal{X} \to \mathbb{R}^d$ be a feature mapping. Let $\mathcal{H} = \{\boldsymbol{x} \to \boldsymbol{w} \circ \Phi(\boldsymbol{x}) : \boldsymbol{w} \in \mathbb{R}^{C\times d}, \|\boldsymbol{w}\|_F \leq \Lambda\}$ and $\mathcal{X} \subseteq \{\boldsymbol{x} : \|\Phi(\boldsymbol{x})\|_2 \leq r\}$. Let $M > 0$ be fixed. Then for any $\delta > 0$, with probability at least $1 - \delta$ over the random sampling of the training set, for all $h \in \mathcal{H}$ and all $\rho \in (0, M]$ we have:*

$$\mathcal{R}(h) \leq \hat{\mathcal{R}}_\rho(h) + 3\sqrt{\frac{\log \frac{4}{\delta}}{2n}} + \frac{4C}{\rho}\sqrt{\frac{r^2\Lambda^2}{n}} + \sqrt{\frac{\log\log_2 \frac{2M}{\rho}}{n}} \tag{6.10}$$

Again, by union bound we can get a generalization bound similar to Corollary 26. Thus, the MSG-risk for multi-class classification is still defined as (6.8), with $\tilde{\ell}_{\rho_k}$ defined as a surrogate of (6.9).

The main issue with the MSG-risk is that it is non-convex *w.r.t.* $\rho_1, \cdots, \rho_K$, which means that we need to use some non-convex optimization methods to minimize it. In the following sections we will introduce two approaches that work under different settings.

6.3.2 Alternating Minimization for the Domain-incomplete Setting

In post-hoc weight normalization, we have a pretrained feature mapping Φ and only need to find an optimal linear classifier $h(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle$ that minimizes the worst-group or balanced risk. First, we formulate the task of minimizing (6.8) as the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{w},s,\tau,\rho}{\text{minimize}} \quad & s \quad (\text{or } \sum_{k\in[K]} s_k) \\ \text{s.t.} \quad & \tau_{k,i} \geq \tilde{\ell}_{\rho_k}(\langle \boldsymbol{w}, \Phi(\boldsymbol{x}_{k,i}) \rangle, y_{k,i}), \quad \forall k \in [K], \forall i \in [n_k] \\ & \frac{1}{n_k} \sum_{i=1}^{n_k} \tau_{k,i} + \frac{1}{\sqrt{n_k}} \left(\alpha + \beta \frac{\|\boldsymbol{w}\|_2}{\rho_k} \right) \leq s \quad (\text{or } s_k), \quad \forall k \in [K] \\ & \rho_k \geq 0, \quad \forall k \in [K] \end{aligned} \tag{6.11}$$

In our implementation, we choose $\tilde{\ell}_\rho$ to be the hinge loss. For example, for binary classification this is $\tilde{\ell}_\rho(\hat{y}, y) := \left(1 - \frac{\hat{y}y}{\rho}\right)_+ = \max\left\{1 - \frac{\hat{y}y}{\rho}, 0\right\}$. Let $\delta_k = \frac{1}{\rho_k}$. With this hinge loss, the optimization problem above can then be re-written as:

$$\begin{aligned} \underset{\boldsymbol{w},s,\tau,\delta}{\text{minimize}} \quad & s \quad (\text{or } \sum_{k\in[K]} s_k) \\ \text{s.t.} \quad & \begin{cases} \text{Binary: } \delta_k \langle \boldsymbol{w}, y_{k,i} \boldsymbol{z}_{k,i} \rangle \geq 1 - \tau_{k,i}, & \forall k \in [K], \forall i \in [n_k] \\ \text{Multi-class: } \delta_k \langle \boldsymbol{w}_{y_{k,i}} - \boldsymbol{w}_{y'}, \boldsymbol{z}_{k,i} \rangle \geq 1 - \tau_{k,i}, & \forall y' \neq y_{k,i}, \forall k, \forall i \end{cases} \\ & \frac{1}{n_k} \sum_{i=1}^{n_k} \tau_{k,i} + \frac{1}{\sqrt{n_k}} (\alpha + \beta \delta_k \|\boldsymbol{w}\|_2) \leq s \quad (\text{or } s_k), \quad \forall k \in [K] \\ & \delta_k \geq 0, \tau_{k,i} \geq 0, \quad \forall k \in [K], \forall i \in [n_k] \end{aligned} \tag{6.12}$$

where the first constraint is either one of the two depending on whether it is binary or multi-class classification, and $\boldsymbol{z}_{k,i} = \Phi(\boldsymbol{x}_{k,i})$. This is a bilinear non-convex optimization problem, which is hard to solve in general. Here we solve this problem with *alternating minimization*, a heuristic method that is widely used in matrix completion. Generally speaking, we train the model for several iterations, and for each iteration, we first fix δ and solve (6.12) *w.r.t.* $\boldsymbol{w}, s, \tau$, and then fix $\boldsymbol{w}$ and solve (6.12) *w.r.t.* s, τ, δ. Note that if $\alpha = 0$ and $\beta = \infty$, then the optimal solution of δ_k is $\delta_k \propto n_k^{1/2}$, so we can initialize δ_k in this way. Comparing (6.8) to (6.7), we can see that α and β depend on r, the maximum norm of the features. Thus, we always normalize the features before solving this optimization problem.

The algorithm for post-hoc weight normalization is listed in Algorithm 5, and in our implementation Φ is trained by ERM. Since this method works under the domain-incomplete setting which is in between domain-aware and domain-oblivious, naturally we would expect the performance of Algorithm 5 to be in between the performances of domain-aware and domain-oblivious methods. What is surprising, however, is that we find in our experiments that the performance of Algorithm 5 is always close to, and in many occasions even better than, the performances of state-of-the-art domain-aware methods. Moreover, given a pretrained encoder, our method only takes minutes to run on a CPU compared to domain-aware methods which typically take hours on a GPU.

