ROBUST CLASSIFICATION UNDER ℓ_0 ATTACK FOR THE GAUSSIAN MIXTURE MODEL

PAYAM DELGOSHA*, HAMED HASSANI†, AND RAMTIN PEDARSANI‡

Abstract. It is well-known that machine learning models are vulnerable to small but cleverly-designed adversarial perturbations that can cause misclassification. While there has been major progress in designing attacks and defenses for various adversarial settings, many fundamental and theoretical problems are yet to be resolved. In this paper, we consider classification in the presence of ℓ_0-bounded adversarial perturbations, a.k.a. sparse attacks. This setting is significantly different from other ℓ_p-adversarial settings, with $p \geq 1$, as the ℓ_0-ball is non-convex and highly non-smooth. Under the assumption that data is distributed according to the Gaussian mixture model, our goal is to characterize the optimal robust classifier and the corresponding robust classification error as well as a variety of trade-offs between robustness, accuracy, and the adversary's budget. To this end, we develop a novel classification algorithm called `FilTrun` that has two main modules: Filtration and Truncation. The key idea of our method is to first filter out the *non-robust* coordinates of the input and then apply a carefully-designed truncated inner product for classification. By analyzing the performance of `FilTrun`, we derive an upper bound on the optimal robust classification error. We further find a lower bound by designing a specific adversarial strategy that enables us to derive the corresponding robust classifier and its achieved error. For the case that the covariance matrix of the Gaussian mixtures is diagonal, we show that as the input's dimension gets large, the upper and lower bounds converge; i.e. we characterize the asymptotically-optimal robust classifier. Throughout, we discuss several examples that illustrate interesting behaviors such as the existence of a *phase transition* for adversary's budget determining whether the effect of adversarial perturbation can be fully neutralized or not.

1. Introduction.

Machine learning has been widely used in a variety of applications including image recognition, virtual assistants, autonomous driving, many of which are safety-critical. Adversarial attacks to machine learning models in the form of a small perturbation added to the input have been shown to be effective in causing classification errors [4, 33, 10, 5, 17]. Formally, the adversary aims to perturb the data in a small ℓ_p-neighborhood so that the perturbed data is "close" to the original data (e.g. imperceptible perturbation in the case of an image) and misclassification occurs. There have been a variety of attacks and defenses proposed in the literature which mostly focus on ℓ_2 or ℓ_∞ bounded perturbations [2, 19, 35]. The state-of-the-art empirical defense against adversarial attacks is iterative training with adversarial examples [18]. While adversarial training can improve robustness, it is shown that there is a fundamental tradeoff between robustness and test accuracy, and such defenses typically lack good generalization performance [34, 32, 26, 1, 36, 13].

The focus of this paper is different from such prior work as we consider the problem of robust classification under ℓ_0-bounded attacks. In this setting, given a pre-specified budget k, the adversary can choose up to k coordinates and arbitrarily change the value of the input at those coordinates. In other words, the adversary can change the input within the so-called ℓ_0-ball of radius k. In contrast with ℓ_p-balls ($p \geq 1$), the ℓ_0-ball is non-convex and highly non-smooth. Moreover, the ℓ_0 ball contains inherent discrete (combinatorial) structures that can be exploited by both the learner and the adversary. As a result, the ℓ_0-adversarial setting bears several fundamental challenges that are absent in other adversarial settings commonly studied in the literature and

*Department of Computer Science, University of Illinois at Urbana Champaign, IL, delgosha@illinois.edu

†Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, hassani@seas.upenn.edu

‡Department of Electrical and Computer Engineering, University of California, Santa Barbara, Santa Barbara, CA, ramtin@ece.ucsb.edu

most techniques from prior work do not readily apply in the ℓ_0 setting. Complicating matters further, it can be shown that any piece-wise linear classifier, e.g. a feed-forward deep neural network with ReLu activations, completely fails in the ℓ_0 setting [31]. These all point to the fact that new methodologies are required in the ℓ_0 setting.

The ℓ_0-adversarial setting involves sparse attacks that perturb only a small portion of the input signal. This has a variety of applications including natural language processing [14], malware detection [11], and physical attacks in object detection [16]. Prior work on ℓ_0 adversarial attacks can be divided into two categories of white-box attacks that are gradient-based, e.g. [5, 22, 21], and black-box attacks based on zeroth-order optimization, e.g. [29, 7]. Defense strategies against ℓ_0-bounded attacks have also been proposed, e.g. defenses based on randomized ablation [15] and defensive distillation [23]. Moreover, [31] develops a simple mathematical framework to show the existence of targeted adversarial examples with ℓ_0-bounded perturbation in arbitrarily deep neural networks.

Despite this interesting recent progress and practical relevance, many fundamental theoretical questions in the ℓ_0-setting have so far been unanswered: *What are the key properties of a robust classifier (recall that all piece-wise linear classifiers fail)? What is the optimal robust classifier in standard theoretical settings such the Gaussian mixture model for data? Is there a trade-off between robustness and accuracy? How does the (optimal) robust classification error behave as the adversary's budget k increases? Are there any phase transitions?*

We consider the problem of classification with ℓ_0-adversarially perturbed inputs under the assumption that data is distributed according to the Gaussian mixture model. We formally introduce this setting in Section 2, and address the questions above in the proceeding sections. In particular, instead of searching for the exact form of the optimal robust classifier (which is intractable), we follow a design-based approach: We introduce a novel algorithm for classification as well as strategies for the adversary. We then precisely characterize the error performance of these methodologies, and consequently, analyse the optimal robust classification error, tradeoffs between robustness and accuracy, phase transitions, etc. We envision that our proposed classification method introduces important modules and insights that are necessary to obtain robustness against ℓ_0-adversaries for general data distributions (and practical datasets), going beyond the theoretical setting of this paper.

Summary of Contributions. The main contributions of this paper are as follows:

- We propose a new robust classification algorithm called `FilTrun` that is based on two main modules: **Fil**tration and **Trun**cation (See Section 3.1.1 and Algorithm 3.1 therein). The filtration module removes the *non-robust* coordinates (features) from the input by zeroing out their values. The result is then passed through the truncation module which returns a label by computing a *truncated inner product* with a weight vector whose weights are optimized according to the distribution of un-filtered (surviving) coordinates. The truncation module is inspired by tools from robust statistics and guarantees that major outlier values in the input vector, which are possibly caused by the adversary, do not pass to affect the final decision. We highlight that the proposed classifier is highly nonlinear. This is consistent with the simple observation that any linear classifier fails to be robust in the presence of ℓ_0 attacks.
- We analytically derive the robust classification error of the proposed classifier. This in particular serves as an upper bound on the optimal robust

classification error (See Theorem 3.2 and Corollary 3.5).

- We introduce adversarial strategies which, given sufficient budget, perturb the input in a way that the information about the true label is totally erased within the adversarially modified coordinates. The key idea is to pick a subset of the coordinates and to modify their distribution so that they become independent from the true label. This leads to a lower bound for the optimal robust error. (See Theorems 3.8 and 3.11).
- In the case of having a diagonal covariance matrix for the Gaussian mixtures, we prove that our proposed algorithm `FilTrun` is indeed *asymptotically-optimal*, i.e. as the input dimension d approaches infinity, the upper and lower bounds converge to the same analytical expression (See Theorems 3.13 in Section 3.3.2). To the best of our knowledge, this is the first result that establishes optimality for the robust classification error of any mathematical model with ℓ_0 attack.
- We discuss our results through several example scenarios. In certain scenarios, a phase transition is observed in the sense that for a threshold α_0, when the adversary's budget is asymptotically below d^{α_0}, its effect can be completely neutralized, while if the adversary's budget is above d^{α_0}, no classifier can do better than a naive classifier. In some other scenarios, no sharp phase transition is existent, leading to a trade-off between robustness and accuracy.

2. Problem Formulation. We consider the binary Gaussian mixture model where the distribution for the data generation is specified by the label being $y \sim \text{Unif}\{\pm 1\}$ and $\boldsymbol{x} \sim \mathcal{N}(y\boldsymbol{\mu}, \Sigma)$, i.e. the Gaussian distribution with mean $y\boldsymbol{\mu}$ and covariance matrix Σ, where $\boldsymbol{\mu} \in \mathbb{R}^d$ and Σ is positive definite. Hereafter, we denote this distribution by $(\boldsymbol{x}, y) \sim \mathcal{D}$ and refer to y as the label and to $\boldsymbol{x}$ as the input. Our results correspond to arbitrary choices of $\boldsymbol{\mu}$ and Σ, however, we consider as running example an important special case in which Σ is a diagonal matrix, i.e. the coordinates of $\boldsymbol{x}$ are independent conditioned on y. Focusing on classification, we consider functions of the form $\mathcal{C} : \mathbb{R}^d \to \{-1, 1\}$ that predict the label from the input. As a metric for the discrepancy between the prediction of the classifier on the input $\boldsymbol{x}$ and the true label y, we consider the 0-1 loss $\ell(\mathcal{C}; \boldsymbol{x}, y) = \mathbb{1}\left[\mathcal{C}(\boldsymbol{x}) \neq y\right]$. We consider classification in the presence of an adversary that perturbs the input $\boldsymbol{x}$ within the ℓ_0-ball of radius k:

$$\mathcal{B}_0(\boldsymbol{x}, k) := \{\boldsymbol{x}' \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq k\},$$

where for $\boldsymbol{x} = (x_1, \cdots, x_d)$ we define $\|\boldsymbol{x}\|_0 := \sum_{i=1}^d \mathbb{1}\left[x_i \neq 0\right]$. In other words, the adversary can arbitrarily modify at most k coordinates of $\boldsymbol{x}$ to obtain $\boldsymbol{x}'$, and feed the new vector $\boldsymbol{x}'$ to the classifier. We call k the *budget* of the adversary. In this setting, the robust classification error of a classifier $\mathcal{C}$ is defined to be the following:

$$\mathcal{L}_{\boldsymbol{\mu},\Sigma}(\mathcal{C}, k) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x},k)} \ell(\mathcal{C}; \boldsymbol{x}', y)\right]. \tag{2.1}$$

We aim to design classfiers with minimum robust classification error. Hence, we define the *optimal robust classification error* by minimizing (2.1) over all possible classifiers:

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k) := \inf_{\mathcal{C}} \mathcal{L}_{\boldsymbol{\mu},\Sigma}(\mathcal{C}, k). \tag{2.2}$$

Our goal in this paper is to precisely characterize $\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k)$ parameterized by $\Sigma, \boldsymbol{\mu}$ and in different regimes of the adversary's budget k.

It is well known that in the absence of the adversary, i.e. when $k = 0$, the Bayes optimal classifier is the linear classifier $\mathcal{C}(\boldsymbol{x}) = \operatorname{sgn}\left(\langle \Sigma^{-1}\boldsymbol{\mu}, \boldsymbol{x}\rangle\right)$ which achieves the *optimal standard error* of $\bar{\Phi}(\|\boldsymbol{\nu}\|_2)$ where $\boldsymbol{\nu} := \Sigma^{-1/2}\boldsymbol{\mu}$ and $\bar{\Phi}(x) := 1 - \Phi(x)$ denotes the complementary CDF of a standard normal distribution. In order to fix the baseline, specifically to have a meaningful asymptotic discussion, we may assume without loss of generality that

$$\|\boldsymbol{\nu}\|_2 = \|\Sigma^{-1/2}\boldsymbol{\mu}\|_2 = 1. \tag{2.3}$$

Hence, the optimal standard error, which is a lower bound for (2.2), becomes $\bar{\Phi}(1)$.

To highlight some of the main challenges of the ℓ_0-adversarial setting, we note that linear classifiers in general have been very successful in the Gaussian mixture setting. Apart from the fact that the Bayes-optimal classier is linear (when there is no adversary), even when the adversarial corruptions are chosen in a ℓ_p-ball for $p \geq 1$ it can be shown that the optimal robust classifiers in many cases are also linear (see [3, 9]). In contrast, in the presence of ℓ_0-adversaries, it is not hard to show that *any* linear classifier completely fail. More precisely, when $\mathcal{C}$ is linear and $k \geq 1$, we have $\mathcal{L}_{\boldsymbol{\mu},\Sigma}(\mathcal{C}, k) = \frac{1}{2}$. Such failure of linear classifiers showcases, on the one hand, how powerful the adversary is, and on the other hand, the necessity of new methodologies in designing robust classifiers.

Further Related Work. For ℓ_p adversaries, $p \geq 1$, Gaussian mixture models have been the main setting used in prior work to investigate optimal rules, trade-offs, and various other phenomena for robust classification; See e.g. [28, 3, 9, 12, 27, 8, 25, 6, 20, 24]. Further, [30] considers data to be uniformly distributed on the sphere or cube and shows the inevitability of adversarial examples in ℓ_p-settings, $p \geq 0$. In contrast, to the best of our knowledge, our work provides the first comprehensive study on the ℓ_0-adversarial setting using the Gaussian mixture model.

Notation. Given two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, $\boldsymbol{x} \odot \boldsymbol{y} \in \mathbb{R}^d$ denotes the elementwise product of $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e. $(x_1y_1, \ldots, x_dy_d)$. Moreover, $\operatorname{sort}(\boldsymbol{x})$ denotes the vector containing the elements in $\boldsymbol{x}$ in descending order. For $a \in \mathbb{R}$, $\operatorname{sgn}(a)$ returns the sign of a. We use $[d]$ to denote the set $\{1, \ldots, d\}$ and $[i : j]$ denotes the set $\{i, i+1, \ldots, j\}$. Given a vector $\boldsymbol{x} \in \mathbb{R}^d$ and a subset $A \subseteq [d]$, $\boldsymbol{x}_A = (x_a : a \in A) \in \mathbb{R}^{|A|}$ denotes the subvector of $\boldsymbol{x}$ consisting of the coordinates in A. Given a matrix Σ, its diagonal part, denoted by $\widetilde{\Sigma}$, has the same diagonal entries as Σ and its other entries are 0. Given a matrix $A \in \mathbb{R}^{d\times d}$, $\|A\|_\infty$ denotes the operator norm of A induced by the vector ℓ_∞ norm, i.e. $\|A\|_\infty := \sup_{\boldsymbol{x}\neq 0} \|A\boldsymbol{x}\|_\infty / \|\boldsymbol{x}\|_\infty = \max_{1\leq i\leq d} \sum_{j=1}^{d} |A_{i,j}|$.

