# Sample and Query Complexities of Some Estimation Problems

**Sayantan Sen**

Supervisor: Dr. Sourav Chakraborty

Advanced Computing and Microelectronics Unit
Indian Statistical Institute
203 B. T. Road, Kolkata-700108

A thesis submitted in partial fulfillment of the requirements for the degree of *Doctor of Philosophy* in *Computer Science* at *Indian Statistical Institute*

October, 2023

*To Ma and Baba*

# ACKNOWLEDGEMENTS

Every Ph.D. is a very personal and emotional journey, and mine was no different. First and foremost I would like to thank my advisor Sourav Chakraborty. Sourav introduced me to research and instilled a love for the beautiful area of property testing in me. Over the last five years, we have had innumerable discussions over various topics: both in academics as well as outside it. He has always treated me as a colleague from the first time we started working on. He has been always been available for discussions at almost any time, even during his busy schedule. Especially, in early 2020, when the first wave of COVID-19 had struck, and the whole world was under lockdown, he used to check on me multiple times every week, just to see if everything was okay. I consider myself extremely fortunate to have Sourav as my supervisor.

Next, I would like to thank Eldar Fischer. I have learned a great lot of things about research and property testing from him. Two papers in this thesis have been joint works with him, and his insights into problems continue to amaze me. Special thanks to him for hosting me at Technion for an academic visit, where we had several fruitful discussions.

I would also like to thank Arijit Bishnu and Arijit Ghosh for their help during my Ph.D. They were always available for discussions and provided me with advice when I needed it. I thank Arijit Bishnu for his probability course in my first semester, which has helped me greatly in my research.

Next, I would thank Gopinath Mishra. I met him when he was a senior Ph.D. student in the department. Over the last few years, we have worked on many projects, and I have learned a lot from discussions with him. Thanks to Ansuman Banerjee, Shayak Chakraborty and Kuldeep S. Meel, Simon Apers, Sander Gribling and Dániel Szabó for the fruitful collaborations. I would also thank Uddalok Sarkar and Rahul Raychaudhury for collaborating on two projects with me during my Ph.D. journey.

I now thank all my collaborators: Simon Apers, Ansuman Banerjee, Arijit Bishnu, Sourav Chakraborty, Shayak Chakraborty, Eldar Fischer, Arijit Ghosh, Sander Gribling, Kuldeep S. Meel, Rahul Raychaudhury, Uddalok Sarkar and Dániel Szabó. Special thanks to Sophie Laplante for hosting me for an academic visit at IRIF, Paris, which eventually resulted in a joint paper with Simon, Sander, and Dániel. Also, thanks to Siddharth Barman and Anand Louis for a wonderful visit to Indian Institute of Science (IISc). I also thank Nithin Varma for hosting me at Chennai Mathematical Institute (CMI) for an academic visit.

I would like to thank Indian Statistical Institute, and my department ACMU in particular, to provide me with an excellent environment for research. Specially, I thank my fellow Ph.D. students at ACMU: Manaswi, Gopinath, Chandrima, Avijeet, Soumi, Sourasena, Uddalok, Kaustabh, Arun, Joy, Santlal, Souvik, Swarnalipa, and Lakshmikanta for all the fun and discussions we had over the last five years. Thanks to all of my other unmentioned friends. Please accept my apologies for not being able to put all of your names.

I thank Indian Statistical Institute for funding me during my Ph.D. and to Indian Association for Research in Computing Science (IARCS) for partially sponsoring my travel to Paris to present our work at ICALP 2022.

Finally, saving the best words for the last, I thank my family: my parents, my brothers, my nephew, and my sister-in-law for their never-ending love, and support, especially to my nephew Aratrik, who has filled our lives with love. Finally, I thank my parents for everything. This thesis is dedicated to them.

Sayantan Sen

# Abstract

Given data from some experiment, inferring information from the underlying distribution is of prime importance, and has been extensively studied. However, due to the huge size of the data, traditional methods are often no longer applicable. Thus new tools and techniques are being developed for inferring useful information from large amounts of data. This thesis makes progress in this direction.

The primary goal is to design efficient randomized algorithms aka. testers that can distinguish whether a given unknown object is “close” or “far” from a property of interest with as few accesses as possible. This is referred to as distribution testing when the unknown object is a probability distribution, and graph property testing when it is a graph. The minimum number of samples required to decide a property in distribution testing is referred to as sample complexity, while in graph property testing, it is referred to as query complexity.

In this thesis, we study several fundamental problems in distribution and graph property testing such as (i) Can one design a tolerant tester for any distribution property with only black-box access to a non-tolerant tester? (ii) Does there exist distribution properties with global structure that can be learnt efficiently? (iii) the role of adaptivity in distribution testing, and tolerant testing for (iv) graph isomorphism and (v) bipartiteness.

The results of the thesis are divided into three parts. In the first part, we study the connection between the sample complexities of non-tolerant and tolerant testing of distributions and prove a tight quadratic gap for label-invariant (symmetric) properties, while providing lower bounds for non-concentrated properties. We also present an algorithm that can learn a concentrated distribution even when its support set is unknown apriori.

In the second part, we investigate problems (ii) and (iii) in huge object model, where distributions are defined over n-dimensional Hamming cube and the tester obtains n-bit strings as samples. Since reading the string in its entirety may not be feasible for large $n$, the tester has query access to the sampled strings. We define the notion of index-invariant properties, properties that are invariant under the permutations of the indices $\{1,\ldots,n\}$ and prove that any index-invariant property whose VC-dimension is bounded has a tester whose query complexity is independent of n and depends only on VC-dimension. Moreover, the dependencies of sample and query complexities of our tester on the VC-dimension are tight. We also study the power of adaptiveness in this model and prove a tight quadratic separation between query complexities of adaptive and non-adaptive testers for index-invariant properties, compared to tight exponential separation for its non-index-invariant counterpart.

In the third part, we study property testing of dense graphs and give positive answers to problems (iv) and (v). We prove that tolerant graph isomorphism testing is equivalent to the problem of estimating the Earth Mover Distance of two distributions, constructed from the graphs. Moreover, our equivalence proof is model-independent. Finally, we design a tester for tolerant bipartiteness testing whose query and time complexities are significantly better compared to previous works.

# Contents

# List of Figures

# Chapter 1

# Introduction

In the computer science community, designing algorithms that run in time linear in the size of the input has been the gold standard since its inception. Indeed, it is difficult to imagine designing algorithms for nontrivial problems that perform better, as the algorithm at least needs to read the input to make its decision. However, as larger and larger data sets are becoming more prevalent day by day, reading the input in its entirety is no longer feasible. Thus there has been huge interest in designing algorithms that run in sublinear time and read only a part of the input. Over the past two decades, there has been extensive research, and several tools and techniques have been developed for this purpose.

Often these data sets come with some suitable representation. For example, we can view the data sets as samples from some underlying probability distributions. Examples include data sets of network traffic records, financial transactions, sensors, etc. In most of the scenarios, we do not know the underlying probability distribution explicitly, we just have access to samples from the unknown distribution. To make sense of the data sets, the primary goal is to understand some properties of the underlying probability distribution, by seeing as few samples as possible. For example, we might want to estimate the number of elements in the support of the distribution that have non-zero probability mass. Many of these questions can be answered successfully using classical techniques from Statistics. However, it turns out that by applying techniques from Statistics literature, we often need at least a linear number of samples to understand the properties of the underlying distribution. But the challenge big data proposes is the immense size of the supports of the distributions. This makes several known techniques impractical for practical purposes. In order to tackle this challenge, several tools have been developed over the past few decades.

The field of property testing was started by the work of Rubinfeld and Sudan [RS96], where the authors studied the problem of testing the correctness of programs. Later Goldreich, Goldwasser, and Ron [GGR98] formally introduced the notion of property testing in their seminal work. In that work, they studied several properties of dense graphs like bipartitness, colorability, maximum cut, etc., and designed efficient algorithms for these problems. Later various properties of bounded degree graphs have also been studied [GR97].

The field of distribution testing was initiated implicitly in the works of Goldreich and Ron [GR00, GR11], where they tested whether a bounded degree regular graph is an expander via estimating the $\ell_2$ norm of an underlying distribution, as well as the uniformity property. Later Batu et. al [BFR$^+$00] studied the problem of testing whether two unknown distributions are close as well as the problem of identity and independence testing of distribution in [BFF$^+$01]. The sample complexity bound for testing uniformity was later improved by Paninski [Pan08].

The problem of distinguishing whether an unknown distribution $D$ has some property $\mathcal{P}$ or it is far from all distributions with that property is normally referred to as *non-tolerant testing* of $\mathcal{P}$. The minimum number of samples required for deciding any property $\mathcal{P}$ is defined as the ***Sample Complexity*** of testing $\mathcal{P}$. Several natural properties like uniformity [BFR$^+$00, Pan08], monotonicity [BKR04, ADK15], histogram [ILR12] etc. have been studied over the years, and the design of such testers have used a wide array of techniques.

Another related problem is the problem of *tolerant testing* of distributions. Here we want to distinguish whether $D$ is "*close*", or "*far*" from $\mathcal{P}$. Although it seems that tolerant testing is a generalization of the non-tolerant variant, it is interesting to note that tolerant testing problems are often significantly harder than their non-tolerant counterparts and require new techniques. For example, the problem of non-tolerant testing of whether a distribution is uniform or not requires $\Theta(\sqrt{n})$ samples [Pan08], but tolerant testing of uniformity requires $\Theta(n/\log n)$ samples, where $n$ is the size of the support of the distribution [VV10, VV11].

When the unknown huge object is a graph $G$, the problem of distinguishing if $G$ is "close" or "far" from a property $\mathcal{P}$ is called *graph property testing*. The field of graph property testing was first introduced in the seminal work of Goldreich, Goldwasser, and Ron [GGR98]. Their work introduced the model which is now referred to as *dense graph property testing*. In this model, a graph $G$ on $n$ vertices is represented as an $n \times n$ adjacency matrix, where $(i, j)$-th entry of the matrix is $1$ if there is an edge between the vertices $i$ and $j$, and $0$ otherwise. The tester can perform query to any entry of the adjacency matrix. A graph $G$ is said to be $\varepsilon$-close to $\mathcal{P}$ if at most $\varepsilon n^2$ edges are required to be modified (added or removed) to make the modified graph have the property $\mathcal{P}$. The goal here is to design testers that perform as few queries to the adjacency matrix as possible to decide $\mathcal{P}$ with high probability. The minimum number of queries required to test a property $\mathcal{P}$ is called the ***Query Complexity*** of testing $\mathcal{P}$. In [GGR98], the authors studied various properties like degree regularity, bipartiteness, maximum cut, $k$-colorability, etc. They further studied a more general problem called *graph partition problem* which generalizes $k$-colorability, as well as biclique and maximum cut. Surprisingly, several properties which can be expressed as graph partition problems in this model have very efficient testers, often testers have query complexity independent of $n$, depending only on the proximity parameter $\varepsilon$.

Over the last two decades, property testing in dense graph model has been extensively studied. The problem of $k$-colorability was later studied by Alon and Krivelevich [AK02] who designed a tester with better complexities. This bound was further improved by Sohler [Soh12]. The problem of estimating the size of the maximum cut was later improved by Alon, Vega, Kannan, and Karpinski [AdlVKK03]. Another important problem is the problem of isomorphism testing. The problem of property testing of graph isomorphism was first studied by Fischer and Matsliah [FM08] and subsequently, Babai and Chakraborty [BC10] studied the non-tolerant property testing version of the hypergraph isomorphism problem. Another interesting set of works was initiated by Alon, Fischer, Krivilevich, and Szegedy [AFKS00] who applied Szemeredi Regularity Lemma to design efficient testers for various dense graph properties. Since then, there have been several works that employed the Regularity lemma as well as its several variants for designing efficient testers [AS05, FN07, AS08, AFNS09].

Apart from the dense graph model, graph property testing has also been investigated in the *bounded degree* model initiated by Goldreich and Ron [GR97]. Here the degree of every vertex of the graph is bounded by a constant $d$, and the graphs are represented as adjacency lists, that is, every vertex of the graph $G$ has a list containing its neighbors in $G$ in an arbitrary order. A graph $G$ is said to be $\varepsilon$-far from a property if we need to modify (add or delete) at least $\varepsilon dn$ edges to make the modified graph have the property. See [GR97, GR99, AK02, BOT02, CS10b, CSS09, GR11, Soh12, CGR$^+$14, CPS15, KSS18] for related results in this model. Later Parnas and Ron [PR02] defined the notion of *general graph model*, which bridges the adjacency matrix and bounded degree graph models. Here a graph $G$ with $m$ edges is said to be $\varepsilon$-far from another graph $H$ if we need to modify at least $\varepsilon m$ edges to make $G$ isomorphic to $H$. See the works [PR02, KKR04, Fei04, CRT05, CEF$^+$05, PR07, GR08, NO08, NO08, AKKR08, CS09, YYI09, MR09, ORRR12, ELRS17, ER18, ERS19, Lev21] for several relevant results and techniques.

For detailed references of the results and various related techniques, see the books of Goldreich [Gol17] and Bhattacharyya and Yoshida [BY22], and the surveys of Fischer [Fis04], Ron [Ron08, Ron09], Czumaj and Sohler [CS10a], Rubinfeld [Rub12], Rubinfeld and Shapira [RS11], Cannone [Can20a, Can22] to name a few.

## 1.1 Various models of computation

Now we present a brief introduction to the complexity models that have been studied in this thesis. Let us start with the sampling model.

### 1.1.1 Sampling model

In this model, we assume the unknown distribution $D$ is defined over a finite set $\Omega$ and is represented as an oracle. Often $\Omega$ is set as $[n] = \{1, \dots, n\}$. The tester can get independent samples from the oracle corresponding to $D$. The primary goal is to design a tester that uses as few samples as possible to decide some property of $D$.

