# PRIVATELY LEARNING GAUSSIANS AND THEIR MIXTURES

# ON THE SAMPLE COMPLEXITY OF PRIVATELY LEARNING GAUSSIANS AND THEIR MIXTURES

BY

ISHAQ ADEN-ALI, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF DEPARTMENT OF COMPUTING AND SOFTWARE

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

© Copyright by Ishaq Aden-Ali, July 2021

All Rights Reserved

Master of Science (2021) McMaster University

(department of computing and software) Hamilton, Ontario, Canada

| | |
|---|---|
| TITLE: | On The Sample Complexity of Privately Learning Gaussians and their Mixtures |
| AUTHOR: | Ishaq Aden-Ali<br>B.Eng. (Electrical Engineering),<br>McMaster University, Hamilton, Canada |
| SUPERVISOR: | Dr. Hassan Ashtiani |
| NUMBER OF PAGES: | x, 111 |

# Lay Abstract

Is it possible to estimate an unknown probability distribution given random samples from it? This is a fundamental problem known as distribution learning (or density estimation) that has been studied by statisticians for decades, and in recent years has become a topic of interest for computer scientists. While distribution learning is a mature and well understood problem, in many cases the samples (or data) we observe may consist of sensitive information belonging to individuals and well-known solutions may inadvertently result in the leakage of private information.

In this thesis we study distribution learning under the assumption that the data is generated from high-dimensional Gaussians (or their mixtures) with the aim of understanding how many samples an algorithm needs before it can guarantee a good estimate. Furthermore, in order to protect against leakage of private information, we consider approaches that maintain *differential privacy* – the gold standard for modern private data analysis.

# Abstract

In this thesis we prove sample complexity upper bounds for privately PAC learning two fundamental classes of distributions:

**Multivariate Gaussians.** We provide sample complexity upper bounds for semi-agnostically learning multivariate Gaussians under the constraint of approximate differential privacy. These are the first finite sample upper bounds for general Gaussians which do not impose restrictions on the parameters of the distribution. Our bounds are near-optimal in the case when the covariance is known to be the identity, and conjectured to be near-optimal in the general case. From a technical standpoint, we provide analytic tools for arguing the existence of global "locally small" covers from local covers of the space. These are exploited using modifications of recent techniques for differentially private hypothesis selection.

**Mixtures of Gaussians.** We consider the problem of learning mixtures of Gaussians under the constraint of approximate differential privacy. We provide the first sample complexity upper bounds for privately learning mixtures of unbounded axis-aligned (or even unbounded univariate) Gaussians. To prove our results, we design a new technique for privately learning mixture distributions. A class of distributions

$\mathcal{F}$ is said to be list-decodable if there is an algorithm that, given "heavily corrupted" samples from $f \in \mathcal{F}$, outputs a list of distributions, $\widehat{\mathcal{F}}$, such that one of the distributions in $\widehat{\mathcal{F}}$ approximates $f$. We show that if $\mathcal{F}$ is privately list-decodable then we can privately learn mixtures of distributions in $\mathcal{F}$. Finally, we show axis-aligned Gaussian distributions are privately list-decodable, thereby proving mixtures of such distributions are privately learnable.

**Dedications**

*to my beloved family*

# Acknowledgements

This thesis would not have been possible without the guidance of my brilliant advisor Hassan Ashtiani. Hassan took a chance on me when I was a clueless undergraduate student with no experience working on theoretical research problems. He taught me many things, guided me when I was stuck, answered my questions, and encouraged me when I got things right. I can count on one hand the number of people that I know to be as patient, kind, and caring as Hassan. I am hopeful that this is just the beginning of a long lasting and fruitful collaboration between the two of us.

I would also like to thank my two wonderful co-authors Gautam Kamath and Christopher Liaw for teaching me so much during our collaborations. It is an understatement to say that the content of this thesis benefited from their knowledge and guidance. It was truly wonderful working with them both, and I can confidently say that these two made me thoroughly enjoy the collaborative aspect of research.

I would like to give special thanks to Gautam for also being a mentor to me. He always found the time to answer my questions, many of which went beyond technical topics related to our research. Gautam played a huge role in helping me take the next steps of my academic journey, and I am truly grateful for all that he has done for me.

I would also like to thank Professors Tim Davidson and Antoine Deza for writing me letters of recommendations when I was applying to the masters program at

McMaster. Special thanks to Antoine for also being a mentor to me, starting all the way back when I was in my final year of undergraduate studies. Antoine vehemently encouraged me to pursue graduate studies when I was considering going into industry, and I will forever be grateful to him for his encouragement.

I would like to thank my officemates Hongfeng, Nima, Carlos, Akihiro, Zhi, and Ning. I had the pleasure of interacting with all these folks on a day to day basis before the pandemic hit. I enjoyed the conversations about our personal lives near the kettle, and the discussions about our research in front of the whiteboard.

I would also like to thank the other students in Hassan's research group. This includes Nima, Zhewei, Alireza, Qing, and Jamil. It has been a pleasure learning from each of them through our weekly research meetings and Slack. I can only imagine that our interactions would have been even more enjoyable if they were in person.

I would like to thank my parents and siblings for their unwavering support and encouragement throughout my academic journey so far. Last but certainly not least, I would like to thank my wife Suaad for always cheering me up when a challenging research problem had me down, and for patiently listening to me talk about my half-baked research ideas.

# Contents

# Chapter 1

# Introduction

The fundamental problem of *distribution learning* is concerned with the design of algorithms (i.e., *estimators*) that, given samples generated from an unknown distribution $f$, output an "approximation" of $f$. While the literature on distribution learning is vast and has a long history dating back to the late nineteenth century, in many cases the dataset may consist of sensitive data belonging to individuals, and naive execution of classic methods may inadvertently result in private information leakage.

To address concerns of this nature, in 2006, Dwork, McSherry, Nissim, and Smith introduced the celebrated notion of differential privacy (DP) [DMNS06], which provides a strong standard for data privacy. Roughly speaking, differential privacy guarantees that no single data point can influence the output of an algorithm too much, which intuitively provides privacy by "hiding" the contribution of each individual. Differential privacy has seen practical adoption in many organizations, including Apple [Dif17], Google [EPK14, BEM+17], Microsoft [DKY17], and the US Census Bureau [DLS+17]. At this point, there is a rich body of literature, giving differentially

private algorithms for a wide array of tasks.

It is thus natural to ask whether it is possible to design differentially private distribution learning algorithms. Differentially private algorithms benefit from having access to large amounts of data. Roughly speaking, this is because it is easier to "hide" the contribution of any individual datapoint when the amount of data provided increases. Unfortunately, there exist learning problems where satisfying the constraint of differential privacy provably requires an *infinite* number of datapoints to solve! (see, e.g. [BNSV15].) Put another way, for certain learning problems, there *does not* exist an algorithm that uses a finite number of datapoints to solve the problem accurately, while also respecting differential privacy.

This highlights that for certain problems, the *cost* of privacy on the required amount of data can be insurmountable. In this thesis, we will be interested in studying what the cost of privacy is for learning two extremely fundamental classes of distributions often used to model data: $i$) high-dimensional Gaussians, and $ii$) mixtures of Gaussians. While the amount of data required to learn these classes non-privately is very well understood, our goal will be to better understand how much more data we need to learn these classes under the constraint of differential privacy.

## 1.1 Learning Unbounded Multivariate Gaussians

In recent years, there has been a flurry of activity in differentially private distribution learning. A number of techniques have been developed in the literature for this problem. In the pure differentially private setting, Bun et al. [BKSW19] recently introduced a method to learn a class of distributions when the class admits a finite cover, i.e. when the entire class of distributions can be well-approximated by a finite

number of representative distributions. In fact, they show that this is an exact characterization of distributions which can be learned under pure differential privacy in the sense that a class of distributions is learnable under pure differential privacy if and only if the class admits a finite cover [HT10, BKSW19]. As a consequence of this result, they obtained pure differentially private algorithms for learning Gaussian distributions provided that the mean of the Gaussians are bounded *and* the covariance matrix of the Gaussians are spectrally bounded.[1] Moreover, such restrictions on the Gaussians are necessary under the constraint of pure differential privacy.

One way to remove the requirement of having a finite cover is to relax to a weaker notion of privacy known as approximate differential privacy. With this notion, Bun, Kamath, Steinke, and Wu [BKSW19] introduced another method to learn a class of distributions that, instead of requiring a finite cover, requires a "locally small" cover, i.e. a cover where each distribution in the class is well-approximated by only a small number of elements within the cover. They prove that the class of Gaussians with arbitrary mean and a fixed, known covariance matrix has a locally small cover which implies an approximate differentially private algorithm to learn this class of distributions.

As the most notable omission, Bun, Kamath, Steinke, and Wu do not provide a locally small cover for general multivariate Gaussians. Indeed, for Gaussians with identity covariance, it is easy to reason about the local size of covers, as total variation distance between distributions corresponds to the $\ell_2$-distance between their means. However, when the covariance is not fixed, the total variation distance is characterized by the *Mahalanobis distance*, which has a significantly more sophisticated geometry.

[1] When we say that a matrix $\Sigma$ is spectrally bounded, we mean that there are $0 < a_1 \leq a_2$ such that $a_1 \cdot I \preceq \Sigma \preceq a_2 \cdot I$.

Analyzing these situations to show local smallness appears to be intractable using current analytic techniques, which involve explicitly constructing and analyzing a cover of the space. Given this challenge, up to now it has not been clear even whether a finite sample algorithm exists at all! And this is only for the fundamental case of Gaussians, raising the question of how one would even approach more complex classes of distributions.

### 1.1.1 Contributions to Learning Multivariate Gaussians

We resolve these issues by providing a simpler method for proving the *existence* of locally small covers. This lead to the main results of Chapter 3: we present sample complexity upper bounds for semi-agnostically learning Gaussian distributions under approximate differential privacy. We informally state our results below. In the following theorem and the rest of this thesis we use $\widetilde{O}$ to hide polylogarithmic factors, i.e. $\widetilde{O}(f(x))$ means $O(f(x)\log^c f(x))$ for some $c > 0$.

**Theorem 1.1.1** (Informal version of Theorem 3.6.6)**.** *The sample complexity of semi-agnostically learning a $d$-dimensional Gaussian distribution to $\alpha$-accuracy in total variation distance under $(\varepsilon, \delta)$-differential privacy is*

$$\widetilde{O}\left(\frac{d^2}{\alpha^2} + \frac{d^2}{\alpha\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right).$$

This is the first sample complexity bound for privately learning a multivariate Gaussian with no conditions on the covariance matrix. The first and third terms are known to be tight, and there is strong evidence that the second is as well. The previous best algorithm was that of [KLSU19], which provided the stronger guarantee

of concentrated differential privacy [DR16, BS16] (which is intermediate to pure and approximate DP). However, it required the true covariance matrix to be spectrally bounded as $I \preceq \Sigma \preceq KI$ for some known parameter $K$, and the third term in the sample complexity is instead $O\left(\frac{d^{3/2}\log^{1/2} K}{\varepsilon}\right)$, which is prohibitive for large (or unknown) $K$. In contrast, our result holds for unrestricted Gaussian distributions.

We also provide a better upper bound for the case when the covariance matrix is known.

**Theorem 1.1.2** (Informal version of Theorem 3.6.4)**.** *The sample complexity of semi-agnostically learning a $d$-dimensional Gaussian distribution with known covariance to $\alpha$-accuracy in total variation distance under $(\varepsilon, \delta)$-differential privacy is*

$$\widetilde{O}\left(\frac{d}{\alpha^2} + \frac{d}{\alpha\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right).$$

This is the first bound which achieves a near-optimal dependence simultaneously on all parameters. In particular, it improves upon previous results in which the third term is replaced by $O\left(\frac{\log(1/\delta)}{\alpha\varepsilon}\right)$ [BKSW19] or $O\left(\frac{\sqrt{d}\log^{3/2}(1/\delta)}{\varepsilon}\right)$ [KV18, KLSU19, BKSW19].

## 1.2 Learning Unbounded Mixtures of Gaussians

It is straightforward to see that if a class of distributions admits a finite cover then the class of its mixtures also admits a finite cover. Combined with the aforementioned work of Bun, Kamath, Steinke, and Wu this implies a pure differentially private algorithm for learning mixtures of Gaussians with bounded mean and spectrally bounded covariance matrices. It is natural to wonder whether an analogous statement holds

for locally small covers. In other words, if a class of distributions admits a locally small cover then does the class of mixtures also admit a locally small cover? If so, this would provide a fruitful direction to design differentially private algorithms for learning mixtures of unbounded Gaussians.

Unfortunately, there are simple examples of classes of distributions that admit a locally small cover yet their mixture do *not*. (See Section 4.4 for a formal statement and proof.) This leaves open the question of designing private algorithms for many classes of distributions that are learnable in the non-private setting. One concrete open problem is for the class of mixtures of two univariate Gaussian distributions. A more general problem is private learning of mixtures of $k$ axis-aligned Gaussian distributions. (Recall that an axis-aligned Gaussians is a Gaussian with a diagonal covariance matrix.)

### 1.2.1 Contributions to Learning Mixtures Gaussians

We demonstrate that it is indeed possible to privately learn mixtures of unbounded univariate Gaussians. This leads to the main results of Chapter 4: we provide sample complexity upper bounds for learning mixtures of unbounded $d$-dimensional axis-aligned Gaussians and mixtures of $d$-dimensional Gaussians with the same known covariance matrix. We informally state our results below.

**Theorem 1.2.1** (Informal version of Theorem 4.7.3)**.** *The sample complexity of learning a mixture of $k$ $d$-dimensional axis-aligned Gaussians to $\alpha$-accuracy in total variation distance under $(\varepsilon, \delta)$-differential privacy is*

$$\widetilde{O}\left(\frac{k^2 d \log^{3/2}(1/\delta)}{\alpha^2 \varepsilon}\right).$$

Even for the univariate case, our result is the *first* sample complexity upper bound for learning mixture of Gaussians under differential privacy for which the variances are unknown and the parameters of the Gaussians may be unbounded. In the non-private setting, it is known that $\widetilde{\Theta}(kd/\alpha^2)$ samples are necessary and sufficient to learn an axis-aligned Gaussian in $\mathbb{R}^d$ [SOAJ14, ABH+20].

If the covariance matrix of each component of the mixture is the same and known or, without loss of generality, equal to the identity matrix, then we can improve the dependence on the parameters and obtain a result that is in line with the non-private setting.

**Theorem 1.2.2** (Informal version of Theorem 4.7.1)**.** *The sample complexity of learning a mixture of $k$ $d$-dimensional Gaussians with identity covariance matrix to $\alpha$-accuracy in total variation distance under $(\varepsilon, \delta)$-differential privacy is*

$$\widetilde{O}\left(\frac{kd}{\alpha^2} + \frac{kd\log(1/\delta)}{\alpha\varepsilon}\right).$$

To prove our results, we first construct a two step algorithm for privately learning univariate mixtures. The algorithm first privately estimates the variance of the mixture components, and then uses these estimates of the variance to carefully estimate the mean of the mixture components privately. Both of these steps rely on the celebrated stable histogram algorithm, and require a very careful construction of the histogram bin widths. Finally, we show how to extend this algorithm for univariate mixtures to $d$-dimensional axis-aligned mixtures.

## 1.3 Thesis Organization

In Chapter 2 we go over some basic definitions and well known results that we will make use of in this thesis. Chapter 3 is dedicated to proving our sample complexity upper bounds for learning *unbounded* high-dimensional Gaussians. Finally, in Chapter 4 we prove our sample complexity upper bounds for learning mixtures of Gaussians. We conclude with some open problems in Chapter 5.

# Chapter 2

# Background

This chapter is dedicated to going over the background information required to successfully read Chapters 3 and 4. We begin by going over the notation used in this thesis. We then define differential privacy and standard results related to it. Next, we state the problem of distribution learning and formally define different notions of learning algorithms for this problem. We then state the definition of the Vapnik-Chervonenkis dimension. Finally, we conclude by going over related work in the literature.

## 2.1 Notation

For any $m \in \mathbb{N}$, $[m]$ denotes the set $\{1, 2, \ldots, m\}$. Let $X \sim f$ denote a random variable $X$ sampled from distribution $f$. Let $(X_i)_{i=1}^m \sim f^m$ denote an i.i.d. random sample of size $m$ from distribution $f$.

For a positive integer $d$ let $\mathbb{S}_d \subset \mathbb{R}^{d \times d}$ be the set of all $d$-by-$d$ positive semi-definite real matrices. For a matrix $A \in \mathbb{R}^{m \times n}$, define $\|A\|_{1,1} = \sum_{i=1}^m \sum_{j=1}^n |A_{ij}|$ and

$\|A\|_{\infty,\infty} = \max_{i,j} |A_{ij}|$. The determinant of a square matrix $A$ is given by $\det(A)$.

For a vector $\mu \in \mathbb{R}^d$ and a matrix $\Sigma \in \mathbb{S}_d$, we use $\mathcal{N}(\mu, \Sigma)$ to denote the multivariate normal distribution with mean $\mu$ and covariance matrix $\Sigma$. A useful property of Gaussian distributions is that any linear transformation of a Gaussian random vector is also a Gaussian random vector. In particular, if $X \sim \mathcal{N}(\mu, \Sigma)$ is a $d$-dimensional Gaussian random vector and $A$ and $b$ are a $d$-dimensional square matrix and vector respectively, it follows that

$$AX + b \sim \mathcal{N}(A\mu + b, A\Sigma A^T). \tag{2.1.1}$$

## 2.2 Differential Privacy

Let $X^* = \cup_{i=1}^{\infty} X^i$ be the set of all datasets of arbitrary size over a domain set $X$. We say two datasets $D, D' \in X^*$ are neighbours if $D$ and $D'$ differ by at most one data point. Informally, an algorithm is differentially private if its output on neighbouring databases are similar. Formally, differential privacy (DP)[1] has the following definition.

**Definition 2.2.1** ([DMNS06, DKM+06])**.** *A randomized algorithm $T : X^* \to \mathcal{Y}$ is* $(\varepsilon, \delta)$-differentially private *if for all $n \geq 1$, for all neighbouring datasets $D, D' \in X^n$, and for all measurable subsets $S \subseteq \mathcal{Y}$,*

$$\Pr\left[T(D) \in S\right] \leq e^{\varepsilon} \Pr[T(D') \in S] + \delta\,.$$

*If $\delta = 0$, we say that $T$ is $\varepsilon$-differentially private.*

[1] We will use the acronym DP to refer to both the terms "differential privacy" and "differentially private". Which term we are using will be clear from the specific sentence.

We refer to $\varepsilon$-DP as *pure* DP, and $(\varepsilon, \delta)$-DP for $\delta > 0$ as *approximate* DP. We make use of the following property of differentially private algorithms which asserts that adaptively composing differentially private algorithms remains differentially private. By adaptive composition, we mean that we run a sequence of algorithms $M_1(D), \dots, M_T(D)$ where the choice of algorithm $M_t$ may depend on the outputs of $M_1(D), \dots, M_{t-1}(D)$.

**Lemma 2.2.2** (Composition of DP [DMNS06, DRV10])**.** *If $M$ is an adaptive composition of differentially private algorithms $M_1, \dots, M_T$ then the following two statements hold:*

1. *If $M_1, \dots, M_T$ are $(\varepsilon_1, \delta_1), \dots, (\varepsilon_T, \delta_T)$-differentially private, then $M$ is $(\varepsilon, \delta)$-differentially private for*

$$\varepsilon = \sum_{t=1}^{T} \varepsilon_t \quad \text{and} \quad \delta = \sum_{t=1}^{T} \delta_t.$$

2. *If $M_1, \dots, M_T$ are $(\varepsilon_0, \delta_1), \dots, (\varepsilon_0, \delta_T)$-differentially private for some $\varepsilon_0 \leq 1$, then for any $\delta_0 > 0$, $M$ is $(\varepsilon, \delta)$-differentially private for*

$$\varepsilon = \varepsilon_0 \sqrt{6T \log(1/\delta_0)} \quad \text{and} \quad \delta = \delta_0 + \sum_{t=1}^{T} \delta_t.$$

The first statement in Lemma 2.2.2 is often referred to as *basic* composition and the second statement is often referred to as *advanced* composition. We also make use of the fact that post-processing the output of a differentially private algorithm does not impact privacy.

**Lemma 2.2.3** (Post Processing). *If $M : \mathcal{X}^n \to \mathcal{Y}$ is $(\varepsilon, \delta)$-differentially private, and $P : \mathcal{Y} \to \mathcal{Z}$ is any randomized function, then the algorithm $P \circ M$ is $(\varepsilon, \delta)$-differentially private.*

A fundamental building block of differential privacy is the the exponential mechanism [MT07]. It is used to privately select an approximate "best" candidate from a (finite) set of candidates. The quality of a candidate with respect to the dataset is measured by a score function. Let $\mathcal{R}$ be the set of possible candidates. A score function $S : X^* \times \mathcal{R} \to \mathbb{R}$ maps each pair consisting of a dataset and a candidate to a real-valued score. The *exponential mechanism* $\mathcal{M}_E$ takes as input a dataset $D$, a set of candidates $\mathcal{R}$, a score function $S$, a privacy parameter $\varepsilon$ and outputs a candidate $r \in \mathcal{R}$ with probability proportional to $\exp\left(\frac{\varepsilon S(D,r)}{2\Delta(S)}\right)$, where $\Delta(S)$ is the sensitivity of the score function which is defined as

$$\Delta(S) = \max_{r \in \mathcal{R}, D \sim D'} |S(D, r) - S(D', r)| .$$

**Theorem 2.2.4** ([MT07]). *For any dataset $D$, score function $S$ and privacy parameter $\varepsilon > 0$, the exponential mechanism $\mathcal{M}_E(D, S, \varepsilon)$ is an $\varepsilon$-differentially private algorithm, and with probability at least $1 - \beta$, it selects an outcome $r \in \mathcal{R}$ such that*

$$S(D, r) \geq \max_{r' \in \mathcal{R}} S(D, r') - \frac{2\Delta(S) \log(|\mathcal{R}|/\beta)}{\varepsilon}.$$

## 2.3 Distribution Learning

A *distribution learning method* is a (potentially randomized) algorithm that, given a sequence of i.i.d. samples from a distribution $f$, outputs a distribution $\widehat{f}$ as an estimate of $f$. The focus of this paper is on absolutely continuous probability distributions (distributions that have a density with respect to the Lebesgue measure), so we refer to a probability distribution and its probability density function interchangeably. The specific measure of "closeness" between distributions that we use is the *total variation distance*.

**Definition 2.3.1.** *Let $g$ and $f$ be two probability distributions defined over $\mathcal{X}$ and let $\Omega$ be the Borel sigma-algebra on $\mathcal{X}$. The* total variation distance *between $g$ and $f$ is defined as*

$$d_{\mathrm{TV}}(g,f) = \sup_{S\in\Omega} |\mathbf{P}_g(S) - \mathbf{P}_f(S)| = \frac{1}{2}\int_{x\in\mathcal{X}} |g(x) - f(x)|\mathrm{d}x = \frac{1}{2}\|g-f\|_1 \in [0,1].$$

*where $\mathbf{P}_g(S)$ denotes the probability measure that $g$ assigns to $S$. Moreover, if $\mathcal{F}$ is a set of distributions over a common domain, we define $d_{\mathrm{TV}}(g,\mathcal{F}) = \inf_{f\in\mathcal{F}} d_{\mathrm{TV}}(g,f)$.*

We say distributions $g$ and $f$ are $\alpha$-close if $d_{\mathrm{TV}}(g,f) \leq \alpha$. We also say a distribution $g$ is $\alpha$-close to a set of distributions $\mathcal{F}$ if $d_{\mathrm{TV}}(g,\mathcal{F}) \leq \alpha$. We now formally define distribution learners in a few different settings.

**Definition 2.3.2** (Realizable PAC learner)**.** *We say Algorithm $\mathcal{A}$ is a* realizable PAC-learner *for a class of distributions $\mathcal{F}$ which uses $n(\alpha,\beta)$ samples, if for every $\alpha,\beta\in(0,1)$, every $f\in\mathcal{F}$, and every $n \geq n(\alpha,\beta)$ the following holds: if the algorithm is given parameters $\alpha,\beta$ and a sequence of $n$ i.i.d. samples from $f$ as inputs, then it*

*outputs an approximation* $\widehat{f}$ *such that* $d_{\mathrm{TV}}(\widehat{f}, f) \leq \alpha$ *with probability at least* $1-\beta$.[2]

**Definition 2.3.3** ($C$-agnostic PAC learner)**.** *For* $C > 0$ *we say Algorithm* $\mathcal{A}$ *is* $C$-agnostic PAC learner *for a class of distributions* $\mathcal{F}$ *with sample complexity* $n_C(\alpha, \beta)$ *if for any* $\alpha, \beta \in (0,1)$*, every distribution* $g$ *such that* $d_{\mathrm{TV}}(g, \mathcal{F}) = \mathrm{OPT}$*, and every* $n \geq n_C(\alpha, \beta)$ *the following holds: if the algorithm is given parameters* $\alpha$,$\beta$ *and a sequence of* $n$ *i.i.d. samples from* $g$*, the algorithm outputs an approximation* $\widehat{f}$ *such that* $d_{\mathrm{TV}}(\widehat{f}, g) \leq C \cdot \mathrm{OPT} + \alpha$ *with probability at least* $1-\beta$.

If $C = 1$ we will refer to the algorithm as an agnostic PAC learner and for $C > 1$ we will refer to the algorithm as a *semi-agnostic* PAC learner as is standard in learning theory.

We now define differentially private analogues of distribution learners.

**Definition 2.3.4** ($(\varepsilon, \delta)$-DP realizable PAC learner)**.** *We say algorithm* $\mathcal{A}$ *is an* $(\varepsilon, \delta)$-DP realizable PAC learner *for a class of distributions* $\mathcal{F}$ *that uses* $n(\alpha, \beta, \varepsilon, \delta)$ *samples if:*

1. *Algorithm* $\mathcal{A}$ *is a realizable PAC Learner for* $\mathcal{F}$ *that uses* $n(\alpha, \beta, \varepsilon, \delta)$ *samples.*
2. *Algorithm* $\mathcal{A}$ *satisfies* $(\varepsilon, \delta)$*-DP.*

**Definition 2.3.5** ($(\varepsilon, \delta)$-DP $C$-agnostic PAC learner)**.** *We say algorithm* $\mathcal{A}$ *is an* $(\varepsilon, \delta)$-DP realizable PAC learner *for a class of distributions* $\mathcal{F}$ *that uses* $n_C(\alpha, \beta, \varepsilon, \delta)$ *samples if:*

1. *Algorithm* $\mathcal{A}$ *is a* $C$*-agnostic PAC learner for* $\mathcal{F}$ *that uses* $n_C(\alpha, \beta, \varepsilon, \delta)$ *samples.*
2. *Algorithm* $\mathcal{A}$ *satisfies* $(\varepsilon, \delta)$*-DP.*

[2]The probability is over $n(\alpha, \beta)$ samples drawn from $f$ and the randomness of the algorithm.

A useful object for us to define is the total variation ball.

**Definition 2.3.6** (TV ball)**.** *The total variation ball of radius $\gamma \in (0,1)$, centered at a distribution $g$ with respect to a set of distributions $\mathcal{F}$, written $\mathcal{B}(\gamma, g, \mathcal{F})$, is the following subset of $\mathcal{F}$:*

$$\mathcal{B}(\gamma, g, \mathcal{F}) \;\coloneqq\; \{f \in \mathcal{F} : d_{\mathrm{TV}}(g, \mathcal{F}) \leq \gamma\}.$$

In this paper we consider coverings and packings of sets of distributions with respect to the total variation distance.

**Definition 2.3.7** ($\gamma$-covers and $\gamma$-packings)**.** *For any $\gamma \in (0,1)$ a* $\gamma$-cover *of a set of distributions $\mathcal{F}$ is a set of distributions $\mathcal{C}_\gamma$, such that for every $f \in \mathcal{F}$, there exists some $\widehat{f} \in \mathcal{C}_\gamma$ such that $d_{\mathrm{TV}}(f, \widehat{f}) \leq \gamma$.*

*A* $\gamma$-packing *of a set of distributions $\mathcal{F}$ is a set of distributions $\mathcal{P}_\gamma \subseteq \mathcal{F}$, such that for every pair of distributions $f, f' \in \mathcal{P}_\gamma$, we have that $d_{\mathrm{TV}}(f, f') \geq \gamma$.*

**Definition 2.3.8** ($\gamma$-covering and $\gamma$-packing number)**.** *For any $\gamma \in (0,1)$, the* $\gamma$-covering number *of a set of distributions $\mathcal{F}$, $N(\mathcal{F}, \gamma) := \min\{n \in \mathbb{N} : \exists \mathcal{C}_\gamma \text{ s.t. } |\mathcal{C}_\gamma| = n\}$, is the size of the smallest possible $\gamma$-covering of $\mathcal{F}$. Similarly, the* $\gamma$-packing number *of a set of distributions $\mathcal{F}$, $M(\mathcal{F}, \gamma) := \max\{n \in \mathbb{N} : \exists \mathcal{P}_\gamma \text{ s.t. } |\mathcal{P}_\gamma| = n\}$, is the size of the largest subset of $\mathcal{F}$ that forms a packing for $\mathcal{F}$.*

The following proposition follows directly from a well known relationship between packings and covers of metric spaces (see [Ver18, Lemma 4.2.8]).