Algorithm 5 Post-hoc Weight Normalization with Alternating Minimization

Require: Training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, pretrained encoder Φ, hyperparameters α, β, iterations T

1: Build features for the training samples: $\boldsymbol{z}_i = \Phi(\boldsymbol{x}_i)$ for all i
2: Input normalization: $\tilde{\boldsymbol{z}}_i = \frac{\boldsymbol{z}_i - \mu}{\sigma\sqrt{d}}$, where μ and σ are the mean and standard deviation of $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n$, and d is the dimension of the feature space.
3: Initialization: $\delta_k \propto n_k^{1/2}$ for $k \in [K]$
4: **for** $t = 1, \cdots, T$ **do**
5: Fix δ and solve (6.12) *w.r.t.* $\boldsymbol{w}, s, \tau$ with $\{(\tilde{\boldsymbol{z}}_i, y_i)\}_{i=1}^n$ as input
6: Fix $\boldsymbol{w}$ and solve (6.12) *w.r.t.* s, τ, δ with $\{(\tilde{\boldsymbol{z}}_i, y_i)\}_{i=1}^n$ as input
7: **end for**
8: **return** the final model $f(\boldsymbol{x}) = \boldsymbol{w}^* \cdot \frac{\Phi(\boldsymbol{x}) - \mu}{\sigma\sqrt{d}}$ where $\boldsymbol{w}^*$ is the solution found

6.3.3 End-to-end Training with Stochastic Gradient Descent for the Domain-aware Setting

The second approach is directly minimizing the non-convex MSG-risk (6.8) with SGD *w.r.t.* h and $\rho_1, \cdots, \rho_K$ jointly, so that we can train a neural network end-to-end under the domain-aware setting. Define $\delta_k = \rho_k^{-1}$. We make $\delta_1, \cdots, \delta_K$ trainable parameters, so that it can be optimized together with the weights of h (with different learning rates). More specifically, in our implementation, we choose the surrogate loss of ℓ_ρ to be the logistic loss which is most widely used in practice:

$$\ell_\rho(\hat{y}, y) \leq C \log\left(1 + \exp(-\frac{\hat{y}y}{\rho})\right) := C\tilde{\ell}_\rho(\hat{y}, y) \tag{6.13}$$

for some constant C which we ignore in the implementation. And the final objective function is

$$\hat{\mathcal{R}}_{\alpha,\beta}(h, \delta) = \max_{k \in [K]} \left[\frac{1}{n_k} \sum_{i=1}^{n_k} \log\left(1 + \exp(-\delta_k h(\boldsymbol{x}_{k,i}) y_{k,i})\right) + \frac{1}{\sqrt{n_k}} \left(\alpha + \beta \delta_k \|h\|_2\right)\right] \tag{6.14}$$

where $\|h\|_2$ is defined as the L_2-norm of all its weights when flattened as a 1-dimensional vector. We update the model weights and δ by minimizing this objective with stochastic gradient descent (SGD). Moreover, after each update, we normalize δ so that $\delta_1 + \cdots + \delta_K$ is fixed.

Previous work [25, 117] showed that existing methods like importance weighting and Group DRO suffer from overfitting: At the early stage of training the model trained with these methods can achieve high worst-group or balanced test performance, but the performance gradually drops to the same low level as ERM as training proceeds. On the contrary, our method fixes the overfitting problem as we will show in our experiments, so it does improve DRG over existing methods.

6.4 Experiments

6.4.1 Setup

Datasets. For tasks that require maximizing the worst-group performance, we follow [163] and use two datasets: CelebA and CivilComments-Wilds. For tasks that require maximizing the

balanced performance, we follow [76] and use class-imbalanced CIFAR-10 with two types of class imbalance: Long-Tail (LT) imbalance and Step imbalance. CelebA and CivilComments-Wilds have official train-validation-test splits. For CIFAR-10, we randomly split the test set into two halves, making one the validation set and the other one the test set. See Appendix 6.6.1 for more details.

Models and baselines. Following prior work, we use a wide ResNet-18 for CelebA, a Bert-base-uncased model for CivilComments-Wilds, and a wide ResNet-32 for the imbalanced CIFAR-10 datasets. For the fairness tasks, we compare our method with two baselines: the SOTA domain-oblivious method CVaR and the SOTA domain-aware methods importance weighting (IW) and Group DRO (GDRO). For the class imbalance tasks, since they cannot be domain-oblivious, we only compare with ERM and SOTA domain-aware methods (IW, LDAM-DRW, CDT, LA and VS, see [76] for a summary of these methods). For each method except ours, we train the model for a fixed number of epochs (100 for CelebA, 5 for CivilComments-Wilds and 300 for CIFAR datasets), and select the one with the highest validation worst-group/balanced accuracy. Note that here the "domain-oblivious methods" are not 100% domain-oblivious because we use the group labels during validation (as [163] pointed out, model selection without group labels is too hard, and currently no method is better than ERM). We use the original code from previous work whenever available.

Implementation. In post-hoc weight normalization, we solve convex optimization with MOSEK [3], a commercial optimizer. To train Φ, we run ERM for a fixed number of epochs (as detailed above), and take the encoder of the checkpoint at the end of training as Φ (so that Φ is *completely domain-oblivious*). We only use the training set but not the validation set, while all other methods use the validation set to select the best model. For end-to-end training with SGD, we combine it with importance weighting. See Appendix 6.6.2 for details on hyperparameters.

6.4.2 Results

First, we evaluate the performance of post-hoc weight normalization with alternating minimization. In Tables 6.1 and 6.2 we report the worst-group/balanced test accuracy achieved by post-hoc weight normalization as well as previous methods with different feature space dimensions, which we control by changing the width of the network. For alternating minimization, we always run 10 iterations. Overall, we observe that using the MSG-risk in post-hoc weight normalization achieves performances that are close to, and in many occasions even better than, the performances of state-of-the-art domain-aware methods. This lends additional credence to the emerging empirical understanding [72] that ERM learns sufficiently robust feature encoders. Also note that our method is a very efficient method which only takes minutes to run on a CPU given a pretrained encoder, whereas retraining from scratch takes hours on a GPU. Regarding the effect of the feature dimension, we find that increasing the feature dimension does not always provide higher performances. On CIFAR-10 the performances are higher for all methods when the feature dimension increases, but not so for CelebA.

Table 6.1: Results for CelebA and CivilComments-Wilds. Each experiment is run with 5 different random seeds, and the mean and std. dev. of the worst-group test accuracies (%) are reported.

Dataset	CelebA			CivilComments
Feature Dim	64	128	256	784
ERM	42.44 ± 4.72	46.89 ± 2.96	42.78 ± 5.02	58.74 ± 2.94
CVaR	61.29 ± 7.64	66.07 ± 3.69	71.25 ± 1.96	63.90 ± 4.64
IW	87.00 ± 1.45	86.89 ± 2.47	87.67 ± 1.44	68.05 ± 1.12
Group DRO	85.67 ± 1.49	85.60 ± 2.48	83.67 ± 2.47	68.34 ± 2.40
MSG (Alg. 5)	$\mathbf{87.12 \pm 0.69}$	$\mathbf{87.70 \pm 1.59}$	$\mathbf{87.94 \pm 0.39}$	$\mathbf{71.67 \pm 1.12}$

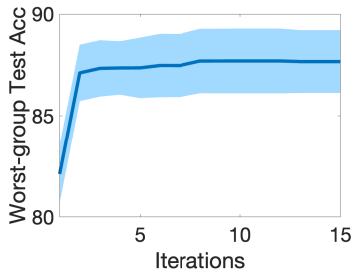
Table 6.2: Results for CIFAR-10 with Long-Tail or Step class imbalance ("-100" means that the size of the largest class is 100 times that of the smallest). Each experiment is run with 5 different random seeds, and the mean and std. dev. of the balanced average test accuracies (%) are reported.