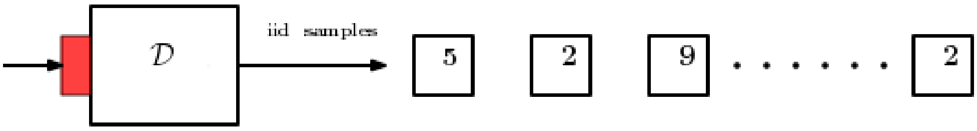

Figure 1.1: Sampling model for distribution testing

### 1.1.2 Huge object model

In the standard distribution testing model, samples are drawn independently from the input distribution. It is implicitly assumed that the size of each sample is small enough that the tester can read it in its entirety. Thus the primary goal has been to minimize the number of samples required by the tester to decide some property.

However when the distributions are defined over some large domain, say the $n$-dimensional Hamming cube $\{0, 1\}^n$ for a large $n$, even reading a few samples is infeasible. To address this, Goldreich and Ron [GR22] have defined a new model called the *huge object model*, where the samples may only be queried in a few places. The primary objective here is to optimize the sample as well as the query complexities of the tester.

### 1.1.3 Adjacency matrix model

This model is used to study various properties of dense graphs. We will assume that the graphs have $n$ vertices numbered as $\{1, \dots, n\}$.

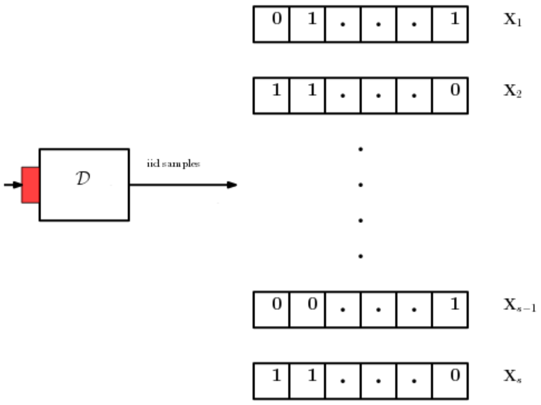

Figure 1.2: Query model for huge object testing

*Adjacency matrix* model is the most studied model in the field of graph property testing when the graphs that are under test are dense graphs. This model was introduced in the seminal work of Goldreich, Goldwasser, and Ron [GGR98], where the authors studied various properties of dense graphs in this model. Here the input graph is stored as an adjacency matrix, and we can perform queries to the matrix. The goal is to minimize the number of queries required to decide whether the unknown graph has some particular property, or is it far from all graphs with this property. Now we formally define the query procedure.

Let us assume that the graph is $G(V, E)$, where $V$ and $E$ denote the set of vertices and edges of $G$ respectively. We will assume that $|V| = n$, and $|E| = m$, where $n$ and $m$ are two non-negative integers. The type of query that can be performed here is said to be *edge-existence* query (aka. *adjacency* query), and is defined as follows:

**Edge-existence query**: For two vertices $u, v \in V$, given $\{u, v\}$ as input to the oracle, the oracle returns $1$ if there is an edge between the vertices $u$ and $v$, and $0$ otherwise.

There are other types of queries defined for the case when the graph is sparse. However, in this thesis, we will only study the properties of dense graphs, and will only use edge-existence query.

$$\begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & \boxed{X_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nn} \end{bmatrix}$$

Figure 1.3: Adjacency matrix model

## 1.2 Our results in this thesis

In this thesis, we study several fundamental problems in the field of property testing. Namely:

**(i)** Given any distribution property $\mathcal{P}$, can one design a tolerant tester for $\mathcal{P}$ when one has black-box access to a non-tolerant tester for $\mathcal{P}$, without knowing any internal details of the non-tolerant tester?

**(ii)** Does there exist distribution properties with global structure that can be learned efficiently?

**(iii)** How does adaptivity play role in designing testers for distribution properties?

**(iv)** How to efficiently test whether a dense graph is close or far from being isomorphic to another graph?

**(v)** Can we test whether a dense graph is close or far from being bipartite efficiently?

The results in this thesis are divided into three parts. In Part I, we study the relation of tolerant and non-tolerant testing of probability distributions in the Sampling Model, where the distribution under test is represented as an oracle, and independent samples can be obtained from it, and answer (i) affirmatively. In Part II, we study distribution testing in the huge object model. Here the distributions are defined over $n$-dimensional Hamming cube, and the tester can obtain samples from the oracle, as well as perform queries to the strings obtained as samples. Here we study problems (ii) and (iii). Finally, in Part III, we study property testing of dense graphs, where we give positive answers to problems (iv) and (v). In the following, we present an overview of the results of this thesis.

### 1.2.1 Part I: Results in the Sampling Model

The problem of distinguishing whether an unknown distribution $D$ has some property $\mathcal{P}$ or it is far from all distributions with that property is normally referred to as *non-tolerant testing* of $\mathcal{P}$. Several natural properties like uniformity [Pan08], monotonicity [BKR04, ADK15], histogram [ILR12] etc. have been studied over the years, and the design of such testers have used a wide array of techniques.

Another related problem is the problem of *tolerant testing* of distributions. Here we want to distinguish whether $D$ is "*close*", or "*far*" from a property $\mathcal{P}$. Although it seems that tolerant testing is a generalization of the non-tolerant variant, it is interesting to note that tolerant testing problems are often significantly harder than their non-tolerant counterparts and require new techniques. For example, the problem of non-tolerant testing of whether a distribution is uniform or far from it requires $\Theta(\sqrt{n})$ samples [BFR$^+$00, Pan08], but tolerant testing of uniformity requires $\Theta(n/\log n)$ samples, where $n$ is the size of the support of the distribution [VV10, VV11]. Although tight bounds for tolerant and non-tolerant testing of several problems are known for a long time, there is no general technique that can construct a tolerant tester from its non-tolerant counterpart. Below we give an outline of our results in this part of the thesis.

**Chapter 4: Construction of tolerant testers for label-invariant properties** For distribution properties that are *label-invariant*, that is, properties that remain invariant under the relabeling of the support elements of the distribution such as uniformity or entropy, we design a tolerant tester from its non-tolerant counterpart that requires at most quadratic number of samples, compared to the non-tolerant tester. This gap is tight since the property of uniformity is known to have an almost quadratic gap. Surprisingly, our tester requires only the existence of the non-tolerant tester, not its details. Our main result is stated as follows:

**Theorem 1.1 (Informal).** *Any label-invariant distribution property that can be non-tolerantly tested using $\Lambda$ samples, can also be tolerantly tested using $\widetilde{\mathcal{O}}(\min\{\Lambda^2, n\})$ samples, where $n$ is the size of the support of the distribution* [1].

Our tolerant tester corresponding to Theorem 1.1 is not constructive. So, we design a tolerant tester for linear properties (properties that can be expressed as a feasible solution to a set of linear inequalities), that uses the same number of samples as the tester corresponding to Theorem 1.1, and runs in polynomial time.

**Theorem 1.2 (Informal).** *Any label-invariant distribution property that can be non-tolerantly tested using $\Lambda$ samples and can be expressed as a feasible solution to $m$ linear inequalities, can also be tolerantly tested using $\widetilde{\mathcal{O}}(\min\{\Lambda^2, n\})$ samples and in time polynomial in $m$ and $n$, where $n$ is the size of the support of the distribution.*

**Chapter 5: Lower bound results for non-concentrated properties** When moving to general, not necessarily label-invariant properties, the situation is more complicated, and we show some partial results. We show that if a property requires the distributions to be *non-concentrated*, that is, if the probability mass of the distribution is sufficiently spread out, then it can not be non-tolerantly tested with $o(\sqrt{n})$ samples, where $n$ denotes the support size. Clearly, this implies at most a quadratic gap, because a distribution can be learned (and hence tolerantly tested against any property) using $\mathcal{O}(n)$ samples.

**Theorem 1.3 (Informal).** *In order to non-tolerantly test any non-concentrated distribution property, $\Omega(\sqrt{n})$ samples are required, where $n$ is the size of the support of the distribution.*

Being non-concentrated is a strong requirement on properties, as we also prove a close to linear lower bound against their tolerant tests.

[1] $\widetilde{\mathcal{O}}(\cdot)$ hides a poly-logarithmic factor.

**Theorem 1.4 (Informal).** *The sample complexity of tolerant testing of any non concentrated label-invariant distribution property is $\Omega(n^{1-o(1)})$, where $n$ is the size of the support of the distribution.*

**Chapter 6: Learning Distributions with Unknown Support** Apart from the case where the distribution is non-concentrated, we also show if an input distribution is very concentrated, in the sense that it is mostly supported on a subset of size $s$ of the universe, then it can be learned using only $\mathcal{O}(s)$ samples. The learning procedure adapts to the input, and works without knowing $s$ in advance.

**Theorem 1.5 (Informal).** *To learn a distribution approximately, $\mathcal{O}(|S|)$ samples are enough, where $S \subseteq [n]$ is an unknown set of minimum cardinality whose mass is close to* 1. *Note that $|S|$ is also unknown, and the algorithm adapts to it.*

Theorem 1.1 and Theorem 1.2 are formally stated and proved in Chapter 4. Later in Chapter 5, we present the proofs of Theorem 1.3 and Theorem 1.4. Finally, in Chapter 6, we discuss the proof of Theorem 1.5.

**This part is based on the following paper:**

1. Sourav Chakraborty, Eldar Fischer, Arijit Ghosh, Gopinath Mishra, and Sayantan Sen. Exploring the Gap Between Tolerant and Non-Tolerant Distribution Testing, In Proceedings of the $26^{th}$ International Conference on Randomization and Computation (RANDOM), Volume 245, 27:1-27:23, 2022, doi: 10.4230/LIPIcs.APPROX/RANDOM.2022.27. Presented in Highlights of Algorithms (HALG), 2023. Submitted to the journal IEEE Transactions on Information Theory.

### 1.2.2 Part II: Results in the Huge Object Model

In the standard distribution testing model, samples are drawn independently from the input distribution. It is implicitly assumed that the size of each sample is small enough that the tester can read it in its entirety. Thus the primary goal has been to minimize the number of samples required by the tester to decide some property.

However when the distributions are defined over some large domain, say the $n$-dimensional Hamming cube $\{0,1\}^n$ for a large $n$, even reading a few samples is infeasible. To address this, Goldreich and Ron [GR22] have defined a new model called the *huge object model*, where the samples may only be queried in a few places. The objective here is to optimize the sample as well as the query complexities of the tester. The authors in [GR22] studied several well-studied properties in the standard sampling model in this new framework.

In a recent work [CFG$^+$23], we initiate the study of a general class of properties in this model, named as *index-invariant* properties. Informally speaking, these are the properties that are invariant under the permutations of the indices $\{1,\ldots,n\}$. Many interesting properties like monotonicity are index-invariant. It is interesting to note that these properties differ from the more common notion of label-invariant properties that we have discussed before. Now we present an outline of our results in this part of the thesis.

**Chapter 8: Learning Clusterable Distributions** In this chapter, we study *clusterable* distributions, that is, distributions whose support set can be partitioned into various parts. We prove that any distribution that is clusterable, can be learned by performing a number of queries that is independent of $n$. Formally, the result is stated as follows:

**Theorem 1.6 (Informal).** *Given sample and query access to an unknown distribution $D$ over $\{0,1\}^n$, there exists a non-adaptive algorithm that makes a number of queries that is independent of $n$, and either reports a full description of a distribution over $\{0,1\}^n$ or reports* FAIL, *satisfying both of the following conditions:*

*(i) If $D$ is clusterable, then with probability at least $\frac{2}{3}$, the algorithm outputs a full description of a distribution $D'$ such that $D$ is $\varepsilon$-close to $D'_\sigma$ for some permutation $\sigma : [n] \to [n]$.*

*(ii) For any $D$, the algorithm will not output a distribution $D'$ such that $D'_\sigma$ is $\varepsilon$-far from $D$ for every permutation $\sigma : [n] \to [n]$, with probability more than $\frac{1}{3}$.*

*However, if the distribution $D$ is not clusterable, the algorithm may output* FAIL *with any probability.*

**Chapter 9: Testing bounded VC-dimension Properties** In this chapter, we show that any index-invariant distribution property whose VC-dimension is bounded, has a tester whose query complexity is independent of the dimension of the underlying Hamming cube, and depends only on the VC-dimension. Our result is stated as follows:

**Theorem 1.7 (Informal).** *For any fixed constant $d \in \mathbb{N}$, given sample and query access to an unknown distribution $D$ over $\{0,1\}^n$ and a proximity parameter $\varepsilon > 0$, there exists an algorithm that makes $\mathrm{poly}(\frac{1}{\varepsilon})$ queries* [2]*, and either outputs the full description of a distribution or* FAIL *satisfying the following conditions:*

*(i) If the support of $D$ is of VC-dimension at most $d$, then with probability at least $2/3$, the algorithm outputs a full description of a distribution $D'$ such that $D$ is $\varepsilon$-close to $D'_\sigma$ for some permutation $\sigma : [n] \to [n]$.*

*(ii) For any $D$, the algorithm will not output a distribution $D'$ such that $D'_\sigma$ is $\varepsilon$-far from $D$ for all permutations $\sigma : [n] \to [n]$, with probability more than $1/3$. However, if the VC-dimension of the support of $D$ is more than $d$, the algorithm may output* FAIL *with any probability.*

Note that the above theorem corresponds to the learnability of any distribution when the VC-dimension of its support is bounded. As a corollary, it implies that any index-invariant distribution property admitting a global VC-dimension bound is testable with a constant number of queries, depending only on the proximity parameter $\varepsilon$ and the VC-dimension $d$. The corollary is stated as follows:

**Corollary 1.8 (Informal).** *Let $\mathcal{P}$ be an index-invariant property such that any distribution $D \in \mathcal{P}$ has VC-dimension at most $d$, where $d$ is some constant. There exists an algorithm, that has sample and query access to an unknown distribution $D$ over $\{0,1\}^n$,*

[2] The degree of the polynomial in $1/\varepsilon$ depends on the parameter $d$.