**Proposition 2.3.9.** *For a set of distributions $\mathcal{F}$ with $\gamma$-covering number $M(\mathcal{F}, \gamma)$*

*and $\gamma$-packing number $N(\mathcal{F}, \gamma)$, the following holds:*

$$M(\mathcal{F}, 2\gamma) \leq N(\mathcal{F}, \gamma) \leq M(\mathcal{F}, \gamma).$$

We now formally define what it means for a set of distributions to be "locally small".

**Definition 2.3.10** ($\gamma$-locally small)**.** *Fix some $\gamma \in (0,1)$. We say a set of distributions $\mathcal{F}$ is* $(k, \gamma)$-locally small *if*

$$\sup_{f \in \mathcal{F}} |\mathcal{B}(\gamma, f, \mathcal{F})| \leq k,$$

*for some $k \in \mathbb{N}$. If no such $k$ exists, we say $\mathcal{F}$ is* not $\gamma$-locally small.

## 2.4 VC Dimension

An important property of a set of binary functions is its Vapnik-Chervonenkis (VC) dimension, which has the following definition:

**Definition 2.4.1** (VC dimension [VC71])**.** *Let $\mathcal{H}$ be a set of binary functions $h : \mathcal{X} \to \{0,1\}$. The VC dimension of $\mathcal{H}$ is defined to be the largest $d$ such that there exist $x_1, \cdots, x_d \in \mathcal{X}$ and $h_1, \cdots, h_{2^d} \in \mathcal{H}$ such that for all $i, j \in [2^d]$ where $i < j$, there exists $k \in [d]$ such that $h_i(x_k) \neq h_j(x_k)$.*

We can define the VC dimension of a set of distributions $\mathcal{F}$ by looking at the VC dimension of a set of binary functions that is defined with respect to $\mathcal{F}$. More precisely:

**Definition 2.4.2** (VC dimension of a set of distributions)**.** *Let $\mathcal{F}$ be a set of probability distributions on a space $\mathcal{X}$. Define the set of binary functions $\mathcal{H}(\mathcal{F}) = \{h_{f_i,f_j} : f_i, f_j \in \mathcal{F}\}$ where $\forall x \in \mathcal{X}$, $h_{f_i,f_j}(x) = 1 \iff f_i(x) > f_j(x)$. We define the VC dimension of $\mathcal{F}$ to be the VC dimension of $\mathcal{H}(\mathcal{F})$.*[3]

## 2.5 Additional Related Work

There has been a flurry of activity on differentially private distribution learning and parameter estimation in recent years for many problem settings [NRS07, BUV14, DHS15, SU17a, SU17b, DSS+15, BSU17, KV18, KLSU19, CWZ19, BKSW19, DFM+20, ASZ20, KSU20, BDKU20, LKKO21]. While many of these focus on settings with parameters bounded by some constant, some pay particular attention to the cost in terms of this bound, including [KV18, KLSU19, BKSW19, BDKU20, DFM+20].

The work of Bun, Kamath, Steinke, and Wu [BKSW19] is built upon classic results in hypothesis selection, combined with the exponential mechanism [MT07]. The underlying non-private approach was pioneered by Yatracos [Yat85], and refined in subsequent work by Devroye and Lugosi [DL96, DL97, DL01]. After this, additional considerations have been taken into account, such as computation, approximation factor, robustness, and more [Mu08, DDS12, DK14, SOAJ14, AJOS14, DKK+16, ABDM18, ABDH+18, AFJ+18, BKM19, AA20]. Notably, these primitives have also been translated to the more restrictive setting of *local* differential privacy [GKK+20]. Similar techniques have also been exploited in a federated setting [LSY+20].

Bun, Kamath, Steinke, and Wu [BKSW19] showed how to learn spherical Gaussian mixtures where each Gaussian component has bounded mean under pure differential

[3] To avoid measurability issues we assume the preimage of 0 is measurable for any $h \in \mathcal{H}(\mathcal{F})$.

privacy. Acharya, Sun and Zhang [ASZ20] were able to obtain lower bounds in the same setting that nearly match the upper bounds of Bun, Kamath, Steinke and Wu [BKSW19]. Both [NRS07, KSSU19] consider differentially private learning of Gaussian mixtures, however their focus is on parameter estimation and therefore require additional assumptions such as separation or boundedness of the components.

There has also been a lot of work on private distribution learning and parameter estimation in the locally private setting [DJW17, WHW+16, KBR16, ASZ19, DR18, DR19, JKMW19, YB18, GRS19]. Other work on differentially private estimation include [DL09, Smi11, BD14, ASZ18, BS19, CKM+19, ZKKW20]. See [KU20] for more coverage of recent works in private statistics.

# Chapter 3

# Privately Learning Unbounded Gaussians

In this chapter, we will prove sample complexity upper bounds for privately learning the class of $d$-dimensional Gaussians where we make *no assumptions* on the parameters of the Gaussians. The content of this chapter is based on joint work with Hassan Ashtiani and Gautam Kamath [AAK21].

## 3.1 Main Results

We prove sample complexity upper bounds for learning high-dimensional Gaussians with a known covariance matrix and unknown covariance matrix. We informally state the two main results of this chapter below.

**Theorem 3.1.1** (Informal version of Theorem 3.6.4)**.** *The sample complexity of semi-agnostically learning a $d$-dimensional Gaussian distribution with known covariance to*

*$\alpha$-accuracy in total variation distance under $(\varepsilon, \delta)$-differential privacy is*

$$\widetilde{O}\left(\frac{d}{\alpha^2} + \frac{d}{\alpha\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right).$$

This is the first bound which achieves a near-optimal dependence simultaneously on all parameters. In particular, it improves upon previous results in which the third term is replaced by $O\left(\frac{\log(1/\delta)}{\alpha\varepsilon}\right)$ [BKSW19] or $O\left(\frac{\sqrt{d}\log^{3/2}(1/\delta)}{\varepsilon}\right)$ [KV18, KLSU19, BKSW19].

**Theorem 3.1.2** (Informal version of Theorem 3.6.6)**.** *The sample complexity of semi-agnostically learning a $d$-dimensional Gaussian distribution to $\alpha$-accuracy in total variation distance under $(\varepsilon, \delta)$-differential privacy is*

$$\widetilde{O}\left(\frac{d^2}{\alpha^2} + \frac{d^2}{\alpha\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right).$$

This is the first sample complexity bound for privately learning a multivariate Gaussian with no restrictions on the condition number of it's covariance matrix. Prior to this result, it was not even known if it were possible to privately learn a Gaussian that has a covariance matrix with unbounded condition number!

## 3.2 Techniques

Our proof techniques build upon the approach of Bun, Kamath, Steinke, and Wu [BKSW19] to provide methods better suited for estimation under the constraint of approximate differential privacy. Their work focuses primarily on pure DP distribution estimation for classes of distributions with a finite cover. Specifically, given a

class of distributions with an $\alpha$-cover of size $\mathcal{C}_\alpha$, they give a pure DP algorithm for learning said class in total variation distance that uses $O(\log |\mathcal{C}_\alpha|)$ samples. Naturally, this gives vacuous bounds for classes with an infinite cover – indeed, packing lower bounds show that this is inherent under pure DP [HT10, BBKN14, BKSW19]. To avoid these lower bounds, they show that learning is still possible if one relaxes to approximate DP and considers a "locally small" cover: one that has at most $k$ elements which are within an $O(\alpha)$-total variation distance ball of any element in the set. The sample complexity of the resulting method does not depend on $|\mathcal{C}_\alpha|$, and instead we pay logarithmically in the parameter $k$. They apply this framework to provide algorithms for estimating general univariate Gaussians, and multivariate Gaussians with identity covariance. However, their arguments construct explicit covers for these cases, and it appears difficult to construct and analyze covers in situations with a rich geometric structure, such as multivariate Gaussians. Indeed, it seems difficult in these settings to reason that a set is simultaneously a cover (i.e., every distribution in the class has a close element) and locally small (i.e., every distribution does not have *too many* close elements).

We avoid this tension by taking a myopic view: in Lemma 3.5.1, we show that if we can construct a cover with few elements for the neighbourhood of each *individual* distribution, then there exists a locally small cover for the *entire space*. This makes it significantly easier to reason about locally small covers, as we only have to consider covering a single distribution at a time, and we do not have to reason about how the elements that cover each distribution overlap with each other. For example: to cover the neighbourhood of a single Gaussian with (full rank) covariance $\Sigma$, we can transform the covariance to the identity by multiplying by $\Sigma^{-1/2}$, cover the neighbourhood

of $N(0, I)$ (which is easier), and transform the cover back to the original domain. This is far simpler than trying to understand how to simultaneously cover multiple Gaussians with differently shaped covariance matrices in a locally small manner. Our results for covering are presented in Section 3.5.

We then go on to apply these locally small covers to derive sample complexity upper bounds for learning high-dimensional Gaussians. As mentioned before, this is done in [BKSW19], though we refine their method to achieve stronger bounds. While this refinement is simple, we believe it to be important both technically (as it allows us to achieve likely near-optimal sample complexities) and conceptually (as we believe it clearly identifies what the "hard part" of the problem is). To elaborate, our approach can be divided into two steps;

1. **Coarse Estimation.** Find any distribution which is 0.99-close to the true distribution, using the approximate DP GAP-MAX algorithm in [BKSW19].

2. **Fine Estimation.** Generate an $O(\alpha)$-cover around the distribution from the previous step, and run the pure DP private hypothesis selection algorithm in [BKSW19].

We are not the first to use this type of two-step approach, as such decomposition has been previously applied, e.g., [KV18, KLSU19, KSU20]. However, it was not applied in the context of the GAP-MAX algorithm in [BKSW19], preventing them from getting the right dependencies on all parameters – in particular, it was not clear how to disentangle the dependencies on $\log(1/\delta)$ and $1/\alpha$ using their method directly.

As another contribution, in Section 3.4, we revisit the generic private hypothesis selection problem. The main result of Bun, Kamath, Steinke, and Wu [BKSW19] is an algorithm for this problem which requires knowledge of the distance to the best

hypothesis. They then wrap this algorithm in another procedure which "guesses" the distance to the best hypothesis, resulting in a semi-agnostic algorithm. However, this loses large factors in the agnostic guarantee and is rather indirect. We instead analyze the privatization of a different algorithm, the minimum distance estimate, which gives a semi-agnostic algorithm directly, with an optimal agnostic constant (i.e., providing a tight factor of 3 [DL01]). In our opinion, the algorithm and proof are even simpler than the non-agnostic algorithm of Bun, Kamath, Steinke, and Wu [BKSW19].

## 3.3 Preliminaries

We state some definitions and simple results that will be useful for this chapter.

We define the set of $d$-dimensional *location Gaussians* as $\mathcal{G}_L^d := \left\{\mathcal{N}(\mu, I) : \mu \in \mathbb{R}^d\right\}$, and the set of $d$-dimensional *scale Gaussians* as $\mathcal{G}_S^d := \{\mathcal{N}(0, \Sigma) : \Sigma \in \mathbb{S}_d\}$. We define the set of (all) $d$-dimensional Gaussians as $\mathcal{G}^d := \left\{\mathcal{N}(\mu, \Sigma) : \mu \in \mathbb{R}^d, \Sigma \in \mathbb{S}_d\right\}$.

The following Lemma gives us the VC dimension of the classes $\mathcal{G}_L^d$ and $\mathcal{G}^d$.

**Lemma 3.3.1.** *The class of $d$-dimensional location Gaussians $\mathcal{G}_L^d$ has VC dimension $d+1$. Furthermore, the class of $d$-dimensional Gaussians $\mathcal{G}^d$ has VC dimension $O(d^2)$.*

*Proof.* For location Gaussians, $\mathcal{H}(\mathcal{G}_L^d)$ corresponds to linear threshold functions (i.e., half-spaces), which have VC dimension $d+1$. Similarly $\mathcal{H}(\mathcal{G}^d)$ corresponds to quadratic threshold functions, which have VC dimension $\binom{d+2}{2} = O(d^2)$ [Ant95]. □

While we have defined the standard notions of realizable and semi-agnostic PAC learning in Chapter 2, we will need the following intermediate notion of learning in this thesis. In this setting, we will be able to handle model-misspecification like in

the semi-agnostic case, however we will require an upper bound on "how wrong" our model is.

**Definition 3.3.2** (($\xi, C$)-robust PAC learner)**.** *For $\xi \in (0,1)$ and $C > 0$ we say Algorithm $\mathcal{A}$ is a* ($\xi, C$)-robust PAC learner *for a class of distributions $\mathcal{F}$ which uses $\widetilde{n}_C(\alpha, \beta)$ samples, if for every $\alpha, \beta \in (0,1)$, every distribution $g$ such that $d_{\mathrm{TV}}(g, \mathcal{F}) \leq \xi$, and every $n \geq \widetilde{n}_C(\alpha, \beta)$ the following holds: if the algorithm is given parameters $\xi, \alpha, \beta$ and a sequence of $n$ i.i.d. samples from $g$ as inputs, then it outputs an approximation $\widehat{f}$ such that $d_{\mathrm{TV}}(\widehat{f}, g) \leq C \cdot \xi + \alpha$ with probability at least $1 - \beta$.*

We can also define a differential private analogue of robust PAC learners.

**Definition 3.3.3** (($\varepsilon, \delta$)-DP ($\xi, C$)-robust PAC learner)**.** *We say algorithm $\mathcal{A}$ is an* ($\varepsilon, \delta$)-DP realizable PAC learner *for a class of distributions $\mathcal{F}$ that uses $\widetilde{n}_C(\alpha, \beta, \varepsilon, \delta)$ samples if:*

1. *Algorithm $\mathcal{A}$ is a ($\xi, C$)-robust PAC learner for $\mathcal{F}$ that uses $\widetilde{n}_C(\alpha, \beta, \varepsilon, \delta)$ samples.*

2. *Algorithm $\mathcal{A}$ satisfies ($\varepsilon, \delta$)-DP.*

## 3.4 Private Hypothesis Selection

The problem of *hypothesis selection* is a classical approach for reducing estimation problems to pairwise comparisons. Roughly speaking, in hypothesis selection we are given a list of distributions $\mathcal{F}$ and sample access to an unknown distributions $g$, and our goal is to pick a distribution $\widehat{f} \in \mathcal{F}$ that is close to $g$ using samples from $g$.

Given this (informal) definition, we can see that the only difference between hypothesis selection and distribution learning is that in the former, we are required to output some distribution $\widehat{f} \in \mathcal{F}$, while in the latter, we are free to output *any* distribution (possibly not in $\mathcal{F}$). As is standard in learning theory, we call these two setting *proper learning* and *improper learning* respectively. While this difference might seem minor, it turns out to have some technical consequences. Bousquet, Kane and Moran [BKM19] recently showed that one can achieve strictly better accuracy with improper learning over proper learning. Nevertheless, for our purposes we are willing to work with the slightly weaker guarantees provided by considering proper learners. Thus, throughout this thesis we will work with proper learning algorithms and refer to hypothesis selection and distribution learning interchangeably.

### 3.4.1 Known Results

Recently, Bun, Kamath, Stenike and Wu [BKSW19] translated the powerful tools for hypothesis selection to the differentially private setting, giving an $\varepsilon$-DP algorithm for hypothesis selection that uses the exponential mechanism with a carefully constructed score function.

Theorem 3.4.1 below is a modified version of a result of Bun, Kamath, Stenike and Wu [BKSW19] where we decouple the accuracy parameter $\alpha$ from the robustness parameter $\xi$, and boost the success probability to be arbitrarily high. The proof of this modified version follows immediately from their proof.

Recall that in the $(\xi, C)$-robust PAC learning setting we have a predetermined class of distributions $\mathcal{F}$ and receive i.i.d. samples from an unknown distribution $g$, where we know the upper bound $d_{\mathrm{TV}}(g, \mathcal{F}) \leq \xi$. For a class of distributions $\mathcal{F}$, we

will denote $f^*$ as the distribution in $\mathcal{F}$ that is closest to the unknown distribution $g$. Finally, recall that we take $\widehat{f} \in \mathcal{F}$ to be the output of the learning algorithm.

**Theorem 3.4.1** ([BKSW19]). *For any $\xi, \varepsilon \in (0,1)$ and class of distributions $\mathcal{F} = \{f_1, \dots, f_m\}$, $\mathrm{PHS}(\xi, \alpha, \beta, \varepsilon, \mathcal{F}, D)$ is an $\varepsilon$-DP $(\xi, 3)$-robust PAC learner for $\mathcal{F}$ that uses*

$$\widetilde{n}_3(\alpha, \beta, \varepsilon, \delta) = O\left(\frac{\log(m/\beta)}{\alpha^2} + \frac{\log(m/\beta)}{\alpha\varepsilon}\right)$$

*samples. Furthermore, when the algorithm succeeds it guarantees that $d_{\mathrm{TV}}(\widehat{f}, f^*) \leq 2\xi + \alpha$.*

We note the guarantee that the algorithm gives with respect to $f^*$ in the theorem statement for technical reasons that will become apparent in the proofs of Section 3.4.3. Unfortunately, to get finite sample guarantees using the above result, we are limited to considering finite classes of distribution.

Using a uniform convergence argument together with a GAP-MAX algorithm [BDRS18], Bun, Kamath, Steinke, and Wu [BKSW19] showed that it may also be possible to get a similar guarantee when the size of the class of distributions is infinite, provided that we relax the notion of privacy to approximate differential privacy. The following is an alternate version of Theorem 4.1 in [BKSW19]. Again, in this version we decouple the accuracy parameter $\alpha$ from the robustness parameter $\xi$. The proof follows directly from the proof of Theorem 4.1 in [BKSW19].

**Theorem 3.4.2** ((alternate) Theorem 4.1 [BKSW19]). *For any $\xi, \varepsilon, \delta \in (0,1)$ and class of distributions $\mathcal{F}$ with VC dimension $d$ that satisfies $|\mathcal{B}(3\xi + \alpha, f^*, \mathcal{F})| \leq k$, $\text{GAP-MAX}(\xi, \alpha, \beta, \varepsilon, \delta, k, \mathcal{F}, D)$ is an $(\varepsilon, \delta)$-DP $(\xi, 4)$-robust PAC learner for $\mathcal{F}$ that*

*uses*

$$\widetilde{n}_4(\alpha,\beta,\varepsilon,\delta) = O\left(\frac{d+\log(1/\beta)}{\alpha^2} + \frac{\log(k/\beta)+\min\{\log(|\mathcal{F}|),\log(1/\delta)\}}{\alpha\varepsilon}\right)$$

*samples.*

*Furthermore, when the algorithm succeeds it guarantees that* $d_{\mathrm{TV}}(\widehat{f}, f^*) \leq 3\xi + \alpha$.

Note that Theorem 3.4.2 requires knowledge of $|\mathcal{B}(3\xi+\alpha, f^*, \mathcal{F})|$, which we likely do not know a priori. In fact, since by definition a $(\xi, C)$-robust PAC learner must work for *any* distribution $g$ satisfying $d_{\mathrm{TV}}(g, \mathcal{F}) \leq \xi$, $f^*$ can change depending on $g$. So, instead of focusing on a single $f^*$, we can find an upper bound on the size of the largest total variation ball centered at *any* $f \in \mathcal{F}$, i.e. $\sup_{f\in\mathcal{F}} |\mathcal{B}(3\xi+\alpha, f, \mathcal{F})| \leq k$. This directly translates to showing $\mathcal{F}$ is $(k, 3\xi+\alpha)$-locally small.

This lays the foundation for the strategy used by Bun, Kamath, Steinke, and Wu [BKSW19] to construct a private distribution learner for an infinite class of distributions $\mathcal{F}$: by using a $\xi$-cover for $\mathcal{F}$ that is $(k, 6\xi+\alpha)$-locally small[1] as the input to the GAP-MAX algorithm, given the right amount of samples (which depends on $k$), with high probability the algorithm outputs a distribution that is $(8\xi+\alpha)$-close to the unknown distribution $g$.

As we will see in Section 3.4.3, Bun, Kamath, Steinke, and Wu [BKSW19] also gave a method that can convert a robust PAC learner to a semi-agnostic learner at the cost of some poly-logarithmic factors and a larger agnostic constant. This leads to the natural question of whether there exists an $\varepsilon$-DP semi-agnostic learner which achieves the *same* bound on the required number of samples as the PHS algorithm with a comparable agnostic constant. We answer this question in the affirmative and

[1] Note that the guarantee we can get from any $\xi$-cover $\mathcal{C}_\xi$ is $d_{\mathrm{TV}}(f^*, \mathcal{C}_\xi) \leq \xi \implies d_{\mathrm{TV}}(g, \mathcal{C}_\xi) \leq 2\xi$.

give the details in Section 3.4.2 below.

### 3.4.2 Semi-Agnostic Private Hypothesis Selection

As a first attempt to directly obtain a semi-agnostic algorithm, Bun, Kamath, Steinke, and Wu [BKSW19] gave an $\varepsilon$-DP 9-agnostic PAC learner based on the Laplace mechanism. This algorithm – which we will refer to as Naïve-PHS – is similar to the PHS algorithm. Unfortunately, the number of samples of the Naïve-PHS algorithm uses is

$$n_9(\alpha, \beta, \varepsilon, 0) = O\left(\frac{\log(|\mathcal{F}|/\beta)}{\alpha^2} + \frac{|\mathcal{F}|^2 \log(|\mathcal{F}|/\beta)}{\alpha\varepsilon}\right),$$

which is exponentially worse than the PHS algorithm. This leads to the interesting question of whether we can do better in the semi-agnostic setting.

The PHS algorithm (Theorem 3.4.1) is based on the celebrated *Scheffé tournament* (see, e.g., Chapter 6 of [DL01]), where the distributions in $\mathcal{F}$ play a round robin tournament against one another. The winner of this tournament is then chosen as the output. One of the technical difficulties in constructing privatized versions of the Scheffé tournament via the exponential mechanism is that a single sample can quite drastically change the outcome of the tournament, which makes choosing score functions based on tournaments challenging. We can sidestep this issue completely by considering another approach to hypothesis selection called the *minimum distance estimate* (MDE).

The MDE approach is based on *maximizing* a particular function of the data and $\mathcal{F}$ as we will see shortly. Fortunately, this estimator is already in the form of a maximization problem and the function we aim to maximize has low sensitivity.

Thus, using the exponential mechanism together with the MDE is a very natural way to privatize semi-agnostic hypothesis selection.

The standard MDE requires $O(m^3)$ computations, where $m$ is the number of hypotheses in $\mathcal{F}$. Mahalanabis and Štefankovič [Mu08] presented a modified MDE that is very similar to the original MDE, but only requires $O(m^2)$ computations. Fortunately, this modified algorithm maintains the guarantee of the original algorithm, so we will privatize the modified MDE instead of the original MDE. We formally state our result below.

**Theorem 3.4.3.** *For any $\varepsilon \in (0,1)$ and class of distributions $\mathcal{F} = \{f_1, \ldots, f_m\}$, there exists an $\varepsilon$-DP 3-agnostic PAC learner for $\mathcal{F}$ that uses*

$$n_3(\alpha, \beta, \varepsilon, \delta) = O\left(\frac{\log(m/\beta)}{\alpha^2} + \frac{\log(m/\beta)}{\alpha\varepsilon}\right)$$

*samples.*

Before we prove the result, we define a few things. For an ordered pair of distributions $(f_i, f_j)$ over a common domain $\mathcal{X}$, we define their *Scheffé set* as $A_{ij} = \{x \in \mathcal{X} : f_i(x) > f_j(x)\}$. Observe that the Scheffé sets "witness" the TV distance between two distributions over the same domain. We will thus find it useful to write the TV distance between $f_i$ and $f_j$ as:

$$2d_{\mathrm{TV}}(f_i, f_j) = \big(\mathbf{P}_{f_i}(A_{ij}) - \mathbf{P}_{f_j}(A_{ij})\big) + \big(\mathbf{P}_{f_i}(A_{ji}) - \mathbf{P}_{f_j}(A_{ji})\big).$$

We note that most of the analysis in our proof of Theorem 3.4.3 is standard in proving the correctness of the MDE (e.g., see the proof of Theorem 6.3 in [DL01]), and is slightly adapted using the analysis of the modified MDE algorithm in Theorem

4 of [Mu08]. The only difference here is our use of the exponential mechanism.

*Proof of Theorem 3.4.3.* Fix some distribution $g$ and class $\mathcal{F}$ with common domain $\mathcal{X}$. For a dataset $D$ (of i.i.d. samples drawn from $g$) and set $A \subseteq \mathcal{X}$, we define $\widehat{P}(A, D) = \frac{1}{n} \cdot |\{x \in D : x \in A\}|$. For a distribution $f \in \mathcal{F}$ and a set $A \subseteq \mathcal{X}$, let $R(f, A) = \mathbf{P}_f(A) - \widehat{P}(A)$. For any $f_i \in \mathcal{F}$, we define the score function

$$\begin{aligned} S(D, f_i) &= - \sup_{j \in [m] \setminus \{i\}} \left| \left(\mathbf{P}_{f_i}(A_{ij}) - \widehat{P}(A_{ij}, D)\right) - \left(\mathbf{P}_{f_i}(A_{ji}) - \widehat{P}(A_{ji}, D)\right) \right| \\ &= - \sup_{j \in [m] \setminus \{i\}} \left| R(f_i, A_{ij}) - R(f_i, A_{ji}) \right|. \end{aligned}$$

With this in place, the algorithm is simple: run the exponential mechanism [MT07] with this score function, on the set of candidates $\mathcal{F}$, with dataset $D$, and return whichever distribution it outputs.