Dataset	CIFAR-10 (LT-100)		CIFAR-10 (STEP-100)	
Feature Dim	64	128	64	128
ERM	72.28 ± 0.44	74.62 ± 0.41	66.00 ± 2.00	68.12 ± 1.46
IW	72.94 ± 0.94	73.92 ± 1.29	68.33 ± 1.27	69.41 ± 1.26
LDAM-DRW	77.30 ± 0.53	78.56 ± 0.45	78.06 ± 0.69	78.65 ± 0.75
CDT	79.87 ± 1.04	81.64 ± 0.36	75.93 ± 0.86	77.76 ± 0.31
LA	$\mathbf{80.68 \pm 0.69}$	82.57 ± 0.53	76.47 ± 0.25	80.58 ± 0.96
VS	80.48 ± 0.49	82.76 ± 0.34	$\mathbf{79.67 \pm 0.64}$	$\mathbf{82.06 \pm 0.65}$
MSG (Alg. 5)	80.44 ± 0.50	$\mathbf{82.78 \pm 0.47}$	78.62 ± 0.95	80.57 ± 0.72

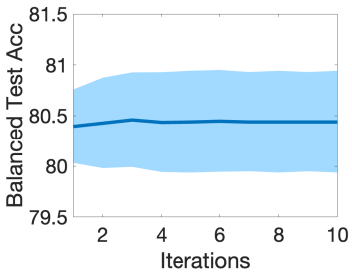
Second, we study the convergence rate of alternating minimization. We run Algorithm 5 with different numbers of iterations on CelebA and imbalanced CIFAR-10 (LT-100) and plot the results in Figure 6.2. We can see that alternating minimization converges very quickly: It reaches the optimal point after around 8 iterations on CelebA and around 3 iterations on CIFAR-10 (LT-100).

Third, we evaluate the performance of end-to-end training with SGD. We run our method together with importance weighting and group DRO on CelebA and plot the worst-group test accuracies achieved during training in Figure 6.3. The plot clearly shows that the performances of importance weighting and group DRO are high at the early stage of training, but as the training proceeds they will start to drop at some point, while the performance of our method continues to rise and maintains high, which implies that using the MSG-risk fixes the overfitting problem of previous methods.

Finally, we investigate what factors decide the optimal margins found by minimizing the MSG-risk. First, we look at the optimal δ_k found by post-hoc weight normalization with alternating minimization and report them in Table 6.3. Note that a smaller δ_k means a larger margin for that group. On CIFAR-10, we can see that as expected smaller groups have smaller δ_k, *i.e.* larger margins. However, to our surprise, on CelebA the smaller groups have larger δ_k, *i.e.* smaller



(a) CelebA.



(b) CIFAR-10 (LT-100).

Figure 6.2: The convergence rate of alternating minimization in post-hoc weight normalization.

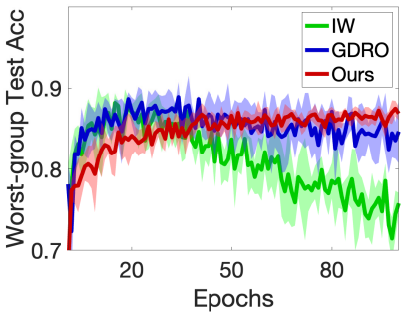


Figure 6.3: End-to-end training with SGD on CelebA.

Table 6.3: The optimal δ_k ($= \rho_k^{-1}$) found by post-hoc weight normalization with alternating minimization. Each experiment is run 5 times.

CIFAR-10 (LT-100)			CelebA		
n_k	$\delta_k = \rho_k^{-1}$	Feat. Stdev.	n_k	$\delta_k = \rho_k^{-1}$	Feat. Stdev.
5000	9.11 ± 0.33	2.37 ± 0.08	1387	28.59 ± 1.69	2.58 ± 0.15
2997	6.93 ± 0.40	2.53 ± 0.13	66874	18.59 ± 1.27	2.94 ± 0.07
83	1.96 ± 0.06	2.73 ± 0.16	22880	26.54 ± 4.52	2.76 ± 0.28
50	1.52 ± 0.05	3.00 ± 0.24	71629	15.72 ± 1.34	3.08 ± 0.09

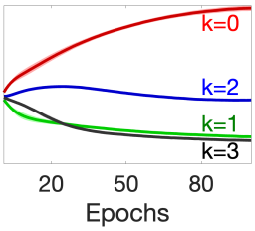


Figure 6.4: δ_k in end-to-end training on CelebA.

margins (even though at initialization their δ_k is smaller). The same phenomenon is observed in end-to-end training. We plot the change of δ_k in end-to-end training on CelebA in Figure 6.4, which shows that the smallest group has the largest δ_k while the two majority groups have small δ_k. We hypothesize that this is because the two majority groups, while larger, are harder to fit. To verify this, we compute the L_2-norm of the standard deviation of the feature vector $\Phi(\boldsymbol{x}_{k,i})$ over i for each group k, and report the results in Table 6.3 (column "Feat. Stdev."). We can see that this quantity aligns with the optimal margin perfectly: Groups with larger feature variances have smaller δ_k, *i.e.* larger margins. Therefore, although the feature variance is not a perfect measure of how hard it is to fit a group, our results show that our method does make the classifier have larger margins on more difficult groups as we desire.

6.5 Conclusion

In this work, we studied the difficult yet important problem of how to improve distributionally robust generalization (DRG), and proposed the MSG-risk that is derived from the margin theory. We used two ways to minimize the MSG-risk: alternating minimization which can be used in post-hoc weight normalization for the domain-incomplete setting, and direct SGD which can be used to train a neural network end-to-end for the domain-aware setting. Our experiments showed that the MSG-risk leads to state-of-the-art robust performance, and thus we believe this work

substantially advances our theoretical and practical understanding of how to improve DRG.

There are two remaining questions from this work: (i) In our experiments we find the optimal α and β via grid search, but can we integrate them into the objective and optimize them jointly? (ii) The margin theory only considers linear models, and for neural networks we replace $\|\boldsymbol{w}\|_2$ with $\|h\|_2$, which lacks a solid theoretical foundation, so can we extend the margin theory to obtain generalization bounds for more complex models? We leave these problems to future work.

6.6 More Experimental Details

6.6.1 Datasets

In our experiments, we use three datasets: CelebA, CivilComments-Wilds and CIFAR-10 with long-tail or step class imbalance.

CelebA [94] is a human face image dataset, where each sample is an image of a human face and has 40 binary attributes. Following [117], we take the blond attribute as the target and the male attribute as the confounding variable. We need to train a classifier to classify whether a person is blond or not, and the two binary attributes form four groups. In this dataset, most males are not blond, so a model trained with ERM would classify most males as not blond, meaning that it would have a poor performance on the male and blond group, while our goal is to train a model that performs well on all four groups.