3. Main Results.

In this section, we state our main results that include (i) the proposed algorithm and its performance analysis that serves as an upper bound on the optimal robust classification error (Section 3.1), (ii) lower bound on the optimal robust classification error (Section 3.2), and (iii) discussion on the optimality of the proposed algorithm (Section 3.3). Throughout, we illustrate our theoretical results and their ramifications via several examples.

3.1. Upper Bound on the Optimal Robust Classification Error: Algorithm Description and Theoretical Guarantees.

In Section 3.1.1, we introduce `FilTrun`, our proposed robust classification algorithm, and in Section 3.1.2, we analyze its performance.

3.1.1. Algorithm Description.

We describe our proposed algorithm `FilTrun`, a robust classifier which is based on two main modules: Truncation and Filtration. We

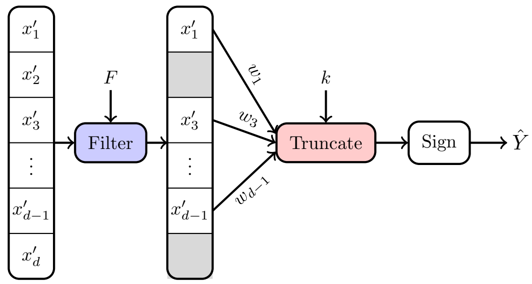

Fig. 1: Schematic of `FilTrun`.

first introduce each of these modules and then proceed with describing the classifier.

Truncation. Given vectors $\boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^d$ and an integer $0 \leq k < d/2$, we define the k–truncated inner product of $\boldsymbol{w}$ and $\boldsymbol{x}$ as the summation of the element-wise product of $\boldsymbol{w}$ and $\boldsymbol{x}$ after removing the top and bottom k elements, and denote it by $\langle \boldsymbol{w}, \boldsymbol{x} \rangle_k$. More precisely, let $\boldsymbol{z} := \boldsymbol{w} \odot \boldsymbol{x} \in \mathbb{R}^d$ be the element-wise product of $\boldsymbol{w}$ and $\boldsymbol{x}$ and let $\mathbf{s} = (s_1, \cdots, s_d) = \mathrm{sort}(\boldsymbol{z})$ be obtained by sorting coordinates of $\boldsymbol{z}$ in descending order. We then define

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle_k := \sum_{i=k+1}^{d-k} s_i. \tag{3.1}$$

Note that when $k = 0$, this reduces to the normal inner product $\langle \boldsymbol{w}, \boldsymbol{x} \rangle$. Truncation is a natural method to remove "outliers" which might exist in the data due to an adversary modifying some coordinates. Therefore, we expect the truncated inner product to be robust against ℓ_0 perturbations. The following lemma formalizes this. The proof of Lemma 3.1 is given in Appendix A.

Lemma 3.1. *Given $\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{w} \in \mathbb{R}^d$, for integer k satisfying $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq k < d/2$, we have*

$$|\langle \boldsymbol{w}, \boldsymbol{x}' \rangle_k - \langle \boldsymbol{w}, \boldsymbol{x} \rangle| \leq 8k \|\boldsymbol{w} \odot \boldsymbol{x}\|_\infty.$$

In the context of our problem, this lemma suggests that if the budget of the adversary is at most k, we can bound the difference between the k–truncated inner product between $\boldsymbol{w}$ and the adversarially modified sample $\boldsymbol{x}'$ and the (non-truncated) inner product between $\boldsymbol{w}$ and the original sample $\boldsymbol{x}$. Recall that in the absence of the adversary, the optimal Bayes classifier is a linear classifier of the form $\mathrm{sgn}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle)$ with $\boldsymbol{w} = \Sigma^{-1}\boldsymbol{\mu}$. Hence, motivated by Lemma 3.1, one can argue that $\mathrm{sgn}(\langle \boldsymbol{w}, \boldsymbol{x}' \rangle_k)$ would be robust against ℓ_0 adversarial attacks with budget at most k assuming we can appropriately control the bound of Lemma 3.1. However, this is not enough–it turns out that in certain cases, we need to *filter out* some of the input coordinates and perform the truncation on the remaining coordinates, which we call the *surviving* coordinates.

Filtration refers to discarding some of the coordinates of the input. Intuitively, these coordinates are the *non-robust* features which do more harm than good when the input is adversarially corrupted. More precisely, given a fixed and nonempty subset of coordinates $F \subseteq [d]$, we define the classifier $\mathcal{C}_F^{(k)}$ as follows:

$$\mathcal{C}_F^{(k)}(\boldsymbol{x}') := \operatorname{sgn}\big(\, \langle \boldsymbol{w}(F), \boldsymbol{x}'_F \rangle_k \,\big), \tag{3.2}$$

where

$$\boldsymbol{w}(F) := \Sigma_F^{-1} \boldsymbol{\mu}_F,$$

and

$$\Sigma_F = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[(\boldsymbol{x}_F - \boldsymbol{\mu}_F)(\boldsymbol{x}_F - \boldsymbol{\mu}_F)^T | y = 1 \right] \tag{3.3}$$

is the covariance matrix of $\boldsymbol{x}_F$ conditioned on y, which is essentially the submatrix of Σ corresponding to the elements in F. Note that $\boldsymbol{w}(F)$ is the optimal Bayes classifier of y given $\boldsymbol{x}_F$ in the absence of the adversary. It is easy to see that when Σ is diagonal, $\boldsymbol{w}(F) = \boldsymbol{w}_F$, but this might not hold in general.

Algorithm 3.1 and Figure 1 illustrate the classification procedure `FilTrun` given in (3.2). So far we have not explained how the set F is chosen and the algorithm works with any such set given as an input. Later we discuss how the set F is chosen (see Remarks 3.4 and 3.15).

Algorithm 3.1 `FilTrun`

Input:
- k: adversary's ℓ_0 budget
- $\boldsymbol{\mu}, \Sigma$: parameters of the Gaussian distribution
- F: the set of surviving coordinates
- $\boldsymbol{x}'$: the corrupted input

Output:
- $\mathcal{C}_F^{(k)}(\boldsymbol{x}')$

1: **function** FilTrun$(k, \boldsymbol{\mu}, \Sigma, F, \boldsymbol{x}')$
2:     **Filtering:** Construct $\boldsymbol{\mu}_F, \Sigma_F$ and $\boldsymbol{x}'_F$ corresponding to the coordinates in F
3:     Compute $\boldsymbol{w}(F) \leftarrow \Sigma_F^{-1} \boldsymbol{\mu}_F$
4:     **Truncation:** Compute $\langle \boldsymbol{w}(F), \boldsymbol{x}'_F \rangle_k$
5:     **Return** $\operatorname{sgn}(\langle \boldsymbol{w}(F), \boldsymbol{x}'_F \rangle_k)$
6: **end function**

3.1.2. Upper bound on the robust classification error of `FilTrun`. Theorem 3.2 below states an upper bound for the robust error associated with the classification algorithm `FilTrun` introduced in Section 3.1.1. In particular, this yields an upper bound on the optimal robust classification error. The proof of Theorem 3.2 is given in Appendix B.

THEOREM 3.2. *Assume that $\boldsymbol{\mu}, \Sigma$ are given such that (2.3) holds. For a given nonempty $F \subseteq [d]$ and $0 \le k < d/2$, we have*

$$\mathcal{L}_{\boldsymbol{\mu},\Sigma}(\mathcal{C}_F^{(k)}, k) \le \frac{1}{\sqrt{2\log d}} + \bar{\Phi}\left(\|\boldsymbol{\nu}(F)\|_2 - \frac{16k\sqrt{2\log d}\,\|\widetilde{\Sigma}_F^{1/2}\Sigma_F^{-1/2}\|_\infty \|\boldsymbol{\nu}(F)\|_\infty}{\|\boldsymbol{\nu}(F)\|_2} \right), \tag{3.4}$$

where Σ_F is defined in (3.3), $\widetilde{\Sigma}_F$ is the diagonal part of Σ_F, and

$$\boldsymbol{\nu}(F) := \Sigma_F^{-1/2} \boldsymbol{\mu}_F.$$

As a consequence, we obtain

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k) \le \frac{1}{\sqrt{2\log d}} + \min_{F\subseteq[d]} \bar{\Phi}\left(\|\boldsymbol{\nu}(F)\|_2 - \frac{16k\sqrt{2\log d}\|\widetilde{\Sigma}_F^{1/2}\Sigma_F^{-1/2}\|_\infty \|\boldsymbol{\nu}(F)\|_\infty}{\|\boldsymbol{\nu}(F)\|_2} \right). \tag{3.5}$$

Remark 3.3. Recall from Section 3.1.1 that F is the set of coordinates used for classification (i.e. the information in the coordinates F^c is discarded). Therefore, we essentially work with $\boldsymbol{x}_F$ as an input. If the adversary is not present, the optimal classification error is achieved via the Bayes linear classifier which has error $\bar{\Phi}(\|\boldsymbol{\nu}(F)\|_2)$. However, due to the existence of an adversary, we need to perform truncation which influences the error through the second term inside the argument of $\bar{\Phi}$ in (3.4).

Remark 3.4. The bound in Theorem 3.2 can be used as a guide to choose the set of surviving coordinates F. More precisely, we can choose F which minimizes the right hand side in (3.5). Later, in Section 3.3, we discuss a simpler mechanism for choosing F when the covariance matrix Σ is diagonal (see Remark 3.15 therein).

Here, we outline the proof of Theorem 3.2. Due to the symmetry, we only need to analyze the classification error when $y = 1$. In this case, an error occurs only when there exists some $\boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x}, k)$ such that $\langle \boldsymbol{w}(F), \boldsymbol{x}'_F\rangle_k \le 0$. But since $\|\boldsymbol{x}'_F - \boldsymbol{x}_F\|_0 \le \|\boldsymbol{x}' - \boldsymbol{x}\|_0 \le k$, Lemma 3.1 implies that for such $\boldsymbol{x}'$, we have $|\langle \boldsymbol{w}(F), \boldsymbol{x}'_F\rangle_k - \langle \boldsymbol{w}(F), \boldsymbol{x}_F\rangle| \le 8k\|\boldsymbol{w}(F)\odot \boldsymbol{x}_F\|_\infty$. Therefore, the robust classification error is upper bounded by $\mathbb{P}\left(\langle \boldsymbol{w}(F), \boldsymbol{x}_F\rangle \le 8k\|\boldsymbol{w}(F)\odot\boldsymbol{x}_F\|_\infty\right)$. But the random variable $\langle \boldsymbol{w}(F), \boldsymbol{x}_F\rangle$ is Gaussian with a known distribution, and the proof follows by bounding $\|\boldsymbol{w}(F)\odot\boldsymbol{x}_F\|_\infty$. See Appendix B for details.

When the covariance matrix Σ is diagonal, Σ_F is also diagonal and $\widetilde{\Sigma}_F^{1/2}\Sigma_F^{-1/2} = I$. Moreover, $\boldsymbol{\nu}(F) = \boldsymbol{\nu}_F$ where $\boldsymbol{\nu} := \Sigma^{-1/2}\boldsymbol{\mu}$. This yields the following corollary of Theorem 3.2.

Corollary 3.5. *Assume that $\boldsymbol{\mu}, \Sigma$ are given such that (2.3) holds and Σ is diagonal. Then, for nonempty $F \subseteq [d]$ we have*

$$\mathcal{L}_{\boldsymbol{\mu},\Sigma}(\mathcal{C}_F^{(k)}, k) \le \frac{1}{\sqrt{2\log d}} + \bar{\Phi}\left(\|\boldsymbol{\nu}_F\|_2 - \frac{16k\sqrt{2\log d}\|\boldsymbol{\nu}_F\|_\infty}{\|\boldsymbol{\nu}_F\|_2} \right),$$

and in particular

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k) \le \frac{1}{\sqrt{2\log d}} + \min_{F\subseteq[d]} \bar{\Phi}\left(\|\boldsymbol{\nu}_F\|_2 - \frac{16k\sqrt{2\log d}\|\boldsymbol{\nu}_F\|_\infty}{\|\boldsymbol{\nu}_F\|_2} \right).$$

Now we discuss the above bounds via two examples, which we use as running examples to discuss our results in the subsequent sections as well. In the following, $I_d \in \mathbb{R}^{d\times d}$ and $\mathbf{1}_d \in \mathbb{R}^d$ denote the $d\times d$ identity matrix and the all-ones vector of size d, respectively.

Example 3.6. *Let $\Sigma = I_d$ and $\boldsymbol{\mu} = \frac{1}{\sqrt{d}}\mathbf{1}_d$. In the absence of the adversary, the optimal Bayes classification error is $\bar{\Phi}(1)$. Moreover, simplifying the bounds in Corollary 3.5, we get*

$$\mathcal{L}_{\boldsymbol{\mu},\Sigma}(\mathcal{C}_F^{(k)}, k) \le \frac{1}{\sqrt{2\log d}} + \bar{\Phi}\left(\sqrt{\frac{|F|}{d}} - \frac{16k\sqrt{2\log d}}{\sqrt{|F|}} \right).$$

This is minimized when $F = [d]$, resulting in

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k) \le \frac{1}{\sqrt{2\log d}} + \bar{\Phi}\left(1 - \frac{16k\sqrt{2\log d}}{\sqrt{d}}\right).$$

Note that if $k = o(\sqrt{d/\log d})$, the upper bound is approximately $\bar{\Phi}(1)$ which is the optimal classification error in the absence of the adversary. This means that for $k = o(\sqrt{d/\log d})$, the effect of the adversary can be completely neutralized. We will show a lower bound for this example later in Section 3.2 (see Example 3.9 therein) which shows that when $k \ge \sqrt{d}\log d$, no classifier can do asymptotically better than a naive classifier. This establishes a phase transition at $k = \sqrt{d}$ up to logarithmic terms.