It is not hard to see that the score function has sensitivity $2/n$. Let $f_k \in \mathcal{F}$ be any distribution that maximizes the score function. From Theorem 2.2.4, it follows that running the exponential mechanism with our dataset $D$, the class of distributions $\mathcal{F}$, privacy parameter $\varepsilon$ and the the score function above outputs a distribution $f_{k'} \in \mathcal{F}$ that guarantees, with probability no less than $1 - \beta/2$,

$$S(D, f_{k'}) \geq S(D, f_k) - \frac{4 \log(2m/\beta)}{n\varepsilon} \geq S(D, f_k) - \alpha,$$

where the second inequality holds so long as $n = \Omega\left(\frac{\log(m/\beta)}{\alpha\varepsilon}\right)$. We condition on this

event, which can equivalently be stated as

$$\sup_{j\in[m]\setminus\{k'\}} |R(f_{k'}, A_{k'j}) - R(f_{k'}, A_{jk'})| \leq \sup_{j\in[m]\setminus\{k\}} |R(f_k, A_{kj}) - R(f_k, A_{jk})| + \alpha. \tag{3.4.1}$$

We can now bound the total variation distance between the unknown distribution $g$ and the output $f_{k'}$. Let $f_l$ be any distribution in $\mathcal{F}$ that satisfies $d_{\mathrm{TV}}(f_l, g) = \mathrm{OPT}$.[2] Using the triangle inequality we have,

$$2d_{\mathrm{TV}}(f_{k'}, g) = \|f_{k'} - g\|_1 \leq \|f_l - g\|_1 + \|f_{k'} - f_l\|_1. \tag{3.4.2}$$

We now look at the right most term in Eq. (3.4.2). By the definition of the TV distance and an application of the triangle inequality we have

$$\begin{aligned}
\|f_{k'} - f_l\|_1 &= \left(\mathbf{P}_{f_{k'}}(A_{k'l}) - \mathbf{P}_{f_l}(A_{k'l})\right) + \left(\mathbf{P}_{f_l}(A_{lk'}) - \mathbf{P}_{f_{k'}}(A_{lk'})\right) \\
&= \left|\left(\mathbf{P}_{f_{k'}}(A_{k'l}) - \mathbf{P}_{f_l}(A_{k'l})\right) + \left(\mathbf{P}_{f_l}(A_{lk'}) - \mathbf{P}_{f_{k'}}(A_{lk'})\right)\right| \\
&\leq |R(f_{k'}, A_{k'l}) - R(f_{k'}, A_{lk'})| + |R(f_l, A_{lk'}) - R(f_l, A_{k'l})| \\
&\leq \sup_{j\in[m]\setminus\{k'\}} |R(f_{k'}, A_{k'j}) - R(f_{k'}, A_{jk'})| + \sup_{j\in[m]\setminus\{l\}} |R(f_l, A_{lj}) - R(f_l, A_{jl})|.
\end{aligned}$$

Using Eq. (3.4.1), the fact that $f_k$ maximizes the score function, and the triangle

[2] Note that this implies that $\|f_l - g\|_1 = 2\mathrm{OPT}$.

inequality all together yields

$$\begin{aligned}\|f_{k'} - f_l\|_1 &\leq \sup_{j\in[m]\setminus\{k\}} |R(f_k, A_{kj}) - R(f_k, A_{jk})| + \sup_{j\in[m]\setminus\{l\}} |R(f_l, A_{lj}) - R(f_l, A_{jl})| + \alpha \\ &\leq 2 \sup_{j\in[m]\setminus\{l\}} |R(f_l, A_{lj}) - R(f_l, A_{jl})| + \alpha \\ &\leq 2 \sup_{j\in[m]\setminus\{l\}} \left|\left(\mathbf{P}_{f_l}(A_{lj}) - \mathbf{P}_g(A_{lj})\right) + \left(\mathbf{P}_g(A_{jl}) - \mathbf{P}_{f_l}(A_{jl})\right)\right| \\ &+ 2 \sup_{j\in[m]\setminus\{l\}} \left|\left(\mathbf{P}_g(A_{lj}) - \widehat{P}(A_{lj}, D)\right) + \left(\widehat{P}(A_{jl}, D) - \mathbf{P}_{f_l}(A_{jl})\right)\right| + \alpha.\end{aligned}$$

Notice that the first term on the right hand side of the final inequality is at most twice the $\ell_1$ distance between $f_l$ and $g$. Let

$$\Delta(g) = 2 \sup_{j\in[m]\setminus\{l\}} \left|\left(\mathbf{P}_g(A_{lj}) - \widehat{P}(A_{lj}, D)\right) + \left(\widehat{P}(A_{jl}, D) - \mathbf{P}_{f_l}(A_{jl})\right)\right|.$$

This gives us

$$\|f_{k'} - f_l\|_1 \leq 2\|f_l - g\|_1 + \Delta(g) + \alpha.$$

Furthermore, notice that the term $\Delta(g)$ is small when the difference between the empirical and the true probability measures assigned by $g$ to the Scheffè sets is small. We can thus upper bound this term by $\alpha$ by using $2\binom{m}{2}$ standard Chernoff bounds together with a union bound to get

$$\|f_{k'} - f_l\|_1 \leq 2\|f_l - g\|_1 + 2\alpha, \tag{3.4.3}$$

with probability no less than $1 - \beta/2$ so long as $n = \Omega\left(\frac{\log(m/\beta)}{\alpha^2}\right)$. Putting Eq. (3.4.2)

and Eq. (3.4.3) together gives us,

$$d_{\mathrm{TV}}(f_{k'}, g) = 3\mathrm{OPT} + \alpha.$$

A union bound together with setting $n = \Omega\left(\frac{\log(m/\beta)}{\alpha^2} + \frac{\log(m/\beta)}{\alpha\varepsilon}\right)$ completes the proof.

□

### 3.4.3 Converting a Robust PAC Learner to a Semi-Agnostic PAC Learner

Unfortunately, the algorithms in Theorem 3.4.1 and Theorem 3.4.2 are not semi-agnostic and require an upper bound on $\mathrm{OPT} = d_{\mathrm{TV}}(g, \mathcal{F})$ via $\xi$. Although we have given an $(\varepsilon, \delta)$-DP semi-agnostic algorithm comparable to PHS in Section 3.4.2, we do not have an analogous $(\varepsilon, \delta)$-DP semi-agnostic algorithm for the GAP-MAX algortihm.

To get around this issue of having to know $\xi$, Bun, Kamath, Steinke, and Wu [BKSW19] gave a simple procedure that takes an $(\varepsilon, \delta)$-DP robust PAC learner and constructs an $(\varepsilon, \delta)$-DP semi-agnostic PAC learner, at the cost of some low order poly-logarithmic factors in the bounds on the required number of samples, and an increase in the agnostic constant. We can thus use the GAP-MAX algorithm together with this procedure to get an $(\varepsilon, \delta)$-DP semi-agnostic PAC leaner given an infinite class of distributions. The procedure [BKSW19] came up with works in the following way: run the $(\varepsilon, \delta)$-DP robust PAC learner with (a small number of) different values for $\xi$ to get a shortlist of candidates. Use the semi-agnostic NaïvePHS algorithm to select a good hypothesis from the short list. As we mentioned earlier, the guarantee of this

approach (Theorem 3.4 in [BKSW19]) is stated specifically in terms of converting the PHS algorithm from Theorem 3.4.1 into an $\varepsilon$-DP semi-agnostic PAC learner, however it can be immediately generalized to construct $(\varepsilon, \delta)$-DP semi-agnostic PAC learners given any $(\varepsilon, \delta)$-DP robust PAC learner. Furthermore, we can replace the Naïve-PHS algorithm with the sample efficient algorithm from Theorem 3.4.3 to reduce the agnostic constant, and also remove some logarithmic factors in the bound on the required number of samples. This yields the following result.

**Lemma 3.4.4.** *Let $\varepsilon, \delta \in (0,1)$ and $T = \lceil \log_2(1/\alpha) \rceil$. Given an $(\varepsilon, \delta)$-DP $(\xi, C)$-robust PAC learner for a class of distributions $\mathcal{F}$ that uses $\widetilde{n}_C(\alpha, \beta, \varepsilon, \delta)$ samples, there exists an $(\varepsilon, \delta)$-DP $6C$-agnostic PAC learner for $\mathcal{F}$ that uses*

$$n_{6C}(\alpha, \beta, \varepsilon, \delta) = \widetilde{n}_C\left(\frac{\alpha}{12}, \frac{\beta}{2(T+4)}, \frac{\varepsilon}{2(T+4)}, \frac{\delta}{T+4}\right) + O\left(\frac{\log(T/\beta)}{\alpha^2} + \frac{\log(T/\beta)}{\alpha\varepsilon}\right)$$

*samples.*

*Proof.* Fix parameters $\varepsilon, \delta, \alpha, \beta \in (0,1)$. We split our dataset $D \sim g^n$ into $D_1$ and $D_2$, where $|D_1| + |D_2| = n$. Let $T = \lceil \log_2(1/\alpha) \rceil$. For all $t \in [T+4]$, let $\xi_t = 2^{t-1}\alpha/12C$.

The algorithm is simple: we run the $(\varepsilon, \delta)$-DP $(\xi, C)$-robust PAC learner $T+4$ times, where each run $t$ uses parameters $\alpha/12$, $\beta/2(T+4)$, $\varepsilon/2(T+4)$, $\delta/2(T+4)$ and dataset $D_1$. We then use the private semi-agnostic learner from Theorem 3.4.3 with parameters $\alpha$, $\beta/2$, $\varepsilon/2$, and dataset $D_2$. By basic composition of DP (Lemma 2.2.2), it follows immediately that our algorithm is $(\varepsilon, \delta)$-DP.

We now argue about the accuracy of the algorithm. Let $f_t$ be the otuput of run $t$ of the robust PAC learner. If $|D_1| = \widetilde{n}_C\left(\frac{\alpha}{12}, \frac{\beta}{2(T+4)}, \frac{\varepsilon}{2(T+4)}, \frac{\delta}{T+4}\right)$, a union bound guarantees that with probability at least $1 - \beta/2$ each $f_t$ is accurate (assuming $\xi_t$ is

a correct upper bound for OPT).

Notice that if run $t$ succeeds and OPT $\in (\xi_{t-1}, \xi_t]$, we have $d_{\mathrm{TV}}(g, f_t) \leq C\xi_t + \alpha/12 \leq 2C \cdot \mathrm{OPT} + \alpha/12$. Similarly, if OPT $\leq \xi_1$ then $f_1$ satisfies $d_{\mathrm{TV}}(g, f_1) \leq C\xi_1 + \alpha_1 = \alpha/12 + \alpha/12 = \alpha/6$. Finally, if OPT $> \xi_{T+4}$ this implies OPT $> 8/12C$, so *any* distribution $f \in \mathcal{F}$ satisfies $d_{\mathrm{TV}}(g, f) \leq 2C \cdot \mathrm{OPT}$ trivially. So regardless of OPT, there is a run $t$ that satisfies $d_{\mathrm{TV}}(g, f_t) \leq 2C \cdot \mathrm{OPT} + \alpha/6$.

Finally, Theorem 3.4.3 guarantees that, with probability greater than $1-\beta/2$, the second step in our above procedure will output a distribution $\widehat{f}$ such that

$$d_{\mathrm{TV}}(g, \widehat{f}) \leq 3 \inf_{f_t} d_{\mathrm{TV}}(g, f_t) + \alpha/2,$$

as long as $|D_2| = \Omega\left(\frac{\log(T/\beta)}{\alpha^2} + \frac{\log(T/\beta)}{\alpha\varepsilon}\right)$.

Thus, by a union bound, we have with probability at least $1-\beta$ that our algorithm outputs a distribution $\widehat{f}$ such that $d_{\mathrm{TV}}(g, \widehat{f}) \leq 3(2C\cdot\mathrm{OPT}+\alpha/6)+\alpha/2 = 6C\cdot\mathrm{OPT}+\alpha$ so long as

$$n = \widetilde{n}_C\left(\frac{\alpha}{12}, \frac{\beta}{2(T+4)}, \frac{\varepsilon}{2(T+4)}, \frac{\delta}{T+4}\right) + \Omega\left(\frac{\log(T/\beta)}{\alpha^2} + \frac{\log(T/\beta)}{\alpha\varepsilon}\right).$$

□

## 3.5 Covering Unbounded Gaussians

In this section, we demonstrate a simple method to prove that a class of distributions has a locally small cover. As an application, we use this result to show that

the set of *unbounded* location Gaussians and scale Gaussians have locally small covers. We use these two results to give the first sample complexity result for privately learning unbounded high dimensional Gaussians in Section 3.6.

### 3.5.1 From Covering TV balls to Locally Small Covers

The biggest roadblock to using Theorem Theorem 3.4.2 is demonstrating the existence of a locally small cover for the class of distributions $\mathcal{F}$. Unfortunately, explicitly constructing a *global* cover (which is locally small) can be complicated, and may require cumbersome calculations even for "simple" distributions (see, e.g., Lemma 6.13 of [BKSW19]). We offer a conceptually simpler alternative to prove a class of distributions $\mathcal{F}$ has a locally small cover: we demonstrate that if for every $f \in \mathcal{F}$ the total variation ball $\mathcal{B}(\gamma, f, \mathcal{F})$ has an $\frac{\xi}{2}$-cover of size no more than $k$, then there exists a $\xi$-cover for $\mathcal{F}$ that is $(k, \gamma)$-locally small.

**Lemma 3.5.1.** *Given a class of distributions $\mathcal{F}$ and $\xi \in (0,1)$, if for* every *distribution $f \in \mathcal{F}$ the total variation ball $\mathcal{B}(\gamma, f, \mathcal{F}) \subseteq \mathcal{F}$ has an $\frac{\xi}{2}$-cover of size no more than $k$, then there exists a $(k, \gamma)$-locally small $\xi$-cover for $\mathcal{F}$.*

*Proof.* Fix some $f \in \mathcal{F}$. By assumption, we have that the set of distributions $\mathcal{B}(\gamma, f, \mathcal{F})$ has an $\frac{\xi}{2}$-cover of size no more than $k$, which by definition implies that the $\frac{\xi}{2}$-covering number of $\mathcal{B}(\gamma, f, \mathcal{F})$ is no more than $k$. By Proposition 2.3.9, the $\xi$-packing number of $\mathcal{B}(\gamma, f, \mathcal{F})$ is also at most $k$.

Now consider a $\xi$-packing $\mathcal{P}_\xi$ for the class of distributions $\mathcal{F}$. We claim any such $\mathcal{P}_\xi$ must be $(k, \gamma)$-locally small, and we prove this by contradiction. Suppose to the contrary that there were a distribution $f' \in \mathcal{P}_\xi$ such that $|\mathcal{B}(\gamma, f', \mathcal{P}_\xi)| > k$. This would imply that there is a $\xi$-packing for $\mathcal{B}(\gamma, f', \mathcal{F})$ with size larger than $k$, which

contradicts the above observation that the packing number of *any* $\mathcal{B}(\gamma, f', \mathcal{F})$ is at most $k$.

A $\xi$-packing for $\mathcal{F}$ is called maximal if it is impossible to add a new element of $\mathcal{F}$ to it without violating the $\xi$-packing property. We claim that any maximal packing $\mathcal{P}'_\xi$ of $\mathcal{F}$ is also a $\xi$-cover of $\mathcal{F}$. We can prove this by contradiction. Suppose to the contrary that there were a distribution $f'' \in \mathcal{F}$ with $d_{\mathrm{TV}}(f'', \mathcal{P}'_\xi) > \xi$. Then we could add $f''$ to $\mathcal{P}'_\xi$ to produce a strictly larger packing, contradicting the maximality of $\mathcal{P}'_\xi$. Thus taking $\mathcal{P}_\xi$ to be a maximal packing gives us a $(k, \gamma)$-locally small $\xi$-cover. Therefore, it only remains to show that a maximal packing actually exists, which follows from a simple application of Zorn's Lemma.[3] □

### 3.5.2 Locally Small Gaussian Covers

We now prove that both the class of $d$-dimensional location Gaussians and scale Gaussians can be covered in a locally small fashion. Our first result shows that the class of $d$-dimensional location Gaussians $\mathcal{G}_L^d$ has a locally small cover. Our second result is proving the existence of a locally small cover for the class of $d$-dimensional scale Gaussians $\mathcal{G}_S^d$.

#### Covering Location Gaussians

It is not too difficult to come up with an explicit locally small cover for the set of location Gaussians without using Lemma 3.5.1 as is demonstrated in [BKSW19,

[3]Let $M$ be the set of all $\gamma$-packings of $\mathcal{F}$. Define a partial order on $M$ by the relation $\mathcal{P}_1 \leq \mathcal{P}_2 \iff \mathcal{P}_1 \subseteq \mathcal{P}_2$ where $\mathcal{P}_1, \mathcal{P}_2 \in M$. We claim that every chain in this partially ordered set has an upper bound in $M$; by Zorn's lemma, this would imply that $M$ has a maximal element which concludes the proof. To see why every (possibly infinite) chain $\mathcal{P}_1 \leq \mathcal{P}_2 \leq \ldots$ has an upper bound in $M$, we consider the following upper bound $U = \cup_i \mathcal{P}_i$. Note that $U \in M$ since otherwise there would be an index $i$ such that $\mathcal{P}_i \notin M$.

Lemma 6.12]. Nonetheless, we choose to do so as a warmup before attempting to solve the (harder) problem for scale Gaussians. In the case of location Gaussians, our proof is very similar to Lemma 6.12 in [BKSW19]. Constructing an explicit cover is not too difficult in this case because the geometry of $\mathcal{G}_L^d$ is "simple", given that the TV distance between any two distributions is determined by the $\ell_2$ distance of their means. Unfortunately the situation is not that simple in the scale Gaussian case as we will see shortly. We begin by showing that the TV ball centered at any Gaussian $\mathcal{N}(\mu, I)$ with respect to $\mathcal{G}_L^d$ can be covered, as long as the radius is not too large.

**Lemma 3.5.2.** *For any $d \in \mathbb{N}$, $\mu \in \mathbb{R}^d$, $\gamma \in (0, c_1)$ and $\xi \in (0, \gamma)$ where $c_1$ is a universal constant, there exists a $\xi$-cover for the set of distributions $\mathcal{B}\left(\gamma, \mathcal{N}(\mu, I), \mathcal{G}_L^d\right)$ of size*

$$\left(\frac{\gamma}{\xi}\right)^{O(d)}.$$

*Proof.* Fix some $\mathcal{N}(\mu, I) \in \mathcal{G}_L^d$. From [DMR18, Theorem 1.2] we have

$$\frac{1}{200}\min\left\{1, \|\mu_1 - \mu_2\|_2\right\} \leq d_{\mathrm{TV}}\left(\mathcal{N}(\mu_1, I), \mathcal{N}(\mu_2, I)\right) \leq \frac{9}{2}\min\left\{1, \|\mu_1 - \mu_2\|_2\right\}. \tag{3.5.1}$$

For any $\gamma$ smaller than the universal constant $c_1$, the lower bound in Eq. (3.5.1) implies that any $\mathcal{N}(\tilde{\mu}, I) \in \mathcal{B}\left(\gamma, \mathcal{N}(\mu, I), \mathcal{G}_L^d\right)$ must satisfy $\|\mu - \tilde{\mu}\|_2 \leq 200\gamma$. We thus propose the following cover:

$$\mathcal{C}_\xi = \left\{\mathcal{N}(\mu + \hat{z}, I) : \hat{z} \in \left(\frac{2\xi}{9\sqrt{d}}\right)\mathbb{Z}^d, \|\hat{z}\|_2 \leq 200\gamma\right\}.$$

We now prove that $\mathcal{C}_\xi$ is a valid $\xi$-cover. Fix some $\mathcal{N}(\tilde{\mu}, I) \in \mathcal{B}\left(\gamma, \mathcal{N}(\mu, I), \mathcal{G}_L^d\right)$

and define $z = \tilde{\mu} - \mu$. We know $\|z\|_2 \leq 200\gamma$. Let $\hat{z} = \left(\frac{2\xi}{9\sqrt{d}}\right) \lfloor \left(\frac{9\sqrt{d}}{2\xi}\right) z \rfloor$ and $\hat{\mu} = \mu + \hat{z}$. Note that we have $\mathcal{N}(\hat{\mu}, I) \in \mathcal{C}_\xi$. Furthermore, $z$ and $\hat{z}$ are element-wise close ($\|z - \hat{z}\|_\infty \leq 2\xi/9\sqrt{d}$) therefore we have

$$\begin{aligned} d_{\mathrm{TV}}\left(\mathcal{N}(\tilde{\mu}, I), \mathcal{N}(\hat{\mu}, I)\right) &\leq \frac{9}{2}\|\tilde{\mu} - \hat{\mu}\|_2 \\ &= \frac{9}{2}\|z - \hat{z}\|_2 \\ &\leq \frac{9\sqrt{d}}{2}\|z - \hat{z}\|_\infty \\ &\leq \frac{9\sqrt{d}}{2} \cdot \frac{2\xi}{9\sqrt{d}} \\ &= \xi, \end{aligned}$$

where the first inequality follows from Eq. (3.5.1). We now bound the size of this cover.

$$\begin{aligned} |\mathcal{C}_\xi| &= \left| \left\{ \mathcal{N}(\mu + \hat{z}, I) : \hat{z} \in \left(\frac{2\xi}{9\sqrt{d}}\right) \mathbb{Z}^d, \|\hat{z}\|_2 \leq 200\gamma \right\} \right| \\ &\leq \left| \left\{ \hat{z} : \hat{z} \in \mathbb{Z}^d, \|\hat{z}\|_2 \leq \frac{900\sqrt{d}\gamma}{\xi} \right\} \right| \leq \left| \left\{ \hat{z} : \hat{z} \in \mathbb{Z}^d, \|\hat{z}\|_1 \leq \frac{900d\gamma}{\xi} \right\} \right| \\ &\leq \left| \left\{ z_1 - z_2 : z_1, z_2 \in \mathbb{Z}^d_+, \|z_1\|_1 \leq \left\lceil \frac{900d\gamma}{\xi} \right\rceil, \|z_2\|_1 \leq \left\lceil \frac{900d\gamma}{\xi} \right\rceil \right\} \right| \\ &\leq \left| \left\{ z : z \in \mathbb{Z}^d_+, \|z\|_1 \leq \left\lceil \frac{900d\gamma}{\xi} \right\rceil \right\} \right|^2 \\ &\leq \left( \sum_{i=1}^{\lceil 900d\gamma/\xi \rceil} \binom{i+d-1}{d-1} \right)^2 \\ &\leq \left( \left\lceil \frac{900d\gamma}{\xi} \right\rceil \binom{\lceil 900d\gamma/\xi \rceil + d - 1}{d-1} \right)^2 \leq \left(\frac{\gamma}{\xi}\right)^{O(d)}, \end{aligned}$$

where the third last inequality follows from the standard solution to the stars and bars problem. □

Combining Lemma 3.5.1 with Lemma 3.5.2 immediately gives us the following corollary:

**Corollary 3.5.3.** *For any $d \in \mathbb{N}$, $\gamma \in (0, c_1)$, and $\xi \in (0, \gamma)$ where $c_1$ is a universal constant, there exists a $\xi$-cover $\mathcal{C}_\xi$ for the class of $d$-dimensional location Gaussians $\mathcal{G}_L^d$ that is $\left((2\gamma/\xi)^{O(d)}, \gamma\right)$-locally small.*

### Covering Scale Gaussians

It is not a trivial exercise to come up with an explicit cover for the class of scale Gaussians due to the complicated nature of the geometry of $\mathcal{G}_S^d$. Fortunately for us, Lemma 3.5.1 simplifies things significantly. It turns out that if we want to cover the TV ball centered at any $\mathcal{N}(0, \Sigma)$, we can use a cover for $\mathcal{N}(0, I)$ and "stretch" the covariance matrices of every distribution in the cover (using $\Sigma$) so that the modified cover becomes a valid cover for the TV ball centered at $\mathcal{N}(0, \Sigma)$. The following lemma tells us that we can cover the total variation ball centered at $\mathcal{N}(0, I)$ with respect to $\mathcal{G}_S^d$ as long as the radius is not too large.

**Lemma 3.5.4.** *For any $d \in \mathbb{N}$, $\gamma \in (0, c_2)$ and $\xi \in (0, \gamma)$ where $c_2$ is a universal constant, there exists a $\xi$-cover for the set of distributions $\mathcal{B}\left(\gamma, \mathcal{N}(0, I), \mathcal{G}_S^d\right)$ of size*

$$\left(\frac{\gamma}{\xi}\right)^{O(d^2)}.$$

*Proof.* From [DMR18, Theorem 1.1] we have,

$$d_{\mathrm{TV}}(\mathcal{N}(0,I),\mathcal{N}(0,\Sigma)) \geq \frac{1}{100}\min\left\{1, \sqrt{\sum_{i=1}^{d}\lambda_i^2}\right\}, \tag{3.5.2}$$

where $\lambda_1 \dots \lambda_d$ are the eigenvalues of $\Sigma - I$, and it holds that $\sqrt{\sum_{i=1}^{d}\lambda_i^2} = \|\Sigma - I\|_F$.

For any $\gamma$ smaller than the universal constant $c_2'$, the lower bound in Eq. (3.5.2) implies two things: 1) for any $\mathcal{N}(0,\Sigma) \in \mathcal{B}\left(\gamma, \mathcal{N}(0,I), \mathcal{G}_S^d\right)$, $\|\Sigma - I\|_F \leq 100\gamma$ and 2) the minimum eigenvalue of $\Sigma$, $\lambda_{\min}$, satisfies $\lambda_{\min} \geq 1 - 100\gamma$.

We thus propose the following cover:

$$\mathcal{C}_\xi = \left\{\mathcal{N}(0, I + \hat{\Delta}) : \hat{\Delta} \in \rho\mathbb{Z}^{d\times d} \cap \mathbb{S}_d, \|\hat{\Delta}\|_F \leq 100\gamma\right\},$$

where $\rho = \frac{\xi\sqrt{2\pi e}(1-100\gamma)}{d+\xi\sqrt{2\pi e}}$. First we will show that this is a valid cover. Consider an arbitrary $\mathcal{N}(0,\Sigma) \in \mathcal{B}\left(\gamma, \mathcal{N}(0,I), \mathcal{G}_S^d\right)$. We want to show that there is a distribution $\mathcal{N}(0,\hat{\Sigma}) \in \mathcal{C}_\xi$ that is $\xi$-close to $\mathcal{N}(0,\Sigma)$. Let $\Delta = \Sigma - I$, let $\hat{\Delta} = \rho\lfloor\Delta/\rho\rfloor$, and let $\hat{\Sigma} = I + \hat{\Delta}$. Since $\|\Delta\|_F = \|\Sigma - I\|_F \leq 100\gamma$, $\mathcal{N}(0,\hat{\Sigma})$ is indeed in the cover.

Next we show that $d_{\mathrm{TV}}(\mathcal{N}(0,\Sigma),\mathcal{N}(0,\hat{\Sigma})) \leq \xi$. We use Proposition 32 in [VV10], which states for any two positive definite matrices $\Sigma$ and $\hat{\Sigma}$, if $\|\Sigma - \hat{\Sigma}\|_{\infty,\infty} \leq \rho'$ and the smallest eigenvalue of $\Sigma$ satisfies $\lambda_{\min} > \eta$, then we have

$$d_{\mathrm{TV}}(\mathcal{N}(0,\Sigma),\mathcal{N}(0,\hat{\Sigma})) \leq \frac{d\rho'}{\sqrt{2\pi e}(\eta - \rho')}. \tag{3.5.3}$$

By the definition of $\mathcal{C}_\xi$, $\|\hat{\Delta} - \Delta\|_{\infty,\infty} \leq \rho$. Since any valid $\Sigma$ must satisfy $\lambda_{\min} \geq$

$1-100\gamma$, our choice of setting $\rho = \frac{\xi\sqrt{2\pi e}(1-100\gamma)}{d+\xi\sqrt{2\pi e}}$ implies that

$$d_{\mathrm{TV}}(\mathcal{N}(0,\Sigma),\mathcal{N}(0,\hat{\Sigma})) \leq \xi,$$

for any $\gamma$ smaller than the universal constant $c_2'$.

We now bound the size of the cover in a similar manner to the case of location Gaussians.

$$\begin{aligned}
|\mathcal{C}_\xi| &= \left|\left\{\hat{\Delta} \in \rho\mathbb{Z}^{d\times d} \cap \mathbb{S}_d : \|\hat{\Delta}\|_F \leq 100\gamma\right\}\right| \\
&\leq \left|\left\{\hat{\Delta} \in \mathbb{Z}^{d\times d} : \|\hat{\Delta}\|_F \leq 100\gamma/\rho\right\}\right| \\
&\leq \left|\left\{\hat{\Delta} \in \mathbb{Z}^{d\times d} : \|\hat{\Delta}\|_{1,1} \leq 100\gamma d/\rho\right\}\right| \\
&\leq \left|\left\{\hat{\Delta} \in \mathbb{Z}_+^{d\times d} : \|\hat{\Delta}\|_{1,1} \leq \lceil 100\gamma d/\rho\rceil\right\}\right|^2 \\
&\leq \left(\left\lceil \frac{100\gamma d(d+\xi\sqrt{2\pi e})}{\xi\sqrt{2\pi e}(1-100\gamma)}\right\rceil \cdot \binom{\left\lceil \frac{100\gamma d(d+\xi\sqrt{2\pi e})}{\xi\sqrt{2\pi e}(1-100\gamma)}\right\rceil + d^2 - 1}{d^2-1}\right)^2,
\end{aligned}$$

for any $\gamma$ smaller than the universal constant $c_2''$ we have,

$$|\mathcal{C}_\xi| \leq \left(\frac{\gamma}{\xi}\right)^{O(d^2)}.$$

Setting $c_2 = \min\{c_2', c_2''\}$ completes the proof. □

The following corollary is a direct consequence of Lemma 3.5.4 and Proposition A.0.1.

**Corollary 3.5.5.** *For any $d \in \mathbb{N}$, $\gamma \in (0, c_2)$, $\xi \in (0,\gamma)$ where $c_2$ is a universal constant, and $\Sigma \in \mathbb{S}_d$, there exists a $\xi$-cover for the set of distributions $\mathcal{B}\left(\gamma, \mathcal{N}(0,\Sigma), \mathcal{G}_S^d\right)$*

*of size*

$$\left(\frac{\gamma}{\xi}\right)^{O(d^2)}.$$

*Proof.* Fix $\gamma \in (0, c_2)$, $\xi \in (0, \gamma)$ and $\Sigma \in \mathbb{S}_d$. We first consider the case that $\Sigma$ has full rank.