*takes a proximity parameter $\varepsilon > 0$, and distinguishes whether $D \in \mathcal{P}$ or $D$ is $\varepsilon$-far from $\mathcal{P}$ with probability at least $2/3$, by making only* $\mathrm{poly}(\frac{1}{\varepsilon})$ *queries.*

It turns out that our tester for testing VC-dimension property takes $\exp(d)$ samples, and performs $\exp(\exp(d))$ queries for VC-dimension $d$. We show that this bound is tight, in the sense that there exists an index-invariant property with VC-dimension $d$ such that any tester for the property requires an exponential number of samples and a doubly-exponential number of queries on $d$.

**Theorem 1.9 (Informal).** *Let $d, n \in \mathbb{N}$. There exists an index-invariant property $\mathcal{P}_{\mathsf{vc}}$ with VC-dimension at most $d$ such that any (non-adaptive) tester for $\mathcal{P}_{\mathsf{vc}}$ requires $2^{\Omega(d)}$ samples and $2^{2^{d-\mathcal{O}(1)}}$ queries.*

**Chapter 10: Role of adaptivity for general properties:** We also study another important feature of testers: the notions of adaptivity and non-adaptivity. Informally, *non-adaptive* testers are testers that perform all the queries to the input together, and depending upon the answers to its queries, decide of accepting or rejecting the input. This is in contrast to the notion of *adaptive* testing, where the tester performs a set of queries, and based upon the answers, performs the second set of queries, and so on. It is clear that adaptive testers are at least as powerful as non-adaptive testers, and the goal is to understand their relative powers.

In the standard model of distribution testing, since the model is inherently non-adaptive, there is essentially no gap between adaptive and non-adaptive testers. However, in the context of testing properties of dense graphs, it is well known that the complexity of non-adaptive testers can be at most quadratic compared to the adaptive testers for any property, which is also tight [GT03, GW21]. For graphs with bounded-degree, for some properties like bipartiteness, this gap is constant vs. $\Omega(\sqrt{n})$, where $n$ denotes the number of vertices of the graph [GR97].

Thus it is natural to study the relative powers of adaptive and non-adaptive testers in the huge object model [GR22]. In our work [CFG$^+$23], we show that for general properties, there is a tight exponential separation between the query complexities of non-adaptive and adaptive testers. The results are as follows:

**Theorem 1.10 (Informal).** *For any non-index-invariant property $\mathcal{P}$, there is at most an exponential gap between the query complexities of adaptive and non-adaptive testers.*

**Theorem 1.11 (Informal).** *There exists a property of distributions over strings that can be $\varepsilon$-tested adaptively using $\mathcal{O}(\log n)$ queries for any $\varepsilon \in (0, 1)$, but $\Omega(\sqrt{n})$ queries are necessary for any non-adaptive algorithm to $\varepsilon$-test it for some $\varepsilon \in (0, 1)$.*

**Chapter 11: Power of adaptivity for index-invariant properties** In contrast to non-index-invariant properties, for *index-invariant* properties defined before, we prove that there is at most a quadratic gap between the query complexities of adaptive and non-adaptive testers, as follows:

**Theorem 1.12 (Informal).** *For any index-invariant property $\mathcal{P}$, there can be at most a quadratic gap between the query complexities of adaptive and non-adaptive testers.*

We also prove that the above gap is almost tight, in the sense that there exists an index-invariant property that can be $\varepsilon$-tested using $\widetilde{\mathcal{O}}(n)$ adaptive queries, while $\widetilde{\Omega}(n^2)$ non-adaptive queries are required to $\varepsilon$-test it.

**Theorem 1.13 (Informal).** *There exists an index-invariant property $\mathcal{P}_{\text{Gap}}$ that can be $\varepsilon$-tested adaptively using $\widetilde{\mathcal{O}}(n)$ queries for any $\varepsilon \in (0, 1)$, while there exists an $\varepsilon \in (0, 1)$ for which $\widetilde{\Omega}(n^2)$ queries are necessary for any non-adaptive $\varepsilon$-tester.*

Theorem 1.6 is formally stated and proved in Chapter 8. Later in Chapter 9, we present the proofs of Theorem 1.7, Corollary 1.8 and Theorem 1.9. In Chapter 10, we prove Theorem 1.10 and Theorem 1.11. Finally, in Chapter 11, we discuss the proofs of Theorem 1.12 and Theorem 1.13.

**This part is based on the following paper:**

1. Sourav Chakraborty, Eldar Fischer, Arijit Ghosh, Gopinath Mishra, and Sayantan Sen, Testing of Index-Invariant Properties in the Huge Object Model, in Proceedings of the $36^{th}$ Conference on Learning Theory (COLT) 2023, Volume 195,

pages 3065–3136, url: https://proceedings.mlr.press/v195/chakraborty23a.html. Featured in Oded Goldreich's Choices (https://www.wisdom.weizmann.ac.il/õded /MC/335.html).

### 1.2.3 Part III: Results in the Adjacency Matrix Model

When the unknown huge object is a graph $G$, the problem of distinguishing if $G$ is "close" or "far" from a property $\mathcal{P}$ is called *graph property testing*. When $G$ is a dense graph, it is stored as an adjacency matrix, and the tester can query any entry of the adjacency matrix. Similar to distribution testing, the goal here is to design testers that perform as few queries as possible to decide $\mathcal{P}$ with high probability. The minimum number of queries required to test a property $\mathcal{P}$ is called the **Query Complexity** of testing $\mathcal{P}$.

The field of graph property testing was first introduced in the seminal work of Goldreich, Goldwasser, and Ron [GGR98]. Since then there has been a flurry of interesting works. Below we give an outline of our results in this part.

**Chapter 13 & Chapter 14: Tolerant Graph Isomorphism Testing:** Graph isomorphism has been one of the most celebrated problems in computer science. Roughly speaking, the graph isomorphism problem asks whether two graphs are structure preserving. One central open problem in complexity theory is whether the graph isomorphism problem can be solved in polynomial time. Recently in a breakthrough result, Babai [Bab16] proved that graph isomorphism problem can be decided in quasi-polynomial time. For a central problem like graph isomorphism, naturally, its (and related problems) computational complexity for various models of computation (see the Dagstuhl Report [BDST15]).

For two graphs $G$ and $H$, their *graph isomorphism distance* denotes the fraction of entries that need to be changed in the adjacency matrix of $G$ to make $G$ and $H$ isomorphic. The problem of non-tolerant testing of graph isomorphism was first studied by Fischer and Matsliah [FM08], and they gave tight bounds for several settings.

In a recent work [CGMS21], we studied the *tolerant graph isomorphism testing* problem. Here the goal is to distinguish whether $G$ and $H$ are "close" or "far" from being isomorphic, by performing as few queries as possible. We proved that tolerant graph isomorphism testing is equivalent to the problem of estimating the Earth Mover Distance of two distributions, constructed from the two graphs.

**Theorem 1.14.** *Let $G_k$ and $G_u$ denote the known and the unknown graphs on $n$ vertices, respectively, and $Q_{GI}(G_u, G_k)$ denotes the number of adjacency queries to $G_u$, required by the best algorithm that takes two constants $\gamma_1, \gamma_2$ with $0 \leq \gamma_1 < \gamma_2 \leq 1$ and decides whether $d(G_u, G_k) \leq \gamma_1 n^2$ or $d(G_u, G_k) \geq \gamma_2 n^2$ with probability at least $2/3$. Then*

$$Q_{GI}(G_u, G_k) = \widetilde{\Theta}\left(\textsc{QWoR}_{\text{EMD}}(n)\right)$$

*where $\widetilde{\Theta}(\cdot)$ hides polynomial factors in $\frac{1}{\gamma_2 - \gamma_1}$ and $\log n$.*

In fact, our equivalence proof is model-independent, in the sense that the equivalence also holds for other models, like the communication complexity model. We prove the lower bound of Theorem 1.14 (tolerant GI testing is as hard as tolerant EMD testing) in Chapter 13 and the upper bound of Theorem 1.14 (tolerant EMD testing is as hard as tolerant GI testing) in Chapter 14.

**Chapter 15: Tolerant Bipartiteness Testing in Dense Graphs** The problem of testing whether a graph $G$ is *bipartite* or not has been one of the fundamental problems in computer science. Naturally, it has also been studied in the query complexity framework. In the seminal work of Goldreich, Goldwasser, and Ron [GGR98] that started the field of graph property testing, the authors designed a tester for non-tolerant variant of this problem. The query complexity of their tester is independent of the number of vertices of the graph and depends only on the proximity parameter. The tolerant variant of bipartiteness testing was studied by Alon, Vega, Kannan and Karpinski [AdlVKK03], where they studied the more general problem of estimating the size of the maximum cut of the graph. Note that any tester for maximum cut translate to a tester for tolerant bipartiteness testing.

In a recent work [GMRS22], we designed a tester for tolerant bipartiteness testing, whose sample and query complexities are better compared to [AdlVKK03]. Moreover, the running time of our tester is significantly improved from prior works.

**Theorem 1.15.** *There exists an algorithm* $\textsc{Tol-Bip-Dist}(G, \varepsilon)$ *that given adjacency query access to a dense graph G with* $n$ *vertices and a parameter* $\varepsilon \in (0, 1)$*, decides with probability at least* $\frac{9}{10}$*, whether* $d_{bip}(G) \leq \varepsilon n^2$ *or* $d_{bip}(G) \geq (2 + k)\varepsilon n^2$*, by sampling* $\mathcal{O}(\frac{1}{k^5\varepsilon^2} \log \frac{1}{k\varepsilon})$ *vertices in* $2^{\mathcal{O}(\frac{1}{k^3\varepsilon} \log \frac{1}{k\varepsilon})}$ *time, using* $\mathcal{O}(\frac{1}{k^8\varepsilon^3} \log^2 \frac{1}{k\varepsilon})$ *queries to the adjacency matrix of G, where* $d_{bip}(G)$ *denotes the distance of G from being bipartite.*

We will prove Theorem 1.14 in Chapter 13 Chapter 14, where in Chapter 13, we prove the lower bound part of Theorem 1.14 and then we prove the upper bound part of Theorem 1.14 in Chapter 14. Finally we prove Theorem 15 in Chapter 15.

**This part is based on the following papers:**

1. Sourav Chakraborty, Arijit Ghosh, Gopinath Mishra and Sayantan Sen. Interplay between Graph Isomorphism and Earth Mover's Distance in the Query and Communication Worlds, In Proceedings of the $25^{th}$ International Conference on Randomization and Computation (RANDOM), 2021, Volume 207, 34:1-34:23, doi: 10.4230/LIPIcs.APPROX/RANDOM.2021.34. Presented in Highlights of Algorithms (HALG), 2022. Submitted to the journal ACM Transactions on Computation Theory (TOCT).

2. Arijit Ghosh, Gopinath Mishra, Rahul Raychaudhury, and Sayantan Sen, Tolerant Bipartiteness Testing in Dense Graphs, In Proceedings of the $49^{th}$ International Colloquium on Automata, Languages and Programming (ICALP), 2022, Volume 229, 69:1-69:19, doi: 10.4230/LIPIcs.ICALP.2022.69. Presented in Highlights of Algorithms (HALG), 2023. Submitted to the journal Combinatorics, Probability and Computing (CPC).

# Chapter 2

# Preliminaries

A probability distribution $D$ over a universe $\Omega = [n]$ is a non-negative function $D : \Omega \to [0, 1]$ such that $\sum_{i \in \Omega} D(i) = 1$. For $S \subseteq \Omega$, the mass of $S$ is defined as $D(S) = \sum_{i \in S} D(i)$, where $D(i)$ is the mass of $i$ in $D$. The support of a probability distribution $D$ on $\Omega$ is denoted by $\textsc{Supp}(D)$. For any distribution $D$, by the top $t$ elements of $D$, we refer to the first $t$ elements in the support of $D$ when the elements in the support are sorted according to a non-increasing order of their probability masses in $D$. When we write $\widetilde{\mathcal{O}}(\cdot)$, it sometimes suppresses a poly-logarithmic term in $n$ and the inverse of the proximity parameter(s), as well as the inverse of the difference of two proximity parameters.

For an integer $n$, we will denote the set $\{1, \ldots, n\}$ as $[n]$. Given two vectors $\mathbf{X}$ and $\mathbf{Y}$ in $\{0, 1\}^n$, we will denote by $d_H(\mathbf{X}, \mathbf{Y})$ the normalized Hamming distance between $\mathbf{X}$ and $\mathbf{Y}$, that is,

$$d_H(\mathbf{X}, \mathbf{Y}) := \frac{|\{i \in [n] \, : \, \mathbf{X}_i \neq \mathbf{Y}_i\}|}{n}.$$

Unless stated otherwise, all the distance measures that we will be considering in this thesis will be the normalized distances. For two vectors $\mathbf{X}, \mathbf{Y} \in \{0, 1\}^n$, $\delta_H(\mathbf{X}, \mathbf{Y}) = n \cdot d_H(\mathbf{X}, \mathbf{Y})$ will be used to denote the absolute Hamming distance between $\mathbf{X}$ and $\mathbf{Y}$ in the few places where we will need to refer to it.

## 2.1 Various distance measures of distributions

We will first define $\ell_1$ distance between two distributions.

**Definition 2.1** (**$\ell_1$ distance and variation distance between two distributions).** Let $D_1$ and $D_2$ be two probability distributions over a set $S$. The $\ell_1$ distance between $D_1$ and $D_2$ is defined as

$$||D_1 - D_2||_1 = \sum_{a \in S} |D_1(a) - D_2(a)|.$$

The variation distance between $D_1$ and $D_2$ is defined as:

$$d_{TV}(D_1, D_2) = \frac{1}{2} \cdot ||D_1 - D_2||_1.$$

Throughout this thesis, the Earth Mover Distance (EMD) is the central metric for testing "closeness" and "farness" of a distribution from a given property. It is formally defined below.