EXAMPLE 3.7. *Let $\Sigma = I_d$ and $\boldsymbol{\mu} = (d^{-\frac{1}{3}}, cd^{-\frac{1}{2}}, cd^{-\frac{1}{2}}, \dots, cd^{-\frac{1}{2}})$ where c is chosen such that $\|\boldsymbol{\mu}\|_2 = 1$, resulting in an optimal standard error of $\bar{\Phi}(1)$ in the absence of the adversary. It turns out that the set F that optimizes the bound in Corollary 3.5 is the set $[2:d]$, i.e. we need to discard the first coordinate. In addition to this, we can see that if the classifier does not discard the first coordinate, it can neutralize adversarial attacks with budget of at most $d^{\frac{1}{3}-\epsilon}$, while discarding the first coordinate makes the classifier immune to adversarial budgets up to $d^{\frac{1}{2}-\epsilon}$. In fact, although the first coordinate is more informative compared to the other coordinates, due to this very same reason it is more susceptible to adversarial attacks, and it can do more harm than good when the input is adversarially corrupted. This example highlights the importance of the filtration phase.*

3.2. Lower Bound on Optimal Robust Classification Error: Strategies for the Adversary. In this section, we provide a lower bound on the optimal robust classification error. This is accomplished by introducing an attack strategy for the adversary, and showing that given such a fixed attack, no classifier can achieve better than the lower bound that we introduce. The strategy is best understood when the covariance matrix is diagonal. Therefore, we first assume that Σ is diagonal and denote the diagonal elements of Σ by $\sigma_1^2, \dots, \sigma_d^2$. We later use our strategy for diagonal covariance matrices to get a general lower bound for arbitrary Σ (see Theorem 3.11 at the end of this section).

Assume that the adversary observes realizations $(\boldsymbol{x}, y) \sim \mathcal{D}$ generated from the Gaussian mixture model with parameters $\boldsymbol{\mu}, \Sigma$, where Σ is diagonal. A randomized strategy for the adversary with budget k is identified by a probability distribution which upon observing such realizations $(\boldsymbol{x}, y)$, generates a random vector $\boldsymbol{x}'$ that satisfies $\mathbb{P}\left(\|\boldsymbol{x}' - \boldsymbol{x}\|_0 \le k \mid \boldsymbol{x}, y\right) = 1$. The goal of the adversary is to design this randomized strategy in a way that the corrupted vector $\boldsymbol{x}'$ bears very little information (or even no information) about the label y. In this way, the loss in (2.2) will be maximized. Before rigorously defining our proposed strategy for the adversary, we illustrated its main idea when $d = 1$ in Figure 2.

Recall that $\boldsymbol{\nu} = \Sigma^{-1/2}\boldsymbol{\mu}$. Since Σ is diagonal, $\nu_i = \mu_i/\sigma_i$. We will fix a set of coordinates $A \subseteq [d]$ and a specific value for the budget $k(A) = \|\boldsymbol{\nu}_A\|_1 \log d$. We introduce a randomized strategy for the adversary with the following properties: (i) it can change up to $k(A)$ coordinates of the input; and (ii) all the changed coordinates belong to A, i.e. the coordinates in A^c are left untouched. We denote this adversarial strategy by $\mathsf{Adv}(A)$. Given $A \subset [d]$, having observed $(\boldsymbol{x}, y)$, $\mathsf{Adv}(A)$ follows the procedure explained below. Let $\boldsymbol{Z} = (Z_1, \cdots, Z_d) \in \mathbb{R}^d$ be a random vector that $\mathsf{Adv}(A)$ constructs using the true input $\boldsymbol{x}$. First of all, recall that $\mathsf{Adv}(A)$ does not touch the

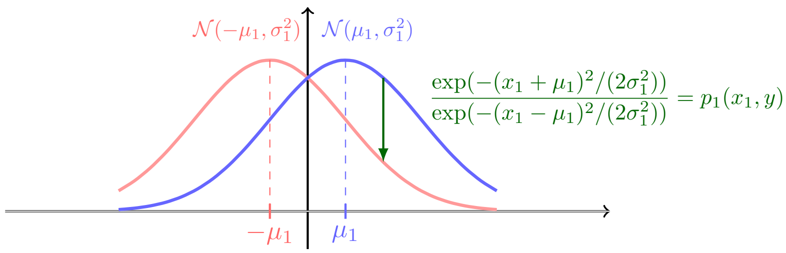

Fig. 2: The idea behind our proposed strategy for the adversary when $d = 1$. Assume $\mu_1 > 0$ and the adversary observes a realization (x_1, y) such that $y = 1$, meaning that x_1 is a realization of $\mathcal{N}(\mu_1, \sigma_1^2)$ (i.e. the blue curve). If $x_1 \leq 0$, the adversary leaves it unchanged, i.e. $x_1' = x_1$. On the other hand, if $x_1 > 0$, we compute the ratio between the two densities (which is precisely $p_1(x_1, y)$ shown in the figure), and with probability $p_1(x_1, y)$ we pick x_1' from an arbitrary distribution (e.g. Uniform$[-1, 1]$). When $y = -1$, we follow a similar procedure, but reversed. It is easy to see that by doing so, the distribution of x_1' is the same when $y = 1$ and $y = -1$, hence x_1' bears no information about y.

coordinates that are not in A, i.e. for $i \in A^c$ we let $Z_i = x_i$. For each $i \in A$, the adversary's act is simple: it either leaves the value unchanged, i.e. $Z_i = x_i$, or it erases the value, i.e. $Z_i \sim \text{Unif}[-1, 1]$–a completely random value between -1 and $+1$. This binary decision is encoded through a Bernoulli random variable I_i taking value 0 with probability $p_i(x_i, y)$ and value 1 otherwise. Here $p_i(x_i, y)$ is defined as

$$p_i(x_i, y) := \begin{cases} \frac{\exp(-(x_i+y\mu_i)^2/2\sigma_i^2)}{\exp(-(x_i-y\mu_i)^2/2\sigma_i^2)} & \text{if } \operatorname{sgn}(x_i) = \operatorname{sgn}(y\mu_i) \\ 0 & \text{otherwise} \end{cases}$$

Note that the condition $\operatorname{sgn}(x_i) = \operatorname{sgn}(y\mu_i)$ ensures that $p_i(x_i, y) \leq 1$. In summary, for each $i \in A$, $\mathsf{Adv}(A)$ lets

$$Z_i = x_i \times I_i + \text{Unif}[-1, 1] \times (1 - I_i), \tag{3.6}$$

where $I_i = \text{Bernoulli}\,(1 - p_i(x_i, y_i))$, and the random variables I_i are generated completely independently w.r.t. all the other variables. It is easy to see that the following holds for the conditional density of $\boldsymbol{Z}_A$ given y

$$\begin{aligned} f_{\boldsymbol{Z}_A|y}(\boldsymbol{z}_A|1) &= f_{\boldsymbol{Z}_A|y}(\boldsymbol{z}_A|-1) \\ &= \prod_{i \in A} \left[\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(|z_i| + |\mu_i|)^2}{2\sigma_i^2} \right) + \frac{\alpha_i}{2} \mathbb{1}\left[z_i \in [-1, 1] \right] \right], \end{aligned} \tag{3.7}$$

where for $i \in A$

$$\alpha_i := \mathbb{P}\left(I_i = 1 | y = 1 \right) = \mathbb{P}\left(I_i = 1 | y = -1 \right) = \int_0^\infty [1 - p_i(t, 1)] f_{x_i|y}(t|1) dt.$$

In other words, α_i is the probability of changing coordinate i. Finally, $\mathsf{Adv}(A)$ checks if the vectors $\boldsymbol{Z}$ and $\boldsymbol{x}$ differ within the budget constraint $k(A) := \|\boldsymbol{\nu}_A\|_1 \log d$. Define $\boldsymbol{x}'$ as follows:

$$\boldsymbol{x}' := \begin{cases} \boldsymbol{Z} & \text{if } \sum_{i \in A} I_i \leq \|\boldsymbol{\nu}_A\|_1 \log d \\ \boldsymbol{x} & \text{o.t.w.} \end{cases} \tag{3.8}$$

It can be shown that with high probability, $\boldsymbol{Z}$ is indeed within the specified budget and $\boldsymbol{x}' = \boldsymbol{Z}$. From this definition, it is evident that with probability one we have

$$\|\boldsymbol{x}' - \boldsymbol{x}\|_0 \leq \|\boldsymbol{\nu}_A\|_1 \log d, \tag{3.9}$$

and hence $\mathsf{Adv}(A)$ is a randomized adversarial strategy that only changes the coordinates in A and has budget $k(A) = \|\boldsymbol{\nu}_A\|_1 \log d$. Now we use this adversarial strategy to show the following result. The proof of Theorem 3.8 is given in Appendix C.

THEOREM 3.8. *Assume that the covariance matrix Σ is diagonal and let $\boldsymbol{\nu} = \Sigma^{-1/2}\boldsymbol{\mu}$. Then for any subset $A \subseteq [d]$, we have*

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}\big(\|\boldsymbol{\nu}_A\|_1 \log d\big) \geq \bar{\Phi}(\|\boldsymbol{\nu}_{A^c}\|_2) - \frac{1}{\log d}.$$

The main idea behind this result and the above adversarial strategy is that due to (3.7), $\boldsymbol{Z}_A$ is independent from y and since the coordinates of the input are independent from each other, and since with high probability $\boldsymbol{x}' = \boldsymbol{Z}$, the coordinates in A have no useful information for the classifier. Hence, the classifier can do no better than the optimal Bayes classifier for the remaining coordinates in A^c, which results in a classification error of $\bar{\Phi}(\|\boldsymbol{\nu}_{A^c}\|_2)$.

We now apply the bound of Thm 3.8 to Examples 3.6, 3.7 that we discussed in Section 3.1.2.

EXAMPLE 3.9. *Assume that $\boldsymbol{\mu}$ and Σ are as in Example 3.6. Applying the bound in Theorem 3.8, we get*

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}\left(\frac{|A|}{\sqrt{d}} \log d\right) \geq \bar{\Phi}\left(\sqrt{1 - \frac{|A|}{d}}\right) - \frac{1}{\log d}.$$

Therefore, setting $A = [d]$, we obtain a lower bound of almost $\bar{\Phi}(0) = 1/2$ for adversarial budget $\sqrt{d}\log d$. In other words, if the adversarial budget is more than $\sqrt{d}\log d$, asymptotically no classifier can do better than a random guess. This together with the discussion in Example 3.6 establishes a phase transition around $\sqrt{d}$ (modulo logarithmic terms).

EXAMPLE 3.10. *Assume that $\boldsymbol{\mu}$ and Σ are as in Example 3.7. Applying the bound of Theorem 3.8 with $A = [d]$, we obtain $\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k) \geq \bar{\Phi}(0) - 1/\log d \approx 1/2$ where $k = (d^{-\frac{1}{3}} + c(d-1)/\sqrt{d}) \log d \approx \sqrt{d}\log d$. Hence, comparing this to Example 3.7, we find similar to Example 3.9 above that a phase transition occurs around adversarial budget $\sqrt{d}$ up to logarithmic terms.*

Now we state our general lower bound which holds for an arbitrary covariance matrix. This is Theorem 3.11 below, whose proof is provided in Appendix D. Given $\boldsymbol{\mu}$ and Σ, we define the $d \times d$ matrix R where the i,j entry in R is $R_{i,j} = \Sigma_{i,j}/\sqrt{\Sigma_{i,i}\Sigma_{j,j}}$. In other words, $R_{i,j}$ is the correlation coefficient between the ith and the jth coordinates in our Gaussian noise. Equivalently, with $\widetilde{\Sigma}$ being the diagonal part of Σ, we may write

$$R := \widetilde{\Sigma}^{-\frac{1}{2}} \Sigma \widetilde{\Sigma}^{-\frac{1}{2}}. \tag{3.10}$$

It is evident that since Σ is assumed to be positive definite, R is also positive definite. Furthermore, we define $\boldsymbol{u} = (u_1, \ldots, u_d)$ where

$$u_i = \frac{\mu_i}{\sqrt{\Sigma_{i,i}}} \qquad 1 \leq i \leq d. \tag{3.11}$$

THEOREM 3.11. *With $\boldsymbol{u}$ and R defined as in (3.10) and (3.11) respectively, for all $A \subseteq [d]$, we have*

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}\left(\frac{1}{\sqrt{\zeta_{min}}}\|\boldsymbol{u}_A\|_1 \log d\right) \geq \bar{\Phi}(\|\boldsymbol{u}_{A^c}\|_2) - \frac{1}{\log d},$$

where $\zeta_{min} > 0$ denotes the minimum eigenvalue of R.

Remark 3.12. Note that when Σ is diagonal, we have $R = I_d$, $\zeta_{\min} = 1$, and $\boldsymbol{u} = \boldsymbol{\nu} = \Sigma^{-1/2}\boldsymbol{\mu}$. Therefore, the bound in Theorem 3.11 reduces to that of Theorem 3.8.

3.3. Optimality of `FilTrun` in the diagonal regime. We have already seen for our two running examples that up to logarithmic terms, our lower and upper bounds match (Examples 3.6 and 3.7 for upper bound, and their matching lower bounds in Examples 3.9 and 3.10, respectively). First, in Section 3.3.1, we show that our lower and upper bounds indeed match up to logarithmic terms in the *diagonal regime*, i.e. when the covariance matrix is diagonal. Then, in Section This in particular implies that our robust classification algorithm `FilTrun` is optimal in this regime.