Let $\Sigma^{1/2}$ be the unique matrix square root of $\Sigma$. Our cover is simple; we take the cover $\mathcal{C}_\xi$ from Lemma 3.5.4 and replace every Gaussian $\mathcal{N}(0, \Sigma_1) \in \mathcal{C}_\xi$ with $\mathcal{N}(0, \Sigma^{1/2}\Sigma_1\Sigma^{1/2})$ to get the modified cover $\widehat{\mathcal{C}}_\xi$. Note that the size of the modified cover remains the same.

We now argue that this is a valid cover for $\mathcal{B}\left(\gamma, \mathcal{N}(0,\Sigma), \mathcal{G}_S^d\right)$ and we prove this by contradiction. Suppose to the contrary that there is a distribution $\mathcal{N}(0,\Sigma') \in \mathcal{B}\left(\gamma, \mathcal{N}(0,\Sigma), \mathcal{G}_S^d\right)$ such that $d_{\mathrm{TV}}(\mathcal{N}(0,\Sigma'), \widehat{\mathcal{C}}_\xi) > \xi$. It follows from Corollary A.0.2, Eq. (2.1.1), and our assumption on $\mathcal{N}(0,\Sigma')$ that

$$d_{\mathrm{TV}}\left(\mathcal{N}\left(0, \Sigma^{-1/2}\Sigma'\Sigma^{-1/2}\right), \mathcal{C}_\xi\right) = d_{\mathrm{TV}}\left(\mathcal{N}\left(0,\Sigma'\right), \widehat{\mathcal{C}}_\xi\right) > \xi,$$

which implies that $\mathcal{N}\left(0, \Sigma^{-1/2}\Sigma'\Sigma^{-1/2}\right) \notin \mathcal{B}\left(\gamma, \mathcal{N}(0,I), \mathcal{G}_S^d\right)$ since $\mathcal{C}_\xi$ is a $\xi$-cover for $\mathcal{B}\left(\gamma, \mathcal{N}(0,I), \mathcal{G}_S^d\right)$. However, by a similar observation, we have that

$$d_{\mathrm{TV}}\left(\mathcal{N}\left(0, \Sigma^{-1/2}\Sigma'\Sigma^{-1/2}\right), \mathcal{N}\left(0,I\right)\right) = d_{\mathrm{TV}}\left(\mathcal{N}\left(0,\Sigma'\right), \mathcal{N}\left(0,\Sigma\right)\right) \leq \gamma,$$

which implies that $\mathcal{N}\left(0, \Sigma^{-1/2}\Sigma'\Sigma^{-1/2}\right) \in \mathcal{B}\left(\gamma, \mathcal{N}(0,I), \mathcal{G}_S^d\right)$, a contradiction. Thus we have proven that $\widehat{\mathcal{C}}_\xi$ is a $\xi$-cover for $\mathcal{B}\left(\gamma, \mathcal{N}(0,\Sigma), \mathcal{G}_S^d\right)$.

We now consider the case that $\Sigma$ has rank $r < d$. It is a well known fact that any Gaussian $g \in \mathcal{G}_S^d$ that satisfies $d_{\mathrm{TV}}(g, \mathcal{N}(0,\Sigma)) < 1$ must have a covariance matrix with the same range as $\Sigma$. Thus, every Gaussian in $\mathcal{B}\left(\gamma, \mathcal{N}(0,\Sigma), \mathcal{G}_S^d\right)$ must have a

covariance matrix with the same range as $\Sigma$.

Let $\Pi$ be a $d \times r$ matrix whose columns form an orthonormal basis for range($\Sigma$) and let $\widehat{\Sigma} = \Pi^\top \Sigma \Pi$. Notice that $\widehat{\Sigma}$ is an $r \times r$ positive definite matrix. We first obtain a $\xi$-cover for $\mathcal{B}\left(\gamma, \mathcal{N}(0, \widehat{\Sigma}), \mathcal{G}_S^r\right)$ and modify this cover to be a $\xi$-cover for $\mathcal{B}\left(\gamma, \mathcal{N}(0, \Sigma), \mathcal{G}_S^d\right)$. Since $\widehat{\Sigma}$ has full rank, we can use the cover $\widehat{\mathcal{C}}_\xi$ we derived in the full rank case above as cover for $\mathcal{B}\left(\gamma, \mathcal{N}(0, \widehat{\Sigma}), \mathcal{G}_S^r\right)$. To modify this to be a $\xi$-cover for $\mathcal{B}\left(\gamma, \mathcal{N}(0, \Sigma), \mathcal{G}_S^d\right)$, we replace every distribution $\mathcal{N}(0, \Sigma_2) \in \widehat{\mathcal{C}}_\xi$ with $\mathcal{N}(0, \Pi \Sigma_2 \Pi^\top)$ to get the modified cover $\widetilde{\mathcal{C}}_\xi$. A simple proof by contradiction similar to the one above shows that this is a valid cover for $\mathcal{B}\left(\gamma, \mathcal{N}(0, \Sigma), \mathcal{G}_S^d\right)$. Finally, it is easy to see that $|\widetilde{\mathcal{C}}_\xi| = O(\gamma/\xi)^{O(r^2)} < O(\gamma/\xi)^{O(d^2)}$. This completes the proof.

□

We can now combine Lemma 3.5.1 with Corollary 3.5.5 to get the following:

**Corollary 3.5.6.** *For any $d \in \mathbb{N}$, $\gamma \in (0, c_2)$, and $\xi \in (0, \gamma)$ where $c_2$ is a universal constant, there exists an $\xi$-cover $\mathcal{C}_\xi$ for the set of scale Gaussians $\mathcal{G}_S^d$ that is $\left((2\gamma/\xi)^{O(d^2)}, \gamma\right)$-locally small.*

## 3.6 Boosting Weak Hypotheses

As we mentioned before, by using a $(k, 6\xi + \alpha)$-locally small $\xi$-cover for an infinite set of distributions $\mathcal{F}$, one can utilize Theorem 3.4.2 to privately learn a distribution to low error. Unfortunately, this approach will yield a sample complexity bound that has a term of order $O(\log(1/\delta)/\alpha\varepsilon)$. In the case of learning an unbounded univariate Gaussian in the realizable setting, it is known that the sample complexity is $O(1/\alpha^2 + \log(1/\delta)/\varepsilon)$ [KV18], however the upper bound on the sample complexity

achieved by Theorem 3.4.2 (together with an appropriate locally smaller cover) is $O(1/\alpha^2 + \log(1/\delta)/\alpha\varepsilon)$ [BKSW19, Corollary 6.15]. In order to overcome the poor dependence on $\log(1/\delta)$, we can instead aim for a two step approach:

1. Use the GAP-MAX algorithm in Theorem 3.4.2 but with constant accuracy $C$ to learn a distribution $f'$ that is roughly $C$-close to the true Gaussian for some appropriately selected constant $C < 1$.

2. Build a *finite* cover for $\mathcal{B}(C, f', \mathcal{F})$ and use the private hypothesis selection algorithm (Theorem 3.4.1) to learn a distribution $\widehat{f}$ that is $\alpha$-close to the true Gaussian.

Running the GAP-MAX algorithm with constant accuracy $C$ thus removes the dependence on $\alpha$ in the $O(\log(1/\delta)/\varepsilon)$ term. Intuitively, this approach learns a "rough" estimate of the right distribution using approximate differential privacy. Since we know that we are roughly $C$-close to the true Gaussian, we can cover $\mathcal{B}(C, f', \mathcal{F})$ with a finite cover, and use the $\varepsilon$-differentially private hypothesis selection algorithm. This two step approach which we dub *boosting* gets us a much better dependence on the privacy parameter $\delta$ in our sample complexity bounds, and as we will see it holds more generally in the robust learning setting.

**Remark 3.6.1.** *We note that the first step in the above approach may only need to produce an exceptionally coarse estimate to the true distribution – one to which it bears very little resemblance at all! We illustrate this with the simple problem of privately estimating a univariate Gaussian $\mathcal{N}(\mu, 1)$ (in the realizable case).*

*We work backwards: our overall target is an algorithm with sample complexity $\widetilde{O}(1/\alpha^2 + 1/\alpha\varepsilon + \log(1/\delta)/\varepsilon)$. Since using the pure DP hypothesis selection algorithm*

*of Theorem 3.4.1 takes* $O(\log|\mathcal{C}_\alpha|(1/\alpha^2+1/\alpha\varepsilon))$ *samples, we require only that* $|\mathcal{C}_\alpha|$ *is less than some quasi-polynomial in* $1/\alpha$*. For the sake of exposition, suppose we restrict further and require* $|\mathcal{C}_\alpha| \leq 1/\alpha^{101}$*. This can be achieved by starting at any point which is at most* $1/\alpha^{100}$ *from the true mean* $\mu$ *and taking an* $\alpha$*-additive grid over this space. But if we only require a starting point* $\hat{\mu}$ *which is* $1/\alpha^{100}$*-close to the true mean* $\mu$*, this corresponds (by Gaussian tail bounds) to a distribution whose total variation distance is roughly* $1-\exp(-1/\alpha^{200})$ *with respect to the true distribution.*

*We can see that the first step in the procedure truly requires an exceptionally coarse estimate of the distribution. The estimate of the mean described is significantly further from the true mean than any individual point will be. Interestingly, note that if one requires a more accurate final distribution, the distribution output in the first step is allowed to be less accurate.*

### 3.6.1 Warmup: Learning Location Gaussians

As a first step, we can show that Algorithm 1 can achieve a slightly more general guarantee than a robust PAC learner. We make Algorithm 1 more general than it needs to be to give a robust learning guarantee for $\mathcal{G}_L^d$ in order to make use of it as a subroutine in Algorithm 2 which robustly learns $\mathcal{G}^d$.

**Lemma 3.6.2.** *For any* $b \geq 1$*,* $\beta,\varepsilon,\delta \in (0,1)$*,* $\alpha \in (0,\frac{4c_1}{3b+12})$ *and* $\xi \in (0,\frac{c_1}{3b+4})$ *where* $c_1$ *is a universal constant, given dataset* $D \sim g^n$ *where* $g$ *satisfies* $d_{\mathrm{TV}}(g,\mathcal{G}_L^d) \leq b\xi+\alpha/4$*, Algorithm 1 is an* $(\varepsilon,\delta)$*-DP algorithm which outputs some* $\widehat{f} \in \mathcal{G}_L^d$ *such that* $d_{\mathrm{TV}}(\widehat{f},g) \leq 3(b+1)\xi+\alpha$ *with probability no less than* $1-\beta$*, so long as*

$$n = \Omega\left(\frac{d\log(1/\xi)+\log(1/\beta)}{\alpha^2}+\frac{d\log(1/\xi)+\log(1/\beta)}{\alpha\varepsilon}+\frac{\log(1/\beta\delta)}{\varepsilon}\right).$$

**Algorithm 1:** Boosting for learning $\mathcal{G}_L^d$: $\text{BOOST}_1(b, \xi, \alpha, \beta, \varepsilon, \delta, D)$.

**Input :** Parameters $b \geq 1$, $\beta, \varepsilon, \delta \in (0,1)$, $\alpha \in (0, \frac{4c_1}{3b+12})$, $\xi \in (0, \frac{c_1}{3b+4})$ and dataset $D$ of size $n$.

**Output:** Distribution $\widehat{f} \in \mathcal{G}_L^d$.

1 Split $D$ into $D_1, D_2$ where $|D_1| = n_1$, $|D_2| = n - n_1$

2 Set $\mathcal{C}_\xi$ as a locally small $\xi$-cover for $\mathcal{G}_L^d$

3 $f' = \text{GAP-MAX}\left((b+1)\xi + \frac{\alpha}{4}, \frac{c_1}{3b+4} - \frac{3\alpha}{4}, \frac{\beta}{2}, \frac{\varepsilon}{2}, \delta, k, \mathcal{C}_\xi, D_1\right)$ // $f' = \mathcal{N}(\mu', I)$

4 Set $\tilde{\mathcal{C}}_\xi$ as $\xi$-cover for $\mathcal{B}\left(\left(3(b+1)\xi + \frac{c_1}{3b+4}\right), f', \mathcal{G}_L^d\right)$

5 **Return** $\widehat{f} = \text{PHS}((b+1)\xi + \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\beta}{2}, \frac{\varepsilon}{2}, \tilde{\mathcal{C}}_\xi, D_2)$ // $\widehat{f} = \mathcal{N}(\widehat{\mu}, I)$

*Proof of Lemma 3.6.2.*

**Privacy.** We first show Algorithm 1 satisfies $(\varepsilon, \delta)$-differential privacy. Line 3 of the algorithm is $(\varepsilon/2, \delta)$-differentially private by the guarantee of Theorem 3.4.2. Line 4 maintains $(\varepsilon/2, \delta)$-privacy by post-processing (Lemma 2.2.3). Finally, line 5 is $(\varepsilon/2, 0)$-differentially private by Theorem 3.4.1. By basic composition (Lemma 2.2.2), the entire algorithm is $(\varepsilon, \delta)$-differentially private.

**Accuracy.** We now argue about the accuracy of the algorithm. Recall that for any $\gamma < c_1$, Corollary 3.5.3 guarantees the existence of a $\xi$-cover that is $(k, \gamma)$-locally small where $k = (2\gamma/\xi)^{O(d)}$. Thus, for any $\xi < \frac{c_1}{3b+4}$, we can set $\mathcal{C}_\xi$ to be a $\xi$-cover that is $(k, \gamma)$-locally small where $\gamma = 3(b+1)\xi + \frac{c_1}{3b+4}$ in line 2. Note that $d_{\text{TV}}(g, \mathcal{G}_L^d) \leq b\xi + \alpha/4$ implies $d_{\text{TV}}(g, \mathcal{C}_\xi) \leq (b+1)\xi + \alpha/4$.

By an upper bound on the VC-dimension of $\mathcal{C}_\xi$ (Lemma 3.3.1) and the guarantee of the GAP-MAX algorithm (Theorem 3.4.2), we have with probability at least $1 - \beta/2$ that the output of the GAP-MAX algorithm on line 3 is a distribution $f'$ that is $(3(b+1)\xi + \frac{c_1}{3b+4})$-close to $f^*$ so long as $|D_1| = \Omega\left(\frac{d\log(1/\xi) + \log(1/\beta\delta)}{\varepsilon}\right)$ and $\alpha < \frac{4c_2}{9b+12}$.

We condition on this occurring.

On line 4 we build a $\xi$-cover $\tilde{\mathcal{C}}_\xi$ for $\mathcal{B}\left(\left(3(b+1)\xi + \frac{c_1}{3b+4}\right), f', \mathcal{G}_L^d\right)$ that satisfies $d_{\mathrm{TV}}(f^*, \tilde{\mathcal{C}}_\xi) \leq \xi$, which implies $d_{\mathrm{TV}}(g, \tilde{\mathcal{C}}_\xi) \leq (b+1)\xi + \alpha/4$. By Lemma 3.5.2, we can indeed construct such a $\tilde{\mathcal{C}}_\xi$ satisfying $|\tilde{\mathcal{C}}_\xi| \leq \left(3(b+1) + \frac{c_1}{\xi(3b+4)}\right)^{O(d)}$ for any $\xi < \frac{c_1}{3b+4}$. By the stated accuracy and size of $\tilde{\mathcal{C}}_\xi$, with probability no less than $1 - \beta/2$ Theorem 3.4.1 guarantees that line 5 outputs $\widehat{f}$ satisfying

$$d_{\mathrm{TV}}(g, \widehat{f}) \leq 3((b+1)\xi + \alpha/4) + \alpha/4 = 3(b+1)\xi + \alpha,$$

so long as $|D_2| = \Omega\left(\frac{d\log(1/\xi)+\log(1/\beta)}{\alpha^2} + \frac{d\log(1/\xi)+\log(1/\beta)}{\alpha\varepsilon}\right)$.

A union bound and setting $n = \Omega\left(\frac{d\log(1/\xi)+\log(1/\beta)}{\alpha^2} + \frac{d\log(1/\xi)+\log(1/\beta)}{\alpha\varepsilon} + \frac{\log(1/\beta\delta)}{\varepsilon}\right)$ completes the proof. □

The following result can be derived by using an algorithm very similar to Algorithm 1. The proof is nearly identical to the proof of Lemma 3.6.2.

**Lemma 3.6.3.** *For any $\varepsilon, \delta \in (0,1)$ and $\xi \in (0, c_3)$ where $c_3$ is a universal constant, there exists an $(\varepsilon,\delta)$-DP $(\xi, 6)$-robust PAC learner for $\mathcal{G}_L^d$ that uses*

$$\widetilde{n}_6(\alpha,\beta,\varepsilon,\delta) = O\left(\frac{d\log(1/\xi)+\log(1/\beta)}{\alpha^2} + \frac{d\log(1/\xi)+\log(1/\beta)}{\alpha\varepsilon} + \frac{\log(1/\beta\delta)}{\varepsilon}\right)$$

*samples.*

We can now convert our $(\varepsilon,\delta)$-DP robust PAC learner (Lemma 3.6.3) into an $(\varepsilon,\delta)$-DP semi-agnostic PAC learner using Lemma 3.4.4. Note that while the bound in Lemma 3.6.3 has a dependence on $\log(1/\xi)$, we can replace this with $\log(1/\alpha)$ in the semi-agnostic bounds. This is because when we convert from robust to semi-agnostic

learners (Lemma 3.4.4) we run the robust learner $T$ times with different robustness parameter $\xi_t$, where $\forall t \in [T]$, $\xi_t = \Omega(\alpha)$. Thus, $\log(1/\xi_t)$ terms simplify to $\log(1/\alpha)$.

**Theorem 3.6.4.** *For any $\varepsilon, \delta \in (0,1)$ and* OPT *smaller than a universal constant, there exists an $(\varepsilon,\delta)$-DP 36-agnostic PAC learner for $\mathcal{G}_L^d$ that uses*

$$n_{36}(\alpha,\beta,\varepsilon,\delta) = \widetilde{O}\left(\frac{d+\log(1/\beta)}{\alpha^2} + \frac{d+\log(1/\beta)}{\alpha\varepsilon} + \frac{\log(1/\beta\delta)}{\varepsilon}\right)$$

*samples.*

### 3.6.2 Learning Gaussians

We can now show that Algorithm 2 achieves the following upper bound on the sample complexity of robustly learning $\mathcal{G}^d$.

**Lemma 3.6.5.** *For any $\varepsilon, \delta \in (0,1)$, and $\xi \in (0, c_4)$ where $c_4$ is a universal constant, Algorithm 2 is an $(\varepsilon,\delta)$-DP $(\xi, 24)$-robust PAC learner for $\mathcal{G}^d$ that uses*

$$\widetilde{n}_{24}(\alpha,\beta,\varepsilon,\delta) = O\left(\frac{d^2\log(1/\xi)+\log(1/\beta)}{\alpha^2} + \frac{d^2\log(1/\xi)+\log(1/\beta)}{\alpha\varepsilon} + \frac{\log(1/\beta\delta)}{\varepsilon}\right)$$

*samples.*

**Algorithm 2:** Boosting for learning $\mathcal{G}^d$: $\text{BOOST}_2(\xi, \alpha, \beta, \varepsilon, \delta, D)$.

**Input :** Parameters $\alpha, \beta, \varepsilon, \delta \in (0,1)$, $\xi \in (0, c_4)$ and dataset $D$ of size $2n$.

**Output:** Distribution $\widehat{f} \in \mathcal{G}_S^d$.

1 Split $D$ into $D_1$, $D_2$, $D_3$ where $|D_1| = 2n_1$, $|D_2| = n_2$, $|D_3| = n_3$

2 Set $\mathcal{C}_\xi$ as a locally small cover for $\mathcal{G}_S^d$

3 Set $D_1' = \{Y^1, \dots, Y^{n_1}\}$ where $Y^i = \frac{1}{\sqrt{2}}(X^{2i} - X^{2i-1})$, $X^i \in D_1$

4 $f' = \text{GAP-MAX}\left(3\xi, \frac{c_2}{10}, \frac{\beta}{4}, \frac{\varepsilon}{4}, \frac{\delta}{2}, k, \mathcal{C}_\xi, D_1'\right)$ // $f' = \mathcal{N}(0, \Sigma')$

5 Set $\tilde{\mathcal{C}}_\xi$ a a $\xi$-cover for $\mathcal{B}\left(\left(9\xi + \frac{c_3}{10}\right), f', \mathcal{G}_S^d\right)$

6 $\widehat{f}_1 = \text{PHS}(3\xi, \frac{\alpha}{4}, \frac{\beta}{4}, \frac{\varepsilon}{4}, \tilde{\mathcal{C}}_\xi, D_2)$ // $\widehat{f}_1 = \mathcal{N}(0, \widehat{\Sigma})$

7 Set $D_3' = \{W^1, \dots, W^{n_3}\}$ where $W^i = \widehat{\Sigma}^{-1/2} X^i$, $X^i \in D_3$

8 $\widehat{f}_2 = \text{BOOST}_1\left(7, \xi, \alpha, \frac{\beta}{2}, \frac{\varepsilon}{2}, \frac{\delta}{2}, D_3'\right)$ // $\widehat{f}_2 = \mathcal{N}(\widehat{\mu}, I)$

9 **Return:** $\widehat{f} = \mathcal{N}\left(\widehat{\Sigma}^{1/2}\widehat{\mu}, \widehat{\Sigma}\right)$

*Proof of Lemma 3.6.5.*

**Privacy.** We first show Algorithm 2 satisfies $(\varepsilon, \delta)$-differential privacy. Line 4 of the algorithm is $(\varepsilon/4, \delta/2)$-differentially private by the guarantee of Theorem 3.4.2. Line 5 maintains $(\varepsilon/4, \delta/2)$-privacy by post-processing (Lemma 2.2.3). Line 6 is $(\varepsilon/4, 0)$-differentially private by Theorem 3.4.1. Line 7 maintains privacy by post processing (Lemma 2.2.3). Finally, line 8 is $(\varepsilon/2, \delta/2)$-differentially private by the privacy of Algorithm 1 proved in Lemma 3.6.2. By basic composition (Lemma 2.2.2) the entire algorithm is $(\varepsilon, \delta)$-differentially private.

**Accuracy.** We now argue about the accuracy of the algorithm. Recall that for any $\gamma < c_2$, Corollary 3.5.6 guarantees the existence of a $\xi$-cover that is $(k, \gamma)$-locally small where $k = (2\gamma/\xi)^{O(d^2)}$. Thus, on line 2 we can set $\mathcal{C}_\xi$ to be a $\xi$-cover that is $(k, \gamma)$-locally small where $\gamma = 9\xi + \frac{c_2}{10}$, so long as $\xi < \frac{c_2}{10}$.

Let $f^* = \mathcal{N}(\mu^*, \Sigma^*)$ be a distribution that satisfies $d_{\mathrm{TV}}(f^*, g) \leq \xi$. By Proposition A.0.4, for every $Y \in D_1'$ on line 3 we have that $Y \sim q_1$ where $d_{\mathrm{TV}}(q_1, \mathcal{N}(0, \Sigma^*)) \leq 2\xi$. This implies that $d_{\mathrm{TV}}(q_1, \mathcal{C}_\xi) \leq 3\xi$. By an upper bound on the VC dimension of $\mathcal{C}_\xi$ (Lemma 3.3.1) and the guarantee of the GAP-MAX algorithm (Theorem 3.4.2), we have with probability at least $1 - \beta/4$ that the output of the GAP-MAX algorithm on line 4 is a distribution $f' = \mathcal{N}(0, \Sigma')$ that is $\left(9\xi + \frac{c_2}{10}\right)$-close to $\mathcal{N}(0, \Sigma^*)$, so long as $|D_1'| = \Omega\left(\frac{d^2 \log(1/\xi) + \log(1/\beta\delta)}{\varepsilon}\right)$. We condition on this occurring.

On line 5, we build a $\xi$-cover $\tilde{\mathcal{C}}_\xi$ for $\mathcal{B}\left(9\xi + \frac{c_2}{10}, f', \mathcal{G}_S^d\right)$ that satisfies $d_{\mathrm{TV}}(\mathcal{N}(0, \Sigma^*), \tilde{\mathcal{C}}_\xi) \leq \xi$, which implies that $d_{\mathrm{TV}}(q_1, \tilde{\mathcal{C}}_\xi) \leq 3\xi$. By Corollary 3.5.5, we can indeed construct such a $\tilde{\mathcal{C}}_\xi$ satisfying $|\tilde{\mathcal{C}}_\xi| \leq (9 + \frac{c_2}{10\xi})^{O(d^2)}$ for any $\xi < \frac{c_2}{10}$. By the stated accuracy and size of $\tilde{\mathcal{C}}_\xi$, with probability no less than $1 - \beta/4$, Theorem 3.4.1 guarantees that line 6 outputs $\widehat{f}_1 = \mathcal{N}(0, \widehat{\Sigma})$ such that $d_{\mathrm{TV}}(\widehat{f}_1, \mathcal{N}(0, \Sigma^*)) \leq 6\xi + \alpha/4$ so long as $|D_2| = \Omega\left(\frac{d^2 \log(1/\xi) + \log(1/\beta)}{\alpha^2} + \frac{d^2 \log(1/\xi) + \log(1/\beta)}{\alpha\varepsilon}\right)$. We further condition on this occurring.

For samples $W \in D_3'$ on line 7, let $q_2$ be the distribution satisfying $W \sim q_2$.[4] By Eq. (2.1.1) and Corollary A.0.2,

$$d_{\mathrm{TV}}(q_2, \mathcal{N}(\widehat{\Sigma}^{-1/2}\mu^*, \widehat{\Sigma}^{-1/2}\Sigma^*\widehat{\Sigma}^{-1/2})) \leq \xi.$$

Furthermore, using the triangle inequality, Corollary A.0.2, Eq. (2.1.1), and the above

[4] If $\widehat{\Sigma}$ is not invertible, the range of $\widehat{\Sigma}$ is an $r$-dimensional linear subspace of $\mathbb{R}^d$, for some $r < d$. Let $\Pi$ be a $d \times r$ matrix whose columns form a orthonormal basis for the range of $\widehat{\Sigma}$. It follows that $\widetilde{\Sigma} = \Pi^\top \widehat{\Sigma} \Pi$ is a $r \times r$ positive definite covariance matrix. Moreover, by construction, $\widetilde{\Sigma}$ is identical to $\widehat{\Sigma}$ after projection on to the subspace defined by the range of $\widehat{\Sigma}$. We can thus project our data onto the range of $\widehat{\Sigma}$ and continue the algorithm using $\widetilde{\Sigma}$.

inequality, we have

$$\begin{aligned} TV(q_2, \mathcal{N}(\widehat{\Sigma}^{-1/2}\mu^*, I)) &\leq d_{\mathrm{TV}}(q_2, \mathcal{N}(\widehat{\Sigma}^{-1/2}\mu^*, \widehat{\Sigma}^{-1/2}\Sigma^*\widehat{\Sigma}^{-1/2})) \\ &\quad + d_{\mathrm{TV}}(\mathcal{N}(\widehat{\Sigma}^{-1/2}\mu^*, \widehat{\Sigma}^{-1/2}\Sigma^*\widehat{\Sigma}^{-1/2}), \mathcal{N}(\widehat{\Sigma}^{-1/2}\mu^*, I)) \\ &\leq \xi + d_{\mathrm{TV}}(\mathcal{N}(\mu^*, \Sigma^*), \mathcal{N}(\mu^*, \widehat{\Sigma})) \\ &= \xi + d_{\mathrm{TV}}(\mathcal{N}(0, \Sigma^*), \widehat{f}_1) \\ &\leq 7\xi + \alpha/4, \end{aligned}$$

which proves that $q_2$ is $(7\xi + \alpha/4)$-close to $\mathcal{G}_L^d$.

Thus, by the guarantee of Lemma 3.6.2, when we run $\mathrm{BOOST}_1(7, \xi, \alpha, \beta/2, \varepsilon/2, \delta/2, D_3')$ on line 8, we have with probability no less than $1 - \beta/2$ that the output $\widehat{f}_2 = \mathcal{N}(\widehat{\mu}, I)$ satisfies $d_{\mathrm{TV}}(\widehat{f}_2, q_2) \leq 24\xi + \alpha$, so long as $\xi < \frac{c_1}{25}$, $\alpha < \frac{4c_1}{33}$,[5] and $|D_3'| = \Omega\left(\frac{d\log(1/\xi)+\log(1/\beta)}{\alpha^2} + \frac{d\log(1/\xi)+\log(1/\beta)}{\alpha\varepsilon} + \frac{\log(1/\beta\delta)}{\varepsilon}\right)$.