**Definition 2.2 (Earth Mover Distance** (EMD)**).** Let $D_1$ and $D_2$ be two probability distributions over $\{0,1\}^n$. The EMD between $D_1$ and $D_2$ is denoted by $d_{EM}(D_1, D_2)$, and defined as the solution to the following linear program:

$$\begin{aligned}
\text{Minimize} \quad & \sum_{\mathbf{X},\mathbf{Y} \in \{0,1\}^n} f_{\mathbf{XY}} d_H(\mathbf{X}, \mathbf{Y}) & \\
\text{Subject to} \quad & \sum_{\mathbf{Y} \in \{0,1\}^n} f_{\mathbf{XY}} = D_1(\mathbf{X}), & \forall\, \mathbf{X} \in \{0,1\}^n \\
& \sum_{\mathbf{X} \in \{0,1\}^n} f_{\mathbf{XY}} = D_2(\mathbf{Y}), & \forall\, \mathbf{Y} \in \{0,1\}^n \\
& 0 \leq f_{\mathbf{XY}} \leq 1, & \forall \mathbf{X}, \mathbf{Y} \in \{0,1\}^n
\end{aligned}$$

Intuitively, the variable $f_{\mathbf{XY}}$ stands for the amount of probability mass transferred from $\mathbf{X}$ to $\mathbf{Y}$.

Directly from the definitions of $d_{EM}(D_1, D_2)$ and $d_H(\mathbf{X}, \mathbf{Y})$, we get the following simple yet useful observation connecting $\ell_1$ distance and EMD between two distributions.

**Observation 2.3 (Relation between EMD and $\ell_1$ distance).** Let $D_1$ and $D_2$ be two distributions over the $n$-dimensional Hamming cube $\{0,1\}^n$. Then we have the following relation between the Earth Mover Distance and $\ell_1$ distance between $D_1$ and $D_2$:

$$d_{EM}(D_1, D_2) \leq \frac{||D_1 - D_2||_1}{2}.$$

Now we formally define the notions of "closeness" and "farness" of two distributions with respect to the Earth Mover Distance.

**Definition 2.4 (Closeness and farness with respect to** EMD**).** Given two proximity parameters $\varepsilon_1$ and $\varepsilon_2$ with $0 \leq \varepsilon_1 < \varepsilon_2 \leq 1$, two distributions $D_1$ and $D_2$ over the $n$-dimensional Hamming cube $\{0,1\}^n$ are said to be $\varepsilon_1$-*close* if $d_{EM}(D_1, D_2) \leq \varepsilon_1$, and $\varepsilon_2$-*far* if $d_{EM}(D_1, D_2) \geq \varepsilon_2$.

Now we proceed to define the notion of distribution properties over the Hamming cube below.

**Definition 2.5 (Distribution property over the Hamming cube).** Let $\mathcal{D}$ denote the set of all distributions over the $n$-dimensional Hamming cube $\{0,1\}^n$. A *distribution property* $\mathcal{P}$ is a topologically closed subset of $\mathcal{D}$. [1] A distribution $D \in \mathcal{P}$ is said to be *in the property* or to *satisfy* the property. Any other distribution is said to be *not in the property* or *to not satisfy* the property.

Now we are now ready to define the notion of distance of a distribution from a property.

**Definition 2.6 (Distance of a distribution from a property).** The distance of a distribution $D$ from a property $\mathcal{P}$ is the minimum Earth Mover Distance between $D$ and any distribution in $\mathcal{P}$. [2] For $\varepsilon \in [0,1]$, a distribution $D$ is said to be $\varepsilon$-*close* to $\mathcal{P}$ if the

[1] We put this restriction to avoid formalism issues. In particular, the investigated distribution properties that we know of (such as monotonicity and being a k-histogram) are topologically closed.

[2] The assumption that $\mathcal{P}$ is closed indeed makes it a minimum rather than an infimum.

distance of $D$ from $\mathcal{P}$ is at most $\varepsilon$. Analogously, for $\varepsilon \in [0, 1]$, a distribution $D$ is said to be *$\varepsilon$-far* from $\mathcal{P}$ if the distance of $D$ from $\mathcal{P}$ is more than $\varepsilon$.

## 2.2 Formal definitions of various kinds of property testers

First we let us discuss the query procedure.

**Definition 2.7 (Query to sampled vectors).** Let $\mathcal{A}$ be a tester with a set of sampled vectors $\mathbf{V}_1, \ldots, \mathbf{V}_s$, drawn independently from an input distribution $D$ over $\{0, 1\}^n$, where $\mathbf{V}_i = (v_{i,1}, \ldots, v_{i,n})$ for every $i \in [s]$. In order to perform a query, the tester will provide $i$ and $j$, and will receive $v_{i,j}$ as the answer to the query.

In the following, we formally describe the notion of a tester.

**Definition 2.8 ($\varepsilon$-test).** Let $\varepsilon \in (0, 1)$ be a proximity parameter, and $\delta \in (0, 1)$. A probabilistic algorithm $\mathcal{A}$ is said to *$\varepsilon$-test* a property $\mathcal{P}$ with probability at least $1 - \delta$, if any input in $\mathcal{P}$ is accepted by $\mathcal{A}$ with probability at least $1 - \delta$, and any input that is $\varepsilon$-far from $\mathcal{P}$ is rejected by $\mathcal{A}$ with probability at least $1 - \delta$. Unless explicitly stated, we assume that $\delta = 1/3$.

Now we define two different types of testers, *adaptive* testers and *non-adaptive* testers, which will be used throughout the thesis. We begin by describing the adaptive testers. Informally, adaptive testers correspond to algorithms that perform queries depending on the answers to previous queries. Formally:

**Definition 2.9 (Adaptive tester).** Let $\mathcal{P}$ be a property over $\{0, 1\}^n$. An adaptive tester for $\mathcal{P}$ with query complexity $q$ and sample complexity $s$ is a randomized algorithm $\mathcal{A}$ that $\varepsilon$-tests $\mathcal{P}$ by performing the following:

- $\mathcal{A}$ first draws some random coins and samples $s$ vectors from the unknown distribution $D$, denoted by $S = \{\mathbf{V}_1, \ldots, \mathbf{V}_s\}$.
- $\mathcal{A}$ then queries the $j_1$-th index of $\mathbf{V}_{i_1}$, for some $j_1 \in [n]$ and $i_1 \in [s]$ depending on the random coins.

- Suppose that $\mathcal{A}$ has executed $k$ steps and has queried the $j_\ell$-th index of $\mathbf{V}_{j_\ell}$, where $1 \leq \ell \leq k$. At the $(k+1)$-th step, depending upon the random coins and the answers to the queries till the $k$-th step, $\mathcal{A}$ will perform a query for the $j_{k+1}$-th bit of $\mathbf{V}_{i_{k+1}}$, where $j_{k+1} \in [n]$ and $i_{k+1} \in S$.
- After $q$ steps, $\mathcal{A}$ reports ACCEPT or REJECT depending on the random coins and the answers to all $q$ queries.

Now we define the more restricted *non-adaptive testers*. Informally, non-adaptive testers decide the set of queries to be performed on the input even before performing the first query. Formally:

**Definition 2.10 (Non-adaptive tester).** Let $\mathcal{P}$ be a property over $\{0,1\}^n$. A non-adaptive tester for $\mathcal{P}$ with query complexity $q$ and sample complexity $s$ is a randomized algorithm $\mathcal{A}$ that $\varepsilon$-tests $\mathcal{P}$ by performing the following:

- $\mathcal{A}$ tosses some random coins, and depending on the answers constructs a sequence of subsets of indices $J_1, \ldots, J_s \subset [n]$ such that $\sum\limits_{i=1}^{s} J_i \leq q$.
- $\mathcal{A}$ takes $s$ samples $\mathbf{V}_1, \ldots, \mathbf{V}_s$ from the unknown distribution $D$.
- $\mathcal{A}$ queries for the coordinates of $\mathbf{V}_i$ that are in $J_i$, for each $i \in [s]$.
- $\mathcal{A}$ reports either ACCEPT or REJECT based on the answers from the queries to the vectors, that is, $\mathbf{V}_1 \mid_{J_1}, \mathbf{V}_2 \mid_{J_2}, \ldots, \mathbf{V}_s \mid_{J_s}$, and the random coins.

## 2.3 Some probability results

Now we state some probability results used in this thesis.

**Lemma 2.11 (Multiplicative Chernoff bound** [DP09]**).** *Let $X_1, \ldots, X_n$ be independent random variables such that $X_i \in [0,1]$. For $X = \sum\limits_{i=1}^{n} X_i$ and $\mu = \mathbb{E}[X]$, the following holds for any $0 \leq \delta \leq 1$.*

$$\mathbb{P}(|X-\mu| \geq \delta\mu) \leq 2\exp\left(-\mu\delta^2/3\right).$$

**Lemma 2.12 (Additive Chernoff bound** [DP09]**).** *Let $X_1, \ldots, X_n$ be independent random variables such that $X_i \in [0,1]$. For $X = \sum\limits_{i=1}^{n} X_i$ and $\mu_l \leq \mathbb{E}[X] \leq \mu_h$, the following hold for any $\delta > 0$.*

**(i)** $\mathbb{P}\left(X \geq \mu_h + \delta\right) \leq \exp\left(-2\delta^2/n\right)$.

**(ii)** $\mathbb{P}\left(X \leq \mu_l - \delta\right) \leq \exp\left(-2\delta^2/n\right)$.

**Lemma 2.13 (Chernoff-Hoeffding bound [DP09]).** *Let $X_1, \ldots, X_n$ be independent random variables such that $X_i \in [0,1]$. For $X = \sum\limits_{i=1}^{n} X_i$ and $\mu_l \leq \mathbb{E}[X] \leq \mu_h$, the followings hold for any $0 < \varepsilon < 1$:*

**(i)** $\mathbb{P}\left(X \geq (1+\varepsilon)\mu_h\right) \leq \exp\left(\frac{-\varepsilon^2\mu_h}{3}\right)$.

**(ii)** $\mathbb{P}\left(X \leq (1-\varepsilon)\mu_l\right) \leq \exp\left(\frac{-\varepsilon^2\mu_l}{3}\right)$.

**Lemma 2.14 (Hoeffding's Inequality** [DP09]**).** *Let $X_1, \ldots, X_n$ be independent random variables such that $a_i \leq X_i \leq b_i$ and $X = \sum\limits_{i=1}^{n} X_i$. Then, for all $\delta > 0$,*

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq \delta\right) \leq 2\exp\left(-2\delta^2/\sum_{i=1}^{n}(b_i - a_i)^2\right).$$

**Lemma 2.15 (Hoeffding's Inequality for sampling without replacement** [Hoe94]**).** *Let $n$ and $m$ be two integers such that $1 \leq n \leq m$, and $x_1, \ldots, x_m$ be real numbers, with $a \leq x_i \leq b$ for every $i \in [m]$. Suppose that $I$ is a set that is drawn uniformly from all subsets of $[m]$ of size $n$, and let $X = \sum\limits_{i \in I} x_i$. Then, for all $\delta > 0$,*

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq \delta\right) \leq 2\exp\left(-2\delta^2/n \cdot (b-a)^2\right).$$

Now let us consider the following observation which states that if the normalized Hamming distance between two vectors $\mathbf{X}$ and $\mathbf{Y}$ are small, the same also holds with high probability when $\mathbf{X}$ and $\mathbf{Y}$ are projected on a set of random indices $K$. A similar result also holds when the distance is large between the two vectors $\mathbf{X}$ and $\mathbf{Y}$.

**Observation 2.16 (Approximating-string-distances).** For $\mathbf{U}, \mathbf{V} \in \{0,1\}^n$ and assume that $K \subseteq [n]$ is a set of indices chosen uniformly at random without replacement. Then the following holds with probability at least $1 - e^{-\mathcal{O}(\delta^2|K|)}$:

$$|d_H(\mathbf{U}, \mathbf{V}) - d_H(\mathbf{U}\mid_K, \mathbf{V}\mid_K)| \leq \delta.$$

*Proof.* Follows from the fact that sampling without replacement is as good as sampling with replacement (Lemma 2.15). □

**Lemma 2.17 (Chernoff bound for bounded dependency** [Jan04]**).** *Let $X_1, \ldots, X_n$ be random variables such that $a_i \leq X_i \leq b_i$ and $X = \sum\limits_{i=1}^{n} X_i$. Let $\mathcal{D}$ be the (directed)* dependency *graph, where $V(\mathcal{D}) = \{X_1, \ldots, X_n\}$ and $X_i$ is completely independent of all variables $X_j$ for which $(X_i, X_j)$ is not a directed edge. Then for any $\delta > 0$,*

$$\mathbb{P}(|X - \mathbb{E}[X]| \geq \delta) \leq 2e^{-2\delta^2/\chi^*(\mathcal{D})\sum\limits_{i=1}^{n}(b_i - a_i)^2}.$$

*where $\chi^*(\mathcal{D})$ denotes the* fractional chromatic number *of $\mathcal{D}$.*

**Corollary 2.18 (Corollary of Lemma 2.17).** *Let $X_1, \ldots, X_n$ be indicator random variables such that the dependency graph is a disjoint union of $n/k$ many $k$ size cliques. For $X = \sum\limits_{i=1}^{n} X_i$ and $\mu_l \leq \mathbb{E}[X] \leq \mu_h$, the followings hold for any $\delta > 0$:*

**(i)** $\mathbb{P}\left(X \geq \mu_h + \delta\right) \leq \exp\left(\frac{-2\delta^2}{kn}\right)$,

**(ii)** $\mathbb{P}\left(X \leq \mu_\ell - \delta\right) \leq \exp\left(\frac{-2\delta^2}{kn}\right)$.

*Proof.* Follows from the fact that the dependency graph has chromatic number $k$, and the fractional chromatic number of a graph is at most the chromatic number of any graph. □

# Part I

# Results in the Sampling Model

# Chapter 3

# Testing in the Sampling Model

## 3.1 Introduction

Let $D$ be a distribution over a finite set $\Omega$, and $\mathcal{P}$ be a property, that is, a set of distributions over $\Omega$. Given access to independent random samples from $\Omega$ according to the distribution $D$, we are interested in the problem of distinguishing whether the distribution $D$ is $\eta$-close to having the property $\mathcal{P}$, or is $\varepsilon$-far from having the property $\mathcal{P}$, where $\eta$ and $\varepsilon$ are two fixed proximity parameters such that $0 \leq \eta < \varepsilon \leq 2$. The distance of the distribution $D$ from the property $\mathcal{P}$ is defined as $\min\limits_{D' \in \mathcal{P}} ||D - D'||_1$, where $||D - D'||_1$ denotes the $\ell_1$-*distance* between the distributions $D$ and $D'$ [1]. A distribution $D$ is said to be $\eta$-close to $\mathcal{P}$ if the distance of $D$ from $\mathcal{P}$ is at most $\eta$. Similarly, $D$ is said to be $\varepsilon$-far from $\mathcal{P}$, if the distance of $D$ from $\mathcal{P}$ is at least $\varepsilon$. The goal is to design a tester that uses as few samples as possible. For $\eta > 0$, the problem of distinguishing the two cases is referred to as the *tolerant distribution testing* problem of $\mathcal{P}$, and the particular case where $\eta = 0$ is referred to as the *non-tolerant distribution testing* problem of $\mathcal{P}$. The sample complexity (tolerant and non-tolerant testing) is the number of samples required by the best algorithm that can distinguish with high probability (usually with probability at least $\frac{2}{3}$) whether the distribution $D$ is $\eta$-close to having the property $\mathcal{P}$, or is $\varepsilon$-far from

[1] Strictly speaking it is an infimum, but since all properties we consider are compact sets, it is equal to the minimum.

having the property $\mathcal{P}$.