CivilComments-Wilds is one of the datasets in the Wilds package [78] and is based on CivilComments [20]. It is a language sentiment dataset, where each sample is an online text comment and the label is whether the comment was rated as toxic. There are 8 demographic identities considered: male, female, LGBTQ, Christian, Muslim, other religions, black and white. These 8 binary attributes together with the binary label form 16 groups. Note that a sample can appear in multiple groups: a comment can contain contents of both LGBTQ and Christian.

The CIFAR-10 dataset [80] is a image dataset with 10 classes. In the original dataset, each class has 5000 training samples. To make the classes imbalanced, we have two methods: The Long-Tail (LT) method makes the sizes of the classes decrease exponentially, and the Step method keeps 5 classes unchanged and remove a equal number of samples from each of the other 5 classes. In our experiments we consider LT-100 and Step-100, meaning that the size of the biggest group is 100 times that of the smallest one. We randomly remove training samples with a fixed random seed.

6.6.2 Training Hyperparameters

Alternating minimization. For CelebA, following [117], we use the learning rate 10^{-4} for every method. We use a weight decay factor of 10^{-4} for ERM, CVaR and importance weighting, and 0.1 for group DRO. For CVaR, we always use $\alpha = 0.1$ (note that this is the α for CVaR, not the one in the MSG-risk). For group DRO, we use $\nu = 0.01$. For each of the above methods, we train 100 epochs and select the one with the highest worst-group validation accuracy. For post-hoc weight normalization, we choose Φ to be the feature encoder at the end of the 100 epochs. Regarding the hyperparameters in the MSG-risk, for the feature dimensions 64 and 128, we use

$\alpha = 4.0$ and $\beta = 0.02$; and for the feature dimension 256, we use $\alpha = 7.0$ and $\beta = 0.3$. The optimal α and β are found by grid search.

For CivilComments-Wilds, for all methods, we use the same hyperparameters as in [78] and [163]. Regarding the hyperparameters in the MSG-risk, we use $\alpha = 16.0$ and $\beta = 2.0$.

For class imbalanced CIFAR-10, for all methods, we use the same hyperparameters as in [76] except that we train 300 epochs and perform learning rate decay at epochs 220 and 280[1]. Regarding the hyperparameters in the MSG-risk, we use $\alpha = 0$ and $\beta = 2.0$ (note that α does not matter in class imbalance tasks).

End-to-end training with SGD. In Figure 6.3, in order to make the models overfit faster, we use a larger learning rate 10^{-3}. Under the original learning rate, the models would still overfit, but much slower. For the MSG-risk, we use $\alpha = 4.0$ and $\beta = 0.02$.

6.7 Proof of Theorem 24

A generalized reweighting (GRW) algorithm minimizes the following objective at time t:

$$\hat{\mathcal{R}}_{\boldsymbol{q}^{(t)}}(f) = \sum_{i=1}^{n} q_i^{(t)} \ell(f(\boldsymbol{x}_i), y_i) \tag{6.15}$$

for some loss function ℓ. And a first-order differentiable function f is called L-smooth over $\mathcal{D}$ if

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{D} \tag{6.16}$$

The proof of this theorem is based on the following theorem:

Theorem 28 (Theorem 8 in [164])**.** *Consider a linear model $f(\boldsymbol{x}) = \langle \theta, \boldsymbol{x} \rangle$. For any loss ℓ that is convex, differentiable, L-smooth in $\hat{y}$ and strictly monotonically decreasing to zero as $y\hat{y} \to +\infty$, and any GRW such that $q_i^{(t)} \to q_i$ as $t \to \infty$ for some positive $q_1, \cdots, q_n$, denote*

$$F(\theta) = \sum_{i=1}^{n} q_i \ell(\langle \theta, \boldsymbol{x}_i \rangle, y_i) \tag{6.17}$$

If $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ are linearly independent, and f is trained under gradient descent with a sufficiently small learning rate η, then we will have the following results for the model weight $\theta^{(t)}$:

1. *$F(\theta^{(t)}) \to 0$ as $t \to \infty$.*
2. *$\left\|\theta^{(t)}\right\|_2 \to \infty$ as $t \to \infty$.*
3. *Define $\theta_R = \operatorname{argmin}_\theta \{F(\theta) : \|\theta\|_2 \leq R\}$. For any R such that $\min_{\|\theta\|_2 \leq R} F(\theta) < \min_i q_i \ell(0, y_i)$, θ_R is unique. And if $\lim_{R\to\infty} \frac{\theta_R}{R}$ exists, then $\lim_{t\to\infty} \frac{\theta^{(t)}}{\left\|\theta^{(t)}\right\|_2}$ also exists and the two limits are equal.*

[1] In the original paper, the authors train 200 epochs for LT-100 and 300 epochs for STEP-100. In our experiments, we train 300 epochs for both for consistency. Moreover, the original paper does not mention how to decay the learning rate when training for 300 epochs, and we confirmed with the authors that they decayed the learning rate at epochs 220 and 280 to achieve their results on STEP-100.

Denote $\boldsymbol{x}_i' = \delta(\boldsymbol{x}_i, y_i)\boldsymbol{x}_i$, then $\boldsymbol{x}_1', \cdots, \boldsymbol{x}_n'$ are linearly independent because $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ are linearly independent, so for any τ there exists θ_0 such that $\langle \theta_0, \boldsymbol{x}_i' \rangle = \tau(\boldsymbol{x}_i, y_i)$ for all i. Denote $\theta = \boldsymbol{w} + \theta_0$ where $\boldsymbol{w}$ is the weight of the original model, then the GLA-loss can be rewritten as

$$\ell_{\text{GLA}}(f; \boldsymbol{x}_i, y_i) = q(\boldsymbol{x}_i, y_i)\ell_{\log}(\langle \theta, \boldsymbol{x}_i' \rangle, y_i) = q(\boldsymbol{x}_i, y_i)\log(1 + \exp(-y_i\langle \theta, \boldsymbol{x}_i' \rangle)) \tag{6.18}$$

where $\ell_{\log}$ is the logistic loss. It is easy to show that $\ell_{\log}$ satisfies the conditions of Theorem 28, and by condition $q(\boldsymbol{x}_i, y_i) > 0$ for all i. Also we can see that $\theta^{(t)}$ is trained under gradient descent with $\boldsymbol{x}_1', \cdots, \boldsymbol{x}_n'$ as inputs. Thus, by Theorem 28, we have as $t \to \infty$, $\|\theta^{(t)}\|_2 \to \infty$. And for the logistic loss, we can show that (as shown in Appendix B.5.3 in [164])

$$\lim_{R\to\infty} \frac{\theta_R}{R} = \underset{\|\theta\|_2=1}{\operatorname{argmax}} \left\{ \min_{1\le i\le n} y_i \langle \theta, \boldsymbol{x}_i' \rangle \right\} \tag{6.19}$$