Note that the above guarantees hold for any $\xi < \min\{\frac{c_2}{10}, \frac{c_1}{25}\} = c_4$. Thus, from Corollary A.0.2 and Eq. (2.1.1), it follows that the output of line 8 satisfies

$$d_{\mathrm{TV}}(\widehat{f}, g) = d_{\mathrm{TV}}(\widehat{f}_2, q_2) \leq 24\xi + \alpha.$$

Setting $n = \Omega\left(\frac{d^2\log(1/\xi)+\log(1/\beta)}{\alpha^2} + \frac{d^2\log(1/\xi)+\log(1/\beta)}{\alpha\varepsilon} + \frac{\log(1/\beta\delta)}{\varepsilon}\right)$ together with a union bound completes the proof. □

Finally, we can combine the above result with Lemma 3.4.4 to get a semi-agnostic PAC learner that can handle modest levels of model misspecification.

[5] For any target $\alpha \geq \frac{4c_1}{33}$, we can run the algorithm with $\alpha' = \frac{4c_1}{34}$ and the guarantee will trivially hold with respect to $\alpha$.

**Theorem 3.6.6.** *For any $\varepsilon, \delta \in (0,1)$ and* OPT *smaller than a universal constant, there exists an $(\varepsilon, \delta)$-DP 144-agnostic PAC learner for $\mathcal{G}^d$ that uses*

$$n_{144}(\alpha, \beta, \varepsilon, \delta) = \widetilde{O}\left(\frac{d^2 + \log(1/\beta)}{\alpha^2} + \frac{d^2 + \log(1/\beta)}{\alpha\varepsilon} + \frac{\log(1/\beta\delta)}{\varepsilon}\right)$$

*samples.*

### 3.6.3 Bounds for the Realizable Setting

The following sample complexity bounds hold for $(\varepsilon, \delta)$-DP (realizable) PAC learning. The proofs are very similar to the proofs for $(\varepsilon, \delta)$-DP robust PAC learning, where the slight difference is that we can build the covers directly with accuracy $\alpha$ (instead of $\xi$) since we assume realizability. The first bound is nearly tight (up to the $\log(1/\alpha)$ factors) and the second one is conjectured to be nearly tight.

**Lemma 3.6.7.** *For any $\varepsilon, \delta \in (0,1)$ there exists an $(\varepsilon, \delta)$-DP PAC learner for $\mathcal{G}_L^d$ that uses*

$$n(\alpha, \beta, \varepsilon, \delta) = O\left(\frac{d\log(1/\alpha) + \log(1/\beta)}{\alpha^2} + \frac{d\log(1/\alpha) + \log(1/\beta)}{\alpha\varepsilon} + \frac{\log(1/\beta\delta)}{\varepsilon}\right)$$

*samples.*

**Lemma 3.6.8.** *For any $\varepsilon, \delta \in (0,1)$ there exists an $(\varepsilon, \delta)$-DP PAC learner for $\mathcal{G}^d$ that uses*

$$n(\alpha, \beta, \varepsilon, \delta) = O\left(\frac{d^2\log(1/\alpha) + \log(1/\beta)}{\alpha^2} + \frac{d^2\log(1/\alpha) + \log(1/\beta)}{\alpha\varepsilon} + \frac{\log(1/\beta\delta)}{\varepsilon}\right)$$

*samples.*

# Chapter 4

# Privately Learning Mixtures of Axis-Aligned Gaussians

In this chapter, we will prove sample complexity upper bounds for privately learning the class of mixtures of high-dimensional Gaussians where $(i)$ all the component have the same known covariance matrix and $(ii)$ all the components are axis-aligned Gaussians. The content of this chapter is based on joint work with Hassan Ashtiani and Christopher Liaw [AAL21].

## 4.1 Main Results

We prove sample complexity upper bounds for learning mixtures of unbounded $d$-dimensional axis-aligned Gaussians and mixtures of $d$-dimensional Gaussians with the same known covariance matrix. We informally state these two results below.

**Theorem 4.1.1** (Informal version of Theorem 4.7.3)**.** *The sample complexity of learning a mixture of $k$ $d$-dimensional axis-aligned Gaussians to $\alpha$-accuracy in total variation distance under $(\varepsilon, \delta)$-differential privacy is*

$$\widetilde{O}\left(\frac{k^2 d \log^{3/2}(1/\delta)}{\alpha^2 \varepsilon}\right).$$

Even for the univariate case, this is the *first* sample complexity upper bound for learning mixture of Gaussians under differential privacy where the variances are unknown and the parameters of the Gaussians may be unbounded.

If the covariance matrix of each component of the mixture is the same and known or, without loss of generality, equal to the identity matrix, then we can improve the dependence on the parameters and obtain a sample complexity upper bound that is similar to the non-private setting.

**Theorem 4.1.2** (Informal version of Theorem 4.7.1)**.** *The sample complexity of learning a mixture of $k$ $d$-dimensional Gaussians with identity covariance matrix to $\alpha$-accuracy in total variation distance under $(\varepsilon, \delta)$-differential privacy is*

$$\widetilde{O}\left(\frac{kd}{\alpha^2} + \frac{kd \log(1/\delta)}{\alpha \varepsilon}\right).$$

## 4.2 Techniques

To prove our results, we devise a novel technique which reduces the problem of privately learning mixture distributions to the problem of private list-decodable learning of distributions. The framework of list-decodable learning was introduced by Balcan, Blum, and Vempala [BBV08] and Balcan, Röglin, and Teng [BRT09] in

the context of clustering but has since been studied extensively in the literature in a number of different contexts [CSV17, DKS18, KKK19, CMY20, DKK20, RY20a, RY20b, BK21]. The problem of list-decodable learning of distributions is as follows. There is a distribution $f$ of interest that we are aiming to learn. However, we do not receive samples from $f$; rather we receive samples from a *corrupted* distribution $g = (1-\gamma)f + \gamma h$ where $h$ is some arbitrary distribution. In our application, $\gamma$ will be quite close to 1. In other words, *most* of the samples are corrupted. The goal in list-decodable learning is to output a *short* list of distributions $f_1, \ldots, f_m$ with the requirement that $f$ is close to at least one of the $f_i$'s. The formal definition of list-decodable learning can be found in Definition 4.3.6. Informally, the reduction can be summarized by the following theorem which is formalized in Section 4.5.

**Theorem 4.2.1** (Informal)**.** *If a class of distributions $\mathcal{F}$ is privately list-decodable then mixtures of distributions from $\mathcal{F}$ are privately learnable.*

Roughly speaking, the reduction from learning mixtures of distribution to list-decodable learning works as follows. Suppose that there is an unknown distribution $f$ which is a mixture of $k$ distributions $f_1, \ldots, f_k$. A list-decodable learner would then receive samples from $f$ as input and output a short list of distributions $\widehat{\mathcal{F}}$ so that for every $f_i$ there is some element in $\widehat{\mathcal{F}}$ that is close to $f_i$. In particular, some mixture of distributions from $\widehat{\mathcal{F}}$ must be close to the true distribution $f$. Since $\widehat{\mathcal{F}}$ is a small finite set, the set of possible mixtures must also be relatively small. This last observation allows us to make use of private hypothesis selection which selects a good hypothesis from a small set of candidate hypotheses [BKSW19, AAK21]. In Section 4.5, we formally describe the aforementioned reduction. We note that a similar connection between list-decodable learning and learning mixture distributions was also used by

Diakonikolas et al. [DKS18]. However, our reduction is focused on the private setting.

The reduction shows that to privately learn mixtures, it is sufficient to design differentially private list-decodable learning algorithms that work for (corrupted versions of) the individual mixture components. To devise list-decodable learners for (corrupted) univariate Gaussian, we utilize "stability-based" histograms [KKMN09, BNS16] that satisfy approximate differential privacy.

To design a list-decodable learner for corrupted univariate Gaussians, we follow a three-step approach that is inspired by the seminal work of Karwa and Vadhan [KV18]. First, we use a histogram to output a list of variances one of which approximates the true variance of the Gaussian. As a second step, we would like to output a list of means which approximate the true mean of the Gaussian. This can be done using histograms provided that we roughly know the variance of the Gaussian. Since we have candidate variances from the first step, we can use a sequence of histograms where the width of the bins of each of the histograms is determined by the candidate variances from the first step. As a last step, using the candidate variances and means from the first two steps, we are able to construct a small set of distributions one of which approximates the true Gaussian to within accuracy $\alpha$. In the axis-aligned Gaussians setting, we use our solution for the univariate case as a subroutine on each dimension separately. Now that we have a list-decodable learner for axis-aligned Gaussians, we use our reduction to obtain a private learning algorithm for learning mixtures of axis-aligned Gaussians.

## 4.3 Preliminaries

We state some definitions and simple results that will be useful for this chapter.

We define $\mathcal{G}$ to be the class of univariate Gaussians and $\mathcal{G}_A^d = \{\mathcal{N}(\mu, \Sigma) : \Sigma_{ij} = 0 \ \forall i \neq j \text{ and } \Sigma_{ii} > 0 \ \forall i\}$ to be the class of axis-aligned Gaussians.

**Definition 4.3.1** ($\alpha$-net)**.** *Let $(X, d)$ be a metric space. A set $N \subseteq X$ is an $\alpha$-net for $X$ under the metric $d$ if for all $x \in X$, there exists $y \in N$ such that $d(x, y) \leq \alpha$.*

The following result is very standard. We add a proof for completeness.

**Proposition 4.3.2.** *For any $\alpha \in (0, 1]$ and $k \geq 2$, there exists an $\alpha$-net of $\Delta_k$ under the $\ell_\infty$-norm of size at most $(3/\alpha)^k$.*

*Proof.* We will give an algorithmic proof of this fact. Let $r = \lceil 1/\alpha \rceil$ and fix $x \in \Delta_k$. Let $\ell = \sum_{i=1}^k rx_i - \lfloor rx_i \rfloor$. Note that $\sum_{i=1}^k rx_i = r$ and $rx_i - \lfloor rx_i \rfloor \in [0, 1)$ so $\ell$ is an integer in the interval $[0, r-1]$. Now define $\hat{x}$

$$\hat{x}_i = \begin{cases} \frac{\lfloor rx_i \rfloor + 1}{r} & i \leq \ell \\ \frac{\lfloor rx_i \rfloor}{r} & i > \ell \end{cases}.$$

Clearly, $\|x - \hat{x}\|_\infty \leq 1/r \leq \alpha$. It remains to check that $\hat{x} \in \Delta_k$. Indeed,

$$\sum_{i=1}^k \hat{x}_i = \sum_{i=1}^k \frac{\lfloor rx_i \rfloor}{r} + \frac{\ell}{r} = \sum_{i=1}^k \frac{\lfloor rx_i \rfloor}{r} + \sum_{i=1}^k \frac{rx_i - \lfloor rx_i \rfloor}{r} = 1,$$

where in the second equality, we used the definition of $\ell$. Note that for each $i$, $\hat{x}_i \in \{0, 1/r, 2/r, \ldots, 1\}$ so this shows that

$$\widehat{\Delta}_k = \{(t_1/r, \ldots, t_k/r) : t \in \mathbb{Z}_{\geq 0}^k, \|t\|_1 = r\},$$

is an $\alpha$-net for $\Delta_k$ of size $(r+1)^k$. To obtain the bound as asserted in the claim, note that $r + 1 = \lceil 1/\alpha \rceil + 1 \leq 1/\alpha + 2 \leq 3/\alpha$ for $\alpha \in (0, 1]$. □

**Definition 4.3.3** ($k$-mix($\mathcal{F}$))**.** *Let $\mathcal{F}$ be a class of probability distributions. Then the class of $k$-mixtures of $\mathcal{F}$, written $k$-mix($\mathcal{F}$), is defined as*

$$k\text{-mix}(\mathcal{F}) \coloneqq \{ \textstyle\sum_{i=1}^{k} w_i f_i \,:\, (w_1, \dots, w_k) \in \Delta_k,\ f_1, \dots, f_k \in \mathcal{F} \}.$$

We will refer to realizable PAC learners and $(\varepsilon, \delta)$-DP realizable PAC learners as PAC learners and $(\varepsilon, \delta)$-DP PAC learners. We omit any reference to realizability since we will focus on realizability for the entirety of this chapter.

We will work with a standard additive corruption model often studied in the list-decodable setting that is inspired by the work of Huber [Hub64]. In this model, a sample is drawn from a distribution of interest with some probability, and with the remaining probability is drawn from an arbitrary distribution. Our list-decodable learners take samples from these "corrupted" distributions as input.

**Definition 4.3.4** ($\gamma$-corrupted distributions)**.** *Fix some distribution $f$ and let $\gamma \in (0, 1)$. We define a* $\gamma$-corrupted distribution *of $f$ as as any distribution $g$ such that*

$$g = (1 - \gamma) f + \gamma h,$$

*for an arbitrary distribution $h$. We define $\mathcal{H}_\gamma(f)$ to be the set of all $\gamma$-corrupted distributions of $f$ .*

**Remark 4.3.5.** *Observe that $\mathcal{H}_\gamma(f)$ is monotone increasing in $\gamma$, i.e. $\mathcal{H}_\gamma(f) \subset \mathcal{H}_{\gamma'}(f)$ for all $\gamma' \in (\gamma, 1)$. To see this, note that if $g = (1-\gamma)f + \gamma h$ then we can also rewrite*

$$g = (1-\gamma')f + (\gamma' - \gamma)f + \gamma h = (1-\gamma')f + \gamma' \left( \frac{(\gamma' - \gamma)}{\gamma'} f + \frac{\gamma}{\gamma'} h \right) = (1-\gamma')f + \gamma' h',$$

*where* $h' = \frac{\gamma'-\gamma}{\gamma} f + \frac{\gamma}{\gamma'} h$. *Hence,* $g \in C_{\gamma'}(f)$.

We note that in this work, we will most often deal with $\gamma$-corrupted distribution where $\gamma$ is quite close to 1; in other words, the vast majority of the samples are corrupted.

Now we define list-decodable learning. In this setting, the goal is to learn a distribution $f$ given samples from a $\gamma$-corrupted distribution $g$ of $f$. Since $\gamma$ is close to 1, instead of finding a single distribution $\widehat{f}$ that approximates $f$, our goal is to output a list of distributions, one of which is accurate. This turns out to be a useful primitive to design algorithms for learning mixture distributions.

**Definition 4.3.6** (list-decodable learner)**.** *We say algorithm* $\mathcal{A}_{\text{LIST}}$ *is an* $L$-list-decodable learner *for a class of distributions* $\mathcal{F}$ *using* $n_{\text{LIST}}(\alpha, \beta, \gamma)$ *samples if for every* $\alpha, \beta, \gamma \in (0, 1)$, $n \geq n_{\text{List}}(\alpha, \beta, \gamma)$, $f \in \mathcal{F}$, *and* $g \in \mathcal{H}_\gamma(f)$, *the following holds: given parameters* $\alpha, \beta, \gamma$ *and a sequence of* $n$ *i.i.d. samples from* $g$ *as inputs,* $\mathcal{A}_{\text{LIST}}$ *outputs a set of distributions* $\widetilde{\mathcal{F}}$ *with* $|\widetilde{\mathcal{F}}| \leq L$ *such that with probability no less than* $1 - \beta$ *we have* $d_{\text{TV}}(f, \widetilde{\mathcal{F}}) \leq \alpha$.

We now define the private version of list-decodable learners.

**Definition 4.3.7** ($(\varepsilon, \delta)$-DP list-decodable learner)**.** *We say algorithm* $\mathcal{A}_{\text{LIST}}$ *is an* $(\varepsilon, \delta)$*-DP* $L$*-list-decodable learner for a class of distributions* $\mathcal{F}$ *that uses* $n_{\text{List}}(\alpha, \beta, \gamma, \varepsilon, \delta)$ *samples if:*

1. *Algorithm* $\mathcal{A}_{\text{LIST}}$ *is a* $L$*-list-decodable learner for* $\mathcal{F}$ *that uses* $n_{\text{LIST}}(\alpha, \beta, \gamma, \varepsilon, \delta)$ *samples.*

2. *Algorithm* $\mathcal{A}_{\text{LIST}}$ *satisfies* $(\varepsilon, \delta)$*-DP.*

## 4.4 Locally Small Covers for Mixtures

A natural approach one might suggest to prove sample complexity upper bounds for mixture classes is to use the local cover based techniques in Chapter 3. Unfortunately, we cannot hope to do so because it is not possible to construct locally small covers for mixture classes in general. While univariate Gaussians admit locally small covers [BKSW19], the following simple proposition shows that mixtures of univariate Gaussians do not.

**Proposition 4.4.1.** *For every $\gamma \in (0,1)$, any $(\gamma/2)$-cover for 2-mix$(\mathcal{G})$ is not $\gamma$-locally small.*

*Proof.* Fix some $\gamma \in (0,1)$. Let $f = \mathcal{N}(0,1)$ and define $g(\mu) := (1-\gamma)\mathcal{N}(0,1) + \gamma\mathcal{N}(\mu,1)$ (note that $f = g(0)$). We will show that the following two statements hold for every $\mu, \mu' \in \mathbb{R}$:

1. $d_{\mathrm{TV}}(g(\mu), g(\mu')) \leq \gamma$, and
2. If $|\mu - \mu'| \geq C$ for a sufficiently large constant $C$, $d_{\mathrm{TV}}(g(\mu), g(\mu')) \geq \gamma/2$.

Consider the set of distributions $\mathcal{F} = \{g(\mu) : \mu \in \{C, 2C, \dots\}\}$ for some large positive constant $C$. For every $g, g' \in \mathcal{F}$, it follows from claim 1 that $g, g' \in \mathcal{B}(\gamma, f, 2\text{-mix}(\mathcal{G}))$ and from claim 2 that $d_{\mathrm{TV}}(g, g') \geq \gamma/2$ for sufficiently large $C$. Thus, the $(\gamma/2)$-packing number of $\mathcal{B}(\gamma, f, 2\text{-mix}(\mathcal{G}))$ is unbounded, and by Proposition 2.3.9, the $(\gamma/2)$-covering number of $\mathcal{B}(\gamma, f, 2\text{-mix}(\mathcal{G}))$ is also unbounded. This implies that *every* $(\gamma/2)$-cover for 2-mix$(\mathcal{G})$ is not $\gamma$-locally small by definition.

It remains to prove the two claims above. From the definition of the TV distance

we have

$$\begin{aligned} d_{\mathrm{TV}}(g(\mu), g(\mu')) &= \frac{1}{2} \left\| (1-\gamma)\mathcal{N}(0,1) + \gamma\mathcal{N}(\mu,1) - (1-\gamma)\mathcal{N}(0,1) - \gamma\mathcal{N}(\mu',1) \right\|_1 \\ &= \frac{\gamma}{2} \left\| \mathcal{N}(\mu,1) - \mathcal{N}(\mu',1) \right\|_1 \\ &= \gamma d_{\mathrm{TV}}(\mathcal{N}(\mu,1), \mathcal{N}(\mu',1)). \end{aligned} \tag{4.4.1}$$

Using the trivial upper bound on the TV distance between any two distributions, we have from Eq. (4.4.1) that $d_{\mathrm{TV}}(g(\mu), g(\mu')) \leq \gamma$, which proves the first claim. If $|\mu - \mu'| \geq C$ for sufficiently large $C$, it follows from Gaussian tail bounds that $d_{\mathrm{TV}}(\mathcal{N}(\mu,1), \mathcal{N}(\mu',1)) = 1 - \exp(-\Omega(C^2))$. Thus, by choosing $C$ to be sufficiently large, it follows from Eq. (4.4.1) that $d_{\mathrm{TV}}(g(\mu), g(\mu')) \geq \gamma/2$. □

## 4.5 List-decodability and Learning Mixtures

In this section, we describe our general technique which reduces the problem of private learning of mixture distributions to private list-decodable learning of distributions. We show that if we have a differentially private list-decodable learner for a class of distributions then this can be transformed, in a black-box way, to a differentially private PAC learner for the class of *mixtures* of such distributions. In the next section, we describe private list-decodable learners for the class of Gaussians and thereby obtain private algorithms for learning mixtures of Gaussians.

First, let us begin with some intuition in the *non*-private setting. Suppose that we have a distribution $g$ which can be written as $g = \sum_{i=1}^{k} \frac{1}{k} f_i$. Then we can view $g$ as a $\frac{k-1}{k}$-corrupted distribution of $f_i$ for each $i \in [k]$. Any list-decodable algorithm that receives samples from $g$ as input is very likely to output a candidate set $\widehat{\mathcal{F}}$

which contains distributions that are close to $f_i$ for each $i \in [k]$. Hence, if we let $\mathcal{K} = \{\sum_{i\in[k]} \frac{1}{k}\widehat{f}_i : \widehat{f}_i \in \widehat{\mathcal{F}}\}$, then $g$ must be close to some distribution in $\mathcal{K}$. The only remaining task is to find a distribution in $\mathcal{K}$ that is close to $g$; this final task is known as hypothesis selection and has a known solution [DL01]. We note that the above argument can be easily generalized to the setting where $g$ is a non-uniform mixture, i.e. $g = \sum_{i=1}^{k} w_i f_i$ where $(w_1, \ldots, w_k) \in \Delta_k$.

The above establishes a blueprint that we can follow in order to obtain a private learner for mixture distributions. In particular, we aim to come up with a private list-decoding algorithm which receives samples from $g$ to produce a set $\widehat{\mathcal{F}}$. Thereafter, one can construct a candidate set $\mathcal{K}$ as mixtures of distributions from $\widehat{\mathcal{F}}$. Note that this step does not access the samples and therefore maintains privacy. In order to choose a good candidate from $\mathcal{K}$, we make use of private hypothesis selection algorithms first studied by Bun, Kamath, Steinke, and Wu [BKSW19] that we improved upon in Section 3.4.2.

We now formalize the above argument. Algorithm 3 shows how a list-decodable learner can be used as a subroutine for learning mixture distributions. In the algorithm, we also make use of a subroutine for private hypothesis selection from Section 3.4.2. In hypothesis selection, an algorithm is given i.i.d. sample access to some unknown distribution as well as a list of distributions to pick from. The goal of the algorithm is to output a distribution in the list that is close to the unknown distribution. The following Corollary follows immediately from Theorem 3.4.3. Note that we have named the algorithm in the following Corollary "PHS". This is not to be confused with the algorithm from Theorem 3.4.1 that has the same name; we have overloaded this name for convenience.

**Corollary 4.5.1.** *Let $n \in \mathbb{N}$. There exist an $(\varepsilon/2)$-DP algorithm* PHS$(\varepsilon, \alpha, \beta, \mathcal{F}, D)$ *with the following property: for every $\varepsilon, \alpha, \beta \in (0,1)$, and every set of distributions $\mathcal{F} = \{f_1, \ldots, f_M\}$, when* PHS *is given $\varepsilon, \alpha, \beta, \mathcal{F}$, and a dataset $D$ of $n$ i.i.d. samples from an unknown (arbitrary) distribution $g$ as input, it outputs a distribution $f_j \in \mathcal{F}$ such that*

$$d_{\mathrm{TV}}(g, f_j) \leq 3 \cdot d_{\mathrm{TV}}(g, \mathcal{F}) + \alpha/2,$$

*with probability no less than $1 - \beta/2$ so long as*

$$n = \Omega\left(\frac{\log(M/\beta)}{\alpha^2} + \frac{\log(M/\beta)}{\alpha\varepsilon}\right).$$

We now formally relate the two problems via the theorem below.

**Algorithm 3:** `Learn-Mixture`$(\alpha, \beta, \varepsilon, \delta, k, D)$.

**Input :** Parameters $\alpha, \beta, \varepsilon, \delta > 0$, $k \in \mathbb{N}$ and dataset $D$ of $n$ i.i.d. samples generated $g$.

**Output:** mixture $\widehat{g} = \sum_{i=1}^{n} \widehat{w}_i \widehat{f}_i$.

1 Split $D$ into $D_1, D_2$ where $|D_1| = n_1$, $|D_2| = n - n_1$
// $n_1 = n_{\texttt{List}}\left(\frac{\varepsilon}{2}, \delta, \frac{\alpha}{18}, \frac{\beta}{2k}, 1 - \frac{\alpha}{18k}\right)$.

2 $\widehat{\mathcal{F}} = \{\widehat{f}_1, \ldots, \widehat{f}_L\} \leftarrow \mathcal{A}_{\mathrm{List}}(\alpha/18, \beta/2k, 1 - \alpha/18k, \varepsilon/2, \delta, D_1)$ // $\left(\frac{\varepsilon}{2}, \delta\right)$-`DP` $L$-`list-decodable learner.`

3 Set $\widehat{\Delta}_k$ as $(18k/\alpha)$-net of $\Delta_k$ from Proposition 4.3.2

4 Set $\mathcal{K} = \{\sum_{i=1}^{k} \widehat{w}_i \widehat{f}_i : \widehat{w} \in \widehat{\Delta}_k, \widehat{f}_i \in \mathcal{K}\}$

5 $\widehat{g} \leftarrow$ PHS$(\varepsilon/2, \alpha, \beta/2, \mathcal{K}, D_2)$

6 **Return** $\widehat{g}$

**Theorem 4.5.2.** *Let $k \in \mathbb{N}$ and $\varepsilon, \delta \in (0,1)$. Suppose that $\mathcal{F}$ is $(\varepsilon/2, \delta)$-DP $L$-list-decodable using $n_{\mathrm{List}}$ samples. Then Algorithm 3 is an $(\varepsilon, \delta)$-DP PAC learner for*

$k$-mix $(\mathcal{F})$ *that uses*

$$n(\alpha,\beta,\varepsilon,\delta) = n_{\text{List}}\left(\frac{\alpha}{18},\frac{\beta}{2k},1-\frac{\alpha}{18k},\frac{\varepsilon}{2},\delta\right) + O\left(\frac{k\log(Lk/\alpha)+\log(1/\beta)}{\alpha^2}+\frac{k\log(Lk/\alpha)+\log(1/\beta)}{\alpha\varepsilon}\right)$$

*samples.*

*Proof.* We begin by briefly showing that Algorithm 3 satisfies $(\varepsilon,\delta)$-DP before arguing about its accuracy.

**Privacy.** We first prove that Algorithm 3 is $(\varepsilon,\delta)$-DP. Step 2 of the algorithm satisfies $(\varepsilon/2,\delta)$-DP by the fact that $\mathcal{A}_{\text{List}}$ is an $(\varepsilon/2,\delta)$-DP $L$-list-decodable learner. Steps 3 and 4 maintain $(\varepsilon/2,\delta)$-DP by post processing (Lemma 2.2.3). Finally, step 5 satisfies $(\varepsilon/2)$-DP by Corollary 4.5.1. By basic composition (Lemma 2.2.2) the entire algorithm is $(\varepsilon,\delta)$-DP.

**Accuracy.** We now proceed to show that Algorithm 3 PAC learns $k$-mix$(\mathcal{F})$. In step 2 of Algorithm 3, we use the $(\varepsilon/2,\delta)$-DP $L$-list-decodable learner to obtain a set of distributions $\widehat{\mathcal{F}}$ of size at most $L$. Note that for any mixture component $f_j$, $g$ is a $(1-w_j)$-corrupted distribution of $f_j$ since

$$g = w_j f_j + \textstyle\sum_{i\neq j} w_i f_i = w_j f_j + (1-w_j)\sum_{i\neq j}\frac{w_i f_i}{1-w_j} = w_j f_j + (1-w_j)h,$$

where $h = \sum_{i\neq j}\frac{w_i f_i}{1-w_j}$.

Let $N = \{i\in[k] \,:\, w_i \geq \alpha/18k\}$ denote the set of *non-negligible* components. We first show that for any non-negligible component $i\in N$, there exists $\widehat{f}\in\widehat{\mathcal{F}}$ that is

close to $f_i$.

**Claim 4.5.3.** *If* $|D_1| \geq n_{\text{List}}(\alpha/18, \beta/2k, 1-\alpha/18k, \varepsilon/2, \delta)$ *then* $d_{\text{TV}}(f_i, \widehat{\mathcal{F}}) \leq \alpha/18$ *for all* $i \in N$ *with probability at least* $1-\beta/2$.