While results and techniques from distribution testing are already interesting in their own right, they have also found numerous applications in central problems in Theoretical Computer Science, and in particular in property testing, e.g. graph isomorphism testing [FM08, Gol19] and function isomorphism testing [ABC$^+$13], learning theory [BBC$^+$10, DKS17, DK16], and differential privacy [ADKR19, GKK$^+$20, Zha21, ACF$^+$21]. Thus, understanding the tolerant and non-tolerant sample complexity of distribution testing is a central problem in theoretical computer science.

There have been extensive studies of non-tolerant and tolerant testing of some specific distribution properties like uniformity, identity with a fixed distribution, equality of two distributions and independence of a joint distribution [BFR$^+$00, BFF$^+$01, Pan08, Val11, VV11, VV17a]. Various other specific distribution properties have also been studied [BC17, DKS18]. Then, some works investigated general tests for the large class of all shape-restricted properties of distributions, which contains properties like monotonicity, log-concavity, modality etc. [CDGR18, FLV17]. This work proves general results about the gap between tolerant and non-tolerant distribution testing that hold for large classes of properties.

## 3.2 Our results

We now informally present our results. The formal definitions are presented in Section 3.4. We assume that distributions are supported over a set $\Omega = [n] = \{1, 2, \ldots, n\}$. We first prove a result about label-invariant distribution properties (properties that are invariant under all permutations of $\Omega$). We show that, for any label-invariant distribution property, there is at most a quadratic blowup in its tolerant sample complexity as compared to its non-tolerant counterpart, ignoring poly-logarithmic factors.

**Theorem 3.1 (Informal).** *Any label-invariant distribution property that can be non-tolerantly tested using $\Lambda$ samples, can also be tolerantly tested using $\widetilde{\mathcal{O}}(\min\{\Lambda^2, n\})$*

*samples, where $n$ is the size of the support of the distribution* [2].

This result gives a unified way for obtaining tolerant testers from their non-tolerant counterparts. The above result will be stated and proved formally in Section 4.2. Moreover, in Section 4.3, we give a constructive variant of the tolerant tester of Theorem 3.1, when the property can be expressed as the feasible solution to a set of linear inequalities.

**Theorem 3.2 (Informal).** *Any label-invariant distribution property that can be non-tolerantly tested using $\Lambda$ samples and can be expressed as a feasible solution to $m$ linear inequalities, can also be tolerantly tested using $\widetilde{\mathcal{O}}(\min\{\Lambda^2, n\})$ samples and in time polynomial in $m$ and $n$, where $n$ is the size of the support of the distribution.*

Note that if $\Lambda = \Omega(\sqrt{n})$, Theorem 3.1 is obvious. It is only interesting if $\Lambda = o(\sqrt{n})$. Now we present a property for which this connection is useful. Consider a natural distribution property: given a distribution $D$ and a parameter $k$, we want to decide whether the support size of $D$ is at most $k$ or $\varepsilon$-far from having support at most $k$. If $k = o(\sqrt{n})$, the query complexity for testing this problem is $\mathcal{O}(\frac{k}{\log k})$ [VV17b].

It is a natural question to investigate the extent to which the above theorem can be generalized. Though we are not resolving this question completely, as a first step in the direction of extending the above theorem for properties that are not necessarily label-invariant, we consider the notion of *non-concentrated* properties. By the notion of a non-concentrated distribution, intuitively, we mean that there is no significant portion of the base set of the distribution that carries only a negligible weight, making the probability mass of the distribution well distributed among its indices. Specifically, any subset $X \subseteq [n]$, for which $|X|$ is above some threshold (say $\beta n$ with $\beta \in (0, \frac{1}{2})$), has probability mass of at least another threshold (say $\alpha$ with $\alpha \in (0, \frac{1}{2})$). A property is said to be non-concentrated if only non-concentrated distributions can satisfy the property. We prove a lower bound on the testing of any non-concentrated property (not necessarily label-invariant).

[2] $\widetilde{\mathcal{O}}(\cdot)$ hides a poly-logarithmic factor.

**Theorem 3.3 (Informal).** *In order to non-tolerantly test any non-concentrated distribution property, $\Omega(\sqrt{n})$ samples are required, where $n$ is the size of the support of the distribution.*

The quadratic gap between tolerant testing and non-tolerant testing for any non-concentrated property follows from the above theorem, since by a folklore result, only $\mathcal{O}(n)$ samples are required to learn any distribution approximately.

The proof of Theorem 3.3 for label-invariant non-concentrated properties is a generalization of the proof of the $\Omega(\sqrt{n})$ lower bound for classical uniformity testing, while for the whole theorem, that is, for the general (not label-invariant) non-concentrated properties, a more delicate argument is required. The formal proof is presented in Section 5.3.

The next natural question is about the sample complexity of any tolerant tester for non-concentrated properties. We address this question for properties which are label-invariant non-concentrated by proving the following theorem in Section 5.2.2. However, the question is left open for non-label-invariant properties.

**Theorem 3.4 (Informal).** *The sample complexity of tolerant testing of any non concentrated label-invariant distribution property is $\Omega(n^{1-o(1)})$, where $n$ is the size of the support of the distribution.*

A natural question related to tolerant testing is:

*How many samples are required to learn a distribution?*

As pointed out earlier, any distribution can be learnt using $\mathcal{O}(n)$ samples. But what if the distribution happens to be *very concentrated*? We present an upper bound result for learning a distribution, in which the sample complexity depends on the minimum cardinality of any set $S \subseteq [n]$ over which the unknown distribution is concentrated.

**Theorem 3.5 (Informal).** *To learn a distribution approximately, $\mathcal{O}(|S|)$ samples are enough, where $S \subseteq [n]$ is an unknown set of minimum cardinality whose mass is close to* 1*. Note that $|S|$ is also unknown, and the algorithm adapts to it.*

Observe that we cannot learn a distribution supported on the set $S$ using $o(|S|)$ samples, so the above result is essentially tight.

## 3.3 Related works

Several forms of distribution testing have been investigated for over a century in statistical theory [Kin97, CF14], while combinatorial properties of distributions have been explored over the last two decades in Algorithm Theory, Machine Learning and Information Theory [Gol17, Mac03, CT01, BY22]. In Algorithm Theory, the investigation into testing properties of distributions started with the work of Goldreich and Ron [GR00], even though it was not directly stated there in these terms. Batu, Fortnow, Rubinfeld, Smith and White [BFR$^+$00] launched the intensive study of property testing of distributions with the problem of equivalence testing [3]. Later, Batu, Fischer, Fortnow, Kumar, Rubinfeld and White [BFF$^+$01] studied the problems of identity and independence testing of distributions [4]. Since then there has been a flurry of interesting works in this model. For example, Paninski [Pan08] proved tight bounds on uniformity testing, Valiant and Valiant [VV11] resolved the tolerant sample complexity for a large class of label-invariant properties that includes uniformity testing, Acharya, Daskalakis and Kamath [ADK15] proved various optimal testing results under several distance measures, and Valiant and Valiant [VV17a] studied the sample complexity of instance optimal identity testing. In [BC17], Batu and Cannone studied the problem of *generalized uniformity testing*, where the distribution is promised to be supported on an unknown set $S$, and proved a tight bound of $\tilde{\Theta}(|S|^{2/3})$ samples for non-tolerant uniformity testing. This is in contrast to the non-tolerant uniformity testing of a distribution supported over $[n]$, whose sample complexity is $\Theta(\sqrt{n})$, ignoring the dependence on the proximity parameter. Daskalakis, Kamath and Wright [DKW18] studied the problem of tolerant

[3] Given two unknown probability distributions that can be accessed via samples from their respective oracles, equivalence testing refers to the problem of distinguishing whether they are same or far from each other.

[4] Given an unknown distribution accessible via samples, the problem of identity testing refers to the problem of distinguishing whether it is identical to a known distribution or far from it.

testing under various distance measures. Very recently, Canonne, Jain, Kamath and Li [CJKL22] revisited the problem of determining the sample complexity of tolerant identity testing, where they proved the optimal dependence on the proximity parameters. Going beyond studying specific properties, Canonne, Diakonikolas, Gouleakis and Rubinfeld [CDGR18] studied the class of *shape-restricted* properties of a distribution, a condition general enough to contain several interesting properties like monotonicity, log-concavity, $t$-modality etc. Their result was later improved by Fischer, Lachish and Vasudev [FLV17]. See the surveys of Cannone [Can20c, Can22] for a more exhaustive list.

While the most studied works concentrate on non-tolerant testing of distributions, a natural extension is to test such properties tolerantly. Since the introduction of tolerant testing in the pioneering work of Parnas, Ron and Rubinfeld [PRR06], that defined this notion for classical (non-distribution) property testing, there have been several works in this framework. Note that it is nontrivial in many cases to construct tolerant testers from their non-tolerant counterparts, as in the case of tolerant junta testing [BCE$^+$19] for example. In a series of works, it has been proved that tolerant testing of the most natural distribution properties, like uniformity, requires an almost linear number of samples [Val11, VV11] [5]. Now a natural question arises about how the sampling complexity of tolerant testing is related to non-tolerant testing of distributions in general. To the best of our knowledge, there is no known example with more than a quadratic gap.

It would also be interesting to bound the gap for sample-based testing as defined in the work of Goldreich and Ron [GR16]. This model was investigated further in the work of Fischer, Lachish and Vasudev [FLV15], where a general upper bound for non-tolerant sample-based testing of strongly testable properties was provided.

[5] To be precise, the exact lower bounds for non-tolerant uniformity testing is $\Omega(\sqrt{n})$, and for tolerant uniformity testing it is $\Omega(\frac{n}{\log n})$, where $n$ is the support size of the distribution and the proximity parameter $\varepsilon$ is constant.

### Organization of the part

Section 3.4 contains the definitions used throughout the chapters in this part of the thesis. In Chapter 4, we present the overviews of the proofs as well as formally state and prove Theorem 3.1 and Theorem 3.2. Later in Chapter 5, we prove Theorem 3.3 and Theorem 3.4. Finally Theorem 3.5 is proved in Chapter 6.

## 3.4 Preliminaries

Here we present some relevant definitions required in this part of the thesis.

**Definition 3.6 (Label-invariant property).** Let us consider a property $\mathcal{P}$. For a distribution $D$ and a permutation $\sigma : \Omega \to \Omega$, consider the distribution $D_\sigma$ defined as $D_\sigma(\sigma(i)) = D(i)$ (equivalently, $D_\sigma(i) = D(\sigma^{-1}(i))$) for each $i \in \Omega$. If for every distribution $D$ in $\mathcal{P}$, $D_\sigma$ is also in $\mathcal{P}$ for every permutation $\sigma$, then the property $\mathcal{P}$ is said to be *label-invariant*.

Although there are several other distance measures, in this part, we mainly focus on the $\ell_1$ distance. Let us recall the following two definitions which will be crucially used here.

**Definition 3.7 (Distance between two distributions).** The distance between two distributions $D_1$ and $D_2$ over $\Omega$ is the standard $\ell_1$ distance between them, which is defined as $||D_1 - D_2||_1 := \sum\limits_{i \in \Omega} |D_1(i) - D_2(i)|$. For $\eta \in [0, 2]$, $D_1$ and $D_2$ are said to be $\eta$-close to each other if $||D_1 - D_2||_1 \leq \eta$. Similarly, for $\varepsilon \in [0, 2]$, $D_1$ and $D_2$ are said to be $\varepsilon$-far from each other if $||D_1 - D_2||_1 \geq \varepsilon$.

**Definition 3.8 (Distance of a distribution from a property).** The distance of a distribution $D$ from a property $\mathcal{P}$ is the minimum $\ell_1$-distance between $D$ and any distribution in $\mathcal{P}$. For $\eta \in [0, 2]$, a distribution $D$ is said to be $\eta$-close to $\mathcal{P}$ if the distance of $D$ from $\mathcal{P}$ is at most $\eta$. Analogously, for $\varepsilon \in [0, 2]$, a distribution $D$ is said to be $\varepsilon$-far from $\mathcal{P}$ if the distance of $D$ from $\mathcal{P}$ is at least $\varepsilon$.