3.3.1. Comparing the Bounds. In Theorem 3.13 below, in the diagonal regime we compare our upper bound of Corollary 3.5 and our lower bound of Theorem 3.8. Proof of Theorem 3.13 is given in Appendix E. Recall that $\boldsymbol{\nu} := \Sigma^{-1/2}\boldsymbol{\mu}$ and we assume (2.3) holds. When Σ is diagonal and its diagonal entries are $\sigma_1^2, \ldots, \sigma_d^2$, we have $\nu_i = \mu_i/\sigma_i$. Without loss of generality, we may assume that the coordinates of $\boldsymbol{\nu}$ are decreasingly ordered such that

$$|\nu_1| \geq |\nu_2| \geq \cdots \geq |\nu_d|. \tag{3.12}$$

Given $c \in [0, 1]$, we define

$$\lambda_c := \min\{\lambda : \|\boldsymbol{\nu}_{[1:\lambda]}\|_2 \geq c\}. \tag{3.13}$$

THEOREM 3.13. *If Σ is diagonal and the coordinates in $\boldsymbol{\nu}$ are sorted as in (3.12), then:*

1. *For $0 \leq c < 1$, we have*

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}\left(\frac{\|\boldsymbol{\nu}_{[1:\lambda_c]}\|_1}{\log d}\right) \leq \frac{1}{\sqrt{2\log d}} + \bar{\Phi}\left(\sqrt{1-c^2} - \frac{16\sqrt{2}}{\sqrt{1-c^2}\sqrt{\log d}}\right).$$

2. *For $0 < c \leq 1$, we have*

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(\|\boldsymbol{\nu}_{[1:\lambda_c]}\|_1 \log d) \geq \bar{\Phi}(\sqrt{1-c^2}) - \frac{1}{\log d}.$$

Remark 3.14. Roughly speaking, Theorem 3.13 says that up to logarithmic terms, we have

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(\|\boldsymbol{\nu}_{[1:\lambda_c]}\|_1) \approx \bar{\Phi}(\sqrt{1-c^2}).$$

Recall from our previous discussion that we are interested in studying adversarial budgets scaling as d^α, which justifies neglecting the multiplicative logarithmic terms. Furthermore, following the proof of Theorem 3.13, the upper bound in the first part is obtained by our robust classifier by setting $F = \{\lambda_c, \ldots, d\}$. Roughly speaking, the classifier discards the coordinates in $\boldsymbol{\nu}$ which constitute fraction c of the ℓ_2 norm of $\boldsymbol{\nu}$,

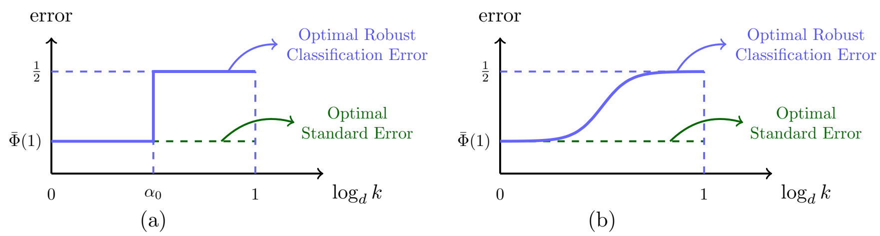

Fig. 3: Asymptotic behavior in the diagonal regime: Illustration of scenarios with (a) a phase transition, and (b) no phase transition

and performs a truncated inner product classification on the remaining coordinates. But the ℓ_2 norm of the remaining coordinates is roughly $\sqrt{1-c^2}$, and the effect of truncation is vanishing as long as the adversarial power is below $\|\boldsymbol{\nu}_{[1:\lambda_c]}\|_1$ by a logarithmic factor. Note that although the top coordinates in $\boldsymbol{\nu}$ are relatively more important in terms of the classification power, due to the same reason, they are more susceptible to adversarial attack.

Remark 3.15. In view of Theorem 3.13 and Remark 3.14, we can introduce the following mechanism for choosing the surviving set F for the adversary given adversarial power k. Let $r(k) = \min\{r : \|\boldsymbol{\nu}_{[1:r]}\|_1 \geq k \log d\}$ and set $F = [r(k) : d]$. Then the classifier $\mathcal{C}_F^{(k)}$ achieves the optimal robust classification error of almost $\bar\Phi(\sqrt{1-c^2})$ where $c = \|\boldsymbol{\nu}_{[1:r(k)]}\|_2$.

3.3.2. Asymptotic Analysis, Phase Transitions, and Trade-offs. In this section, we perform a thorough analysis when the adversarial budget scales as d^α using our results in the diagonal regime. Here, we describe the main messages. (i) We show that our bounds asymptotically match in the diagonal regime and `FilTrun` is indeed optimal. (ii) Through the asymptotic analysis, we observe that in some scenarios, a sharp phase transition on the optimal robust error occurs as we increase $\alpha := \log_d k$ (See Figure 3-(a)). We have already given examples of such scenarios (e.g. Example 3.6). In such cases, below the transition, i.e. when $\alpha < \alpha_0$, the optimal robust error is the same as the optimal standard error. And when we are above the transition, i.e. when $\alpha > \alpha_0$, any classifier becomes useless as the robust error becomes $\frac{1}{2}$. As a result, asymptotically speaking, there exists no tradeoff between robustness and standard accuracy in scenarios where there is a sharp transition.

However, there are other scenarios where instead of a sharp phase transition, in the asymptotic regime, the optimal robust error continuously increases as a function of adversary's budget (see Figure 3-(b)). In such scenarios, there exists a non-trivial tradeoff between robustness and standard accuracy. I.e. to achieve optimal robust error it is necessary to filter many informative coordinates which hurts the standard accuracy. See Example 3.21 below.

In order to perform an asymptotic analysis, we assume that the dimension of the space, d, goes to infinity. More precisely, we assume that we have a sequence $(\boldsymbol{\mu}^{(d)}, \Sigma^{(d)})$ where for each d, $\boldsymbol{\mu}^{(d)} \in \mathbb{R}^d$ and $\Sigma^{(d)}$ is a diagonal covariance matrix with nonzero diagonal entries. We define

$$\boldsymbol{\nu}^{(d)} := (\Sigma^{(d)})^{-1/2} \boldsymbol{\mu}^{(d)}.$$

As usual, as in (2.3), in order to keep the optimal classification error in the absence of the adversary fixed, we assume that

$$\|\boldsymbol{\nu}^{(d)}\|_2 = 1 \qquad \forall d. \tag{3.14}$$

Furthermore, without loss of generality, we assume that the coordinates in $\boldsymbol{\nu}$ are sorted in a descending order with respect to their magnitude, i.e.

$$|\nu_1^{(d)}| \geq |\nu_2^{(d)}| \geq \cdots \geq |\nu_d^{(d)}| \qquad \forall d. \tag{3.15}$$

To simplify the notation, we use $\mathcal{L}_d^*(.)$ as a shorthand for $\mathcal{L}^*_{\boldsymbol{\mu}^{(d)},\Sigma^{(d)}}(.)$. We are mainly interested in studying the asymptotic behavior of $\mathcal{L}_d^*(k_d)$ when k_d is a sequence of adversarial budgets so that k_d behaves like d^α. Motivated by Theorem 3.13, it is natural to define

$$\lambda_c^{(d)} := \min\{\lambda : \|\boldsymbol{\nu}^{(d)}_{[1:\lambda]}\|_2 \geq c\} \qquad \text{for } 0 < c \leq 1. \tag{3.16}$$

Furthermore, for $0 < c \leq 1$, we define

$$\Psi_d(c) := \log_d \|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_1. \tag{3.17}$$

Note that since $c > 0$, $\lambda_c^{(d)} \geq 1$ and $\|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_1 > 0$. Therefore, $\Psi_d(c)$ is well-defined. Furthermore, it is easy to verify the following properties for the function $\Psi_d(.)$:

LEMMA 3.16. *$\Psi_d(.)$ is nonincreasing and $\Psi_d(c) \in [-1/2, 1/2]$ for all $c \in (0,1]$.*

Proof. Note that

$$\Psi_d(c) = \log_d \|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_1 \leq \log_d \|\boldsymbol{\nu}^{(d)}\|_1 \leq \log_d(\sqrt{d}\|\boldsymbol{\nu}^{(d)}\|_2) = \log_d \sqrt{d} = \frac{1}{2}.$$

On the other hand, note that for $c > 0$, we have $\lambda_c^{(d)} \geq 1$ and $\Psi_d(c) \geq \log_d |\nu_1^{(d)}| = \log_d \|\boldsymbol{\nu}\|_\infty$. Furthermore, we have $1 = \|\boldsymbol{\nu}^{(d)}\|_2^2 \leq d\|\boldsymbol{\nu}^{(d)}\|_\infty$ which implies that $\|\boldsymbol{\nu}^{(d)}\|_\infty \geq 1/\sqrt{d}$. Consequently, $\Psi_d(c) \geq \log_d 1/\sqrt{d} = -1/2$. This completes the proof. ∎

Roughly speaking, Theorem 3.13 implies that if k_d behaves like $d^{\Psi_d(c)}$, then $\mathcal{L}^*(k_d) \approx \bar{\Phi}(\sqrt{1-c^2})$. In order to transform this into a formal asymptotic argument, we assume that for all $c \in (0,1]$, the sequence $\Psi_d(c)$ is convergent, and we define $\Psi_\infty(c) := \lim_{d\to\infty} \Psi_d(c)$ as the limit. Since $\Psi_d(.)$ is nondecreasing, if the pointwise limit $\Psi_\infty(.)$ exists, it is also nondecreasing and we may define

$$\Psi_\infty(0) := \lim_{c \downarrow 0} \Psi_d(c).$$

Additionally, we can show the following lemma.

LEMMA 3.17. *If $\Psi_\infty(.)$ exists as above, then $\Psi_\infty(c) \in [0, 1/2]$ for all $c \in [0,1]$.*

Proof. For all $c > 0$ and all d, we have

$$\|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_1 \geq \|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_2^2 \geq c^2.$$

Therefore

$$\Psi_\infty(c) = \lim_{d\to\infty} \Psi_d(c) = \lim_{d\to\infty} \log_d \|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_1 \geq \liminf_{d\to\infty} 2\log_d c = 0.$$

Sending c to zero we also realize that $\Psi_\infty(0) \geq 0$. ∎

Given these, we can formalize the following asymptotic behavior for the optimal robust classification error. The proof of Theorem 3.18 below is given in Appendix F.

THEOREM 3.18. *If $\Psi_d(.)$ converges pointwise to a nondecreasing function $\Psi_\infty : [0,1] \to [0,1/2]$ as above, then the following hold for all $c \in [0,1]$:*

1. *If $\limsup_{d\to\infty} \log_d k_d < \Psi_\infty(c)$, then $\limsup_{d\to\infty} \mathcal{L}_d^*(k_d) \le \bar{\Phi}(\sqrt{1-c^2})$.*
2. *If If $\liminf_{d\to\infty} \log_d k_d > \Psi_\infty(c)$, then $\liminf_{d\to\infty} \mathcal{L}_d^*(k_d) \ge \bar{\Phi}(\sqrt{1-c^2})$.*

It is sometimes more convenient to state the above theorem in terms of the pseudo inverse of the function $\Psi_\infty(.)$ defined as follows. For $\alpha \in [0,1]$, we define

$$\Psi_\infty^{-1}(\alpha) := \inf\{\bar{\Phi}(\sqrt{1-c^2}) : \Psi_\infty(c) \ge \alpha\} \wedge \frac{1}{2}. \tag{3.18}$$

Note that since $\Psi_\infty(c) \le 1/2$ for all $c \in [0,1]$, we have

$$\Psi_\infty^{-1}(\alpha) = \frac{1}{2} \qquad \forall c > \frac{1}{2}.$$

With this, we can restate Theorem 3.18 as follows.

COROLLARY 3.19. *In the setup of Theorem 3.18, for $\alpha \in [0,1]$ we have*

1. *If $\limsup \log_d k_d < \alpha$ then $\limsup \mathcal{L}_d^*(k_d) \le \Psi_\infty^{-1}(\alpha)$.*
2. *If $\liminf \log_d k_d > \alpha$ then $\liminf \mathcal{L}_d^*(k_d) \ge \Psi_\infty^{-1}(\alpha)$.*

We now discuss this asymptotic result through some examples.

EXAMPLE 3.20. *Let $\boldsymbol{\mu}^{(d)}$ and $\Sigma^{(d)}$ be as in Example 3.6, i.e. $\Sigma^{(d)} = I_d$ and $\boldsymbol{\mu}^{(d)} = \frac{1}{\sqrt{d}}\mathbf{1}_d$. Therefore, we have*

$$\boldsymbol{\nu}^{(d)} = (\Sigma^{(d)})^{-\frac{1}{2}}\boldsymbol{\mu}^{(d)} = \left(\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}, \dots, \frac{1}{\sqrt{d}}\right).$$

Using (3.16), we have $\lambda_c^{(d)} = \lfloor dc^2 \rfloor$ and

$$\Psi_d(c) = \log_d \|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_1 = \log_d \frac{\lfloor dc^2 \rfloor}{\sqrt{d}} = \frac{1}{2} + o(1).$$

Therefore, sending $d \to \infty$, we realize that

$$\Psi_\infty(c) = \frac{1}{2} \qquad \forall c \in [0,1].$$

Moreover, using (3.18), we get

$$\Psi_\infty^{-1}(\alpha) = \begin{cases} \bar{\Phi}(1) & \alpha \le \frac{1}{2} \\ \frac{1}{2} & \alpha > \frac{1}{2}. \end{cases}$$

Figure 4 illustrates $\Psi_\infty(.)$ and $\Psi_\infty^{-1}(.)$ for this example. Therefore, employing Corollary 3.19, we realize that

1. *If $\limsup \log_d k_d < 1/2$ then $\limsup \mathcal{L}_d^*(k_d) \le \bar{\Phi}(1)$*
2. *If $\liminf \log_d k_d > 1/2$ then $\mathcal{L}^*(k_d) \ge 1/2$.*

In other words, we observe a phase transition around $\sqrt{d}$ in the sense that if the adversary's budget is asymptoticallly below $\sqrt{d}$, the classifier can achieve the robust classification error $\bar{\Phi}(1)$, i.e. as if there is no adversary, while if the adversary's budget is asymptotically above $\sqrt{d}$, no classifier can achieve a robust classification error better than that of a trivial classifier. This is consistent with the previous observations in this case, i.e. Examples 3.6 and 3.9.

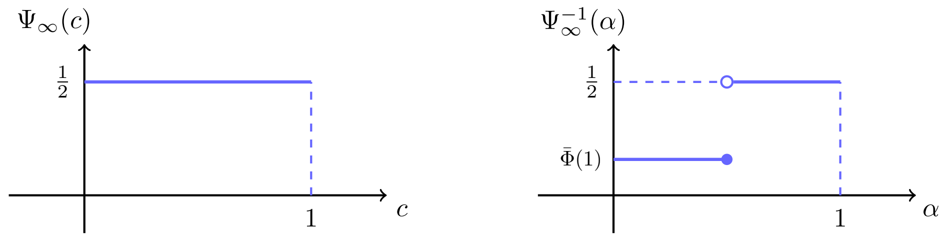

Fig. 4: $\Psi_\infty(.)$ and $\Psi_\infty^{-1}(.)$ for Example 3.20. This observe a phase transition at $\sqrt{d}$ where below this threshold, adversary's effect can completely be neutralized, while above this threshold, the classifier can only achieve the trivial bound.