Therefore, by (iii) and the definition of $\boldsymbol{x}_i'$ and θ, we can see that

$$\lim_{t\to\infty} \frac{\boldsymbol{w}^{(t)}}{\|\boldsymbol{w}^{(t)}\|_2} = \underset{\|w\|_2=1}{\operatorname{argmax}} \left\{ \min_{1\le i\le n} y_i \langle \boldsymbol{w}, \delta(\boldsymbol{x}_i, y_i)\boldsymbol{x}_i \rangle \right\} \tag{6.20}$$

which is the result we need. □

Bibliography

[1] Allen-Zhu, Z., Li, Y., and Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR. 1.1.3, 5.1.1

[2] Angelidakis, H., Awasthi, P., Blum, A., Chatziafratis, V., and Dan, C. (2019). Bilu-linial stability, certified algorithms and the independent set problem. In *27th Annual European Symposium on Algorithms (ESA 2019)*, volume 27. 1.3

[3] ApS, M. (2022). *MOSEK Optimizer API for Python. Version 9.3.17*. 6.4.1

[4] Aragam, B., Dan, C., Xing, E. P., and Ravikumar, P. (2020). Identifiability of nonparametric mixture models and bayes optimal clustering. *The Annals of Statistics*, 48(4):2277–2302. 1.3

[5] Attias, I., Kontorovich, A., and Mansour, Y. (2018). Improved generalization bounds for robust learning. *arXiv preprint arXiv:1810.02180*. 1.1.1, 2.1.2

[6] Aubin, B., Krzakala, F., Lu, Y. M., and Zdeborová, L. (2020). Generalization error in high-dimensional perceptrons: Approaching bayes error with convex optimization. *arXiv preprint arXiv:2006.06560*. 4.2.2

[7] Awasthi, P., Dutta, A., and Vijayaraghavan, A. (2019). On robustness to adversarial examples and polynomial optimization. In *Advances in Neural Information Processing Systems*, pages 13760–13770. 2.1.2

[8] Awasthi, P., Frank, N., and Mohri, M. (2020). Adversarial learning guarantees for linear hypotheses and neural networks. *arXiv preprint arXiv:2004.13617*. 1.1.1, 2.1.2

[9] Azizyan, M., Singh, A., and Wasserman, L. (2013). Minimax theory for high-dimensional gaussian mixtures with sparse mean separation. In *Advances in Neural Information Processing Systems*, pages 2139–2147. 2.1.2

[10] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*. 1, 2.1

[11] Balakrishnan, S., Wainwright, M. J., and Yu, B. (2017). Statistical guarantees for the em algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120. 4.1.3

[12] Barocas, S. and Selbst, A. D. (2016). Big data's disparate impact. *Calif. L. Rev.*, 104:671. 3.1

[13] Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. (2020). Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070. 4.1.3

[14] Belkin, M. (2021). Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248. 1.1.3

[15] Bhagoji, A. N., Cullina, D., and Mittal, P. (2019). Lower bounds on adversarial robustness from optimal transport. In *Advances in Neural Information Processing Systems*, pages 7496–7508. 2.1, 2.1.2, 2.2, 2.2.1, 1, 2.3, 2.4, 2.6.2, 2.6.2, 2.7

[16] Bickel, S., Brückner, M., and Scheffer, T. (2007). Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, pages 81–88. 3.1

[17] Birgé, L. (2001). An alternative point of view on lepski's method. *Lecture Notes-Monograph Series*, pages 113–133. 4.3

[18] Blodgett, S. L., Green, L., and O'Connor, B. (2016). Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas. Association for Computational Linguistics. 5.1

[19] Blum, A., Dan, C., and Seddighin, S. (2021). Learning complexity of simulated annealing. In *International conference on artificial intelligence and statistics*, pages 1540–1548. PMLR. 1.3

[20] Borkan, D., Dixon, L., Sorensen, J., Thain, N., and Vasserman, L. (2019). Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 491–500. 3.6.1, 6.6.1

[21] Brent, R. P. (1971). An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4):422–425. 3.4

[22] Bubeck, S., Lee, Y. T., Price, E., and Razenshteyn, I. (2018a). Adversarial examples from cryptographic pseudo-random generators. *arXiv preprint arXiv:1811.06418*. 2.1.2

[23] Bubeck, S., Price, E., and Razenshteyn, I. (2018b). Adversarial examples from computational constraints. *arXiv preprint arXiv:1805.10204*. 2.1.2

[24] Buda, M., Maki, A., and Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259. 1, 4.1

[25] Byrd, J. and Lipton, Z. (2019). What is the effect of importance weighting in deep learning? In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 872–881. PMLR. 6.1, 6.3.3

[26] Cai, T. and Zhang, L. (2019). High dimensional linear discriminant analysis: optimality, adaptive algorithm and missing data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(4):675–705. 2.1.2, 2.2, 2.6.2

[27] Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. (2019). Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, pages 1567–1578. 4.1, 4.1.3, 6.1, 6.2.3

[28] Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. S. (2019). Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*,

pages 11190–11201. 1.1.1, 2.1.2, 4.1.3

[29] Celentano, M., Montanari, A., and Wei, Y. (2020). The lasso with general gaussian designs with applications to hypothesis testing. *arXiv preprint arXiv:2007.13716*. 4.1.3

[30] Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6. 1, 4.1, 4.1.3

[31] Chizat, L., Oyallon, E., and Bach, F. (2019). On lazy training in differentiable programming. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. 5.1.1

[32] Cressie, N. and Read, T. R. (1984). Multinomial goodness-of-fit tests. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(3):440–464. 3.2.2

[33] Cullina, D., Bhagoji, A. N., and Mittal, P. (2018). Pac-learning in the presence of adversaries. In *Advances in Neural Information Processing Systems*, pages 230–241. 1.1.1, 2.1.2

[34] Dan, C., Hansen, K. A., Jiang, H., Wang, L., and Zhou, Y. (2018a). Low rank approximation of binary matrices: Column subset selection and generalizations. In *43rd International Symposium on Mathematical Foundations of Computer Science*. 1.3

[35] Dan, C., Leqi, L., Aragam, B., Ravikumar, P. K., and Xing, E. P. (2018b). The sample complexity of semi-supervised learning with nonparametric mixture models. *Advances in Neural Information Processing Systems*, 31. 1.3

[36] Dan, C., Wang, H., Zhang, H., Zhou, Y., and Ravikumar, P. K. (2019). Optimal analysis of subset-selection based l_p low-rank approximation. *Advances in Neural Information Processing Systems*, 32. 1.3

[37] Dan, C., Wei, Y., and Ravikumar, P. (2020). Sharp statistical guarantees for adversarially robust gaussian classification. *arXiv preprint arXiv:2006.16384*. 4.1.3

[38] Deev, A. (1970). Representation of statistics of discriminant analysis, and asymptotic expansion when space dimensions are comparable with sample size. In *Doklady Akademii Nauk*, volume 195, pages 759–762. Russian Academy of Sciences. 4.1.2, 4.1.2, 4.2.1, 4.5