*Proof.* Fix $i \in N$. Note that $1 - w_i \leq 1 - \alpha/18k$ so $f \in \mathcal{H}_{1-\alpha/18k}(f_i)$. Since step 2 of Algorithm 3 makes use of a list-decodable learner, as long as $|D_1| \geq n_{\text{List}}(\alpha/18, \beta/2k, 1-\alpha/18k, \varepsilon/2, \delta)$ we have $d_{\text{TV}}(f_i, \widehat{\mathcal{F}}) \leq \alpha/18$ with probability at least $1-\beta/2k$. Since this is true for any fixed $i \in N$, a union bound gives that $d_{\text{TV}}(f_i, \widehat{\mathcal{F}}) \leq \alpha/18$ for all $i \in N$ with probability at least $1-\beta/2$. □

Steps 3 and 4 of Algorithm 3 constructs a candidate set $\mathcal{K}$ of mixture distributions using $\widehat{\mathcal{F}}$ and a net of the probability simplex $\Delta_k$. The next claim shows that as long as $d_{\text{TV}}(f_i, \widehat{\mathcal{F}})$ is small for every non-negligible $i \in N$, $d_{\text{TV}}(g, \mathcal{K})$ is small as well.

**Claim 4.5.4.** *If* $d_{\text{TV}}(f_i, \widehat{\mathcal{F}}) \leq \alpha/18$ *for every* $i \in N$, *then* $d_{\text{TV}}(g, \mathcal{K}) \leq \alpha/6$. *In addition,* $|\mathcal{K}| \leq \left(\frac{54Lk}{\alpha}\right)^k$.

*Proof.* Step 3 constructs a set $\widehat{\Delta}_k$ which is an $(18k/\alpha)$-net of the probability simplex $\Delta_k$ in the $\ell_\infty$-norm. By the hypothesis of the claim, for each $i \in N$, there exists $\widehat{f}_i \in \widehat{\mathcal{F}}$ such that $d_{\text{TV}}(f_i, \widehat{f}_i) \leq \alpha/18$. Recall that $g = \sum_{i\in[k]} w_i f_i$. Let $\widehat{w} \in \widehat{\Delta}_k$ such that $\|\widehat{w} - w\|_\infty \leq \alpha/18k$. Now let $\widetilde{g} = \sum_{i\in[k]} \widehat{w}_i \widehat{f}_i$. Note that $\widetilde{g} \in \mathcal{K}$. Moreover, a straightforward calculation shows that $d_{\text{TV}}(g, \widetilde{g}) \leq \alpha/6$ (see Proposition B.1.1 for the detailed calculations). This proves that $d_{\text{TV}}(g, \mathcal{K}) \leq \alpha/6$.

Lastly, to bound $|\mathcal{K}|$ we have $|\mathcal{K}| \leq |\widehat{\mathcal{F}}|^k \cdot |\widehat{\Delta}_k|$. Note that $|\widehat{\mathcal{F}}| \leq L$ since it is the output of an $L$-list-decodable learner and $|\widehat{\Delta}_k| \leq (54k/\alpha)^k$ by Proposition 4.3.2. This implies the claimed bound on $|\mathcal{K}|$. □

The only remaining step is to select a good hypothesis from $\mathcal{K}$. This is achieved using the private hypothesis selection algorithm from Corollary 4.5.1 which guarantees that step 5 of Algorithm 3 returns $\widehat{g}$ satisfying $d_{\mathrm{TV}}(g, \widehat{g}) \leq 3 \cdot d_{\mathrm{TV}}(g, \mathcal{K}) + \alpha/2$ with probability $1 - \beta/2$ as long as

$$\begin{aligned} |D_2| &= \Omega\left(\frac{\log(|\mathcal{K}|/\beta)}{\alpha^2} + \frac{\log(|\mathcal{K}|/\beta)}{\alpha\varepsilon}\right) \\ &= \Omega\left(\frac{k\log(Lk/\alpha) + \log(1/\beta)}{\alpha^2} + \frac{k\log(Lk/\alpha) + \log(1/\beta)}{\alpha\varepsilon}\right). \end{aligned} \tag{4.5.1}$$

Combining this with Claim 4.5.3, Claim 4.5.4, and a union bound, we have that with probability $1 - \beta$,

$$d_{\mathrm{TV}}(g, \widehat{g}) \leq 3 \cdot d_{\mathrm{TV}}(g, \mathcal{K}) + \alpha/2 \leq \alpha,$$

where the first inequality follows from private hypothesis selection and the second inequality follows from Claim 4.5.3 and Claim 4.5.4.

Finally, the claimed sample complexity bound follows from the samples required to construct $\widehat{\mathcal{F}}$ (which follows from Claim 4.5.3) and the samples required for private hypothesis selection which is given in Eq. (4.5.1). □

This reduction is quite useful because it is conceptually much simpler to devise list-decodable learners for a given class $\mathcal{F}$. In what follows, we will devise such list-decodable learners for certain classes and use Theorem 4.5.2 to obtain private PAC learners for mixtures of these classes.

## 4.6 Learning Mixtures of Univariate Gaussians

Let $\mathcal{G}$ be the class of all univariate Gaussians. In this section we consider the problem of privately learning univariate Guassian Mixtures, $k$-mix$(\mathcal{G})$. In the previous section, we showed that it is sufficient to design private list-decodable learners for univariate Gaussians. As a warm-up and to build intuition about our techniques, we begin with the simpler problem of constructing private list-decodable learners for Gaussians with a single known variance $\sigma^2$. In what follows, we often use "tilde" (e.g. $\widetilde{M}, \widetilde{V}$) to denote sets that are meant to be *coarse*, or *constant*, approximations and "hat" (e.g. $\widehat{\mathcal{F}}, \widehat{M}, \widehat{V}$) to denote sets that are meant to be *fine*, say $O(\alpha)$, approximations.

### 4.6.1 Warm-up: Learning Gaussian Mixtures with a Known, Shared Variance

In this sub-section we will construct a private list-decodable learner for univariate Gaussians with a known variance $\sigma^2$. A useful algorithmic primitive that we will use throughout this section and the next is the *stable histogram* algorithm. In the following lemma and the remainder of the thesis, $n$ denotes the number of samples that is given to the algorithm.

**Lemma 4.6.1** (Histogram learner [KKMN09, BNS16])**.** *Let $n \in \mathbb{N}$, $\eta, \beta, \varepsilon \in (0,1)$ and $\delta \in (0, 1/n)$. Let $D$ be a dataset of $n$ points over a domain $\mathcal{X}$. Let $K$ be a countable index set and $\mathbf{B} = \{B_i\}_{i \in K}$ be a collection of disjoint bins defined on $\mathcal{X}$, i.e. $B_i \subseteq \mathcal{X}$ and $B_i \cap B_j = \emptyset$ for $i \neq j$. Finally, let $\overline{p}_i = \frac{1}{n} \cdot |D \cap B_i|$. There is an $(\varepsilon, \delta)$-DP algorithm* `Stable-Histogram`$(\varepsilon, \delta, \eta, \beta, D, \mathbf{B})$ *that takes as input parameters*

*$\varepsilon, \delta, \eta, \beta$, dataset $D$ and bins $\mathbf{B}$, and outputs estimates $\{\widetilde{p}_i\}_{i \in K}$ such that for all $i \in K$,*

$$|\overline{p}_i - \widetilde{p}_i| \leq \eta,$$

*with probability no less than $1 - \beta$ so long as*

$$n = \Omega\left(\frac{\log(1/\beta\delta)}{\eta\varepsilon}\right).$$

We note that the condition above on $\delta \in (0, 1/n)$ is standard in the differential privacy literature. Indeed, for useful privacy, $\delta$ should be "cryptographically small", i.e., $\delta \ll 1/n$.

For any fixed $\sigma^2 > 0$ we define $\mathcal{G}_\sigma$ to be the set of all univariate Gaussians with variance $\sigma^2$. For the remainder of this section, we let $g = \mathcal{N}(\mu, \sigma^2) \in \mathcal{G}_\sigma$ and $g' \in \mathcal{H}_\gamma(g)$. (Recall that $g' \in \mathcal{H}_\gamma(g)$ means that $g' = (1-\gamma)g + \gamma h$ for some distribution $h$.) Algorithm 4 shows how we privately output a list of real numbers, one of which is close to the mean of $g$ given samples from $g'$.

**Algorithm 4:** `Univariate-Mean-Decoder`$(\beta, \gamma, \varepsilon, \delta, \widetilde{\sigma}, D)$.

**Input :** Parameters $\varepsilon, \beta, \gamma \in (0,1)$, $\delta \in (0, 1/n)$, $\widetilde{\sigma}$ and dataset $D$

**Output:** Set of approximate means $\widetilde{M}$.

1 Partition $\mathbb{R}$ into bins $\mathbf{B} = \{B_i\}_{i \in \mathbb{N}}$ where $B_i = ((i-0.5)\widetilde{\sigma}, (i+0.5)\widetilde{\sigma}]$.

2 $\{\widetilde{p}_i\}_{i \in \mathbb{N}} \leftarrow$ `Stable-Histogram`$(\varepsilon, \delta, (1-\gamma)/24, \beta/2, D, \mathbf{B})$.

3 $H \leftarrow \{i : \widetilde{p}_i > (1-\gamma)/8\}$

4 If $|H| > 12/(1-\gamma)$ **fail** and return $\widetilde{M} = \emptyset$

5 $\widetilde{M} \leftarrow \{i\widetilde{\sigma} : i \in H\}$

6 **Return** $\widetilde{M}$.

The following lemma shows that the output of Algorithm 4 is a list of real numbers

with the guarantee that at least one element in the list is close to the true mean of a Gaussian which has been corrupted. Note that the lemma assumes the slightly weaker condition where the algorithm receives an approximation to the standard deviation instead of the true standard deviation. This additional generality is used in the next section.

**Lemma 4.6.2.** *Algorithm 4 is an $(\varepsilon, \delta)$-DP algorithm such that for any $g = \mathcal{N}(\mu, \sigma^2)$ and $g' \in \mathcal{H}_\gamma(g)$, when it is given parameters $\varepsilon, \beta, \gamma \in (0,1)$, $\delta \in (0, 1/n)$, $\widetilde{\sigma} \in [\sigma, 2\sigma)$ and dataset $D$ of $n$ i.i.d. samples from $g'$ as input, it outputs a set $\widetilde{M}$ of real numbers of size*

$$|\widetilde{M}| \leq \frac{12}{1-\gamma}.$$

*Furthermore, with probability no less than $1-\beta$ there is an element $\widetilde{\mu} \in \widetilde{M}$ such that*

$$|\widetilde{\mu} - \mu| \leq \sigma,$$

*so long as*

$$n = \Omega\left(\frac{\log(1/\beta\delta)}{(1-\gamma)\varepsilon}\right).$$

Let us begin by gathering several straightforward observations about the algorithm. Let $p_i = \mathbf{P}_{X \sim g'}[X \in B_i]$ be the probability that a sample drawn from $g'$ lands in bin $B_i$. Let $\overline{p}_i = \frac{1}{n}|D \cap B_i|$ be the actual number of samples drawn from $g'$ that have landed in $B_i$. Let $j = \lceil \mu/\widetilde{\sigma} \rfloor$. It is a simple calculation to check that $|j\widetilde{\sigma} - \mu| \leq \sigma$. Thus, we would like to show that $j\widetilde{\sigma} \in \widetilde{M}$ or, equivalently, that $j \in H$. As a first step, we show that many samples actually land in bin $B_j$.

**Claim 4.6.3.** *If $n = \Omega(\log(1/\beta)/(1-\gamma))$ then $\overline{p}_j > (1-\gamma)/6$ with probability at least $1 - \beta/2$.*

*Proof.* First, observe that for a bin $B_i = ((i-0.5)\widetilde{\sigma}, (i+0.5)\widetilde{\sigma}]$ and $X \sim g'$, we have (recalling Definition 4.3.4), $p_i = \mathbf{P}_{X\sim g'}[X \in B_i] \geq (1-\gamma)\mathbf{P}_{X\sim g}[X \in B_i]$. A fairly straightforward calculation (see Proposition B.2.1) gives that $\mathbf{P}_{X\sim g}[X \in B_j] \geq 1/3$ so that $p_j \geq (1-\gamma)/3$.

A standard Chernoff bound (Lemma A.0.5) implies that $|\overline{p}_j - p_j| < p_j/2$ with probability at least $1-\beta/2$ provided $n \geq C\log(1/\beta)/(1-\gamma)$ for some constant $C > 0$. As $p_j \geq (1-\gamma)/3$ this implies $\overline{p}_j > (1-\gamma)/6$. □

Next, we claim that the output of the stable histogram approximately preserves the weight of all the bins and, moreover, that the output does not have too many heavy bins. The first assertion implies that since bin $B_j$ is heavy, the stable histogram also determines that bin $B_j$ is heavy. The second assertion implies that the algorithm does not fail. Let $\{\widetilde{p}_i\}_{i\in\mathbb{N}}$ be the output of the stable histogram, as defined in Algorithm 4.

**Claim 4.6.4.** *If $n = \Omega(\log(1/\beta\delta)/(1-\gamma)\varepsilon)$ then with probability $1-\beta/2$, we have (i) $|\overline{p}_i - \widetilde{p}_i| \leq (1-\gamma)/24$ for all $i \in \mathbb{N}$ and (ii) $|H| = |\{i \in \mathbb{N} : \widetilde{p}_i > (1-\gamma)/8\}| \leq 12/(1-\gamma)$.*

*Proof.* The first assertion directly follows from Lemma 4.6.1 with $\eta = (1-\gamma)/24$. In the event that $|\overline{p}_i - \widetilde{p}_i| \leq (1-\gamma)/24$, we now show that $|H| \leq 12/(1-\gamma)$. Note that it suffices to argue that if $i \in H$ then $\overline{p}_i > (1-\gamma)/12$. Since $\sum_{i\in\mathbb{N}} \overline{p}_i = 1$, this implies that $|H| \leq 12/(1-\gamma)$. Indeed, we argue the contrapositive. If $\overline{p}_i \leq (1-\gamma)/12$ then $\widetilde{p}_i \leq \overline{p}_i + (1-\gamma)/24 \leq (1-\gamma)/8$ and, hence, $i \notin H$. □

With Claim 4.6.3 and Claim 4.6.4 in hand, we are now ready to prove Lemma 4.6.2.

*Proof of Lemma 4.6.2.* We briefly prove that the algorithm is private before proceeding to the other assertions of the lemma.

**Privacy.** Line 2 is the only part of the algorithm that looks at the data and it is $(\varepsilon, \delta)$-DP by Lemma 4.6.1. The remainder of the algorithm can be viewed as post-processing (Lemma 2.2.3) so it does not affect the privacy.

**Bound on $|\widetilde{M}|$.** For the bound on $|\widetilde{M}|$, observe that if $|H| > 12/(1-\gamma)$ then the algorithm fails so $|\widetilde{M}| \leq 12/(1-\gamma)$ deterministically.

**Accuracy.** Let $g, g', \mu$ be as defined in the statement of the lemma. We now show that there exists $\widetilde{\mu} \in \widetilde{M}$ such that $|\widetilde{\mu} - \mu| \leq \sigma$. Let $j = \lceil \mu/\widetilde{\sigma} \rfloor$. For the remainder of the proof, we assume that $n = \Omega(\log(1/\beta\delta)/(1-\gamma)\varepsilon)$.

Claim 4.6.3 asserts that, with probability $1 - \beta/2$, we have $\overline{p}_j > (1-\gamma)/6$. Claim 4.6.4 asserts that, with probability $1 - \beta/2$, $\widetilde{p}_j \geq \overline{p}_j - (1-\gamma)/24$ *and* that $|H| \leq 12/(1-\gamma)$. By a union bound, with probability $1-\beta$, we have that $\overline{p}_j > (1-\gamma)/8$ and the algorithm does not fail. This implies that $j \in H$ so $j\widetilde{\sigma} \in \widetilde{M}$. Finally, note that $|j\widetilde{\sigma} - \mu| \leq \widetilde{\sigma}/2 \leq \sigma$ where the last inequality uses the assumption that $\widetilde{\sigma} \leq 2\sigma$. □

**Corollary 4.6.5.** *For any $\varepsilon \in (0,1)$ and $\delta \in (0, 1/n)$, there is an $(\varepsilon, \delta)$-DP $L$-list-decodable learner for $\mathcal{G}_\sigma$ with known $\sigma > 0$ where $L = O(1/(1-\gamma)\alpha)$, and the number of samples used is*

$$n_{\text{LIST}}(\alpha, \beta, \gamma, \varepsilon, \delta) = O\left(\frac{\log(1/\beta\delta)}{(1-\gamma)\varepsilon}\right).$$

*Proof.* The algorithm is simple; we run `Univariate-Mean-Decoder`$(\varepsilon, \delta, \beta, \gamma, \sigma, D)$ and obtain the set $\widetilde{M}$. Let $\widehat{M}$ be an $\alpha\sigma$-net of the set of intervals $\{[\widetilde{\mu} - \sigma, \widetilde{\mu} + \sigma] : \widetilde{\mu} \in \widetilde{M}\}$ of size $|\widetilde{M}| \cdot (2 \cdot \lceil 1/2\alpha \rceil + 1)$, i.e.

$$\widehat{M} = \{\widetilde{\mu} + 2j\alpha\sigma : \widetilde{\mu} \in \widetilde{M},\ j \in \{0, \pm 1, \ldots, \pm\lceil 1/2\alpha \rceil\}\}.$$

We then return $\widehat{\mathcal{F}} = \{\mathcal{N}(\widehat{\mu}, \sigma^2) : \widehat{\mu} \in \widehat{M}\}$. Finally, Lemma 4.6.2 and post-processing (Lemma 2.2.3) imply that the algorithm is $(\varepsilon, \delta)$-DP while Lemma 4.6.2 and Proposition A.0.6 imply the accuracy guarantee.[1] □

Finally, we use Corollary 4.6.5 and Theorem 4.5.2 to construct an $(\varepsilon, \delta)$-DP PAC learner for $k$-mix$(\mathcal{G}_\sigma)$.

**Theorem 4.6.6.** *For any $\varepsilon \in (0,1)$ and $\delta \in (0, 1/n)$, there is an $(\varepsilon, \delta)$-DP PAC learner for $k$-mix$(\mathcal{G}_\sigma)$ with known $\sigma > 0$ that uses*

$$\begin{aligned} n(\alpha, \beta, \varepsilon, \delta) &= O\left(\frac{k\log(k/\alpha) + \log(1/\beta)}{\alpha^2} + \frac{k\log(k/\alpha\beta\delta)}{\alpha\varepsilon}\right) \\ &= \widetilde{O}\left(\frac{k + \log(1/\beta)}{\alpha^2} + \frac{k\log(1/\beta\delta)}{\alpha\varepsilon}\right) \end{aligned}$$

*samples.*

### 4.6.2 Learning Arbitrary Univariate Gaussian Mixtures

In this section, we construct a list-decodable learner for $\mathcal{G}$, the class of all univariate Gaussians. First, in Algorithm 5, we design an $(\varepsilon, \delta)$-DP algorithm that receives samples from $g' \in \mathcal{H}_\gamma(g)$ where $g \in \mathcal{G}$ and outputs a list of candidate values for the standard deviation, one of which approximates the standard deviation of $g$ with high probability. Then, in Algorithm 6, we use Algorithm 4 and Algorithm 5 to design an $(\varepsilon, \delta)$-DP list-decoder for $\mathcal{G}$.

[1] Note that we can only use Proposition A.0.6 for target $\alpha$ as large as $2/3$. For any target $\alpha > 2/3$, we can simply run the algorithm with $\alpha = 2/3$.

### Estimating the variance

We begin with a method to estimate the variance. Algorithm 5 shows how to take a set of samples and output a list of standard deviations, one of which approximates the true standard deviation up to a factor of 2.

**Algorithm 5:** `Univariate-Variance-Decoder`$(\beta, \gamma, \varepsilon, \delta, D)$.

**Input :** Parameters $\varepsilon, \beta, \gamma \in (0,1)$, $\delta \in (0, 1/n)$, and a dataset $D$

**Output:** Set of approximate standard deviations $\widetilde{V} = \{\widetilde{\sigma}_1, \dots, \widetilde{\sigma}_L\}$.

$Y_k \leftarrow |(X^{2k} - X^{2k-1})/\sqrt{2}|$ for $k \in [n]$. // $X^i$s from Dataset $D = \{X^1, \dots, X^{2n}\}$

$D' \leftarrow \{Y_1, \dots, Y_n\}$.

Partition $\mathbb{R}_{>0}$ into bins $\mathbf{B} = \{B_i\}_{i \in \mathbb{Z}}$ where $B_i = (2^i, 2^{i+1}]$.

$\{\widetilde{p}_i\}_{i \in \mathbb{Z}} \leftarrow$ `Stable-Histogram`$(\varepsilon, \delta, (1-\gamma)^2/24, \beta/2, D', \mathbf{B})$.

$H \leftarrow \{i : \widetilde{p}_i > (1-\gamma)^2/8\}$

If $|H| > 12/(1-\gamma)^2$ **fail** and return $\widetilde{V} = \emptyset$

$\widetilde{V} \leftarrow \{2^{i+1} : i \in H\}$.

**Return** $\widetilde{V}$

**Lemma 4.6.7.** *Algorithm 5 is an $(\varepsilon, \delta)$-DP algorithm such that for any $g = \mathcal{N}(\mu, \sigma^2)$ and $g' \in \mathcal{H}_\gamma(g)$, when it is given parameters $\varepsilon, \beta, \gamma \in (0,1)$, $\delta \in (0, 1/n)$ and dataset $D$ of $2n$ i.i.d. samples from $g'$ as input, it outputs a set $\widetilde{V}$ of positive real numbers of size*

$$|\widetilde{V}| \leq \frac{12}{(1-\gamma)^2}.$$

*Furthermore, with probability no less than $1-\beta$ there is an element $\widetilde{\sigma} \in \widetilde{V}$ such that*

$$\sigma \leq \widetilde{\sigma} < 2\sigma,$$

*so long as*

$$n = \Omega\left(\frac{\log(1/\beta\delta)}{(1-\gamma)^2\varepsilon}\right).$$

The proof of Lemma 4.6.7 mirrors that of Lemma 4.6.2. Let $g = \mathcal{N}(\mu, \sigma^2)$ and $g' \in \mathcal{H}_\gamma(g)$. Let $X, X' \sim g'$ and let $Y = |X - X'|/\sqrt{2}$. For an integer $i$, let $p_i = \mathbf{P}[Y \in B_i]$ where $B_i = (2^i, 2^{i+1}]$. Let $j$ be the (unique) integer such that $\sigma \in (2^j, 2^{j+1}]$.

**Claim 4.6.8.** *If $n = \Omega(\log(1/\beta)/(1-\gamma)^2)$ then $\overline{p}_j > (1-\gamma)^2/6$ with probability $1 - \beta/2$.*

*Proof.* Since, $X, X' \sim g'$ and $Y = |X - X'|/\sqrt{2}$, a straightforward calculation shows that $p_j \geq (1-\gamma)^2/4$ (see Proposition B.2.2 and Proposition B.2.3 for details).

Next, a standard Chernoff bound (Lemma A.0.5) implies that $|\overline{p}_j - p_j| < p_j/3$ with probability at least $1 - \beta/2$ provided $n \geq C\log(1/\beta)/(1-\gamma)^2$ for some constant $C > 0$. As $p_j \geq (1-\gamma)^2/4$ this implies $\overline{p}_j > (1-\gamma)^2/6$. □

**Claim 4.6.9.** *If $n = \Omega(\log(1/\beta\delta)/(1-\gamma)^2\varepsilon)$ then with probability $1 - \beta/2$, we have (i) $|\overline{p}_i - \widetilde{p}_i| \leq (1-\gamma)^2/24$ for all $i \in \mathbb{N}$ and (ii) $|H| = |\{i \in \mathbb{N} : \widetilde{p}_i > (1-\gamma)^2/8\}| \leq 12/(1-\gamma)^2$.*

*Proof.* The first assertion directly follows from Lemma 4.6.1 with $\eta = (1-\gamma)^2/24$. In the event that $|\overline{p}_i - \widetilde{p}_i| \leq (1-\gamma)^2/24$, we now show that $|H| \leq 12/(1-\gamma)^2$. Note that it suffices to argue that if $i \in H$ then $\overline{p}_i > (1-\gamma)^2/12$. Since $\sum_{i\in\mathbb{N}} \overline{p}_i = 1$, this implies that $|H| \leq 12/(1-\gamma)^2$. Indeed, we argue the contrapositive. If $\overline{p}_i \leq (1-\gamma)^2/12$ then $\widetilde{p}_i \leq \overline{p}_i + (1-\gamma)^2/24 \leq (1-\gamma)^2/12$ and, hence, $i \notin H$. □

Given Claim 4.6.8 and Claim 4.6.9, we now prove Lemma 4.6.7.

*Proof of Lemma 4.6.7.* We briefly prove that the algorithm is private before proceeding to the other assertions of the lemma.

**Privacy.** Line 4 is the only part of the algorithm that looks at the data and it is $(\varepsilon, \delta)$-DP by Lemma 4.6.1. The remainder of the algorithm can be viewed as post-processing (Lemma 2.2.3) so does not affect the privacy.

**Bound on $|\widetilde{V}|$.** For the bound on $|\widetilde{V}|$, observe that if $|H| > 12/(1-\gamma)^2$ then the algorithm fails so $|\widetilde{V}| \leq 12/(1-\gamma)^2$ deterministically.

**Accuracy.** Let $g, g', \sigma$ be as defined in the statement of the lemma. We now show that there exists $\widetilde{\sigma} \in \widetilde{V}$ such that $\widetilde{\sigma} \in [\sigma, 2\sigma)$. Let $j$ be the unique integer such that $\sigma \in (2^j, 2^{j+1}]$. For the remainder of the proof, we assume that $n = \Omega(\log(1/\beta\delta)/(1-\gamma)^2\varepsilon)$.

Claim 4.6.8 asserts that, with probability $1 - \beta/2$, we have $\overline{p}_j > (1-\gamma)^2/6$. Claim 4.6.9 asserts that, with probability $1 - \beta/2$, $\widetilde{p}_j \geq \overline{p}_j - (1-\gamma)^2/24$ *and* that $|H| \leq 12/(1-\gamma)^2$. By a union bound, with probability $1 - \beta$, we have that $\overline{p}_j > (1-\gamma)^2/8$ and the algorithm does not fail. This implies that $j \in H$ so $2^{j+1} \in \widetilde{V}$ and, by the choice of $j$, $\sigma \leq 2^{j+1} < 2\sigma$. This completes the proof. □

### A list-decodable learner for univariate Gaussians

Finally, in this this section, we use Algorithm 4 and Algorithm 5 to design a list-decodable learner for $\mathcal{G}$. The list-decodable learner is formally described in Algorithm 6.

**Lemma 4.6.10.** *Algorithm 6 is an $(\varepsilon, \delta)$-DP algorithm such for any $g = \mathcal{N}(\mu, \sigma^2)$ and $g' \in \mathcal{H}_\gamma(g)$, when it is given parameters $\varepsilon, \alpha, \beta, \gamma \in (0,1)$, $\delta \in (0, 1/n)$ and dataset $D$ of $n$ i.i.d. samples from $g'$ as inputs, it outputs a set $\widehat{M}$ of real numbers*

**Algorithm 6:** `Univariate-Gaussian-Decoder`$(\alpha, \beta, \gamma, \varepsilon, \delta, D)$.

**Input :** Parameters $\varepsilon, \alpha, \beta, \gamma \in (0,1)$, $\delta \in (0, 1/n)$ and a dataset $D$

**Output:** Set of approximate means $\widehat{M}$ and variances $\widehat{V}$.

Set $T = 12/(1-\gamma)^2$

Set $\varepsilon' = \varepsilon/(2\sqrt{6T\log(2(T+1)/\delta)})$ and $\delta' = \delta/2(T+1)$

Split $D$ into $D_1, D_2$ where $|D_1| = n_1$, $|D_2| = n_2 = n - n_1$

// $n_1 = \Theta(\log(1/\beta\delta)/(1-\gamma)^2\varepsilon)$.