It is interesting to observe that not always we have a phase transition as in the above example. Below we discuss an example in which we have no phase transition, and the asymptotic robust classification error gradually increases as a function of the adversary's budget.

Example 3.21. *Let $\Sigma = I_d$. Assume that $d = 2^n - 1$ for some integer n and define*

$$\boldsymbol{\mu}^{(d)} = \left(\frac{\sqrt{1/n}}{1}, \frac{\sqrt{1/n}}{\sqrt{2}}, \frac{\sqrt{1/n}}{\sqrt{2}}, \ldots, \frac{\sqrt{1/n}}{\sqrt{d/2}}, \ldots, \frac{\sqrt{1/n}}{\sqrt{d/2}} \right).$$

More precisely, we split the unit ℓ_2 norm of $\boldsymbol{\mu}^{(d)}$ into n blocks, where the first block is the first coordinate, the second block is the second two coordinate, the ith block constitutes of 2^i coordinates, and the final block is the last $d/2$ coordinates. Moreover, the power is uniformly distributed within each block. It is easy to see that for $c = \sqrt{m/n}$ for $1 \le m \le n$, we have $\lambda_c^{(d)} = 2^m - 1$ and

$$\Psi_d(c) = \Psi_d\left(\sqrt{\frac{m}{n}}\right) = \log_d\left(\sqrt{\frac{1}{n}}\frac{\sqrt{2}^m - 1}{\sqrt{2} - 1}\right) = \frac{c^2}{2} + o(1).$$

Therefore, $\Psi_d(.)$ converges pointwise to $\Psi_\infty(.)$ such that $\Psi_\infty(c) = c^2/2$ for $0 \le c \le 1$. Thereby, we have

$$\Psi_\infty^{-1}(\alpha) = \begin{cases} \bar{\Phi}(1 - 2\alpha) & 0 \le \alpha \le 1/2 \\ \frac{1}{2} & 1/2 < \alpha \le 1. \end{cases}$$

Figure 5 illustrates $\Psi_\infty(.)$ and $\Psi_\infty^{-1}(.)$ in this examples. As we can see, unlike Example 3.20, we do not have a phase transition here. In fact, the asymptotic optimal robust classification error continuously increases as a function of adversarial ℓ_0 budget.

4. Conclusion. In this paper, we studied the binary Gaussian mixture model under ℓ_0 attack. We developed a novel nonlinear classifier called `FilTrun` that first cleverly selects the robust coordinates of the input and then classifies based on a truncated inner product operation. Analyzing the performance of our proposed method, we derived an upper bound on optimal robust classification error. We further derived a lower bound on this, and showed the efficacy of `FilTrun`: when the covariance matrix of Gaussian mixtures is diagonal, `FilTrun` is asymptotically optimal.

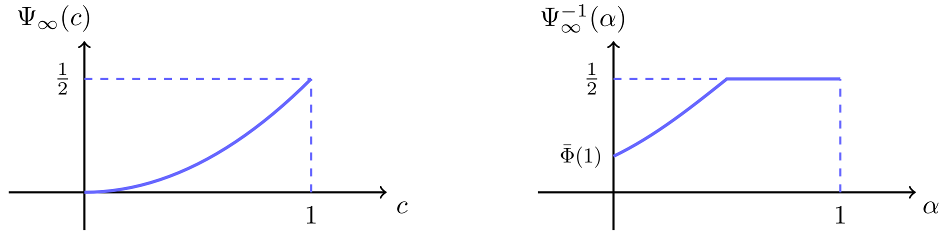

Fig. 5: $\Psi_\infty(.)$ and $\Psi_\infty^{-1}(.)$ for Examples 3.21. Unlike Example 3.20, we do not have a phase transition here and the asymptotic optimal robust classification error continuously increases as a function of the adversarial ℓ_0 budget.

There are many directions to be pursued. Deriving a tighter lower bound and resolving the optimality gap for the case of non-diagonal covariance matrices remains open. Applying the key ideas of `FilTrun`, filtration and truncation, to a more complicated setting (e.g. neural networks) can be of great importance from a practical viewpoint. A crucial message of this paper is to emphasize the importance of non-linear operations such as truncation for designing defense against ℓ_0 attacks. Finally, analyzing robust classification error with ℓ_0 attacks for more complex stylized models such as multi-class Gaussian mixtures, two-layer neural networks, neural tangent kernel models, etc. is a promising future direction.

REFERENCES

[1] A. A. Al Makdah, V. Katewa, and F. Pasqualetti, *A fundamental performance limitation for adversarial classification*, IEEE Control Systems Letters, 4 (2019), pp. 169–174.

[2] A. Athalye, N. Carlini, and D. A. Wagner, *Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples*, in Proceedings of the 35th International Conference on Machine Learning, ICML, Stockholm, Sweden, July 10-15, 2018, pp. 274–283, http://proceedings.mlr.press/v80/athalye18a.html.

[3] A. N. Bhagoji, D. Cullina, and P. Mittal, *Lower bounds on adversarial robustness from optimal transport*, in Advances in Neural Information Processing Systems, 8-14 December 2019, Vancouver, BC, Canada, 2019, pp. 7496–7508, http://papers.nips.cc/paper/8968-lower-bounds-on-adversarial-robustness-from-optimal-transport.

[4] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, *Evasion attacks against machine learning at test time*, in Joint European conference on machine learning and knowledge discovery in databases, Springer, 2013, pp. 387–402.

[5] N. Carlini and D. A. Wagner, *Towards evaluating the robustness of neural networks*, in 2017 IEEE Symposium on Security and Privacy, San Jose, CA, USA, May 22-26,, 2017, pp. 39–57, https://doi.org/10.1109/SP.2017.49, https://doi.org/10.1109/SP.2017.49.

[6] L. Chen, Y. Min, M. Zhang, and A. Karbasi, *More data can expand the generalization gap between adversarially robust and standard models*, in International Conference on Machine Learning, PMLR, 2020, pp. 1670–1680.

[7] F. Croce, M. Andriushchenko, N. D. Singh, N. Flammarion, and M. Hein, *Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks*, arXiv preprint arXiv:2006.12834, (2020).

[8] C. Dan, Y. Wei, and P. Ravikumar, *Sharp statistical guaratees for adversarially robust gaussian classification*, in International Conference on Machine Learning, PMLR, 2020, pp. 2345–2355.

[9] E. Dobriban, H. Hassani, D. Hong, and A. Robey, *Provable tradeoffs in adversarially robust classification*, arXiv preprint arXiv:2006.05161, (2020).

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and harnessing adversarial examples*, arXiv preprint arXiv:1412.6572, (2014).

[11] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, *Adversarial perturbations against deep neural networks for malware classification*, arXiv preprint arXiv:1606.04435, (2016).

[12] J. Hayes, *Provable trade-offs between private & robust machine learning*, arXiv preprint arXiv:2006.04622, (2020).

[13] A. Javanmard, M. Soltanolkotabi, and H. Hassani, *Precise tradeoffs in adversarial training for linear regression*, in Conference on Learning Theory, PMLR, 2020, pp. 2034–2078.

[14] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, *Is bert really robust? natural language attack on text classification and entailment*, arXiv preprint arXiv:1907.11932, 2 (2019).

[15] A. Levine and S. Feizi, *Robustness certificates for sparse adversarial attacks by randomized ablation.*, in AAAI, 2020, pp. 4585–4593.

[16] J. Li, F. Schmidt, and Z. Kolter, *Adversarial camera stickers: A physical camera-based attack on deep learning systems*, in International Conference on Machine Learning, PMLR, 2019, pp. 3896–3904.

[17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards deep learning models resistant to adversarial attacks*, arXiv preprint arXiv:1706.06083, (2017).

[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards deep learning models resistant to adversarial attacks*, in 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018, https://openreview.net/forum?id=rJzIBfZAb.

[19] Z. Marzi, S. Gopalakrishnan, U. Madhow, and R. Pedarsani, *Sparsity-based defense against adversarial attacks on linear classifiers*, in 2018 IEEE International Symposium on Information Theory, ISIT, Vail, CO, USA, June 17-22, 2018, pp. 31–35, https://doi.org/10.1109/ISIT.2018.8437638, https://doi.org/10.1109/ISIT.2018.8437638.

[20] Y. Min, L. Chen, and A. Karbasi, *The curious case of adversarially robust models: More data can help, double descend, or hurt generalization*, arXiv preprint arXiv:2002.11080, (2020).

[21] A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard, *Sparsefool: a few pixels make a big difference*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9087–9096.

[22] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, *The limitations of deep learning in adversarial settings*, in 2016 IEEE European symposium on security and privacy (EuroS&P), IEEE, 2016, pp. 372–387.

[23] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, *Distillation as a defense to adversarial perturbations against deep neural networks*, in 2016 IEEE Symposium on Security and Privacy (SP), IEEE, 2016, pp. 582–597.

[24] B. Puranik, U. Madhow, and R. Pedarsani, *Adversarially robust classification based on glrt*, arXiv preprint arXiv:2011.07835, (2020).

[25] M. S. Pydi and V. Jog, *Adversarial risk via optimal transport and optimal couplings*, in International Conference on Machine Learning, PMLR, 2020, pp. 7814–7823.

[26] A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang, *Adversarial training can hurt generalization*, arXiv preprint arXiv:1906.06032, (2019).

[27] E. Richardson and Y. Weiss, *A bayes-optimal view on adversarial examples*, arXiv preprint arXiv:2002.08859, (2020).

[28] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, *Adversarially robust generalization requires more data*, in Advances in Neural Information Processing Systems, 2018, pp. 5014–5026.

[29] L. Schott, J. Rauber, M. Bethge, and W. Brendel, *Towards the first adversarially robust neural network model on mnist*, arXiv preprint arXiv:1805.09190, (2018).

[30] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein, *Are adversarial examples inevitable?*, arXiv preprint arXiv:1809.02104, (2018).

[31] A. Shamir, I. Safran, E. Ronen, and O. Dunkelman, *A simple explanation for the existence of adversarial examples with small hamming distance*, arXiv preprint arXiv:1901.10861, (2019).

[32] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao, *Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models*, in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 631–648.

[33] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, *Intriguing properties of neural networks*, in International Conference on Learning Representations, 2014, Banff, AB, Canada, April 14-16, 2014, http://arxiv.org/abs/1312.6199.

[34] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, *Robustness may be at odds with accuracy*, in International Conference on Learning Representations, no. 2019, 2019.

[35] E. Wong and J. Z. Kolter, *Provable defenses against adversarial examples via the convex outer adversarial polytope*, in Proceedings of the 35th International Conference on Machine Learning, ICML, Stockholm, Sweden, July 10-15, 2018, http://proceedings.mlr.press/v80/wong18a.html.

[36] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, *Theoretically principled trade-off between robustness and accuracy*, in International Conference on Machine Learning, PMLR, 2019, pp. 7472–7482.

Appendix A. Proof of Lemma 3.1.

In this section, we prove Lemma 3.1. First we need to define some notations and discuss some lemmas.

Given $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$, we define the sample average of $\boldsymbol{x}$ as $\mathsf{Mean}(\boldsymbol{x}) := \sum_{i=1}^{d} x_i/d$. Moreover, we define *truncated sum* $\mathsf{TSum}_k(\boldsymbol{x})$ for $k < n/2$ as follows. Let $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$ be the set of sorted values in $\boldsymbol{x}$. We define

$$\mathsf{TSum}_k(\boldsymbol{x}) := \sum_{i=k+1}^{d-k} x_{(i)},$$

which is the truncated sum of the elements in $\boldsymbol{x}$ after removing the top and bottom k values. For instance, $\mathsf{TSum}_1(1, 1, 2, 3, 4, 5) = 1 + 2 + 3 + 4 = 10$. Moreover, we define the truncated mean of $\boldsymbol{x}$ as follows:

$$\mathsf{TMean}_k(\boldsymbol{x}) := \frac{\mathsf{TSum}_k(S)}{d - 2k}.$$

Note that when $k = 0$, the above quantities reduce to the sum and the sample average, respectively. It is straightforward to see that

$$\text{(A.1)} \qquad \left| \mathsf{TSum}_k(\boldsymbol{x}) - \sum_{i=1}^{n} x_i \right| \leq 2kM \qquad \text{given } |x_i| \leq M \quad \forall 1 \leq i \leq n.$$

Lemma A.1. *Assume that $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and $\boldsymbol{x}' = (x'_1, \ldots, x'_d) \in \mathbb{R}^d$ are given such that $\boldsymbol{x}'$ is identical to $\boldsymbol{x}$ in all but at most $k < d/2$ coordinates, i.e. $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq k$. Moreover, assume that for some $M < \infty$, we have $|x_i| \leq M$ for all $1 \leq i \leq d$. Then, if $x'_{(1)} \leq x'_{(2)} \leq \cdots \leq x'_{(d)}$ are the sorted coordinates in $\boldsymbol{x}'$, we have*

$$|x'_{(i)}| \leq M \qquad \forall\, k + 1 \leq i \leq d - k.$$

Essentially, what Lemma A.1 states is that if we modify at most k coordinates in a vector whose elements are bounded by M, in the resulting vector, after truncating the top and bottom k coordinates, all the surviving values are also bounded by M.