[39] Defazio, A. (2016). A simple practical accelerated method for finite sums. In *Advances in neural information processing systems*, pages 676–684. 4.8.1, 4.8.1

[40] Deng, Z., Kammoun, A., and Thrampoulidis, C. (2019). A model of double descent for high-dimensional binary linear classification. *arXiv preprint arXiv:1911.05822*. 4.1.3, 2

[41] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. 3.6.1

[42] Diakonikolas, I., Kamath, G., Kane, D., Li, J., Moitra, A., and Stewart, A. (2019). Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864. 1, 3.1

[43] Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. (2017). Being robust (in high dimensions) can be practical. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 999–1008, International Convention Centre, Sydney, Australia. 1, 3.1

[44] Dobriban, E., Wager, S., et al. (2018). High-dimensional asymptotics of prediction: Ridge regression and classification. *The Annals of Statistics*, 46(1):247–279. 4.1.3, 4.1.3

[45] Donini, M., Oneto, L., Ben-David, S., Shawe-Taylor, J., and Pontil, M. (2018). Empirical risk minimization under fairness constraints. *arXiv preprint arXiv:1802.08626*. 4.1.3

[46] Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. (2019). Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR. 1.1.3, 5.1.1

[47] Duchi, J. and Namkoong, H. (2018). Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*. 1.1.2, 3.1, 3.2.2, 3.2.2, 5.1, 5.1.1, 6.1

[48] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. 3.1, 3.1, 5.1.1

[49] Elkhalil, K., Kammoun, A., Couillet, R., Al-Naffouri, T. Y., and Alouini, M.-S. (2017). A large dimensional study of regularized discriminant analysis classifiers. *arXiv preprint arXiv:1711.00382*. 4.1.3

[50] Fithian, W. and Hastie, T. (2014). Local case-control sampling: Efficient subsampling in imbalanced data sets. *Annals of statistics*, 42(5):1693. 4.1.3

[51] Friedler, S. A., Scheidegger, C., and Venkatasubramanian, S. (2016). On the (im) possibility of fairness. *arXiv preprint arXiv:1609.07236*. 4.1.3

[52] Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., and Herrera, F. (2011). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484. 3.1

[53] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*. 1, 2.1

[54] Gordon, Y. (1985). Some inequalities for gaussian processes and applications. *Israel Journal of Mathematics*, 50(4):265–289. 4.2.2

[55] Gulrajani, I. and Lopez-Paz, D. (2021). In search of lost domain generalization. In *International Conference on Learning Representations*. 3.7, 6.1

[56] Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. (2018). Characterizing implicit bias in terms of optimization geometry. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1832–1841. PMLR. 6.1

[57] Hardt, M., Price, E., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised

learning. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29, pages 3315–3323. Curran Associates, Inc. 3.1, 4.1.3, 5.1.1

[58] Hashimoto, T., Srivastava, M., Namkoong, H., and Liang, P. (2018). Fairness without demographics in repeated loss minimization. In Dy, J. and Krause, A., editors, *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1929–1938, Stockholmsmässan, Stockholm Sweden. PMLR. 1, 1.1.2, 3.1, 3.1, 3.2.1, 3.2.2, 3.7, 5.1, 5.1.1, 5.6, 6.1, 6.2.1

[59] Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. (2019). Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*. 4.1.3

[60] He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284. 1, 4.1, 4.1.3

[61] 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*, pages 770–778. 3.6.1

[62] Hovy, D. and Søgaard, A. (2015). Tagging performance correlates with author age. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers)*, pages 483–488. 5.1

[63] Hu, W., Niu, G., Sato, I., and Sugiyama, M. (2018). Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR. 1.1.2, 3.1, 3.1, 3.3, 3.7, 3.9.2, 3.9.2, 6.1

[64] Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. (2006). Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19:601–608. 3.1

[65] Huber, P. J. (1992). Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer. 1, 1.1.2, 3.1, 3.4

[66] Hutchinson, B. and Mitchell, M. (2019). 50 years of test (un) fairness: Lessons for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 49–58. 4.1.3

[67] Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc. 1.1.3, 5.1.1, 5.3.2, 5.3.3, 19, 5.8.1

[68] Japkowicz, N. (2000). The class imbalance problem: Significance and strategies. In *Proc. of the Int'l Conf. on Artificial Intelligence*, volume 56. Citeseer. 3.1

[69] Javanmard, A., Soltanolkotabi, M., and Hassani, H. (2020). Precise tradeoffs in adversarial training for linear regression. *arXiv preprint arXiv:2002.10477*. (document), 2.1.2, 4.1.3, 4.1.3, 4.2.2, 4.7.1, 4.7.1, 4.7.1

[70] Ji, Z. and Telgarsky, M. (2018). Risk and parameter convergence of logistic regression.

arXiv preprint arXiv:1803.07300. 6.1

[71] Johnson, R. A., Wichern, D. W., et al. (2002). *Applied multivariate statistical analysis*, volume 5. Prentice hall Upper Saddle River, NJ. 2.2

[72] Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., and Kalantidis, Y. (2020). Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*. 6.2.2, 6.4.2

[73] Khim, J. and Loh, P.-L. (2018). Adversarial risk bounds for binary classification via function transformation. *arXiv preprint arXiv:1810.09519*, 2. 1.1.1, 2.1.2

[74] Kim, S.-J., Magnani, A., and Boyd, S. (2006). Robust fisher discriminant analysis. In *Advances in neural information processing systems*, pages 659–666. 2.1.2

[75] Kini, G. R., Paraskevas, O., Oymak, S., and Thrampoulidis, C. (2021a). Label-imbalanced and group-sensitive classification under overparameterization. *arXiv preprint arXiv:2103.01550*. 4.1.3, 4.1.3

[76] Kini, G. R., Paraskevas, O., Oymak, S., and Thrampoulidis, C. (2021b). Label-imbalanced and group-sensitive classification under overparameterization. In *Thirty-Fifth Conference on Neural Information Processing Systems*. 6.1, 6.2.2, 6.2.3, 6.2.3, 6.2.3, 6.4.1, 6.4.1, 6.6.2

[77] Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Beery, S., et al. (2020). Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*. 3.6.1, 3.9.1

[78] Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S. M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. (2021). Wilds: A benchmark of in-the-wild distribution shifts. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR. 6.1, 6.6.1, 6.6.2

[79] Kothari, P. K., Steinhardt, J., and Steurer, D. (2018). Robust moment estimation and improved clustering via sum of squares. In Diakonikolas, I., Kempe, D., and Henzinger, M., editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1035–1046. ACM. 3.5

[80] Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. 6.6.1

[81] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105. 1, 2.1

[82] Kumar, A., Ma, T., and Liang, P. (2020). Understanding self-training for gradual domain adaptation. *arXiv preprint arXiv:2002.11361*. 4.1.3

[83] Kusner, M. J., Loftus, J., Russell, C., and Silva, R. (2017). Counterfactual fairness. In *Advances in neural information processing systems*, pages 4066–4076. 3.1, 5.1.1