$\widetilde{V} \leftarrow$ `Univariate-Variance-Decoder`$(\beta/2, \gamma, \varepsilon/2, \delta/2, D_1)$

Initialize $\widehat{M} \leftarrow \emptyset$

For $\widetilde{\sigma}_i \in \widetilde{V}$ **do**

  $\widetilde{M_i} =$ `Univariate-Mean-Decoder`$(\beta/2, \gamma, \varepsilon', \delta', \widetilde{\sigma}_i, D_2)$

  $\widehat{M_i} \leftarrow \{\widetilde{\mu} + j\alpha\widetilde{\sigma}_i \,:\, \widetilde{\mu} \in \widetilde{M_i},\, j \in \{0, \pm 1, \pm 2, \ldots, \pm\lceil 1/\alpha\rceil\}\}$

  $\widehat{M} \leftarrow \widehat{M} \cup \widehat{M_i}$

$C \leftarrow \{\log_2(1+\alpha), 2\log_2(1+\alpha), \ldots, \lceil 1/\log_2(1+\alpha)\rceil \cdot \log_2(1+\alpha)\}$

$\widehat{V} \leftarrow \{\widetilde{\sigma} \cdot 2^{c-1} \,:\, \widetilde{\sigma} \in \widetilde{V},\, c \in C\}$

**Return** $\widehat{M}, \widehat{V}$

*and a set* $\widehat{V}$ *of positive real numbers such that*

$$|\widehat{M}| \leq \frac{144 \cdot (2 \cdot \lceil 1/\alpha \rceil + 1)}{(1-\gamma)^3} \quad \textit{and} \quad |\widehat{V}| \leq \frac{12 \cdot \lceil \log_{1+\alpha}(2) \rceil}{(1-\gamma)^2}.$$

*Furthermore, with probability no less than* $1-\beta$*, we have the following:*

1. $\exists \widehat{\mu} \in \widehat{M}$ *such that* $|\widehat{\mu} - \mu| \leq \alpha\sigma$
2. $\exists \widehat{\sigma} \in \widehat{V}$ *such that* $|\widehat{\sigma} - \sigma| \leq \alpha\sigma$

*so long as*

$$n = \Omega\left(\frac{\log(1/\beta\delta)}{(1-\gamma)^2\varepsilon} + \frac{\log(1/(1-\gamma)\beta\delta)\sqrt{\log(1/(1-\gamma)\delta)}}{(1-\gamma)^2\varepsilon}\right) = \widetilde{\Omega}\left(\frac{\log^{3/2}(1/\beta\delta)}{(1-\gamma)^2\varepsilon}\right).$$

Before we prove the lemma, we make a few simple observations. Fix $g = \mathcal{N}(\mu, \sigma^2)$ and $g' \in \mathcal{H}_\gamma(g)$. We assume that the algorithm receives $D \sim (g')^{2n}$ as input.

**Claim 4.6.11.** *If* $n_1 = \Omega(\log(1/\beta\delta)/(1-\gamma)^2\varepsilon)$ *then with probability* $1-\beta/2$*, (i) there exists* $\widetilde{\sigma} \in \widetilde{V}$ *such that* $\widetilde{\sigma} \in [\sigma, 2\sigma)$ *and (ii) there exists* $\widehat{\sigma} \in \widehat{V}$ *such that that* $|\widehat{\sigma} - \sigma| \leq \alpha\sigma$*.*

*Proof.* Lemma 4.6.7 directly implies that in line 4, with probability $1-\beta/2$, there is some $\widetilde{\sigma} \in \widetilde{V}$ such that $\widetilde{\sigma} \in [\sigma, 2\sigma)$.

For the final assertion, suppose that $\widetilde{\sigma} \in [\sigma, 2\sigma)$. In particular, $\log_2(2\sigma/\widetilde{\sigma}) \in (0, 1]$. Note that $C$ is $\log_2(1+\alpha)$-net of the interval $[0, 1]$. Hence, there exists some $c \in C$ such that $|c - \log_2(2\sigma/\widetilde{\sigma})| \leq \log_2(1+\alpha)$. For such a value of $c$, we have $(\widetilde{\sigma}/\sigma) \cdot 2^{c-1} \in [1/(1+\alpha), 1+\alpha]$, which upon rearranging gives $\widetilde{\sigma}2^{c-1} \in [\sigma/(1+\alpha), \sigma(1+\alpha)]$. As $1/(1+\alpha) \geq 1-\alpha$, this shows that $|\widetilde{\sigma}2^{c-1} - \sigma| \leq \alpha\sigma$. This completes the proof since $\widetilde{\sigma}2^{c-1} \in \widehat{V}$. □

**Claim 4.6.12.** *Let $\varepsilon', \delta'$ be as defined in Algorithm 6. Suppose that there exists $\widetilde{\sigma}_i \in \widetilde{V}$ such that $\widetilde{\sigma}_i \in [\sigma, 2\sigma)$. If $n_2 = \Omega(\log(1/\beta\delta')/(1-\gamma)\varepsilon')$ then with probability $1-\beta/2$ there exists $\widehat{\mu} \in \widehat{M}$ such that $|\widehat{\mu} - \mu| \leq \alpha\sigma$.*

*Proof.* The condition that there exists $\widetilde{\sigma}_i \in \widetilde{V}$ such that $\widetilde{\sigma}_i \in [\sigma, 2\sigma)$ implies that one of the runs of `Univariate-Mean-Decoder` on line 7 uses $\widetilde{\sigma}_i \in [\sigma, 2\sigma)$. The guarantee of Lemma 4.6.2 shows that with probability $1-\beta/2$, there is some $\widetilde{\mu} \in \widetilde{M}_i$ satisfying $|\widetilde{\mu} - \mu| \leq \sigma$. Finally, on line 8, the algorithm constructs $\widehat{M}_i$ which is a $(\alpha\widetilde{\sigma}_i/2)$-net of the interval $[\widetilde{\mu} - \widetilde{\sigma}_i, \widetilde{\mu} + \widetilde{\sigma}_i] \supset [\widetilde{\mu} - \sigma, \widetilde{\mu} + \sigma]$. Hence, there exists $\widehat{\mu} \in \widehat{M}_i$ such that $|\widehat{\mu} - \mu| \leq \alpha\widetilde{\sigma}/2 < \alpha\sigma$ where the latter inequality used that $\widetilde{\sigma} < 2\sigma$. Since $\widehat{M}_i \subset \widehat{M}$, this implies the claim. □

*Proof of Lemma 4.6.10.*

**Privacy.** We first prove that the algorithm is $(\varepsilon, \delta)$-DP. By Lemma 4.6.2, line 4 satisfies $(\varepsilon/2, \delta/2)$-DP. The loop on line 6 runs at most $12/(1-\gamma)^2$ times since $|\widetilde{V}| \leq 12/(1-\gamma)^2$ (see Lemma 4.6.7). So, by our choice of $\varepsilon'$, $\delta'$ (line 2) and advanced composition (Lemma 2.2.2), all the iterations of line 7 collectively satisfy $(\varepsilon/2, \delta/2)$-DP. No subsequent part of the algorithm accesses the data so by basic composition (Lemma 2.2.2) and post processing (Lemma 2.2.3), the entire algorithm is $(\varepsilon, \delta)$-DP.

**Bound on $|\widehat{M}|$ and $|\widehat{V}|$.** We now prove the claimed upper bounds on the sizes of $\widehat{M}$ and $\widehat{V}$. First, we have $|\widetilde{V}| \leq 12/(1-\gamma)^2$ by Lemma 4.6.7. Since $|C| = \lceil 1/\log_2(1+\alpha) \rceil = \lceil \log_{1+\alpha}(2) \rceil$, this gives $|\widehat{V}| = |\widetilde{V}| \cdot |C| \leq 12 \cdot \lceil \log_{1+\alpha}(2) \rceil / (1-\gamma)^2$. Next, we have that each $|\widehat{M}_i| \leq 12/(1-\gamma)$ in Line 8 by Lemma 4.6.2, so $|\widehat{M}_i| \leq 12 \cdot (2 \cdot \lceil 1/\alpha \rceil + 1)/(1-\gamma)$. Hence, $|\widehat{M}| \leq |\widetilde{V}| \cdot 12 \cdot (2 \cdot \lceil 1/\alpha \rceil + 1)/(1-\gamma) \leq 144 \cdot (2 \cdot \lceil 1/\alpha \rceil + 1)/(1-\gamma)^3$.

**Existence of $\widehat{\mu}$ and $\widehat{\sigma}$.** Claim 4.6.11 asserts that with probability $1-\beta/2$, there is $\widetilde{\sigma} \in \widetilde{V}$ such that $\widetilde{\sigma} \in [\sigma, 2\sigma)$ and that there exists $\widehat{\sigma} \in \widehat{V}$ such that $|\widehat{\sigma} - \sigma| \leq \alpha\sigma$. The latter statement is the bound that we asserted for $\widehat{\sigma}$ in the statement of the lemma.

Next, conditioning on the event that there exists $\widetilde{\sigma} \in \widetilde{V}$ such that $\widetilde{\sigma} \in [\sigma, 2\sigma)$, Claim 4.6.12 implies that with probability $1-\beta/2$, there is some $\widehat{\mu} \in \widehat{M}$ such that $|\widehat{\mu} - \mu| \leq \alpha\sigma$.

To conclude, taking a union bound shows that with probability $1-\beta$, there exists $\widehat{\mu} \in \widehat{M}, \widehat{\sigma} \in \widehat{V}$ satisfying $|\widehat{\mu} - \mu| \leq \alpha\sigma$ and $|\widehat{\sigma} - \sigma| \leq \alpha\sigma$.

**Sample complexity.** Finally, we argue about the sample complexity. For Claim 4.6.11, we needed $n_1 = \Omega(\log(1/\beta\delta)/(1-\gamma)^2\varepsilon)$ samples and for Claim 4.6.12, we needed $n_2 = \Omega(\log(1/\beta\delta')/(1-\gamma)\varepsilon')$ samples. Adding $n_1, n_2$ and plugging in the values for $\varepsilon', \delta'$ as defined in Algorithm 6 gives the claimed bound on the number of samples required. □

**Corollary 4.6.13.** *For any $\varepsilon \in (0,1)$ and $\delta \in (0, 1/n)$, there is an $(\varepsilon, \delta)$-DP $L$-list-decodable learner for $\mathcal{G}$ where*

$$L = O\left(\frac{1}{(1-\gamma)^5\alpha^2}\right),$$

*and the algorithm uses*

$$\begin{aligned} n_{\text{LIST}}(\alpha, \beta, \gamma, \varepsilon, \delta) &= O\left(\frac{\log(1/\beta\delta)}{(1-\gamma)^2\varepsilon} + \frac{\log(1/(1-\gamma)\beta\delta)\sqrt{\log(1/(1-\gamma)\delta)}}{(1-\gamma)^2\varepsilon}\right) \\ &= \widetilde{O}\left(\frac{\log^{3/2}(1/\beta\delta)}{(1-\gamma)^2\varepsilon}\right) \end{aligned}$$

*samples.*

*Proof.* The algorithm is simple; we run `Univariate-Gaussian-Decoder`$(\alpha, \beta, \varepsilon, \delta, \gamma, D)$ and obtain the sets $\widehat{M}$ and $\widehat{V}$. We then output $\widehat{\mathcal{F}} = \{\mathcal{N}(\widehat{\mu}, \widehat{\sigma}) \ : \ \widehat{\mu} \in \widehat{M}, \ \widehat{\sigma} \in \widehat{V}\}$. The algorithm is $(\varepsilon, \delta)$-DP by the guarantee of Lemma 4.6.10 and post processing (Lemma 2.2.3). We have from the guarantee of Lemma 4.6.10 that

$$|\widehat{\mathcal{F}}| = |\widehat{M}| \cdot |\widehat{V}| \leq \left(\frac{1728}{(1-\gamma)^5}\right) \cdot \left\lceil \log_{1+\alpha}(2) \right\rceil \cdot (2\left\lceil 1/\alpha \right\rceil + 1).$$

Note that $\log_{1+\alpha}(2) = \frac{\ln(2)}{\ln(1+\alpha)} \leq \frac{2\ln(2)}{\alpha}$ where the last inequality follows from the inequality $\ln(1+x) \geq x/2$ valid for $x \in [0,1]$. This gives the claimed bound that $L = |\widehat{\mathcal{F}}| = O\left(\frac{1}{(1-\gamma)^5\alpha^2}\right)$.

For any $g \in \mathcal{G}$ and $g' \in \mathcal{H}_\gamma(g)$, given $n$ samples from $g'$ as input, we have from the guarantee of Lemma 4.6.10 and Proposition A.0.6 that the algorithm outputs $\widehat{\mathcal{F}}$ satisfying $d_{\mathrm{TV}}(g, \widehat{\mathcal{F}}) \leq \alpha$ so long as

$$n = \Omega\left(\frac{\log(1/\beta\delta)}{(1-\gamma)^2\varepsilon} + \frac{\log(1/(1-\gamma)\beta\delta)\sqrt{\log(1/(1-\gamma)\delta)}}{(1-\gamma)^2\varepsilon}\right) = \widetilde{\Omega}\left(\frac{\log^{3/2}(1/\beta\delta)}{(1-\gamma)^2\varepsilon}\right).$$

This proves the corollary. □

We can now use Corollary 4.6.13 and Theorem 4.5.2 to immediately get the following Theorem.

**Theorem 4.6.14.** *For any $\varepsilon \in (0,1)$ and $\delta \in (0, 1/n)$, there is an $(\varepsilon, \delta)$-DP PAC learner for $k$-mix$(\mathcal{G})$ that uses*

$$n(\alpha, \beta, \varepsilon, \delta) = \widetilde{O}\left(\frac{k^2 \log^{3/2}(1/\beta\delta)}{\alpha^2\varepsilon}\right)$$

*samples.*

## 4.7 Learning Mixtures of High-dimensional Gaussians

In this section, we prove sample complexity upper bounds for learning mixtures of high dimensional Gaussians where $(i)$ each component has the same known covariance matrix and $(ii)$ each component is an axis-aligned Gaussian.

### 4.7.1 Learning Mixtures with Know Covariance

Let $\mathcal{G}_1^d$ be the class of Gaussians with identity covariance matrix. We use ideas similar to those in section 4.6.1 to prove the following result.

**Theorem 4.7.1.** *For any $\varepsilon \in (0,1)$ and $\delta \in (0, 1/n)$, there is an $(\varepsilon, \delta)$-DP PAC learner for $k$-mix$\left(\mathcal{G}_1^d\right)$ that uses*

$$n(\alpha, \beta, \varepsilon, \delta) = \widetilde{O}\left(\frac{kd\log(1/\beta)}{\alpha^2} + \frac{kd + \log(1/\beta\delta)}{\alpha\varepsilon}\right)$$

*samples.*

Note that the theorem also implies the case where the covariance matrix $\Sigma$ is an arbitrary but known covariance matrix. Indeed, given samples $X_1, \ldots, X_m$, one can apply the algorithm of Theorem 4.7.1 to $\Sigma^{-1/2}X_1, \ldots, \Sigma^{-1/2}X_m$ instead.

The proof of Theorem 4.7.1 follows from Theorem 4.5.2 and Corollary 4.7.2, which is a corollary of Lemma 4.6.2.

**Corollary 4.7.2.** *For any $\varepsilon \in (0,1)$ and $\delta \in (0, 1/n)$, there is an $(\varepsilon, \delta)$-DP $L$-list-decodable learner for $\mathcal{G}_1^d$ where $L = O(d/(1-\gamma)\alpha)^d$, and the number of samples used*

*is*

$$n_{\text{List}}(\alpha, \beta, \gamma, \varepsilon, \delta) = O\left(\frac{d \log(d/\beta\delta)}{(1-\gamma)\varepsilon}\right).$$

*Proof.* For each $i \in [d]$ let $D_i = \{X_i : X \in D\}$ be the dataset consisting of the $i$th coordinate of each element in $D$. We run `Univariate-Mean-Decoder`$(\varepsilon/d, \delta/d, \beta/d, \gamma, \sigma, D_i)$ to obtain the set $\widetilde{M}_i$. Let $\widehat{M}_i$ be an $(\alpha/d)$-net of the set of intervals $\{[\widetilde{\mu}_i - 1, \widetilde{\mu}_i + 1] : \widetilde{\mu}_i \in \widetilde{M}_i\}$ of size $|\widetilde{M}_i| \cdot (2 \cdot \lceil d/2\alpha \rceil + 1)$, i.e.

$$\widehat{M}_i = \{\widetilde{\mu}_i + 2j\alpha/d : \widetilde{\mu}_i \in \widetilde{M}_i,\ j \in \{0, \pm 1, \ldots, \pm\lceil d/2\alpha \rceil\}.$$

Let $\widehat{M} = \{(\widehat{\mu}_1, \ldots, \widehat{\mu}_d) : \widehat{\mu}_i \in \widehat{M}_i\}$. We then return $\widehat{\mathcal{F}} = \{\mathcal{N}(\widehat{\mu}, I) : \widehat{\mu} \in \widehat{M}\}$. Finally, Lemma 4.6.2 (with a union bound over the $d$ coordinates), basic composition (Lemma 2.2.2), and post-processing (Lemma 2.2.3) imply that the algorithm is $(\varepsilon, \delta)$-DP while Lemma 4.6.2, Proposition A.0.3, and Proposition A.0.6 imply the accuracy guarantee. □

### 4.7.2 Learning Mixtures of Axis-Aligned Gaussians

In this section, we prove the following result regarding privately learning the class of mixtures of $k$ axis-aligned Gaussians, $k\text{-mix}(\mathcal{G}_A^d)$.

**Theorem 4.7.3.** *For any $\varepsilon \in (0,1)$ and $\delta \in (0, 1/n)$, there is an $(\varepsilon, \delta)$-DP PAC learner for $k\text{-mix}\left(\mathcal{G}_A^d\right)$ that uses*

$$n(\alpha, \beta, \varepsilon, \delta) = \widetilde{O}\left(\frac{k^2 d \log^{3/2}(1/\beta\delta)}{\alpha^2 \varepsilon}\right)$$

*samples.*

We now demonstrate how to construct an $(\varepsilon, \delta)$-DP list-decodable learner for the class of $d$-dimensional axis-aligned Gaussians, $\mathcal{G}_A^d$. Recall that the class of $d$-dimensional axis-aligned Gaussians is the class of all Gaussians with a diagonal covariance matrix, where the diagonals are arbitrary positive real numbers.

**Algorithm 7:** `Multivariate-Gaussian-Decoder`$(\alpha, \beta, \gamma, \varepsilon, \delta, D)$.

**Input :** Parameters $\varepsilon, \alpha, \beta, \gamma \in (0,1)$, $\delta \in (0, 1/n)$, and a dataset $D$

**Output:** Set of distributions $\widehat{\mathcal{F}} \subset \mathcal{G}_A^d$.

Initialize $\widehat{V}_j \leftarrow \emptyset$, $\widehat{M}_j \leftarrow \emptyset$ for $j \in [d]$

Set $D_i \leftarrow \{X_i \,:\, X \in D\}$ for $i \in [d]$ `// Split dataset by dimension.`

For $i \in [d]$ **do**

  $\widehat{M}_i, \widehat{V}_i \leftarrow$ `Univariate-Gaussian-Decoder`$(\alpha/d, \beta/d, \gamma, \varepsilon/d, \delta/d, D_i)$

$\widehat{M} \leftarrow \{(\widehat{\mu}_1, \dots, \widehat{\mu}_d) \,:\, \widehat{\mu}_i \in \widehat{M}_i,\ i \in [d]\}$

$\widehat{\Lambda} \leftarrow \{\mathrm{diag}(\widehat{\sigma}_1^2, \dots, \widehat{\sigma}_d^2) \,:\, \widehat{\sigma}_i \in \widehat{V}_i,\ i \in [d]\}$

$\widehat{\mathcal{F}} \leftarrow \left\{\mathcal{N}(\widehat{\mu}, \widehat{\Sigma}) \,:\, \widehat{\mu} \in \widehat{M},\ \widehat{\Sigma} \in \widehat{\Lambda}\right\}$

**Return** $\widehat{\mathcal{F}}$

**Lemma 4.7.4.** *For any $\varepsilon \in (0,1)$ and $\delta \in (0, 1/n)$, Algorithm 7 is an $(\varepsilon, \delta)$-DP $L$-list-decodable learner for $\mathcal{G}_A^d$ where*

$$L = O\left(\frac{d^2}{(1-\gamma)^5\alpha^2}\right)^d,$$

*and the algorithm uses*

$$\begin{aligned} n_{\mathrm{List}}(\alpha, \beta, \gamma, \varepsilon, \delta) &= O\left(\frac{d\log(d/\beta\delta)}{(1-\gamma)^2\varepsilon} + \frac{d\log(d/(1-\gamma)\beta\delta)\sqrt{\log(d/(1-\gamma)\delta)}}{(1-\gamma)^2\varepsilon}\right) \\ &= \widetilde{O}\left(\frac{d\log^{3/2}(1/\beta\delta)}{(1-\gamma)^2\varepsilon}\right) \end{aligned}$$

*samples.*

*Proof.*

**Privacy.** We first prove the algorithm is $(\varepsilon, \delta)$-DP. By the guarantee of Lemma 4.6.10, each run of line 4 in the loop is $(\varepsilon/d, \delta/d)$-DP. No subsequent part of the algorithm accesses the data, so by post processing (Lemma 2.2.3) and basic composition (Lemma 2.2.2) the entire algorithm is $(\varepsilon, \delta)$-DP.

**Bound on $|\widehat{\mathcal{F}}|$.** We now prove the claimed upper bound on the size of $\widehat{\mathcal{F}}$. By the guarantee of Lemma 4.6.10, each $\widehat{M}_i$ and $\widehat{V}_i$ obtained on line 4 satisfy $|\widehat{M}_i| \leq 144 \cdot (2 \cdot \lceil d/\alpha \rceil + 1)/(1-\gamma)^3$ and $|\widehat{V}_i| \leq 12 \cdot \lceil \log_{1+\alpha/d}(2) \rceil / (1-\gamma)^2$. This immediately gives us

$$|\widetilde{\mathcal{F}}| = |\widehat{M}| \cdot |\widehat{\Lambda}| = \left( \prod_{i=1}^{d} |\widehat{M}_i| \right) \cdot \left( \prod_{i=1}^{d} |\widehat{V}_i| \right) \leq \left( \left( \frac{1728}{(1-\gamma)^5} \right) \cdot \lceil \log_{1+\alpha/d}(2) \rceil \cdot (2 \cdot \lceil d/\alpha \rceil + 1) \right)^d .$$

To get the bound on $L = |\widehat{\mathcal{F}}|$ as stated in the lemma, we use the fact that $\log_{1+\alpha/d}(2) = \frac{\ln(2)}{\ln(1+\alpha/d)} \leq \frac{2\ln(2)}{\alpha/d}$, where the inequality uses the fact that $\ln(1+x) \geq x/2$ for $x \in [0,1]$.

**Accuracy and sample complexity.** We now prove that the algorithm is a list-decodable learner. Fix some $g = \prod_{i=1}^{d} \mathcal{N}(\mu_i, \sigma_i^2) \in \mathcal{G}_A^d$ and $g' \in \mathcal{H}_\gamma(g)$. By our choice of parameters and the guarantee of Lemma 4.6.10, a single run of algorithm `Univariate-Gaussian-Decoder` on line 4 outputs lists $\widehat{M}_i$ and $\widehat{V}_i$ such that there exist $\widehat{\mu}_i \in \widehat{M}_i$ and $\widehat{\sigma}_i \in \widehat{V}_i$ satisfying $|\widehat{\mu}_i - \mu_i| \leq \alpha\sigma_i/d$ and $|\widehat{\sigma}_i - \sigma_i| \leq \alpha\sigma_i/d$ with

probability at least $1 - \beta/d$ so long as

$$n = \Omega\left(\frac{d\log(d/\beta\delta)}{(1-\gamma)^2\varepsilon} + \frac{d\log(d/(1-\gamma)\beta\delta)\sqrt{\log(d/(1-\gamma)\delta)}}{(1-\gamma)^2\varepsilon}\right).$$

By a union bound, we have with probability no less than $1-\beta$ that for all $i \in [d]$, $|\widehat{\mu}_i - \mu_i| \leq \alpha\sigma_i/d$ and $|\widehat{\sigma}_i - \sigma_i| \leq \alpha\sigma_i/d$. By a standard argument, this implies that with probability at least $1-\beta$ there is some $\widehat{g} \in \widehat{\mathcal{F}}$ such that $d_{\mathrm{TV}}(\widehat{g}, g) \leq \alpha$ (see Proposition A.0.6 and Proposition A.0.3). □

We can now put together Lemma 4.7.4 and Theorem 4.5.2 to immediately get Theorem 4.7.3.

# Chapter 5

# Conclusion

In thesis, we proved upper bounds on the amount of data required to privately learn distributions from two fundamental classes. More specifically, we proved sample complexity upper bounds for privately learning arbitrary high-dimensional Gaussians and mixtures of axis-aligned Gaussians under the rigid constraint of differential privacy.

Many interesting and important questions remain open related to the results in Chapters 3 and 4. We conclude this thesis by stating the most important ones.

**Learning Gaussians.** While our algorithms in Chapter 3 are *statistically efficient*, they leave much to be desired in terms of computational efficiency. Thus, a major open problem is whether there exists computationally efficient algorithms that achieves the same or comparable sample complexity bounds as ours, even for the realizable setting.

Recall that it is known that the first and last terms in Theorem 3.6.6 are nearly tight (there exist matching lower bounds up to log-factors). It is thus an interesting question whether the second term in our bound is also nearly tight.

**Learning Mixtures of Gaussians.** Many interesting open problems remain for privately learning mixtures of Gaussians. The simplest problem is to understand the exact sample complexity (up to constants) for learning mixtures of univariate Gaussians under approximate differential privacy. We make the following conjecture based on known bounds for privately learning a single Gaussian [KV18].

**Conjecture 5.0.1** (Informal)**.** *The sample complexity of learning a mixture of $k$, univariate Gaussians to within total variation distance $\alpha$ with high probability under $(\varepsilon, \delta)$-DP is*

$$\Theta\left(\frac{k}{\alpha^2} + \frac{k}{\alpha\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right).$$

Another wide open question is whether it is even possible to privately learn mixtures of high-dimensional Gaussians when each Gaussian can have an arbitrary covariance matrix. We believe it is possible, and make the following conjecture based on our results in Chapter 3 for privately learning a single high-dimensional Gaussian with no assumptions on the parameters.