**Definition 3.9** (**$(\eta, \varepsilon)$-tester).** An $(\eta, \varepsilon)$-*tester* for a distribution property is a randomized algorithm that has sample access to the unknown distribution (upon query it can receive elements of $\Omega$, each drawn according to the unknown distribution, independently of any previous query or the algorithm's private coins), and distinguishes whether the distribution is $\eta$-close to the property or $\varepsilon$-far from the property, with probability at least $\frac{2}{3}$, where $\eta$ and $\varepsilon$ are proximity parameters such that $0 \leq \eta < \varepsilon \leq 2$. The tester is said to be *tolerant* when $\eta > 0$, and *non-tolerant* when $\eta = 0$.

Now we define the notions of non-concentrated distributions and non-concentrated properties.

**Definition 3.10 (Non-Concentrated distribution).** A distribution $D$ over the domain $\Omega = [n]$ is said to be $(\alpha, \beta)$-*non-concentrated* if for any set $S \subseteq \Omega$ with size $\beta n$, the probability mass on $S$ is at least $\alpha$, where $\alpha$ and $\beta$ are two parameters such that $0 < \alpha \leq \beta < \frac{1}{2}$.

**Definition 3.11 (Non-Concentrated property).** Let $0 < \alpha \leq \beta < \frac{1}{2}$. A distribution property $\mathcal{P}$ is defined to be $(\alpha, \beta)$-*non-concentrated*, if all distributions in $\mathcal{P}$ are $(\alpha, \beta)$-non-concentrated.

Note that the uniform distribution is $(\alpha, \alpha)$-non-concentrated for every $\alpha$, and so is the property of being identical to the uniform distribution. Also, for any $0 < \alpha < \frac{1}{2}$ such that $\alpha n$ is an integer, the uniform distribution is the only $(\alpha, \alpha)$-non-concentrated one. Finally, observe that any arbitrary distribution is both $(0, \beta)$-non-concentrated and $(\alpha, 1)$-non-concentrated, for any $\alpha, \beta \in (0, 1)$.

## 3.5 Technical overview of our results

In this section, we give an overview of our results as follows:

### 3.5.1 Construction of tolerant testers for label-invariant properties

Here we present an overview of the proofs of Theorem 3.1 and Theorem 3.2. We first show that for any label-invariant distribution property, the sample complexities of tolerant and non-tolerant testing are separated by at most a quadratic factor, ignoring poly-logarithmic terms. More specifically, in Theorem 3.1, we prove that for any label-invariant distribution property $\mathcal{P}$ that has a non-tolerant tester with sample complexity $\Lambda$, there exists a tolerant tester for $\mathcal{P}$ that uses $\widetilde{\mathcal{O}}(\Lambda^2)$ samples, ignoring poly-logarithmic factors. Since we can learn a distribution using $\mathcal{O}(n)$ samples, our proof is particularly useful when $\Lambda = o(\sqrt{n})$, where $n$ is the size of the support of the distribution that is being tested.

To prove Theorem 3.1 (restated as Theorem 4.1), we provide an algorithm for tolerant testing of $\mathcal{P}$ with sample complexity $\widetilde{\mathcal{O}}(\Lambda^2)$, based on the existence of a non-tolerant tester of $\mathcal{P}$ with sample complexity $\mathcal{O}(\Lambda)$. Given the existence of such a non-tolerant tester with sample complexity $\mathcal{O}(\Lambda)$, one crucial observation that we use here is that there cannot be two distributions $D_1$ and $D_2$ that are identical on the elements with mass $\Omega(\frac{1}{\Lambda^2})$ (we call them *high* elements), where $D_1$ is in the property $\mathcal{P}$ while $D_2$ is far from $\mathcal{P}$. This is formally stated as Lemma 4.4.

Given that the two distributions $D_1$ and $D_2$ are identical on all elements with mass $\Omega(\frac{1}{\Lambda^2})$, by the birthday paradox, we can say that $\mathcal{O}(\Lambda)$ samples are not enough to obtain any *low* elements, that is, elements with mass $o(\frac{1}{\Lambda^2})$, that appear more than once. Since the property $\mathcal{P}$ is label-invariant, we can apply uniformly random permutations over the low elements of both $D_1$ and $D_2$, making the samples obtained from both $D_1$ and $D_2$ appear as two uniformly random sequences. Thus, from the view of any tester that takes only $\mathcal{O}(\Lambda)$ samples, $D_1$ and $D_2$ will appear the same, which would contradict the existence of a non-tolerant tester that distinguishes $D_1$ from $D_2$ using $\mathcal{O}(\Lambda)$ samples. At this point, we would like to point out that the proof of Lemma 4.4 only assumes the existence of a non-tolerant tester, and is oblivious to its internal details. Later, in Lemma 4.5, we generalize this idea to show that when $D_1$ and $D_2$ are close with respect to the high elements, it cannot be the case that $D_1$ is in the property $\mathcal{P}$, while $D_2$ is far from $\mathcal{P}$. Although the proof follows a similar line to that of Lemma 4.4, more careful analysis is required to prove Lemma 4.5. Note that Lemma 4.5 is the main technical lemma required to prove Theorem 3.1.

Once we have Lemma 4.4 and Lemma 4.5, we can describe the algorithm of Theorem 3.1. Broadly speaking, we show that *partial learning* of the distribution is sufficient for constructing a tolerant tester for any label-invariant property, as opposed to the more familiar paradigm of *testing by learning* [DLM$^+$07, Ser10]. Using Lemma 4.5, we show that estimating the masses of only the high elements is enough for us, along with the fact that the property $\mathcal{P}$ that we are testing is label-invariant. Roughly, the algorithm has three steps. In the first step, we identify and measure the high elements of the unknown distribution $D$. In the second step, we construct a new distribution $\widetilde{D}$ that adheres to the high mass elements obtained from the first step. Finally, in the third step, we check whether there exists any distribution $D_1$ in $\mathcal{P}$ that is close to $\widetilde{D}$. If such a distribution exists, we accept, and otherwise we reject. In the first step, we need $\widetilde{\mathcal{O}}(\Lambda^2)$ samples to correctly estimate the masses of the high elements, which dominates the sample complexity of our tolerant tester.

It is important to note that the *computational efficiency* of the tolerant tester depends on how fast we can check whether the distribution $\widetilde{D}$ (constructed by the algorithm) is *close* to a known property $\mathcal{P}$, where we have the complete description of $\widetilde{D}$. Later, in Theorem 3.2 (restated as Theorem 4.14), we show that when the property $\mathcal{P}$ can be expressed as a feasible solution to a set of linear inequalities, there exists an algorithm that tolerantly tests for $\mathcal{P}$ in time polynomial in the support size of the distribution and the number of linear inequalities required to represent it. The algorithm is similar to that of Theorem 3.1, whereas its polynomial running time follows by using the Ellipsoid method.

### 3.5.2 Lower bound results for non-concentrated properties

Here we give an overview of our proofs of Theorem 3.3 and Theorem 3.4. In Theorem 3.3, we show that in order to non-tolerantly test any non-concentrated property (defined in Definition 3.11), $\Omega(\sqrt{n})$ samples are required, where $n$ denotes the support size of the distribution. Before directly proceeding to prove the result, as a warm-up, we first show an analogous result for label-invariant non-concentrated properties in Theorem 5.3. To prove the theorem, for any distribution $D_{yes}$ in the label-invariant non-concentrated property $\mathcal{P}$ that we are testing, we construct a new distribution $D_{no}$ that is far from $\mathcal{P}$, whose support is a subset of the support of $D_{yes}$. The two distributions are identical over their high probability elements, and they only differ in their *low* probability elements, where a low probability element is an element with mass $\mathcal{O}(\frac{1}{n})$. Since $D_{yes}$ and $D_{no}$ differ only on the elements with mass $\mathcal{O}(\frac{1}{n})$, by the birthday paradox and the fact that the property is label-invariant, any tester that takes $o(\sqrt{n})$ samples cannot distinguish between $D_{yes}$ and $D_{no}$, and the result follows. We note that the proof of Theorem 5.3 is a generalization of the lower bound proof for uniformity testing.

Though the proof of Theorem 3.3 (restated as Theorem 5.8) follows similarly to that of Theorem 5.3, delicate analysis is required to take care of the fact that the properties are no longer label-invariant. We also discuss briefly the reason why the technique used to prove Theorem 5.3 does not work to prove Theorem 3.3, in the beginning of Section 5.3.

As a step further, in Theorem 3.4 (restated as Theorem 5.5), we show $\Omega(n^{1-o(1)})$ samples are necessary to tolerantly test any non-concentrated label-invariant property. This proof follows from an application of the *low frequency blindness theorem* of Valiant [Val11]. The question of tolerant testing of general non-concentrated properties remains open.

### 3.5.3 Learning Distributions with Unknown Support

Here we give an overview of the proof of Theorem 3.5. We consider the problem of learning a distribution $D$, where $D$ is *concentrated* over a unknown set $S \subseteq \Omega$. In Theorem 3.5 (restated as Theorem 6.2), we give an algorithm that achieves this with $\mathcal{O}(|S|)$ samples, even when $|S|$ is also unknown. Note that this problem is reminiscent of the folklore result of learning a distribution over any set $S$ that takes $\mathcal{O}(|S|)$ samples.

However, the folklore result holds only for the case where the set $S$ is known [6].

Broadly, the algorithm iterates over possible values of $|S|$. Starting from $s = 1$, we first take $s$ samples from the the unknown distribution $D$, and construct a new empirical distribution $D_s$ based upon the samples obtained. Once we have the distribution $D_s$, we apply the result of Valiant and Valiant [VV11] to test whether the unknown distribution $D$ is close to the newly constructed distribution $D_s$, by using number of samples that is slightly smaller than $s$. If $D_s$ is close to $D$, we report the distribution $D_s$ as the output and terminate the algorithm. Otherwise, we double the value of $s$ and perform another iteration of the two steps as mentioned above. Finally, we show that when $s \geq |S|$, where $S$ is the unknown set on which $D$ is concentrated, $D_s$ will be close to $D$ with high probability, and we will output a distribution satisfying the statement of Theorem 3.5. To the best of our knowledge, this is the first result of a tester of this kind that adapts to an unknown support size $|S|$.

[6] There are also prior results where only $|S|$ is known, such as in the work of Acharya, Diakonikolas, Li and Schmidt [ADLS17].

# Chapter 4

# Tolerant & Non-tolerant Testers for Label-Invariant Properties

## 4.1 Introduction

In this chapter, we will prove that for any label-invariant property, the sample complexities of tolerant and non-tolerant testing are separated by at most a quadratic factor (ignoring some poly-logarithmic factors). Formally, the result is stated as follows:

**Theorem 4.1 (Theorem 3.1 formalized).** *Let $\mathcal{P}$ be a label-invariant distribution property, for which there exists an $(0, \varepsilon)$-tester (non-tolerant tester) with sample complexity $\Lambda(n, \varepsilon)$, where $\Lambda \in \mathbb{N}$ and $0 < \varepsilon \leq 2$. Then for any $\gamma_1, \gamma_2$ with $\gamma_1 < \gamma_2$ and $0 < \gamma_2 + \varepsilon < 2$, there exists a $(\gamma_1, \gamma_2 + \varepsilon)$-tester (tolerant tester) that has sample complexity $\mathcal{O}\left(\frac{1}{(\gamma_2-\gamma_1)^3} \cdot \min\{\Lambda^2 \log^2 \Lambda, n\}\right)$, where $\Lambda = \Lambda(n, \epsilon)$, and $n$ is the size of the support of the distribution.*

We will prove this in Section 4.2. Then, in Section 4.3, we prove the following theorem regarding construction of efficient tolerant tester.

**Theorem 4.2 (Theorem 3.2 formalized).** *Let $\mathcal{P}$ be a label-invariant distribution property. If there is a $(0, \varepsilon)$-tester (non-tolerant tester) with sample complexity $\Lambda(n, \varepsilon)$, then* *for any $\gamma_1$, $\gamma_2$ with $\gamma_1 < \gamma_2$ and $0 < \gamma_1 < \gamma_2 + \varepsilon < 2$, there exists a $(\gamma_1, \gamma_2 + \varepsilon)$-tester (tolerant tester) that takes $s = \widetilde{\mathcal{O}}(\Lambda^2)$ samples and makes a single emptiness query to the set $\mathcal{CP} \cap \Delta(\widetilde{\mathcal{O}}(s), \Lambda, \widetilde{D}, \beta)$, where $\widetilde{D}$ is a known probability distribution and $\beta = \gamma_1 + \frac{\gamma_2 - \gamma_1}{3}$.*

## 4.2 Non-tolerant vs. tolerant testing of label-invariant properties

Let us assume that $D$ is the unknown distribution and $\Lambda(n, \epsilon) \geq \Omega(\frac{1}{\varepsilon})$ [1]. First note that if $\Lambda = \Omega(\sqrt{n})$, then we can construct a distribution $\widehat{D}$ such that $||D - \widehat{D}||_1 < \frac{\gamma_2 - \gamma_1 + \varepsilon}{2}$, by using $\mathcal{O}\left(\frac{n}{(\gamma_2 - \gamma_1 + \varepsilon)^2}\right)$ samples from $D$. Thereafter we can report $D$ to be $\gamma_1$-close to the property if and only if $\widehat{D}$ is $\frac{\gamma_2 + \gamma_1 + \varepsilon}{2}$-close to the property. In what follows, we discuss an algorithm with sample complexity $\widetilde{\mathcal{O}}(\Lambda^2)$ when $\Lambda = o(\sqrt{n})$. Also, we assume that $n$ and $\Lambda$ are larger than some suitable constant. Otherwise, the theorem becomes trivial.

The idea behind the proof is to classify the elements of $\Omega$ with respect to their masses in $D$ into *high* and *low*, as formally defined below in Definition 4.3. We argue that since $\mathcal{P}$ is $(0, \varepsilon)$-testable using $\Lambda(n, \varepsilon) = \mathcal{O}(q)$ samples, there cannot be two distributions $D_1$ and $D_2$ that are identical on all elements whose probability mass is at least $\frac{1}{q^2}$, for $q = \theta(\Lambda)$ (the set $\mathsf{High}_{1/q^2}$ defined below), where $D_1 \in \mathcal{P}$ but $D_2$ is $\varepsilon$-far from $\mathcal{P}$. We will formally show this in Lemma 4.4, where we will use the fact that $\mathcal{P}$ is label-invariant. Using Lemma 4.4, we prove Lemma 4.5, that (informally) says that if two distributions are close with respect to the high mass elements, then it is not possible that one distribution is close to $\mathcal{P}$ while the other one is far from it. In our algorithm, we intend to approximate the masses of the set $\mathsf{High}_{1/q^2}$, and the term $\Lambda^2$ in the query complexity of our algorithm corresponds to that.