[84] 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. 3.2.1, 3.7

[85] Lai, K. A., Rao, A. B., and Vempala, S. (2016). Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674. IEEE. 1, 3.1

[86] Larson, J., Mattu, S., Kirchner, L., and Angwin, J. (2016). How we analyzed the compas recidivism algorithm. *ProPublica (5 2016)*, 9(1). 3.3

[87] Laurent, B. and Massart, P. (2000). Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338. 4.3, 4.3, 4.4.2

[88] Lee, J., Park, S., and Shin, J. (2020). Learning bounds for risk-sensitive learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13867–13879. Curran Associates, Inc. 3.1

[89] Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. (2019). Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32:8572–8583. (document), 5.1.1, 5.3.2, 5.3.3, 5.8, 5.8.1

[90] Li, M., Zhang, X., Thrampoulidis, C., Chen, J., and Oymak, S. (2021). Autobalance: Optimized loss functions for imbalanced data. In *Thirty-Fifth Conference on Neural Information Processing Systems*. 6.1

[91] Li, T., Prasad, A., and Ravikumar, P. K. (2015). Fast classification rates for high-dimensional gaussian generative models. In *Advances in Neural Information Processing Systems*, pages 1054–1062. 2.1.2

[92] Li, T., Yi, X., Carmanis, C., and Ravikumar, P. (2017). Minimax gaussian classification & clustering. In *Artificial Intelligence and Statistics*, pages 1–9. 2.1.2, 2.4, 2.4.2, 2.4, 2.9, 4.4.1, 4.8.4

[93] Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. (2021). Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR. 5.1.1, 5.6, 6.2.1

[94] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738. 3.6.1, 6.6.1

[95] Masnadi-Shirazi, H. and Vasconcelos, N. (2010). Risk minimization, probability elicitation, and cost-sensitive svms. In *ICML*. 6.2.3

[96] McLachlan, G. J. and Peel, D. (2004). *Finite mixture models*. John Wiley & Sons. 2.1

[97] Mei, S. and Montanari, A. (2019). The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*. 4.1.3

[98] Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. (2021). Long-tail learning via logit adjustment. In *International Conference on Learning Representations*.

6.1, 6.2.3

[99] Michel, P., Hashimoto, T., and Neubig, G. (2021). Modeling the second player in distributionally robust optimization. In *International Conference on Learning Representations*. 3.7

[100] Mignacco, F., Krzakala, F., Lu, Y. M., and Zdeborová, L. (2020). The role of regularization in classification of high-dimensional noisy gaussian mixture. *arXiv preprint arXiv:2002.11544*. 4.4.2, 4.4.2, 4.4.3

[101] Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of machine learning*. MIT press. 25, 27

[102] Montasser, O., Hanneke, S., and Srebro, N. (2019). Vc classes are adversarially robustly learnable, but only improperly. *arXiv preprint arXiv:1902.04217*. 1.1.1, 2.1.2

[103] Namkoong, H. and Duchi, J. C. (2016). Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in neural information processing systems*, pages 2208–2216. 1.1.2, 3.1

[104] Oren, Y., Sagawa, S., Hashimoto, T., and Liang, P. (2019). Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4227–4237, Hong Kong, China. Association for Computational Linguistics. 3.1, 5.1.1

[105] Owen, A. B. (2007). Infinitely imbalanced logistic regression. *Journal of Machine Learning Research*, 8(4). 4.1.3

[106] Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359. 3.1

[107] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016). The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE. 1, 2.1

[108] Patel, V. M., Gopalan, R., Li, R., and Chellappa, R. (2015). Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69. 3.1

[109] Prasad, A., Balakrishnan, S., and Ravikumar, P. (2020). A robust univariate mean estimator is all you need. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4034–4044. PMLR. 3.5

[110] Prasad, A., Suggala, A. S., Balakrishnan, S., and Ravikumar, P. (2018). Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*. 1, 3.1

[111] Qiao, X. and Liu, Y. (2009). Adaptive weighted learning for unbalanced multicategory classification. *Biometrics*, 65(1):159–168. 4.1.3

[112] Qiao, X., Zhang, H. H., Liu, Y., Todd, M. J., and Marron, J. S. (2010). Weighted distance weighted discrimination and its asymptotic properties. *Journal of the American Statistical Association*, 105(489):401–414. 4.1.3

[113] Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2009). *Dataset shift in machine learning*. The MIT Press. 3.1

[114] Raghunathan, A., Xie, S. M., Yang, F., Duchi, J., and Liang, P. (2020). Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*. 2.1.2

[115] Raudys, Š. and Young, D. M. (2004). Results in statistical discriminant analysis: A review of the former soviet union literature. *Journal of Multivariate Analysis*, 89(1):1–35. 4.1.2

[116] Rawls, J. (2001). *Justice as fairness: A restatement*. Harvard University Press. 1, 3.1, 5.1.1

[117] Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2020a). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*. 1.1.3, 3.1, 3.7, 3.9.1, 5.1, 5.1.1, 5.2.1, 5.2.1, 5.2.2, 5.4, 5.4.1, 5.4.1, 6.1, 6.3.3, 6.6.1, 6.6.2

[118] Sagawa, S., Koh, P. W., Lee, T., Gao, I., Xie, S. M., Shen, K., Kumar, A., Hu, W., Yasunaga, M., Marklund, H., Beery, S., David, E., Stavness, I., Guo, W., Leskovec, J., Saenko, K., Hashimoto, T., Levine, S., Finn, C., and Liang, P. (2022). Extending the WILDS benchmark for unsupervised adaptation. In *International Conference on Learning Representations*. 6.1

[119] Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. (2020b). An investigation of why overparameterization exacerbates spurious correlations. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8346–8356. PMLR. 1.1.3, 3.1, 5.1, 5.1.1, 5.3.3

[120] Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. (2018). Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, pages 5019–5031. (document), 1.1.1, 2.1, 2.1.1, 2.1.2, 2.3, 1, 3, 4, 2.1, 2.4, 4.1.3

[121] Scott, A. and Wild, C. (1991). Fitting logistic regression models in stratified case-control studies. *Biometrics*, pages 497–510. 4.1.3

[122] Shen, Y. and Sanghavi, S. (2019). Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning*, pages 5739–5748. PMLR. 3.4, 3.7

[123] Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244. 3.1, 3.7, 5.1, 5.2.1, 6.1

[124] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484. 1, 2.1

[125] Stanforth, R., Fawzi, A., Kohli, P., et al. (2019). Are labels required for improving adversarial robustness? *arXiv preprint arXiv:1905.13725*. 1.1.1, 2.1.2

[126] Suggala, A. S., Prasad, A., Nagarajan, V., and Ravikumar, P. (2018). Revisiting adversarial risk. *arXiv preprint arXiv:1806.02924*. 2.1.2