Proof of Lemma A.1. Let $i_1, \ldots, i_l$ for $l \leq k$ be the coordinates where $\boldsymbol{x}'$ differs from $\boldsymbol{x}$, i.e. $x_{i_j} \neq x'_{i_j}$ for $1 \leq j \leq l$. Note that if $|x'_{i_j}| > M$ for any of $1 \leq j \leq l$, then x'_{i_j} will definitely fall into the top or bottom k coordinates in the sorted list $x'_{(1)} \leq \cdots \leq x'_{(d)}$, since all the $d - l \geq d - k$ remaining coordinates in $\boldsymbol{x}'$ are bounded by M. This means that all the surviving coordinates $x'_{(k+1)}, \ldots, x'_{(d-k)}$ after truncating top and bottom k coordinates in $\boldsymbol{x}'$ are indeed bounded by M which completes the proof. ∎

Lemma A.2. *Assume that $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ is given such that $|x_i| \leq M$ for all $1 \leq i \leq d$. Also, assume that $\boldsymbol{x}' = (x'_1, \ldots, x'_d) \in \mathbb{R}^d$ is identical to $\boldsymbol{x}$ in all but at most k coordinates, i.e. $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq k$. Then, we have*

$$|\mathsf{TSum}_k(\boldsymbol{x}) - \mathsf{TSum}_k(\boldsymbol{x}')| \leq 6kM.$$

Proof. Let $x_{\sigma(1)} \leq \cdots \leq x_{\sigma(d)}$ and $x'_{\sigma'(1)} \leq \cdots \leq x'_{\sigma'(d)}$ be the sorted elements in

$\boldsymbol{x}$ and $\boldsymbol{x}'$ with permutations σ and σ', respectively. Following the definition, we have

$$
\begin{aligned}
\mathsf{TSum}_k(\boldsymbol{x}) &= \sum_{i=k+1}^{d-k} x_{\sigma(i)} = \sum_{i:\sigma^{-1}(i)\in\{k+1,\dots,d-k\}} x_i \\
&= \sum_{i=1}^{d} \mathbb{1}\left[\sigma^{-1}(i) \in \{k+1,\dots,d-k\}\right] x_i.
\end{aligned}
$$

Similarly, we have

$$
\mathsf{TSum}_k(\boldsymbol{x}') = \sum_{i=1}^{d} \mathbb{1}\left[{\sigma'}^{-1}(i) \in \{k+1,\dots,d-k\}\right] x'_i.
$$

To simplify the notation, for $1 \le i \le d$, we define

$$
y_i := \mathbb{1}\left[\sigma^{-1}(i) \in \{k+1,\dots,d-k\}\right] x_i,
$$

and

$$
y'_i := \mathbb{1}\left[{\sigma'}^{-1}(i) \in \{k+1,\dots,d-k\}\right] x'_i.
$$

Moreover, let

$$
\begin{aligned}
&A_1 := \{1 \le i \le d : \sigma^{-1}(i) \in \{k+1,\dots,d-k\} \text{ and } {\sigma'}^{-1}(i) \notin \{k+1,\dots,d-k\}\} \\
&A_2 := \{1 \le i \le d : \sigma^{-1}(i) \notin \{k+1,\dots,d-k\} \text{ and } {\sigma'}^{-1}(i) \in \{k+1,\dots,d-k\}\} \\
&A_3 := \{1 \le i \le d : \sigma^{-1}(i) \in \{k+1,\dots,d-k\} \\
&\qquad\qquad \text{and } {\sigma'}^{-1}(i) \in \{k+1,\dots,d-k\} \text{ and } x_i \ne x'_i\} \\
&A := A_1 \cup A_2 \cup A_3.
\end{aligned}
$$

Note that if $i \notin A$, either $\sigma^{-1}(i) \notin \{k+1,\dots,d-k\}$ and ${\sigma'}^{-1}(i) \notin \{k+1,\dots,d-k\}$, in which case $y_i = y'_i = 0$; or $\sigma^{-1}(i) \in \{k+1,\dots,d-k\}$, ${\sigma'}^{-1}(i) \in \{k+1,\dots,d-k\}$, and $x_i = x'_i$, in which case $y_i = y'_i = x_i = x'_i$. This means that $y_i = y'_i$ for $i \notin A$ and

$$
\begin{aligned}
|\mathsf{TSum}_k(\boldsymbol{x}) - \mathsf{TSum}_k(\boldsymbol{x}')| &\le \sum_{i\in A} |y_i - y'_i| \\
&\le \sum_{i\in A_1} |y_i - y'_i| + \sum_{i\in A_2} |y_i - y'_i| + \sum_{i\in A_3} |y_i - y'_i|.
\end{aligned}
\tag{A.2}
$$

Note that for $i \in A_1$, we have $y'_i = 0$ and $y_i = x_i$, implying $|y_i - y'_i| = |x_i| \le M$. On the other hand, for $i \in A_2$, $y_i = 0$ and $y'_i = x'_i$. But since ${\sigma'}^{-1}(i) \in \{k+1,\dots,d-k\}$, using Lemma A.1, we have $|y_i - y'_i| = |x'_i| \le M$. Moreover, for $i \in A_3$, we have $y_i = x_i$ and $y'_i = x'_i$. Also, from Lemma A.1, we have $|x'_i| \le M$. Thereby, $|y_i - y'_i| \le |x_i| + |x'_i| \le 2M$. Putting all these together, we get

$$
\sum_{i\in A_1} |y_i - y'_i| + \sum_{i\in A_2} |y_i - y'_i| + \sum_{i\in A_3} |y_i - y'_i| \le M|A_1| + M|A_2| + 2M|A_3|. \tag{A.3}
$$

Observe that

$$
|A_1| \le |\{1 \le i \le d : {\sigma'}^{-1}(i) \notin \{k+1,\dots,d-k\}\}| = 2k. \tag{A.4}
$$

Similarly,

$$|A_2| \leq 2k. \tag{A.5}$$

On the other hand,

$$|A_3| \leq |\{1 \leq i \leq d : x_i \neq x_i'\}| \leq k. \tag{A.6}$$

Using (A.4), (A.5), and (A.6) back into (A.3) and comparing with (A.2), we realize that

$$|\mathsf{TSum}_k(\boldsymbol{x}) - \mathsf{TSum}_k(\boldsymbol{x}')| \leq 6kM,$$

which completes the proof. ∎

The following is a direct consequence of Lemma A.2.

Corollary A.3. *Given* $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ *and integer* k *satisfying* $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq k < d/2$, *we have*

$$|\mathsf{TSum}_k(\boldsymbol{x}) - \mathsf{TSum}_k(\boldsymbol{x}')| \leq 6k \min\{\|\boldsymbol{x}\|_\infty, \|\boldsymbol{x}'\|_\infty\}.$$

We are now ready to give the proof of Lemma 3.1:

Proof of Lemma 3.1. We have

$$\begin{aligned}
|\langle \boldsymbol{w}, \boldsymbol{x}' \rangle_k - \langle \boldsymbol{w}, \boldsymbol{x} \rangle| &\leq |\langle \boldsymbol{w}, \boldsymbol{x}' \rangle_k - \langle \boldsymbol{w}, \boldsymbol{x} \rangle_k| + |\langle \boldsymbol{w}, \boldsymbol{x} \rangle_k - \langle \boldsymbol{w}, \boldsymbol{x} \rangle| \\
&\leq |\langle \boldsymbol{w}, \boldsymbol{x}' \rangle_k - \langle \boldsymbol{w}, \boldsymbol{x} \rangle_k| + 2k\|\boldsymbol{w} \odot \boldsymbol{x}\|_\infty \\
&= |\mathsf{TSum}_k(\boldsymbol{w} \odot \boldsymbol{x}') - \mathsf{TSum}_k(\boldsymbol{w} \odot \boldsymbol{x})| + 2k\|\boldsymbol{w} \odot \boldsymbol{x}\|_\infty \\
&\overset{(a)}{\leq} 6k\|\boldsymbol{w} \odot \boldsymbol{x}\|_\infty + 2k\|\boldsymbol{w} \odot \boldsymbol{x}\|_\infty \\
&= 8k\|\boldsymbol{w} \odot \boldsymbol{x}\|_\infty,
\end{aligned}$$

where in step (a) we have used $\|\boldsymbol{w} \odot \boldsymbol{x}' - \boldsymbol{w} \odot \boldsymbol{x}\|_0 \leq \|\boldsymbol{x}' - \boldsymbol{x}\|_0 \leq k$ together with Corollary A.3. This completes the proof. ∎

Appendix B. Proof of the Upper Bound (Theorem 3.2).

Given $\boldsymbol{x} \in \mathbb{R}^d$ and $y \in \{\pm 1\}$, define

$$\ell^{(k)}(\mathcal{C}_F^{(k)}; \boldsymbol{x}, y) := \max_{\boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x}, k)} \ell(\mathcal{C}_F^{(k)}; \boldsymbol{x}', y).$$

We have

$$\begin{aligned}
\ell^{(k)}(\mathcal{C}_F^{(k)}; \boldsymbol{x}, 1) &= \mathbb{1}\left[\exists \boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x}, k) : \mathcal{C}_F^{(k)}(\boldsymbol{x}') \neq 1\right] \\
&= \mathbb{1}\left[\exists \boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x}, k) : \langle \boldsymbol{w}(F), \boldsymbol{x}_F' \rangle_k \leq 0\right]
\end{aligned}$$

Using Lemma 3.1, for $\boldsymbol{x}'$ such that $\|\boldsymbol{x}' - \boldsymbol{x}\|_0 \leq 0$, since $\|\boldsymbol{x}_F' - \boldsymbol{x}_F\|_0 \leq \|\boldsymbol{x}' - \boldsymbol{x}\|_0 \leq k$, we have

$$|\langle \boldsymbol{w}(F), \boldsymbol{x}_F' \rangle_k - \langle \boldsymbol{w}(F), \boldsymbol{x}_F \rangle| \leq 8k\|\boldsymbol{w}(F) \odot \boldsymbol{x}_F\|_\infty.$$

This means that

$$\mathbb{1}\left[\exists \boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x}, k) : \langle \boldsymbol{w}(F), \boldsymbol{x}_F' \rangle_k \leq 0\right] \leq \mathbb{1}\left[\langle \boldsymbol{w}(F), \boldsymbol{x}_F \rangle \leq 8k\|\boldsymbol{w}(F) \odot \boldsymbol{x}_F\|_\infty\right],$$

and

(B.1)
$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\ell^{(k)}(\mathcal{C}_F^{(k)};\boldsymbol{x},1)|y=1\right] \le \mathbb{P}\left(\langle \boldsymbol{w}(F), \boldsymbol{x}_F\rangle \le 8k\|\boldsymbol{w}(F)\odot \boldsymbol{x}_F\|_\infty \,|\, y=1\right).$$

Let Σ_F be as defined in (3.3) and let $\widetilde{\Sigma}_F$ be the diagonal part of Σ_F. Note that since Σ is positive definite, $\widetilde{\Sigma}_F$ is diagonal with positive diagonal entries. Hence, we may write

(B.2)
$$\|\boldsymbol{w}(F)\odot \boldsymbol{x}_F\|_\infty = \|(\widetilde{\Sigma}^{1/2}\boldsymbol{w}(F))\odot(\widetilde{\Sigma}^{-1/2}\boldsymbol{x}_F)\|_\infty \le \|\widetilde{\Sigma}_F^{1/2}\boldsymbol{w}(F)\|_\infty\|\widetilde{\Sigma}_F^{-1/2}\boldsymbol{x}_F\|_\infty.$$

Let σ_i^2 denote the ith diagonal coordinate of Σ. Fix $i\in F$ and note that conditioned on $y=1$, we have $x_i\sim\mathcal{N}(\mu_i,\sigma_i^2)$. On the other hand, with $\boldsymbol{a} := \widetilde{\Sigma}_F^{-1/2}\boldsymbol{x}_F$, we have $a_i\sim\mathcal{N}(\sigma_i^{-1}\mu_i,1)$. Note that $\bar{\Phi}(\sigma_i^{-1}\mu_i)$ is the optimal Bayes classification error of y given x_i only, which is indeed not smaller than the optimal Bayes classification error of y given the whole vector $\boldsymbol{x}$, which is in turn equal to $\bar{\Phi}(\|\boldsymbol{\nu}\|_2)=\bar{\Phi}(1)$. Since $\bar{\Phi}$ is decreasing, this implies $\sigma_i^{-1}\mu_i\le 1$. Consequently, by union bound, we have

$$\begin{aligned}
\mathbb{P}\left(\|\widetilde{\Sigma}_F^{-1/2}\boldsymbol{x}_F\|_\infty > 1+\sqrt{2\log d}\right) &\le \sum_{i\in F}\mathbb{P}\left(a_i-\sigma_i^{-1}\mu_i > \sqrt{2\log d}\right)\\
&\le d\bar{\Phi}(\sqrt{2\log d})\\
&\le d\frac{1}{\sqrt{2\pi}\sqrt{2\log}}e^{-\log d}\\
&\le \frac{1}{\sqrt{2\log d}}.
\end{aligned}$$

Thereby, we get

$$\mathbb{P}\left(\|\widetilde{\Sigma}_F^{-1/2}\boldsymbol{x}_F\|_\infty > 2\sqrt{2\log d}\,|\, y=1\right) \le \frac{1}{\sqrt{2\log d}}. \tag{B.3}$$

On the other hand, we have

$$\|\widetilde{\Sigma}_F^{1/2}\boldsymbol{w}(F)\|_\infty = \|\widetilde{\Sigma}_F^{1/2}\Sigma_F^{-1/2}\boldsymbol{\nu}(F)\|_\infty \le \|\widetilde{\Sigma}_F^{1/2}\Sigma_F^{-1/2}\|_\infty\|\boldsymbol{\nu}(F)\|_\infty, \tag{B.4}$$

where $\|\widetilde{\Sigma}_F^{1/2}\Sigma_F^{-1/2}\|_\infty$ denotes the operator norm of $\widetilde{\Sigma}_F^{1/2}\Sigma_F^{-1/2}$ induced by the vector ℓ_∞ norm. Using (B.2), (B.3), and (B.4) back into (B.1) and simplifying, we get

$$\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\ell^{(k)}(\mathcal{C}_F^{(k)};\boldsymbol{x},1)|y=1\right]\\
&\quad\le \frac{1}{\sqrt{2\log d}} + \mathbb{P}\left(\langle \boldsymbol{w}(F),\boldsymbol{x}_F\rangle \le 16k\sqrt{2\log d}\|\widetilde{\Sigma}_F^{1/2}\Sigma_F^{-1/2}\|_\infty\|\boldsymbol{\nu}(F)\|_\infty \,|\, y=1\right)
\end{aligned}$$

It is easy to see that conditioned on $y=1$, $\langle \boldsymbol{w}(F),\boldsymbol{x}_F\rangle\sim\mathcal{N}(\|\boldsymbol{\nu}(F)\|_2^2,\|\boldsymbol{\nu}(F)\|_2^2)$. Using this in the above bound, we get

$$\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\ell^{(k)}(\mathcal{C}_F^{(k)};\boldsymbol{x},1)|y=1\right]\\
&\quad\le \frac{1}{\sqrt{2\log d}} + \bar{\Phi}\left(\|\boldsymbol{\nu}(F)\|_2 - \frac{16k\sqrt{2\log d}\|\widetilde{\Sigma}_F^{1/2}\Sigma_F^{-1/2}\|_\infty\|\boldsymbol{\nu}(F)\|_\infty}{\|\boldsymbol{\nu}(F)\|_2}\right).
\end{aligned}$$

Due to the symmetry, we have the same bound conditioned on $y = -1$ which yields the desired result.