**Definition 4.3.** For a distribution $D$ over $\Omega$ and $0 < \kappa < 1$, we define

$$\mathsf{High}_\kappa(D) = \{x \in \Omega \mid D(x) \geq \kappa\}$$

[1] This is a reasonable assumption for any non-trivial property.

Now we define a quantity $q \in \mathbb{N}$ where $q = \Theta(\Lambda)$ [2]. Assume that $c^*$ is a suitable large constant (independent of $\Lambda$) such that, if we take $\Lambda$ samples from a distribution, then with probability at least $\frac{3}{4}$, we will not get any sample $x$ whose mass is at most $(\frac{c^*}{\Lambda})^2$ more than once. We define

$$q := \frac{\Lambda}{c^*}. \tag{4.1}$$

We will complete the proof of Theorem 4.1 by using the following two lemmas which we will prove later.

**Lemma 4.4.** *Let $\mathcal{P}$ be a label-invariant property that is $(0, \varepsilon)$-testable using $\Lambda(n, \varepsilon)$ samples and consider $q$ as defined in Equation 4.1. Let $D_1$ and $D_2$ be two distributions such that $\mathsf{High}_{1/q^2}(D_1) = \mathsf{High}_{1/q^2}(D_2)$, and for all $x \in \mathsf{High}_{1/q^2}(D_1)$, the probability of $x$ is the same for both distributions, that is, $D_1(x) = D_2(x)$. Then it is not possible that $D_1$ satisfies $\mathcal{P}$ while $D_2$ is $\varepsilon$-far from satisfying $\mathcal{P}$.*

**Lemma 4.5.** *Let $\mathcal{P}$ be a label-invariant property that is $(0, \varepsilon)$-testable using $\Lambda(n, \varepsilon)$ samples, and consider $q$ as defined in Equation* (4.1)*. Let $\overline{D}$ and $\widetilde{D}$ be two distributions over $\Omega$, where $|\Omega| > 4q^2$, and let $H$ contain the top $q^2$ elements of $\overline{D}$. Also, assume that $\left|\widetilde{D}(\Omega \setminus H) - \overline{D}(\Omega \setminus H)\right| \leq \gamma$. If*

$$\sum_{x \in H} \left|\overline{D}(x) - \widetilde{D}(x)\right| \leq \alpha, \tag{4.2}$$

*then the following hold:*

1. *If $\overline{D}$ is $\beta$-close to $\mathcal{P}$, there exists a distribution $D_1 \in \mathcal{P}$ such that $\mathsf{High}_{1/q^2}(D_1) \subseteq H$ and*

$$\sum_{x \in H} \left|D_1(x) - \widetilde{D}(x)\right| + \left|D_1(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H)\right| \leq (\alpha + \beta + \gamma). \tag{4.3}$$

2. *If $\overline{D}$ is $(\varepsilon + 3\alpha + \beta + 2\gamma)$-far from $\mathcal{P}$ & $D_1$ is a distribution such that $\mathsf{High}_{1/q^2}(D_1) \subseteq$*

[2] Note that $q$ and $\Lambda$ are of the same order of magnitude. We have introduced $q$ for writing proofs more rigorously.

*$H$ and*

$$\sum_{x \in H} \left| D_1(x) - \widetilde{D}(x) \right| + \left| D_1(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H) \right| \leq (\alpha + \beta + \gamma), \tag{4.4}$$

*then the distribution $D_1$ does not satisfy the property $\mathcal{P}$.*

Using the above two lemmas, we will prove Theorem 4.1 in Section 4.2. We present the proofs of Lemma 4.4 and Lemma 4.5 in Section 4.2.

## Proof of Theorem 4.1

Let $D$ be the unknown distribution that we need to test, and assume that $\zeta = \gamma_1$, $\eta = \gamma_2 - \gamma_1$, and $\eta' = \frac{\eta}{64}$. We now provide a tolerant $(\gamma_1, \gamma_2 + \varepsilon)$-tester, that is, a $(\zeta, \zeta + \varepsilon + \eta)$-tester for the property $\mathcal{P}$, as follows:

1. Draw $W = \mathcal{O}\left(\frac{q^2}{\eta'} \log q\right)$ samples from the distribution $D$. Let $S \subseteq \Omega$ be the set of (distinct) samples obtained.

2. Draw additional $\mathcal{O}\left(\frac{W}{\eta'^2} \log W\right)$ samples $Z$ to estimate the value of $D(x)$ for all $x \in S$[3].

3. Construct a set $H$ as the union of $S$ and arbitrary $q^2$ elements from $\Omega \setminus (S \cup Z)$.

4. Define a distribution $\widetilde{D}$ such that, for $x \in H$,

$$\widetilde{D}(x) = \frac{\#\, x \text{ in the multi-set } Z}{|Z|}.$$

And for each $x \in \Omega \setminus H$,

$$\widetilde{D}(x) = \frac{1 - \sum\limits_{x \in H} \widetilde{D}(x)}{|\Omega| - |H|}.$$

[3] Instead of two sets of random samples (where the first one is to generate the set $S$ and the other one is the multi-set $Z$), one can work with only one set of random samples. But in that case, the sample complexity becomes $\mathcal{O}(q^2 \log n)$, as opposed to $\mathcal{O}(q^2 \log q)$ that we are going to prove.

5. If there exists a distribution $D_1$ in $\mathcal{P}$ that satisfies both the following conditions:

   **(A)** $\sum\limits_{x \in H} \left| D_1(x) - \widetilde{D}(x) \right| + |D_1(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H)| \leq 26\eta' + \zeta$.

   **(B)** $\mathsf{High}_{1/q^2}(D_1) \subseteq H$.

   then ACCEPT $D$.

6. If there does not exist any $D_1$ in $\mathcal{P}$ that satisfies both Conditions **(A)** and **(B)** above, then REJECT $D$.

Note that Step 5 as mentioned above is not completely constructive in a computational sense. In Section 4.3, we give a constructive variant of the tester where the property $\mathcal{P}$ can be expressed as a set of linear inequalities. We also give an example of a natural property that can be expressed as a set of linear inequalities.

**Sample Complexity:** The sample complexity of tester is $\mathcal{O}(\frac{q^2}{\eta^3} \log^2 q) = \mathcal{O}(\frac{\Lambda^2 \log^2 \Lambda}{(\gamma_2 - \gamma_1)^3})$, which follows from the above description.

**Correctness of the algorithm.** The proof of correctness of our algorithm is divided into a sequence of lemmas.

**Lemma 4.6.** *The set $H$ and the distribution $\widetilde{D}$ satisfies the following three properties:*

**(i)** *With probability at least $1 - \frac{1}{q}$, $\mathsf{High}_{\eta'/q^2}(D) \subseteq S \subseteq H$.*

**(ii)** *For any $x \in H$, if $D(x) \geq \frac{\eta'}{10W}$, $(1 - \eta')D(x) \leq \widetilde{D}(x) \leq (1 + \eta')D(x)$ holds with probability at least $1 - \frac{1}{q^4}$.*

**(iii)** *For any $x \in \Omega$ with $D(x) \leq \frac{\eta'}{10W}$, either $x \notin H$, or $\widetilde{D}(x) \leq (1 + \eta')\frac{\eta'}{10W}$ holds with probability at least $1 - \frac{1}{q^4}$.*

*Proof.* Let us prove the three parts one by one:

- **(i)** Consider any $x \in \mathsf{High}_{\eta'/q^2}(D)$, that is, $D(x) \geq \frac{\eta'}{q^2}$. Then the probability that $x \notin H$ is at most $(1 - \frac{\eta'}{q^2})^{|H|} \leq \frac{1}{q^4}$. Applying the union bound over all the elements in $\mathsf{High}_{\eta'/q^2}(D)$ (at most $\frac{q^2}{\eta'} = \mathcal{O}(q^3)$ [4] elements), the claim follows.

[4] This follows from the assumption that $\Lambda(n, \epsilon)$ is at least $\Omega(1/\epsilon)$.

- **(ii)** Since $|Z| = \mathcal{O}(\frac{W}{\eta'^2}\log W)$, applying Chernoff bound, we have $(1-\eta')D(x) \leq \widetilde{D}(x) \leq (1+\eta')D(x)$ does not hold with probability at most $\frac{1}{q^4}$.
- **(iii)** Since $|Z| = \mathcal{O}(\frac{W}{\eta'^2}\log W)$, if $x$ is in $H$ (otherwise, we are already done), applying Chernoff bound (only on one side), the bound follows.

□

We now bound the $\ell_1$-distance between $D$ and $\widetilde{D}$ with respect to $H$.

**Lemma 4.7.** $\sum\limits_{x \in H} \left| D(x) - \widetilde{D}(x) \right| \leq 5\eta'(1+\eta') \leq 10\eta'$ *holds with probability at least* $1 - \frac{3}{q}$.

*Proof.* Recall the definition of $\mathsf{High}_{\eta'/10W}(D)$. Note that

$$\sum_{x \in H} \left| D(x) - \widetilde{D}(x) \right| = \sum_{x \in \mathsf{High}_{\eta'/10W}(D)} \left| D(x) - \widetilde{D}(x) \right| + \sum_{x \in H \setminus \mathsf{High}_{\eta'/10W}(D)} \left| D(x) - \widetilde{D}(x) \right|$$

Applying Lemma 4.6 $(ii)$ for each $x \in \mathsf{High}_{\eta'/10W}(D)$, and then using union bound over all such $x \in \mathsf{High}_{\eta'/10W}(D)$, the first term is bounded by $\eta'$ with probability at least $1 - \frac{1}{q}$.

Now the second term, notice that for each $x \in H \setminus \mathsf{High}_{\eta'/10W}(D)$, $D(x) \leq \frac{\eta'}{10W}$. By Lemma 4.6 (iii), and using the union bound over all elements in $H \setminus \mathsf{High}_{\eta'/10W}(D)$ (note that $|H| \leq 2W = \mathcal{O}(q^3)$), with probability at least $1 - \frac{2}{q}$, $\widetilde{D}(x) \leq \eta'(1+\eta')/10W$ for all $x \in H \setminus \mathsf{High}_{\eta'/10W}(D)$. Since $|H| \leq 2W$, the second term is bounded by $4\eta'(1+\eta')$ with probability at least $1 - \frac{2}{q}$. □

Now we prove a lemma that shows that for every distribution $D$, there is a another distribution $\overline{D}$ that is "similar" to $D$, and for which $H$ contains the top $q^2$ elements of $\overline{D}$.

**Lemma 4.8.** *There exists a distribution $\overline{D}$ such that $H$ contains the top $q^2$ elements of $\overline{D}$. Moreover, the following hold:*

**(i)** $||D - \overline{D}||_1 \leq 2\eta'$, *with probability at least* $1 - \frac{2}{q}$.

**(ii)** $\sum\limits_{x \in H} \left|\overline{D}(x) - \widetilde{D}(x)\right| \leq 12\eta'$, *with probability at least* $1 - \frac{5}{q}$.

**(iii)** $|\overline{D}(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H)| \leq 12\eta'$, *with probability at least* $1 - \frac{5}{q}$.

*Proof.* Let $T$ be the set of $q^2$ largest elements of $D$. If $T \subseteq S$, $H$ (as $S \subset H$) contains the largest $q^2$ elements of $D$. In that case, setting $\overline{D}$ to be $D$ gives us the above results.

Now, let us consider the case where $T \not\subseteq S$. By Lemma 4.6 (part (i)), with probability at least $1 - \frac{2}{q}$, $\mathsf{High}_{\eta'/q^2}(D) \subseteq S$. Thus for any $x \in H \setminus S$, $D(x) < \frac{\eta'}{q^2}$. Consider the set $U = T \setminus H$. Notice that since $|H \setminus S| = q^2$ and $|T| = q^2$, $|U| \leq |H \setminus (T \cup S)|$. Let $U = \{y_1, \ldots, y_{|U|}\} \subset \Omega \setminus H$, and let $z_1 \ldots, z_{|U|}$ be some $|U|$ elements of $H \setminus (T \cup S)$. Note, by definition of $T$ and $U$, the set $\{z_1 \ldots, z_{|U|}\}$ and the set $\{y_1 \ldots, y_{|U|}\}$ are disjoint.

Consider the distribution $\overline{D}$ defined as follows:

- For elements in $\{z_1 \ldots, z_{|U|}\}$, we define $\overline{D}(z_i) = D(y_i)$.
- For elements in $\{y_1 \ldots, y_{|U|}\}$, we define $\overline{D}(y_i) = D(z_i)$.
- For all other $x$, we define $\overline{D}(x) = D(x)$.

Note that since all the elements in the sets $\{z_1 \ldots, z_{|U|}\}$ and $\{y_1 \ldots, y_{|U|}\}$ were from $\Omega \setminus S$, from Lemma 4.6 (part (i)), with probability at least $1 - \frac{2}{q}$, $D(y_i) \leq \frac{\eta'}{q^2}$ and $D(z_i) \leq \frac{\eta'}{q^2}$, for all $i \in \{1, \ldots, |U|\}$. Moreover, as $|U| \leq q^2$, we have condition (i) as well. Furthermore, $H$ contains the largest $q^2$ elements of $\overline{D}$ due to its construction.

Using the triangle inequality (relative to $H$) along with Lemma 4.7 and the above expression, we can say that, with probability at least $1 - \frac{5}{q}$, (ii) follows.