**Conjecture 5.0.2** (Informal)**.** *The sample complexity of learning a mixture of $k$, $d$-dimensional Gaussians to with total variation distance $\alpha$ with high probability under $(\varepsilon, \delta)$-DP is*

$$\Theta\left(\frac{kd^2}{\alpha^2} + \frac{kd^2}{\alpha\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right).$$

# Appendix A

# Useful Inequalities

**Proposition A.0.1.** *Let $X$ and $Y$ be random variables taking values in the same set. For any function $f$, we have $d_{\mathrm{TV}}(f(X), f(Y)) \leq d_{\mathrm{TV}}(X, Y)$.*

*Proof.* For any set $A$ we have,

$$\mathbf{Pr}[f(X) \in A] - \mathbf{Pr}[f(Y) \in A] = \mathbf{Pr}[X \in f^{-1}(A)] - \mathbf{Pr}[Y \in f^{-1}(A)] \leq d_{\mathrm{TV}}(X, Y).$$

Taking the supremum of the left hand side completes the proof. □

**Corollary A.0.2.** *Let $X$ and $Y$ be random variables taking values in the same set. For any* invertible *function $f$, we have $d_{\mathrm{TV}}(f(X), f(Y)) = d_{\mathrm{TV}}(X, Y)$.*

*Proof.* By Proposition A.0.1, $d_{\mathrm{TV}}(f(X), f(Y)) \leq d_{\mathrm{TV}}(X, Y)$ and $d_{\mathrm{TV}}(f^{-1}(f(X)), f^{-1}(f(Y))) \leq d_{\mathrm{TV}}(f(X), f(Y))$. □

**Proposition A.0.3** (Lemma 3.3.7, [Rei89])**.** *For $i \in [d]$ let $p_i$ and $q_i$ be distributions*

*over the same domain $\mathcal{X}$. Then*

$$d_{\mathrm{TV}}\left(\prod_{i=1}^{d} p_i, \prod_{i=1}^{d} q_i\right) \leq \sum_{i=1}^{d} d_{\mathrm{TV}}\left(p_i, q_i\right).$$

**Proposition A.0.4.** *Fix some $\xi \in (0,1)$. Let $g$ be a distribution that satisfies $d_{\mathrm{TV}}(g, \mathcal{N}(\mu,\Sigma)) \leq \xi$. If $X_1, X_2 \sim g^2$, and $Y = (X_1 - X_2)/\sqrt{2} \sim q$, then $d_{\mathrm{TV}}(q, \mathcal{N}(0,\Sigma)) \leq 2\xi$.*

*Proof.* Let $X_1, X_2 \sim P^2$ and $Z_1, Z_2 \sim (\mathcal{N}(\mu,\Sigma))^2$. It follows from Proposition A.0.3 and our assumption on $P$ that

$$d_{\mathrm{TV}}\big((X_1,X_2),(Z_1,Z_2)\big) \leq 2 d_{\mathrm{TV}}(P, \mathcal{N}(\mu,\Sigma)) \leq 2\xi.$$

Let $f(x,y) = (x-y)/\sqrt{2}$ so that $f(X_1,X_2) \sim Q$ and $f(Z_1,Z_2) \sim \mathcal{N}(0,\Sigma)$. Using Corollary A.0.2 with the observation above gives us

$$d_{\mathrm{TV}}(Q, \mathcal{N}(0,\Sigma)) = d_{\mathrm{TV}}\big(f(X_1,X_2), f(Z_1,Z_2)\big) = d_{\mathrm{TV}}\big((X_1,X_2),(Z_1,Z_2)\big) \leq 2\xi.$$

□

**Lemma A.0.5** (Chernoff bound; see [Ver18, Exercise 2.3.6])**.** *Let $X_1, \ldots, X_n$ be independent Bernoulli random variables. Let $S_n = \sum_{i=1}^{n} X_i$ and $\mu = S_n$. Then for any $\delta \in (0,1]$ and some absolute constant $c > 0$*

$$\mathbf{P}[|S_n - \mu| \geq \delta\mu] \leq 2e^{-c\mu\delta^2}.$$

**Proposition A.0.6** (Lemma 2.11, [ABH+20])**.** *For any $\mu, \widetilde{\mu} \in \mathbb{R}$ and $\sigma, \widetilde{\sigma} > 0$ with*

*$|\widetilde{\mu} - \mu| \leq \alpha\sigma$ and $|\widetilde{\sigma} - \sigma| \leq \alpha\sigma$ where $\alpha \in [0, 2/3]$, the Gaussians $\mathcal{N}(\mu, \sigma^2)$ and $\mathcal{N}(\widetilde{\mu}, \widetilde{\sigma}^2)$ statisfy*

$$d_{\mathrm{TV}}\left(\mathcal{N}(\mu, \sigma^2), \mathcal{N}(\widetilde{\mu}, \widetilde{\sigma}^2)\right) \leq \alpha.$$

# Appendix B

# Omitted Results from Chapter 4

## B.1 Omitted Results from Section 4.5

**Proposition B.1.1.** *Let $\alpha \in (0,1)$ and $k \in \mathbb{N}$. Let $g = \sum_{i=1}^{k} w_i f_i$ and $\widetilde{g} = \sum_{i=1}^{k} \widetilde{w}_i \widetilde{f}_i$ be two mixture distributions that satisfy*

1. *$\|w - \widetilde{w}\|_\infty \leq \alpha/k$; and*
2. *$d_{\mathrm{TV}}(f_i, \widetilde{f}_i) \leq \alpha$ for $i \in [k]$ such that $w_i \geq \alpha/k$.*

*Then $d_{\mathrm{TV}}(g, \widetilde{g}) \leq 3\alpha$.*

*Proof.* Let $N = \{i \in [k] : w_i \geq \alpha/k\}$. We have that

$$\begin{aligned}
d_{\mathrm{TV}}(\widehat{g}, g) &= \frac{1}{2}\left\|\sum_{i=1}^{k} \widehat{w}_i \widehat{f}_i - \sum_{i=1}^{k} w_i f_i\right\|_1 \\
&= \frac{1}{2}\left\|\sum_{i=1}^{k} \widehat{w}_i(\widehat{f}_i - f_i) + \sum_{i=1}^{k}(\widehat{w}_i - w_i) f_i\right\|_1 \\
&\leq \frac{1}{2}\left\|\sum_{i=1}^{k} \widehat{w}_i(\widehat{f}_i - f_i)\right\|_1 + \frac{1}{2}\left\|\sum_{i=1}^{k}(\widehat{w}_i - w_i) f_i\right\|_1 \\
&\leq \frac{1}{2}\left\|\sum_{i\notin N} \widehat{w}_i(\widehat{f}_i - f_i)\right\|_1 + \frac{1}{2}\left\|\sum_{i\in N} \widehat{w}_i(\widehat{f}_i - f_i)\right\|_1 + \frac{1}{2}\left\|\sum_{i=1}^{k}(\widehat{w}_i - w_i) f_i\right\|_1 \\
&\leq \frac{1}{2}\sum_{i\notin N} \widehat{w}_i \left\|\widehat{f}_i - f_i\right\|_1 + \frac{1}{2}\sum_{i\in N} \widehat{w}_i \left\|\widehat{f}_i - f_i\right\|_1 + \frac{1}{2}\sum_{i=1}^{k} |\widehat{w}_i - w_i| \left\|\widehat{f}_i\right\|_1 \\
&\leq \sum_{i\notin N} \frac{\alpha}{k} \cdot 1 + \sum_{i\in N} \widehat{w}_i \cdot \alpha + \sum_{i=1}^{k} \frac{\alpha}{k} \cdot 1 \\
&\leq \alpha + \alpha + \alpha = 3\alpha.
\end{aligned}$$

Note that in the second-to-last inequality, we used that for $i \notin N$, $\widehat{w}_i \leq \alpha/k$ and the trivial bound $\|\widehat{f}_i - f_i\|_1 \leq 2$ while for $i \in N$, we have $\|\widehat{f}_i - f_i\|_1 \leq \alpha$. □

## B.2 Omitted Results from Section 4.6

**Proposition B.2.1.** *Fix some univariate Gaussian $g = \mathcal{N}(\mu, \sigma^2)$. Let $\widetilde{\sigma}$ satisfy $\sigma \leq \widetilde{\sigma} < 2\sigma$. Partition $\mathbb{R}$ into disjoint bins $\{B_i\}_{i\in\mathbb{N}}$ where $B_i = ((i-0.5)\widetilde{\sigma}, (i+0.5)\widetilde{\sigma}]$ and let $j = \lceil \mu/\widetilde{\sigma} \rfloor$, where $\lceil \cdot \rfloor$ denotes rounding to the nearest integer. It follows that:*

1. $\mathbf{P}_{X\sim g}[X \in B_j] \geq 1/3$,
2. $\mu \in [(j-0.5)\widetilde{\sigma}, (j+0.5)\widetilde{\sigma}]$.

*Proof.* We first prove item 1.

$$\begin{aligned}\mathbf{P}_{X\sim g}[X \in B_j] &= \Phi\left(\frac{(j+0.5)\widetilde{\sigma}}{\sigma} - \frac{\mu}{\sigma}\right) - \Phi\left(\frac{(j-0.5)\widetilde{\sigma}}{\sigma} - \frac{\mu}{\sigma}\right) \\ &= \Phi\left(\frac{j\widetilde{\sigma}-\mu}{\sigma} + \frac{\widetilde{\sigma}}{2\sigma}\right) - \Phi\left(\frac{j\widetilde{\sigma}-\mu}{\sigma} - \frac{\widetilde{\sigma}}{2\sigma}\right) \\ &:= f\left(\frac{j\widetilde{\sigma}-\mu}{\sigma}\right).\end{aligned}$$

Notice that $f(\xi) = \Phi(\xi + \widetilde{\sigma}/2\sigma) - \Phi(\xi - \widetilde{\sigma}/2\sigma)$ is decreasing with $|\xi|$. Furthermore, by the definition of $j$ we have,

$$\begin{aligned}\left|\frac{j\widetilde{\sigma}-\mu}{\sigma}\right| &= \frac{\widetilde{\sigma}}{\sigma}\left|j' - \frac{\mu}{\widetilde{\sigma}}\right| \\ &\leq \frac{\widetilde{\sigma}}{\sigma}\cdot\frac{1}{2} = \frac{\widetilde{\sigma}}{2\sigma}.\end{aligned}$$

So,

$$\begin{aligned}\mathbf{P}_{X\sim g}[X \in B_j] &= f\left(\frac{j\widetilde{\sigma}-\mu}{\sigma}\right) \\ &\geq f\left(\frac{\widetilde{\sigma}}{2\sigma}\right) \\ &= \Phi\left(\frac{\widetilde{\sigma}}{\sigma}\right) - \Phi(0) \\ &\geq \Phi(1) - \Phi(0) \geq 1/3,\end{aligned}$$

where the second last inequality follows from the fact that $\widetilde{\sigma}/\sigma \geq 1$ together with the monotonicity of the c.d.f. and the last inequality follows from a direct calculation.

We now prove the second claim that $\mu \in [(j-0.5)\widetilde{\sigma}, (j+0.5)\widetilde{\sigma})]$. As we saw above,

it follows that

$$\frac{1}{\sigma}\left|j\widetilde{\sigma}-\mu\right| \leq \frac{\widetilde{\sigma}}{2\sigma} \implies \mu \in [(j-0.5)\widetilde{\sigma}, (j+0.5)\widetilde{\sigma}].$$

□

**Proposition B.2.2.** *Fix some univariate Gaussian $g = \mathcal{N}(0, \sigma^2)$. Partition $\mathbb{R}_{>0}$ into disjoint bins $\{B_i\}_{i\in\mathbb{Z}}$ where $B_i = (2^i, 2^{i+1}]$ and let $j \in \mathbb{N}$ satisfy $2^j < \sigma \leq 2^{j+1}$. It follows that:*

$$\mathbf{P}_{X\sim g}[|X| \in B_j] \geq \frac{1}{4}.$$

*Proof.* Since $2^j < \sigma \leq 2^{j+1}$, we can write $\sigma = 2^{j+c}$ for some $c \in (0, 1]$. Let $x = 2^{-c}$ and notice $x \in [1/2, 1)$. We have the following:

$$\begin{aligned}\mathbf{P}_{X\sim g}[|X| \in B_j] &= 2\left(\Phi\left(\frac{2^{j+1}}{\sigma}\right) - \Phi\left(\frac{2^j}{\sigma}\right)\right)\\ &= 2\left(\Phi\left(2^{1-c}\right) - \Phi\left(2^{-c}\right)\right)\\ &= 2f(2^{-c}),\end{aligned} \tag{B.2.1}$$

where we define $f(x) = \Phi(2x) - \Phi(x)$. We now aim to lower bound $f(x)$. By taking the derivative of $f(x)$ twice, we have that $f''(x) = \sqrt{(1/2\pi)}(x\exp(-x^2/2) - 8x\exp(-2x^2))$. By a simple calculation, we have that $f''(x) \leq 0$ when $x \in [0, 2\ln 8/3] \supset [1/2, 1)$, so $f(x)$ is concave when $x \in [1/2, 1)$. This implies that $f(x) \geq \min\{f(1/2), f(1)\}$ for

any $x \in [1/2, 1)$, so from Eq. (B.2.1) we have

$$\begin{aligned}\mathbf{P}_{X\sim g}[|X| \in B_j] &\geq 2\min\{f(1/2), f(1)\} \\ &= 2\min\left\{\Phi(1) - \Phi\left(\frac{1}{2}\right), \Phi(2) - \Phi(1)\right\} \\ &> \frac{1}{4},\end{aligned}$$

where the last inequality follows from a direct calculation. □

**Proposition B.2.3.** *Fix $g = \mathcal{N}(\mu, \sigma^2)$ and $g' \in \mathcal{H}_\gamma(g)$. Let $Z = (X_1 - X_2)/\sqrt{2}$ where $X_1, X_2 \sim g'$ i.i.d. Let $Y \sim \mathcal{N}(0, \sigma^2)$. Then for any measurable $S \subseteq \mathbb{R}$*

$$\mathbf{P}[|Z| \in S] \geq (1-\gamma)^2 \cdot \mathbf{P}[|Y| \in S].$$

*Proof.* We prove this via a coupling argument. Since $g' \in \mathcal{H}_\gamma(g)$ we have $g' = (1-\gamma)g + \gamma h$ for some distribution $h$.

Let $Y_1, Y_2 \sim g$ i.i.d. so that $Y = \frac{Y_1 - Y_2}{\sqrt{2}} \sim \mathcal{N}(0, \sigma^2)$. Also, let $H_1, H_2 \sim h$ i.i.d. Finally, let $B_1, B_2$ be independent Bernoulli random variables with parameter $1-\gamma$, i.e. $B_i = 1$ with probability $1-\gamma$ and $B_i = 0$ with probability $\gamma$.

Now let $X_i = Y_i \cdot B_i + H_i \cdot (1 - B_i)$ and note that $X_i \sim g'$. If $B_1 = B_2 = 1$ and $|Y| \in S$ then certainly $|Z| = |X_1 - X_2|/\sqrt{2} \in S$. Hence,

$$\mathbf{P}[|Z| \in S] \geq \mathbf{P}[\{B_1 = 1\} \cap \{B_2 = 1\} \cap \{|Y| \in S\}] = (1-\gamma)^2 \mathbf{P}[|Y| \in S],$$

where the last equality uses the fact that $B_1, B_2, Y$ are mutually independent random variables. □

# Bibliography

[AA20] Ishaq Aden-Ali and Hassan Ashtiani. On the sample complexity of learning sum-product networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4508–4518. PMLR, 2020.

[AAK21] Ishaq Aden-Ali, Hassan Ashtiani, and Gautam Kamath. On the sample complexity of privately learning unbounded high-dimensional gaussians. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato, editors, *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pages 185–216. PMLR, 16–19 Mar 2021.

[AAL21] Ishaq Aden-Ali, Hassan Ashtiani, and Christopher Liaw. Privately learning mixtures of axis-aligned gaussians. *arXiv preprint arXiv:2106.02162*, 2021.

[ABDH+18] Hassan Ashtiani, Shai Ben-David, Nicholas Harvey, Christopher Liaw, Abbas Mehrabian, and Yaniv Plan. Nearly tight sample complexity bounds for learning mixtures of Gaussians via sample compression schemes. In *Advances in Neural Information Processing Systems 31*, NeurIPS '18, pages 3412–3421. Curran Associates, Inc., 2018.

[ABDM18] Hassan Ashtiani, Shai Ben-David, and Abbas Mehrabian. Sample-efficient learning of mixtures. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI'18, pages 2679–2686. AAAI Publications, 2018.

[ABH+20] Hassan Ashtiani, Shai Ben-David, Nicholas J. A. Harvey, Christopher Liaw, Abbas Mehrabian, and Yaniv Plan. Near-optimal sample complexity bounds for robust learning of gaussian mixtures via compression schemes. *J. ACM*, 67(6):32:1–32:42, 2020.

[AFJ+18] Jayadev Acharya, Moein Falahatgar, Ashkan Jafarpour, Alon Orlitsky, and Ananda Theertha Suresh. Maximum selection and sorting with adversarial comparators. *Journal of Machine Learning Research*, 19(1):2427–2457, 2018.

[AJOS14] Jayadev Acharya, Ashkan Jafarpour, Alon Orlitsky, and Ananda Theertha Suresh. Sorting with adversarial comparators and application to density estimation. In *Proceedings of the 2014 IEEE International Symposium on Information Theory*, ISIT '14, pages 1682–1686, Washington, DC, USA, 2014. IEEE Computer Society.

[Ant95] Martin Anthony. Classification by polynomial surfaces. *Discrete Applied Mathematics*, 61(2):91–103, 1995.

[ASZ18] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Differentially private testing of identity and closeness of discrete distributions. In *Advances in Neural Information Processing Systems 31*, NeurIPS '18, pages 6878–6891. Curran Associates, Inc., 2018.

[ASZ19] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Hadamard response: Estimating distributions privately, efficiently, and with little communication. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, AISTATS '19, pages 1120–1129. JMLR, Inc., 2019.

[ASZ20] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Differentially private assouad, fano, and le cam. *arXiv preprint arXiv:2004.06830*, 2020.

[BBKN14] Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine Learning*, 94(3):401–437, 2014.

[BBV08] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. A discriminative framework for clustering via similarity functions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 671–680, 2008.

[BD14] Rina Foygel Barber and John C. Duchi. Privacy and statistical risk: Formalisms and minimax bounds. *CoRR*, abs/1412.4451, 2014.

[BDKU20] Sourav Biswas, Yihe Dong, Gautam Kamath, and Jonathan Ullman. Coinpress: Practical private mean and covariance estimation. *arXiv preprint arXiv:2006.06618*, 2020.

[BDRS18] Mark Bun, Cynthia Dwork, Guy N. Rothblum, and Thomas Steinke. Composable and versatile privacy via truncated cdp. In *Proceedings of*

*the 50th Annual ACM Symposium on the Theory of Computing*, STOC '18, pages 74–86, New York, NY, USA, 2018. ACM.

[BEM+17] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th ACM Symposium on Operating Systems Principles*, SOSP '17, pages 441–459, New York, NY, USA, 2017. ACM.

[BK21] Ainesh Bakshi and Pravesh K. Kothari. List-decodable subspace recovery: Dimension independent error in polynomial time. In *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, page 1279–1297, 2021.

[BKM19] Olivier Bousquet, Daniel M. Kane, and Shay Moran. The optimal approximation factor in density estimation. In *Proceedings of the 32nd Annual Conference on Learning Theory*, COLT '19, pages 318–341, 2019.

[BKSW19] Mark Bun, Gautam Kamath, Thomas Steinke, and Zhiwei Steven Wu. Private hypothesis selection. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 156–167. Curran Associates, Inc., 2019.

[BNS16] Mark Bun, Kobbi Nissim, and Uri Stemmer. Simultaneous private learning of multiple concepts. In *Proceedings of the 7th Conference on Innovations in Theoretical Computer Science*, ITCS '16, pages 369–380, New York, NY, USA, 2016. ACM.

[BNSV15] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '15, pages 634–649, Washington, DC, USA, 2015. IEEE Computer Society.

[BRT09] Maria Florina Balcan, Heiko Röglin, and Shang-Hua Teng. Agnostic clustering. In *International Conference on Algorithmic Learning Theory*, pages 384–398. Springer, 2009.

[BS16] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Proceedings of the 14th Conference on Theory of Cryptography*, TCC '16-B, pages 635–658, Berlin, Heidelberg, 2016. Springer.

[BS19] Mark Bun and Thomas Steinke. Average-case averages: Private algorithms for smooth sensitivity and mean estimation. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 181–191. Curran Associates, Inc., 2019.

[BSU17] Mark Bun, Thomas Steinke, and Jonathan Ullman. Make up your mind: The price of online queries in differential privacy. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 1306–1325, Philadelphia, PA, USA, 2017. SIAM.

[BUV14] Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *Proceedings of the*

*46th Annual ACM Symposium on the Theory of Computing*, STOC '14, pages 1–10, New York, NY, USA, 2014. ACM.

[CKM+19] Clément L. Canonne, Gautam Kamath, Audra McMillan, Jonathan Ullman, and Lydia Zakynthinou. Private identity testing for high-dimensional distributions. *arXiv preprint arXiv:1905.11947*, 2019.

[CMY20] Yeshwanth Cherapanamjeri, Sidhanth Mohanty, and Morris Yau. List decodable mean estimation in nearly linear time. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 141–148, 2020.

[CSV17] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM Symposium on the Theory of Computing*, STOC '17, pages 47–60, New York, NY, USA, 2017. ACM.

[CWZ19] T. Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *arXiv preprint arXiv:1902.04495*, 2019.

[DDS12] Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A. Servedio. Learning Poisson binomial distributions. In *Proceedings of the 44th Annual ACM Symposium on the Theory of Computing*, STOC '12, pages 709–728, New York, NY, USA, 2012. ACM.

[DFM+20] Wenxin Du, Canyon Foot, Monica Moniot, Andrew Bray, and Adam Groce. Differentially private confidence intervals. *arXiv preprint arXiv:2001.02285*, 2020.

[DHS15] Ilias Diakonikolas, Moritz Hardt, and Ludwig Schmidt. Differentially private learning of structured discrete distributions. In *Advances in Neural Information Processing Systems 28*, NIPS '15, pages 2566–2574. Curran Associates, Inc., 2015.

[Dif17] Differential Privacy Team, Apple. Learning with privacy at scale. https://machinelearning.apple.com/docs/learning-with-privacy-at-scale/appledifferentialprivacysystem.pdf, December 2017.

[DJW17] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 2017.

[DK14] Constantinos Daskalakis and Gautam Kamath. Faster and sample near-optimal algorithms for proper learning mixtures of Gaussians. In *Proceedings of the 27th Annual Conference on Learning Theory*, COLT '14, pages 1183–1213, 2014.

[DKK+16] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '16, pages 655–664, Washington, DC, USA, 2016. IEEE Computer Society.

[DKK20] Ilias Diakonikolas, Daniel Kane, and Daniel Kongsgaard. List-decodable mean estimation via iterative multi-filtering. *Advances in Neural Information Processing Systems*, 33, 2020.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '06, pages 486–503, Berlin, Heidelberg, 2006. Springer.

[DKS18] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. List-decodable robust mean estimation and learning mixtures of spherical Gaussians. In *Proceedings of the 50th Annual ACM Symposium on the Theory of Computing*, STOC '18, pages 1047–1060, New York, NY, USA, 2018. ACM.

[DKY17] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems 30*, NIPS '17, pages 3571–3580. Curran Associates, Inc., 2017.

[DL96] Luc Devroye and Gábor Lugosi. A universally acceptable smoothing factor for kernel density estimation. *The Annals of Statistics*, 24(6):2499–2512, 1996.

[DL97] Luc Devroye and Gábor Lugosi. Nonasymptotic universal smoothing factors, kernel complexity and Yatracos classes. *The Annals of Statistics*, 25(6):2626–2637, 1997.

[DL01] Luc Devroye and Gábor Lugosi. *Combinatorial methods in density estimation*. Springer, 2001.

[DL09] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the 41st Annual ACM Symposium on the Theory of Computing*, STOC '09, pages 371–380, New York, NY, USA, 2009. ACM.

[DLS+17] Aref N. Dajani, Amy D. Lauger, Phyllis E. Singer, Daniel Kifer, Jerome P. Reiter, Ashwin Machanavajjhala, Simson L. Garfinkel, Scot A. Dahl, Matthew Graham, Vishesh Karwa, Hang Kim, Philip Lelerc, Ian M. Schmutte, William N. Sexton, Lars Vilhuber, and John M. Abowd. The modernization of statistical disclosure limitation at the U.S. census bureau, 2017. Presented at the September 2017 meeting of the Census Scientific Advisory Committee.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, Berlin, Heidelberg, 2006. Springer.

[DMR18] Luc Devroye, Abbas Mehrabian, and Tommy Reddad. The total variation distance between high-dimensional Gaussians. *arXiv preprint arXiv:1810.08693*, 2018.

[DR16] Cynthia Dwork and Guy N. Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016.

[DR18] John C. Duchi and Feng Ruan. The right complexity measure in locally private estimation: It is not the fisher information. *arXiv preprint arXiv:1806.05756*, 2018.

[DR19] John Duchi and Ryan Rogers. Lower bounds for locally private estimation via communication complexity. In *Proceedings of the 32nd Annual Conference on Learning Theory*, COLT '19, pages 1161–1191, 2019.

[DRV10] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, FOCS '10, pages 51–60, Washington, DC, USA, 2010. IEEE Computer Society.

[DSS+15] Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. Robust traceability from trace amounts. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '15, pages 650–669, Washington, DC, USA, 2015. IEEE Computer Society.

[EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM Conference on Computer and Communications Security*, CCS '14, pages 1054–1067, New York, NY, USA, 2014. ACM.

[GKK+20] Sivakanth Gopi, Gautam Kamath, Janardhan Kulkarni, Aleksandar Nikolov, Zhiwei Steven Wu, and Huanyu Zhang. Locally private hypothesis selection. In *Proceedings of the 33rd Annual Conference on*

*Learning Theory*, COLT '20, 2020.

[GRS19] Marco Gaboardi, Ryan Rogers, and Or Sheffet. Locally private confidence intervals: Z-test and tight confidence intervals. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, AISTATS '19, pages 2545–2554. JMLR, Inc., 2019.

[HT10] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the 42nd Annual ACM Symposium on the Theory of Computing*, STOC '10, pages 705–714, New York, NY, USA, 2010. ACM.

[Hub64] Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964.

[JKMW19] Matthew Joseph, Janardhan Kulkarni, Jieming Mao, and Zhiwei Steven Wu. Locally private Gaussian estimation. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 2980–2989. Curran Associates, Inc., 2019.

[KBR16] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. In *Proceedings of the 33rd International Conference on Machine Learning*, ICML '16, pages 2436–2444. JMLR, Inc., 2016.

[KKK19] Sushrut Karmalkar, Adam Klivans, and Pravesh Kothari. List-decodable linear regression. *Advances in neural information processing systems*, 2019.

[KKMN09] Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. Releasing search queries and clicks privately. In *Proceedings of the 18th International World Wide Web Conference*, WWW '09, pages 171–180, New York, NY, USA, 2009. ACM.

[KLSU19] Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan Ullman. Privately learning high-dimensional distributions. In *Proceedings of the 32nd Annual Conference on Learning Theory*, COLT '19, pages 1853–1902, 2019.

[KSSU19] Gautam Kamath, Or Sheffet, Vikrant Singhal, and Jonathan Ullman. Differentially private algorithms for learning mixtures of separated Gaussians. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 168–180. Curran Associates, Inc., 2019.

[KSU20] Gautam Kamath, Vikrant Singhal, and Jonathan Ullman. Private mean estimation of heavy-tailed distributions. In *Proceedings of the 33rd Annual Conference on Learning Theory*, COLT '20, 2020.

[KU20] Gautam Kamath and Jonathan Ullman. A primer on private statistics. *arXiv preprint arXiv:2005.00010*, 2020.

[KV18] Vishesh Karwa and Salil Vadhan. Finite sample differentially private confidence intervals. In *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science*, ITCS '18, pages 44:1–44:9, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[LKKO21] Xiyang Liu, Weihao Kong, Sham M. Kakade, and Sewoong Oh. Robust and differentially private mean estimation. *CoRR*, abs/2102.09159, 2021.

[LSY+20] Yuhan Liu, Ananda Theertha Suresh, Felix Yu, Sanjiv Kumar, and Michael Riley. Learning discrete distributions: User vs item-level privacy. *arXiv preprint arXiv:2007.13660*, 2020.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 94–103, Washington, DC, USA, 2007. IEEE Computer Society.

[Mu08] Satyaki Mahalanabis and Daniel Štefankovič. Density estimation in linear time. In *Proceedings of the 21st Annual Conference on Learning Theory*, COLT '08, pages 503–512, 2008.

[NRS07] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on the Theory of Computing*, STOC '07, pages 75–84, New York, NY, USA, 2007. ACM.

[Rei89] Rolf-Dieter Reiss. *Approximate distributions of order statistics with applications to nonparametric statistics*. Springer Series in Statistics. Springer-Verlag, New York, 1989.

[RY20a] Prasad Raghavendra and Morris Yau. List decodable learning via sum of squares. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 161–180. SIAM, 2020.

[RY20b] Prasad Raghavendra and Morris Yau. List decodable subspace recovery. In *Conference on Learning Theory*, pages 3206–3226. PMLR, 2020.

[Smi11] Adam Smith. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the 43rd Annual ACM Symposium on the Theory of Computing*, STOC '11, pages 813–822, New York, NY, USA, 2011. ACM.

[SOAJ14] Ananda Theertha Suresh, Alon Orlitsky, Jayadev Acharya, and Ashkan Jafarpour. Near-optimal-sample estimators for spherical Gaussian mixtures. In *Advances in Neural Information Processing Systems 27*, NIPS '14, pages 1395–1403. Curran Associates, Inc., 2014.

[SU17a] Thomas Steinke and Jonathan Ullman. Between pure and approximate differential privacy. *The Journal of Privacy and Confidentiality*, 7(2):3–22, 2017.

[SU17b] Thomas Steinke and Jonathan Ullman. Tight lower bounds for differentially private selection. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '17, pages 552–563, Washington, DC, USA, 2017. IEEE Computer Society.

[VC71] Vladimir Naumovich Vapnik and Alexey Yakovlevich Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

[Ver18] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

[VV10] Gregory Valiant and Paul Valiant. A CLT and tight lower bounds for estimating entropy. *Electronic Colloquium on Computational Complexity (ECCC)*, 17(179), 2010.

[WHW+16] Shaowei Wang, Liusheng Huang, Pengzhan Wang, Yiwen Nie, Hongli Xu, Wei Yang, Xiang-Yang Li, and Chunming Qiao. Mutual information optimally local private discrete distribution estimation. *arXiv preprint arXiv:1607.08025*, 2016.

[Yat85] Yannis G. Yatracos. Rates of convergence of minimum distance estimators and Kolmogorov's entropy. *The Annals of Statistics*, 13(2):768–774, 1985.

[YB18] Min Ye and Alexander Barg. Optimal schemes for discrete distribution estimation under locally differential privacy. *IEEE Transactions on Information Theory*, 64(8):5662–5676, 2018.

[ZKKW20] Huanyu Zhang, Gautam Kamath, Janardhan Kulkarni, and Zhiwei Steven Wu. Privately learning Markov random fields. In *Proceedings of the 37th International Conference on Machine Learning*, ICML '20. JMLR, Inc., 2020.