Appendix C. Lower Bound in the Diagonal Regime (Theorem 3.8).

Before giving the proof of Theorem 3.8, we need the following lemma.

Lemma C.1. *For any random adversarial strategy with budget k which has a density function $f_{\boldsymbol{x}'|\boldsymbol{x},y}$, we have*

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k) \geq \frac{1}{2}\mathbb{P}\left(f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|1) = f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|-1)\right) + \mathbb{P}\left(f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|-1) > f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|1)\middle| y=1\right),$$

Proof. Note that the right hand side is indeed the Bayes optimal error associated with the MAP estimator assuming that the classifier knows adversary's strategy. Since the classifier does not know the adversary's strategy in general, the right hand side is indeed a lower bound on the optimal robust classification error. ∎

Now we are ready to prove Theorem 3.8.

Proof of Theorem 3.8. Note that when A is empty, there is no adversarial modification and the standard Bayes analysis implies that $\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(0) = \bar{\Phi}(\|\boldsymbol{\nu}\|_2) = \bar{\Phi}(\|\boldsymbol{\nu}_{A^c}\|_2)$ and the desired bound holds. Hence, we may assume that A is nonempty for the rest of the proof.

Note that due to (3.9), the randomized strategy $\mathsf{Adv}(A)$ is valid for the adversary given the budget $\|\boldsymbol{\nu}\|_1 \log d$. Thereby we may use Lemma C.1 with $\mathsf{Adv}(A)$ to bound $\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(\|\boldsymbol{\nu}_A\|_1 \log d)$ from below. Before that, we show that with high probability under the above randomized strategy for the adversary, recalling the definition of random variables I_i for $i \in A$ from (3.6), we have $\sum_{i\in A} I_i \leq \|\boldsymbol{\nu}_A\|_1 \log d$ and hence $\boldsymbol{x}' = \boldsymbol{Z}$. It is easy to see that for each i, $\mathbb{P}(I_i = 1|y = 1) = \mathbb{P}(I_i = 1|y = -1)$; therefore,

$$\begin{aligned}
\mathbb{P}(I_i = 1) &= \mathbb{P}(I_i = 1|y = \mathrm{sgn}(\mu_i)) \\
&= \int_0^\infty [1 - p_i(t, \mathrm{sgn}(\mu_i))] f_{x_i|y}(t|\mathrm{sgn}(\mu_i))dt \\
&= \int_0^\infty \left[1 - \frac{\exp(-(t+|\mu_i|)^2/2\sigma_i^2)}{\exp(-(t-|\mu_i|)^2/2\sigma_i^2)}\right] \exp\left(-(t-|\mu_i|)^2/2\sigma_i^2\right) dt \\
&= 1 - \bar{\Phi}(|\nu_i|) \\
&= \mathrm{Erf}(|\nu_i|/\sqrt{2}) \\
&\leq \left(\sqrt{\frac{2}{\pi}}|\nu_i|\right) \wedge 1.
\end{aligned}$$

Hence, we have

$$\mathbb{P}(I_i = 1) = \mathbb{P}(I_i = 1|y = 1) = \mathbb{P}(I_i = 1|y = -1) \leq \left(\sqrt{\frac{2}{\pi}}|\nu_i|\right) \wedge 1.$$

Therefore, using Markov's inequality, if I is the indicator of the event $\sum_{i\in A} I_i > \|\boldsymbol{\nu}_A\|_1 \log d$, we have

$$\mathbb{P}(I = 1) = \mathbb{P}(I = 1|y = 1) = \mathbb{P}(I = 1|y = -1) \leq \frac{\sqrt{2/\pi}\sum_{i\in A}|\nu_i|}{\|\boldsymbol{\nu}_A\|_1 \log d} \leq \frac{1}{\log d}. \tag{C.1}$$

Now, we bound $\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(\|\boldsymbol{\nu}_A\|_1 \log d)$ from below in the following two cases.

Case 1: $A = [d]$. In this case, using Lemma C.1, we have

$$\begin{aligned}
\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(\|\boldsymbol{\nu}_A\|_1 \log d) &\geq \frac{1}{2}\mathbb{P}\left(f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|1) = f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|-1)\right) \\
&\overset{(a)}{=} \frac{1}{2}\mathbb{P}\left(f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|1) = f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|-1) \,|\, y = 1\right) \\
&\geq \frac{1}{2}\mathbb{P}\left(f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|1) = f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|-1), I = 0 \,|\, y = 1\right) \\
&\overset{(b)}{=} \frac{1}{2}\mathbb{P}\left(f_{\boldsymbol{Z}|y}(\boldsymbol{Z}|1) = f_{\boldsymbol{Z}|y}(\boldsymbol{Z}|-1) \,|\, y = -1\right) \\
&\geq \frac{1}{2}\mathbb{P}\left(f_{\boldsymbol{Z}|y}(\boldsymbol{Z}|1) = f_{\boldsymbol{Z}|y}(\boldsymbol{Z}|-1) \,|\, y = 1\right) - \frac{1}{2}\mathbb{P}\left(I = 1 \,|\, y = 1\right) \\
&\overset{(c)}{\geq} \frac{1}{2} - \frac{1}{2\log d},
\end{aligned}$$

where (a) uses the symmetry, (b) uses the fact that when $I = 0$, by definition we have $\boldsymbol{x}' = \boldsymbol{Z}$, and (c) uses (3.7) and (C.1).

Case 2: $A \subsetneq [d]$. Using Lemma C.1, we have

$$\begin{aligned}
\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(\|\boldsymbol{\nu}_A\|_1 \log d) &\geq \mathbb{P}\left(f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|-1) > f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|1) \,|\, y = 1\right) \\
&\geq \mathbb{P}\left(f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|-1) > f_{\boldsymbol{x}'|y}(\boldsymbol{x}'|1), I = 0 \,|\, y = 1\right) \\
&\overset{(a)}{=} \mathbb{P}\left(f_{\boldsymbol{Z}|y}(\boldsymbol{Z}|-1) > f_{\boldsymbol{Z}|y}(\boldsymbol{Z}|1), I = 0 \,|\, y = 1\right) \\
&\geq \mathbb{P}\left(f_{\boldsymbol{Z}|y}(\boldsymbol{Z}|-1) > f_{\boldsymbol{Z}|y}(\boldsymbol{Z}|1)|y = 1\right) - \mathbb{P}\left(I = 1 \,|\, y = 1\right) \\
&\overset{(b)}{\geq} \mathbb{P}\left(f_{\boldsymbol{Z}|y}(\boldsymbol{Z}|-1) > f_{\boldsymbol{Z}|y}(\boldsymbol{Z}|1) \,|\, y = 1\right) - \frac{1}{\log d}
\end{aligned} \tag{C.2}$$

where (a) uses the fact that by definition, when $I = 0$, we have $\boldsymbol{x}' = \boldsymbol{Z}$, and (b) uses (C.1). Note that since Z_i are conditionally independent given y, we have

$$f_{\boldsymbol{Z}|y}(\boldsymbol{Z}|y) = f_{\boldsymbol{Z}_A|y}(\boldsymbol{Z}_A|y) f_{\boldsymbol{Z}_{A^c}|y}(\boldsymbol{Z}_{A^c}|y).$$

But from (3.7), we have $f_{\boldsymbol{Z}_A|y}(\boldsymbol{Z}_A|1) = f_{\boldsymbol{Z}_A|y}(\boldsymbol{Z}_A|-1)$ with probability one. Using this in (C.2), we get

$$\begin{aligned}
\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(\|\boldsymbol{\nu}_A\|_1 \log d) &\geq \mathbb{P}\left(f_{\boldsymbol{Z}_{A^c}|y}(\boldsymbol{Z}_{A^c}|-1) > f_{\boldsymbol{Z}_{A^c}|y}(\boldsymbol{Z}_{A^c}|1)|y = 1\right) - \frac{1}{\log d} \\
&= \bar{\Phi}(\|\boldsymbol{\nu}_{A^c}\|_2) - \frac{1}{\log d}.
\end{aligned}$$

We may combine the two cases following the convention that when $A = [d]$, $A^c = \emptyset$ and $\|\boldsymbol{\nu}_{A^c}\|_2 = 0$. This completes the proof. ∎

Appendix D. Proof of the General Lower Bound (Theorem 3.11).

In this section, we prove Theorem 3.11 by providing a general lower bound for the optimal robust classification error which relaxes the diagonal assumption for the covariance matrix. Our strategy is to approximate the covariance matrix by a diagonal matrix and use our lower bound of Theorem 3.8. It turns out that the optimal robust classification error is monotone with respect to the positive definite ordering of the covariance matrix. Lemma D.1 below formalizes this. Intuitively speaking, the reason is that more noise makes the classification more difficult, resulting in an increase in the optimal robust classification error.

Lemma D.1. *Assume that $\boldsymbol{\mu} \in \mathbb{R}^d$ and Σ_1 and Σ_2 are two positive definite covariance matrices such that $\Sigma_1 \preceq \Sigma_2$. Then for $0 \le k \le d$ we have*

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma_1}(k) \le \mathcal{L}^*_{\boldsymbol{\mu},\Sigma_2}(k).$$

Proof. Let $y \sim \text{Unif}(\pm 1)$, $\boldsymbol{x}_1 \sim \mathcal{N}(y\boldsymbol{\mu}, \Sigma_1)$ and $\boldsymbol{x}_2 \sim \mathcal{N}(y\boldsymbol{\mu}, \Sigma_2)$. Since $\Sigma_1 \preceq \Sigma_2$, we may write $\Sigma_2 = \Sigma_1 + A$ such that $A \succeq 0$. In addition to this, we may couple $\boldsymbol{x}_1, \boldsymbol{x}_2$ on the same probability space as $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{Z}$ where $\boldsymbol{Z} \sim \mathcal{N}(0, A)$ is independent from all other variables. Now, fix a classifier $\mathcal{C}_2 : \mathbb{R}^d \to \{\pm 1\}$ and note that

$$\begin{aligned}
\mathcal{L}_{\boldsymbol{\mu},\Sigma_2}(\mathcal{C}_2, k) &= \mathbb{P}\left(\exists \boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x}_2, k) : \mathcal{C}_2(\boldsymbol{x}') \neq y\right) \\
&= \mathbb{P}\left(\exists \boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x}_1 + \boldsymbol{Z}, k) : \mathcal{C}_2(\boldsymbol{x}') \neq y\right) \\
&= \mathbb{P}\left(\exists \boldsymbol{x}'' \in \mathcal{B}_0(\boldsymbol{x}_1, k) : \mathcal{C}_2(\boldsymbol{x}'' + \boldsymbol{Z}) \neq y\right) \\
&\ge \inf_{\widetilde{\mathcal{C}}_2 : \mathbb{R}^d \times \mathbb{R}^d \to \{\pm 1\}} \mathbb{P}\left(\exists \boldsymbol{x}'' \in \mathcal{B}_0(\boldsymbol{x}_1, k) : \widetilde{\mathcal{C}}_2(\boldsymbol{x}'', \boldsymbol{Z}) \neq y\right)
\end{aligned} \tag{D.1}$$

Now, fix $\widetilde{\mathcal{C}}_2 : \mathbb{R}^d \times \mathbb{R}^d \to \{\pm 1\}$ and note that using the independence of Z, we may write

$$\begin{aligned}
&\mathbb{P}\left(\exists \boldsymbol{x}'' \in \mathcal{B}_0(\boldsymbol{x}_1, k) : \widetilde{\mathcal{C}}_2(\boldsymbol{x}'', \boldsymbol{Z}) \neq y\right) \\
&\qquad = \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}\left[\exists \boldsymbol{x}'' \in \mathcal{B}_0(\boldsymbol{x}_1, k) : \widetilde{\mathcal{C}}_2(\boldsymbol{x}'', \boldsymbol{Z}) \neq y\right] \middle| \boldsymbol{Z}\right]\right] \\
&\qquad = \int \mathbb{P}\left(\exists \boldsymbol{x}'' \in \mathcal{B}_0(\boldsymbol{x}_1, k) : \widetilde{\mathcal{C}}_2(\boldsymbol{x}_1, \boldsymbol{z}) \neq y\right) f_{\boldsymbol{Z}}(\boldsymbol{z}) d\boldsymbol{z}
\end{aligned} \tag{D.2}$$

But for $z \in \mathbb{R}^d$, if we let $\widetilde{\mathcal{C}}_{2,\boldsymbol{z}}(\boldsymbol{x}) := \widetilde{\mathcal{C}}_2(\boldsymbol{x}, \boldsymbol{z})$, we get

$$\begin{aligned}
\mathbb{P}\left(\exists \boldsymbol{x}'' \in \mathcal{B}_0(\boldsymbol{x}_1, k) : \widetilde{\mathcal{C}}_2(\boldsymbol{x}_1, \boldsymbol{z}) \neq y\right) &= \mathbb{P}\left(\exists \boldsymbol{x}'' \in \mathcal{B}_0(\boldsymbol{x}_1, k) : \widetilde{\mathcal{C}}_{2,\boldsymbol{z}}(\boldsymbol{x}_1) \neq y\right) \\
&\ge \inf_{\mathcal{C}_1 : \mathbb{R}^d \to \{\pm 1\}} \mathbb{P}\left(\exists \boldsymbol{x}'' \in \mathcal{B}_0(\boldsymbol{x}_1, k) : \widetilde{\mathcal{C}}_1(\boldsymbol{x}_1) \neq y\right) \\
&= \mathcal{L}^*_{\boldsymbol{\mu},\Sigma_1}(k).
\end{aligned}$$