[127] Sur, P., Chen, Y., and Candès, E. J. (2019). The likelihood ratio test in high-dimensional

logistic regression is asymptotically a rescaled chi-square. *Probability theory and related fields*, 175(1):487–558. 4.1.3

[128] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*. 1, 2.1

[129] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer. 3.1

[130] Tatman, R. (2017). Gender and dialect bias in youtube's automatic captions. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 53–59. 5.1

[131] Thompson, A. C. and Thompson, A. C. (1996). *Minkowski geometry*. Cambridge University Press. 2.2

[132] Thrampoulidis, C., Abbasi, E., and Hassibi, B. (2018). Precise error analysis of regularized m-estimators in high dimensions. *IEEE Transactions on Information Theory*, 64(8):5592–5628. 4.1.3, 4.6, 4.8.1

[133] Thrampoulidis, C., Oymak, S., and Hassibi, B. (2015). Regularized linear regression: A precise analysis of the estimation error. In *Conference on Learning Theory*, pages 1683–1709. PMLR. 4.2.2

[134] Thrampoulidis, C., Oymak, S., and Soltanolkotabi, M. (2020). Theoretical insights into multiclass classification: A high-dimensional asymptotic view. *arXiv preprint arXiv:2011.07729*. 4.4.2, 4.4.2, 4.4.3, 4.4.3

[135] Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2018). Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*. 2.1.2

[136] Tukey, J. W. (1960). A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, pages 448–485. 1, 1.1.2, 3.1

[137] Van der Vaart, A. W. (2000). *Asymptotic statistics*, volume 3. Cambridge university press. 2.1.3

[138] Van Horn, G. and Perona, P. (2017). The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*. 1, 4.1

[139] Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*. 22

[140] Viering, T., Mey, A., and Loog, M. (2019). Open problem: Monotonicity of learning. In *Conference on Learning Theory*, pages 3198–3201. 4.1

[141] Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press. 2.6.1, 4.1.2

[142] Wang, C., Jiang, B., et al. (2018a). On the dimension effect of regularized linear discriminant analysis. *Electronic Journal of Statistics*, 12(2):2709–2742. 4.1.3

[143] Wang, H., Zhu, R., and Ma, P. (2018b). Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*, 113(522):829–844. 4.1.3

[144] Wang, K. A., Chatterji, N. S., Haque, S., and Hashimoto, T. (2022). Is importance

weighting incompatible with interpolating classifiers? In *International Conference on Learning Representations*. 6.1

[145] Wang, M. and Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153. 3.1

[146] Wang, Y., Jha, S., and Chaudhuri, K. (2017). Analyzing the robustness of nearest neighbors to adversarial examples. *arXiv preprint arXiv:1706.03922*. 2.1.2

[147] Weng, T.-W., Zhang, H., Chen, H., Song, Z., Hsieh, C.-J., Boning, D., Dhillon, I. S., and Daniel, L. (2018). Towards fast computation of certified robustness for relu networks. *arXiv preprint arXiv:1804.09699*. 2.1.2

[148] Wiles, O., Gowal, S., Stimberg, F., Rebuffi, S.-A., Ktena, I., Dvijotham, K. D., and Cemgil, A. T. (2022). A fine-grained analysis on distribution shift. In *International Conference on Learning Representations*. 6.1

[149] Williamson, R. and Menon, A. (2019). Fairness risk measures. In *International Conference on Machine Learning*, pages 6786–6797. PMLR. 4.1.3

[150] Wu, Y. and Zhou, H. H. (2019). Randomly initialized em algorithm for two-component gaussian mixture achieves near optimality in $o(\sqrt{n})$ iterations. *arXiv preprint arXiv:1908.10935*. 4.1.3

[151] Xu, D., Ye, Y., and Ruan, C. (2021). Understanding the role of importance weighting for deep learning. In *International Conference on Learning Representations*. 6.1

[152] Xu, H., Caramanis, C., and Mannor, S. (2009a). Robust regression and lasso. In *Advances in neural information processing systems*, pages 1801–1808. 2.1.2

[153] Xu, H., Caramanis, C., and Mannor, S. (2009b). Robustness and regularization of support vector machines. *Journal of machine learning research*, 10(7). 2.1.2

[154] Xu, H. and Mannor, S. (2012). Robustness and generalization. *Machine learning*, 86(3):391–423. 2.1.2

[155] Xu, Z., Dan, C., Khim, J., and Ravikumar, P. (2020). Class-weighted classification: Trade-offs and robust approaches. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10544–10554. PMLR. 1.1.2, 1.3, 3.1, 3.1, 4.1, 5.1.1, 5.6, 6.1

[156] Yang, Y. and Xu, Z. (2020). Rethinking the value of labels for improving class-imbalanced learning. *arXiv preprint arXiv:2006.07529*. 4.1

[157] 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*. 6.1, 6.2.3

[158] Yin, D., Ramchandran, K., and Bartlett, P. (2018). Rademacher complexity for adversarially robust generalization. *arXiv preprint arXiv:1810.11914*. 1.1.1, 2.1.2

[159] 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*, pages 1171–1180. 3.1, 4.1.3, 5.1.1

[160] Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. (2013). Learning fair representations. In *International Conference on Machine Learning*, pages 325–333. 3.1, 5.1.1

[161] Zhai, R., Cai, T., He, D., Dan, C., He, K., Hopcroft, J., and Wang, L. (2019). Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*. 1.1.1, 1.3, 2.1.2

[162] Zhai, R., Dan, C., He, D., Zhang, H., Gong, B., Ravikumar, P., Hsieh, C.-J., and Wang, L. (2020). Macer: Attack-free and scalable robust training via maximizing certified radius. In *International Conference on Learning Representations*. 1.3

[163] Zhai, R., Dan, C., Kolter, Z., and Ravikumar, P. (2021a). Doro: Distributional and outlier robust optimization. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12345–12355. PMLR. 5.1.1, 5.6, 6.1, 6.2.1, 6.4.1, 6.4.1, 6.6.2

[164] Zhai, R., Dan, C., Kolter, Z., and Ravikumar, P. (2022). Understanding why generalized reweighting does not improve over erm. *arXiv preprint arXiv:2201.12293*. 6.1, 28, 6.7

[165] Zhai, R., Dan, C., Suggala, A., Kolter, J. Z., and Ravikumar, P. K. (2021b). Boosted CVar classification. In *Thirty-Fifth Conference on Neural Information Processing Systems*. 1.3, 6.1

[166] Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. (2019). Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*. 2.1.2

[167] Zhao, H., Dan, C., Aragam, B., Jaakkola, T. S., Gordon, G. J., and Ravikumar, P. (2020). Fundamental limits and tradeoffs in invariant representation learning. *arXiv preprint arXiv:2012.10713*. 1.3

[168] Zheng, X., Dan, C., Aragam, B., Ravikumar, P., and Xing, E. (2020). Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425. PMLR. 1.3

[169] Zhu, B., Jiao, J., and Steinhardt, J. (2020). Generalized resilience and robust statistics. 1.1.2, 3.1