Comparing this with (D.1) and (D.2), we realize that $\mathcal{L}_{\boldsymbol{\mu},\Sigma_2}(\mathcal{C}_2, k) \ge \mathcal{L}^*_{\boldsymbol{\mu},\Sigma_1}(k)$. Since this holds for arbitrary $\mathcal{C}_2$, optimizing for $\mathcal{C}_2$ yields the desired result. ∎

Note that since Σ is positive definite, we have $\Sigma \succeq \alpha I_d$ where $\alpha > 0$ is the minimum eigenvalue of Σ. Therefore, we may use Lemma D.1 together with the lower bound of Theorem 3.8 for $\mathcal{L}^*_{\boldsymbol{\mu},\alpha I_d}(.)$ to obtain a lower bound for $\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(.)$. However, it turns out that it is more efficient in some scenarios to first normalize the diagonal entries of the covariance matrix. More precisely, define the $d \times d$ matrix R where the i,j entry in R is $R_{i,j} = \Sigma_{i,j}/\sqrt{\Sigma_{ii}\Sigma_{jj}}$. In other words, $R_{i,j}$ is the correlation coefficient between the ith and the jth coordinates in our Gaussian noise. Equivalently, with $\widetilde{\Sigma}$ being the diagonal part of Σ, we may write

$$R := \widetilde{\Sigma}^{-\frac{1}{2}} \Sigma \widetilde{\Sigma}^{-\frac{1}{2}}. \tag{D.3}$$

It is evident that since Σ is assumed to be positive definite, R is also positive definite. In fact, R is the covariance matrix of the normalized random vector $\boldsymbol{x}'$ such that $x'_i = x_i/\sqrt{\Sigma}_{i,i}$ where $\boldsymbol{x} \sim \mathcal{N}(y\boldsymbol{\mu}, \Sigma)$. Also , all the diagonal entries in R are equal to 1, and when Σ is diagonal, $R = I_d$ is the identity matrix. Furthermore, we define $\boldsymbol{u} = (u_1, \dots, u_d)$ where

$$u_i = \frac{\mu_i}{\sqrt{\Sigma_{i,i}}} \qquad 1 \le i \le d. \tag{D.4}$$

In fact, with $\boldsymbol{x}'$ being the normalized of $\boldsymbol{x}$ as above, we have $\boldsymbol{u} = \mathbb{E}\left[\boldsymbol{x}'|y=1\right]$. In Lemma D.2, we show that such coordinate-wise normalization does not affect the optimal robust classiciation error. The main reason for this is that any coordinate-wise product of a vector by positive values does not change the ℓ_0 norm. This property is unique to the combinatorial ℓ_0 norm, and indeed does not hold for ℓ_p norms for $p \ge 1$.

Lemma D.2. *Given a vector $\boldsymbol{a} \in \mathbb{R}^d$ with strictly positive entries, if we define $\boldsymbol{\mu}' \in \mathbb{R}^d$ and $\Sigma' \in \mathbb{R}^{d\times d}$ as ${\mu'}_i = a_i\mu_i$ and ${\Sigma'}_{i,j} = a_i a_j \Sigma_{i,j}$, then we have*

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k) = \mathcal{L}^*_{\boldsymbol{\mu}',\Sigma'}(k) \qquad \forall 0 \le k \le d.$$

In particular, with $\boldsymbol{u}$ and R defined above, we have

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k) = \mathcal{L}^*_{\boldsymbol{u},R}(k) \qquad \forall 0 \le k \le d.$$

Proof. Pick $\epsilon > 0$ together with a classifier $\mathcal{C}$ such that

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k) \ge \mathcal{L}_{\boldsymbol{\mu},\Sigma}(\mathcal{C}, k) - \epsilon. \tag{D.5}$$

Let $\boldsymbol{x} \sim \mathcal{N}(y\boldsymbol{\mu}, \Sigma)$, i.e. $(\boldsymbol{x}, y) \sim \mathcal{D}$, and define $\boldsymbol{x}' := \boldsymbol{a} \odot \boldsymbol{x}$. Note that $\boldsymbol{x}' \sim \mathcal{N}(y\boldsymbol{\mu}', \Sigma')$. Let $\mathcal{D}'$ denote the joint distribution of $(\boldsymbol{x}', Y)$. Recall that by definition $\mathcal{L}_{\boldsymbol{\mu},\Sigma}(\mathcal{C}, k) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\boldsymbol{x}'\in\mathcal{B}_0(\boldsymbol{x},k)} \ell(\mathcal{C}; \boldsymbol{x}', y)\right]$. Note that $\boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x}, k)$ iff $\|\boldsymbol{x}' - \boldsymbol{x}\|_0 \le k$. Since all the entries in $\boldsymbol{a}$ are nonzero, this is equivalent to $\|\boldsymbol{a} \odot \boldsymbol{x}' - \boldsymbol{a} \odot \boldsymbol{x}\|_0 \le k$ which is in turn equivalent to $\boldsymbol{a} \odot \boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{a} \odot \boldsymbol{x}, k)$. Therefore, if $\boldsymbol{a}^{-1}$ denotes the elementwise inverse of $\boldsymbol{a}$, we may write

$$\mathcal{L}_{\boldsymbol{\mu},\Sigma}(\mathcal{C}, k) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\boldsymbol{x}''\in\mathcal{B}_0(\boldsymbol{a}\odot\boldsymbol{x},k)} \ell(\mathcal{C}; \boldsymbol{a}^{-1} \odot \boldsymbol{x}'', y)\right].$$

Let $\mathcal{C}'$ be the classifier defined that $\mathcal{C}'(\boldsymbol{x}) := \mathcal{C}(\boldsymbol{a} \odot \boldsymbol{x})$. With this, we can rewrite the above as

$$\begin{aligned}
\mathcal{L}_{\boldsymbol{\mu},\Sigma}(\mathcal{C}, k) &= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\boldsymbol{x}''\in\mathcal{B}_0(\boldsymbol{a}\odot\boldsymbol{x},k)} \ell(\mathcal{C}'; \boldsymbol{x}'', y)\right] \\
&= \mathbb{E}_{(\boldsymbol{x}',y)\sim\mathcal{D}'}\left[\max_{\boldsymbol{x}''\in\mathcal{B}_0(\boldsymbol{x}',k)} \ell(\mathcal{C}'; \boldsymbol{x}'', y)\right] \\
&= \mathcal{L}_{\boldsymbol{\mu}',\Sigma'}(\mathcal{C}', k) \\
&\ge \mathcal{L}^*_{\boldsymbol{\mu}',\Sigma'}(k).
\end{aligned}$$

Comparing this with (D.5) and sending to zero, we realize that $\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k) \ge \mathcal{L}^*_{\boldsymbol{\mu}',\Sigma'}(k)$. Changing the order of $(\boldsymbol{\mu}, \Sigma)$ and $(\boldsymbol{\mu}', \Sigma')$ and replacing $\boldsymbol{a}$ with $\boldsymbol{a}^{-1}$ yields the other direction and completes the proof. ∎

Using the above tools, we are now ready to prove Theorem 3.11.

Proof of Theorem 3.11. Note that since Σ is positive definite, R is also positive definite and $\zeta_{\min} > 0$. Moreover, we have $R \succeq \zeta_{\min} I_d$. Therefore, using Lemmas D.1 and D.2 above, we realize that for all k, we have

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(k) = \mathcal{L}^*_{\boldsymbol{u},R}(k) \geq \mathcal{L}^*_{\boldsymbol{u},\zeta_{\min} I_d}(k). \tag{D.6}$$

Since $\zeta_{\min} I_d$ is diagonal, we may use our lower bound of Theorem 3.8 with $\boldsymbol{\nu} = (\zeta_{\min} I_d)^{-1/2}\boldsymbol{u} = \boldsymbol{u}/\sqrt{\zeta_{\min}}$ to obtain the following bound with holds for all $A \subseteq [d]$

$$\mathcal{L}^*_{\boldsymbol{u},\zeta_{\min} I_d}\left(\frac{1}{\sqrt{\zeta_{\min}}}\|\boldsymbol{u}_A\|_1 \log d\right) \geq \bar{\Phi}(\|\boldsymbol{u}_{A^c}\|_2) - \frac{1}{\log d}.$$

The proof is complete by comparing this with (D.6). ∎

Appendix E. Proof of Theorem 3.13.

We use the bound in Corollary 3.5 with $F = [\lambda_c : d]$, which simplifies into the following with $k = \|\boldsymbol{\nu}_{[1:\lambda_c]}\|_1 / \log d$:

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}\left(\frac{\|\boldsymbol{\nu}_{[1:\lambda_c]}\|_1}{\log d}\right) \leq \frac{1}{\sqrt{2\log d}} + \bar{\Phi}\left(\|\boldsymbol{\nu}_{[\lambda_c:d]}\|_2 - \frac{\|\boldsymbol{\nu}_{[1:\lambda_c]}\|_1 \|\boldsymbol{\nu}_{[\lambda_c:d]}\|_\infty}{\|\boldsymbol{\nu}_{[\lambda_c:d]}\|_2}\frac{16\sqrt{2}}{\sqrt{\log d}}\right). \tag{E.1}$$

Note that we have

$$\|\boldsymbol{\nu}_{[\lambda_c:d]}\|_2^2 = 1 - \|\boldsymbol{\nu}_{1:\lambda_c - 1}\|_2^2 \geq 1 - c^2. \tag{E.2}$$

On the other hand,

$$\begin{aligned} \|\boldsymbol{\nu}_{[1:\lambda_c]}\|_1 \|\boldsymbol{\nu}_{[\lambda_c:d]}\|_\infty &= \|\boldsymbol{\nu}_{[1:\lambda_c]}\|_1 |\nu_{\lambda_c}| \\ &\leq \|\boldsymbol{\nu}_{[1:\lambda_c]}\|_2^2 \\ &\leq \|\boldsymbol{\nu}\|_2^2 \\ &= 1 \end{aligned} \tag{E.3}$$

Substituting (E.2) and (E.3) back into (E.1), we get

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}\left(\frac{\|\boldsymbol{\nu}_{[1:\lambda_c]}\|_1}{\log d}\right) \leq \frac{1}{\sqrt{2\log d}} + \bar{\Phi}\left(\sqrt{1-c^2} - \frac{16\sqrt{2}}{\sqrt{1-c^2}\sqrt{\log d}}\right) \tag{E.4}$$

Furthermore, with $A = [1 : \lambda_c]$, the bound in Theorem 3.8 implies that

$$\mathcal{L}^*_{\boldsymbol{\mu},\Sigma}(\|\boldsymbol{\nu}_{[1:\lambda_c]}\|_1 \log d) \geq \bar{\Phi}(\sqrt{1-c^2}) - \frac{1}{\log d}. \tag{E.5}$$

This completes the proof.

Appendix F. Proof of Theorem 3.18.

Note that since $\Psi_d(.)$ is nondecreasing for all d, if $\Psi_\infty(c) = \lim \Psi_d(c)$ exists, $\Psi_\infty(.)$ is indeed nondecreasing and $\Psi_\infty(0)$ is well-defined.

<u>Part 1</u> First we assume that $c \in (0, 1)$. Since $\Psi_\infty(c) = \lim \Psi_d(c)$ and $\log\log d/\log d \to 0$, $\limsup \log_d k_d < \Psi_\infty(c)$ implies that for d large enough, we have

$$\log_d k_d < \Psi_d(c) - \frac{\log\log d}{\log d}.$$

Thereby,

$$\log_d k_d < \log_d \|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_1 - \frac{\log\log d}{\log d} = \log_d \frac{\|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_1}{\log d}.$$

Hence, Theorem 3.13 implies that

$$\mathcal{L}^*_d(k_d) \le \mathcal{L}^*_d\left(\frac{\|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_1}{\log d}\right) \le \frac{1}{\sqrt{2\log d}} + \bar{\Phi}\left(\sqrt{1-c^2} - \frac{16\sqrt{2}}{\sqrt{1-c^2}\sqrt{\log d}}\right).$$

Sending d to infinity, we get $\limsup \mathcal{L}^*_d(k_d) \le \bar{\Phi}(\sqrt{1-c^2})$. Next, we consider $c = 0$. Note that since $\Psi_\infty(.)$ is nondecreasing, $\limsup \log_d k_d < \Psi_\infty(0)$ implies that $\limsup \log_d k_d < \Psi_\infty(c)$ for all $c > 0$. Consequently, the above bound implies that $\limsup \mathcal{L}^*(k_d) \le \bar{\Phi}(\sqrt{1-c^2})$ for all $c > 0$. Sending c to zero, we realize that $\limsup \mathcal{L}^*(k_d) \le \bar{\Phi}(0)$. Finally, for $c = 1$, note that the classifier that always outputs 1 has misclassification error at most $1/2$. This implies that irrespective of the sequence k_d, we always have $\limsup \mathcal{L}^*_d(k_d) \le 1/2 = \bar{\Phi}(\sqrt{1-1^2})$ and the bound automatically holds for $c = 1$.

Part 2 First we assume that $c \in (0, 1]$. Similar to the first pare, $\liminf \log_d k_d > \Psi_\infty(c)$ implies that for d large enough, we have

$$\log_d k_d > \Psi_d(c) + \frac{\log\log d}{\log d},$$

and

$$\log_d k_d > \log_d \|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_1 + \frac{\log\log d}{\log d} = \log_d(\log d \|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_1).$$

Hence, Theorem 3.13 implies that

$$\mathcal{L}^*_d(k_d) \ge \mathcal{L}^*_d(\log d \|\boldsymbol{\nu}^{(d)}_{[1:\lambda_c^{(d)}]}\|_1) \ge \bar{\Phi}(\sqrt{1-c^2}) - \frac{1}{\log d}.$$

Sending $d \to \infty$, we get $\liminf \mathcal{L}^*_d(k_d) \ge \bar{\Phi}(\sqrt{1-c^2})$. For the case $c = 0$, note that irrespective of the sequence k_d, we always have $\mathcal{L}^*_d(k_d) \ge \mathcal{L}^*_d(0) = \bar{\Phi}(\sqrt{1-0^2})$. Thereby, the result for $c = 0$ automatically holds.