Let us now prove (iii). Since $\overline{D}$ and $\widetilde{D}$ are distributions, $\sum\limits_{x \in H} \overline{D}(x) + \sum\limits_{x \in \Omega \setminus H} \overline{D}(x) = \sum\limits_{x \in H} \widetilde{D}(x) + \sum\limits_{x \in \Omega \setminus H} \widetilde{D}(x)$. Thus,

$$\left|\overline{D}(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H)\right| = \left|\sum_{x \in H} \widetilde{D}(x) - \sum_{x \in H} \overline{D}(x)\right| \leq \sum_{x \in H} \left|\widetilde{D}(x) - \overline{D}(x)\right| \leq 12\eta'$$

The last inequality follows from (ii). □

Now we finally establish the correctness of the algorithm.

*Proof of correctness of the algorithm.* For completeness, consider the case where $D$ is $\zeta$-close to $\mathcal{P}$. By Lemma 4.8 (i) and the triangle inequality, we know that there exists a distribution $\overline{D}$ that is $(\zeta + 2\eta')$-close to $\mathcal{P}$ and $H$ contain the largest $q^2$ elements of $\overline{D}$. Since $\sum\limits_{x \in H} \left|\overline{D}(x) - \widetilde{D}(x)\right| \leq 12\eta'$ and $\left|\overline{D}(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H)\right| \leq 12\eta'$ hold from Lemma 4.8 (ii) and (iii), following Lemma 4.5 for $\alpha = 12\eta'$, $\beta = \zeta + 2\eta'$ and $\gamma = 12\eta'$, we can say that there exists a distribution $D_1$ in $\mathcal{P}$ satisfying Equation (4.3) (which is same as satisfying **Condition (A)** and **Condition (B)** in Step 5 of the algorithm). Hence, our algorithm accepts $D$ in Step 5.

For soundness, consider a distribution $D$ that is $(\varepsilon + \zeta + \eta)$-far from $\mathcal{P}$. Then following Lemma 4.8 (i), we know that there exists a distribution $\overline{D}$ that is $(\varepsilon + \zeta + \eta - 2\eta')$-far from $\mathcal{P}$, that is, $(\varepsilon + 3\alpha + \beta + 2\gamma)$-far from $\mathcal{P}$, where $\alpha = 12\eta'$, $\beta = \zeta + 2\eta'$. Here, we are using that $\eta = 64\eta'$ and $\gamma = 12\eta'$. Also Lemma 4.8 guarantees that $H$ contains the top $q^2$ elements of $\overline{D}$. Following Lemma 4.5, we know that there does not exist any such distribution $D_1$ in $\mathcal{P}$ that satisfies both **Condition (A)** and **Condition (B)** of Step 5 of the algorithm. Thus the algorithm will REJECT the distribution $D$ in Step 6.

Note that the total failure probability of the algorithm is bounded by the probability that Lemma 4.8 does not hold, which is at most $\frac{12}{q}$. □

## Proof of Lemma 4.4 and Lemma 4.5:

*Proof of Lemma 4.4.* We will prove this by contradiction. Let us assume that there are two distributions $D_{yes}$ and $D_{no}$ such that

- $D_{yes} \in \mathcal{P}$;
- $D_{no}$ is $\varepsilon$-far from $\mathcal{P}$;
- $\mathsf{High}_{1/q^2}(D_{yes}) = \mathsf{High}_{1/q^2}(D_{no}) = A$;
- For all $x \in A$, $D_{yes}(x) = D_{no}(x)$.

Now, we argue that any $(0, \varepsilon)$-non-tolerant tester requires more than $\Lambda(n, \varepsilon)$ samples from the unknown distribution $D$ to distinguish whether $D$ is in the property or $\varepsilon$-far from it.

Let $D_Y$ be a distribution obtained from $D_{yes}$ by permuting the labels of $\Omega \setminus A$ using a uniformly random permutation. Specifically, consider a random permutation $\pi : \Omega \setminus A \to \Omega \setminus A$. The distribution $D_Y$ is as follows:

- $D_Y(x) = D_{yes}(x)$ for each $x \in A$ and
- $D_Y(\pi(x)) = D_{yes}(x)$ for each $x \in \Omega \setminus A$.

Similarly, consider the distribution $D_N$ obtained from $D_{no}$ by permuting the labels of $\Omega \setminus A$ using a uniformly random permutation. Note that $D_Y$ is in $\mathcal{P}$, whereas $D_N$ is $\varepsilon$-far from $\mathcal{P}$, which follows from $\mathcal{P}$ being label-invariant.

We will now prove that $D_Y$ and $D_N$ provide similar distributions over sample sequences. More formally, we will prove that any algorithm that takes at most $\Lambda(n, \varepsilon)$ samples, cannot distinguish $D_Y$ from $D_N$ with probability at least $\frac{2}{3}$. We argue that this claim holds even if the algorithm is provided with additional information about the input: Namely, for all $x \in A$, the algorithm is told the value of $D_Y(x)$ (which is the same as $D_N(x)$). When the algorithm is provided with this information, it can ignore all samples obtained from $A$.

By the definition of $A$, for all $x \in \Omega \setminus A$, both $D_Y(x)$ and $D_N(x)$ are at most $\frac{1}{q^2}$. Let $S_Y$ be a sequence of samples drawn according to $D_Y$. If $|S_Y| \leq \Lambda(n, \varepsilon)$, then with probability at least $\frac{3}{4}$, the sequence $(\Omega \setminus A) \cap S_Y$ has no element that appears twice. In other words, the set $(\Omega \setminus A) \cap S_Y$ is a set of at most $\Lambda(n, \varepsilon)$ distinct elements from $\Omega \setminus A$. Since the elements of $\Omega \setminus A$ were permuted using a uniformly random permutation, with probability at least $\frac{3}{4}$, the sequence $(\Omega \setminus A) \cap S_Y$ is a uniformly random sequence of distinct elements from $\Omega \setminus A$. Similarly, if $S_N$ is a sequence of samples drawn according to $D_N$, then with probability at least $\frac{3}{4}$, the sequence $(\Omega \setminus A) \cap S_N$ is a uniformly random sequence of distinct elements from $\Omega \setminus A$. Thus, the distributions over the received sample sequence obtained from $D_Y$ or $D_N$ are of distance $\frac{1}{4}$ of each other, which is strictly less than $\frac{1}{3}$.

Hence, if the algorithm obtains at most $\Lambda(n, \varepsilon)$ samples from the unknown distribution $D$, it cannot distinguish, with probability at least $\frac{2}{3}$, whether the samples are coming from $D_Y$ or $D_N$. □

For the proof of Lemma 4.5, we will need the following simple claim.

**Claim 4.9.** *Let $\sigma : [n] \to [n]$ be a permutation and let $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ be two sets of $n$ positive real numbers. If $a_1 \geq a_2 \geq \cdots \geq a_n$ and $b_1 \geq b_2 \geq \cdots \geq b_n$, then the sum $\sum\limits_{i \in [n]} \left|a_i - b_{\sigma(i)}\right|$ is minimized when $\sigma$ is the identity permutation.*

*Proof.* First observe that if $a, b, c, d$ are four real numbers with $a \geq b$ and $c \geq d$, then the following holds:

$$|a - c| + |b - d| \leq |a - d| + |b - c| . \tag{4.5}$$

The above can be proved by checking all possible orderings of the numbers $a, b, c, d$.

Once we have the above observation, we can now proceed to prove the claim. Let us consider the set of permutations that minimize $\sum\limits_{i \in [n]} \left|a_i - b_{\sigma(i)}\right|$. Let $\sigma$ be one such minimizing permutation that also minimizes the size for the following set $S$:

$$S = \{(i, j) : i < j \text{ and } \sigma(i) > \sigma(j)\}$$

Let $i$ be an index such that $\sigma(i) < \sigma(i + 1)$ (such an index $i$ exists unless $\sigma$ is the identity permutation). Let $\sigma'$ be the permutation obtained from $\sigma$ by swapping $\sigma(i)$ and $\sigma(i+1)$. Then the sum $\sum\limits_{i \in [n]} \left|a_i - b_{\sigma'(i)}\right|$ does not increase from $\sum\limits_{i \in [n]} \left|a_i - b_{\sigma(i)}\right|$, because of Equation 4.5. However, the size of the set $S$ with respect to the permutation $\sigma'$ strictly decreases, and we have a contradiction. □

Now we present the proof of Lemma 4.5.

*Proof of Lemma 4.5.* We consider the two cases separately.

**(1)** If $\overline{D}$ is $\beta$-close to $\mathcal{P}$, then there exists a distribution $D_1$ in $\mathcal{P}$ such that we have $\sum_x \left|\overline{D}(x) - D_1(x)\right| \leq \beta$. Since $\mathcal{P}$ is label-invariant, any permutation of $D_1$ is also in $\mathcal{P}$. Without loss of generality, let us assume that the domain $\Omega$ is a subset of $\{1, \ldots, n\}$.

By Claim 4.9, the permutation $\sigma$ that minimizes $\sum_x \left|\overline{D}(x) - D_1(\sigma(x))\right| \leq \beta$ is the one that orders the $i$-th largest element of $D_1$ with the $i$-th largest element of $\overline{D}$, that is, if $x$ is the element with the $i$-th largest probability mass in $D_1$, then $\sigma(x)$ has the $i$-th largest probability mass in $\overline{D}$. Consider the distribution $D_1^\sigma$ that is defined by $D_1^\sigma(x) = D_1(\sigma(x))$. Clearly, $H$ contains the largest $q^2$ elements of $D_1^\sigma$, and hence also $\mathsf{High}_{1/q^2}(D_1^\sigma) \subseteq H$.

As $\sum_{x\in\Omega} \left|D_1^\sigma(x) - \overline{D}(x)\right| \leq \beta$, $\sum_{x\in H} \left|\overline{D}(x) - \widetilde{D}(x)\right| \leq \alpha$ and $|\overline{D}(\Omega\backslash H) - \widetilde{D}(\Omega\backslash H)| \leq \gamma$, by the triangle inequality, we obtain

$$
\begin{aligned}
& \sum_{x\in H} \left|D_1^\sigma(x) - \widetilde{D}(x)\right| + \left|D_1^\sigma(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H)\right| \\
\leq\ & \sum_{x\in H} |D_1^\sigma(x) - \overline{D}(x)| + \sum_{x\in H} |\overline{D}(x) - \widetilde{D}(x)| \\
& \qquad + |D_1^\sigma(\Omega \setminus H) - \overline{D}(\Omega \setminus H)| + |\overline{D}(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H)| \\
\leq\ & \sum_{x\in H} |D_1^\sigma(x) - \overline{D}(x)| + \sum_{x\in H} |\overline{D}(x) - \widetilde{D}(x)| \\
& \qquad + \sum_{x\in\Omega\setminus H} |D_1^\sigma(x) - \overline{D}(x)| + |\overline{D}(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H)| \\
=\ & \sum_{x\in\Omega} |D_1^\sigma(x) - \overline{D}(x)| + \sum_{x\in H} |\overline{D}(x) - \widetilde{D}(x)| + |\overline{D}(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H)| \\
\leq\ & \alpha + \beta + \gamma
\end{aligned}
$$

**(2)** We will prove this case by contradiction. Let $D_1 \in \mathcal{P}$ be a distribution such that $\mathsf{High}_{1/q^2}(D_1) \subseteq H$ and $\sum_{x\in H} \left|D_1(x) - \widetilde{D}(x)\right| + \left|D_1(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H)\right| \leq \alpha + \beta + \gamma$. Then, as $\sum_{x\in H} \left|\overline{D}(x) - \widetilde{D}(x)\right| \leq \alpha$, by the triangle inequality, we have

$$
\sum_{x\in H} |D_1(x) - \overline{D}(x)| + \left|D_1(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H)\right| \leq 2\alpha + \beta + \gamma. \qquad (4.6)
$$

Consider the distribution $\widehat{D}$ defined as follows:

- For all $x \in H$, $\widehat{D}(x) = D_1(x)$.

- If $D_1(H) \geq \overline{D}(H)$, then for all $x \in \Omega \setminus H$,

  $$\widehat{D}(x) = \overline{D}(x) \cdot \phi,$$

  where $\phi = \frac{1-D_1(H)}{1-\overline{D}(H)}$. Notice that in this case $\phi \leq 1$.

- If $D_1(H) \leq \overline{D}(H)$, then pick the set $T \subset \Omega \setminus H$ with $|T| = 2q^2$ that minimizes $\overline{D}(T)$. Then for all $x \in T$,

  $$\widehat{D}(x) = \overline{D}(x) + \frac{\overline{D}(H) - D_1(H)}{2q^2}$$

  and for all $x \in \Omega \setminus (T \cup H)$, $\widehat{D}(x) = \overline{D}(x)$

Let us first prove that $\mathsf{High}_{1/q^2}(\widehat{D}) \subseteq H$. In the case where $D_1(H) \geq \overline{D}(H)$, for all $x \in \Omega \setminus H$, $\widehat{D}(x) \leq \overline{D}(x)$. Since $\mathsf{High}_{1/q^2}(\overline{D}) \subseteq H$, $\mathsf{High}_{1/q^2}(\widehat{D}) \subseteq H$. Now, in the case where $D_1(H) \leq \overline{D}(H)$, the only $x \in \Omega \setminus H$ for which $\widehat{D}(x) > \overline{D}(x)$ are those in $T$. Since $|\Omega| > 4q^2$, the lowest $2q^2$ elements on $\overline{D}$ must each have mass less than $\frac{1}{2q^2}$. So even if we add $\frac{1}{2q^2}$ for any element $x \in T$, $\widehat{D}(x) < 1/q^2$. Hence in this case also $\mathsf{High}_{1/q^2}(\widehat{D}) \subseteq H$ since $\mathsf{High}_{1/q^2}(\overline{D}) \subseteq H$ and $\mathsf{High}_{1/q^2}(D_1) \subseteq H$.

Now let us bound the $\ell_1$ distance between $\widehat{D}$ and $\overline{D}$. Observe that

$$\sum_{x \in \Omega \setminus H} \left|\widehat{D}(x) - \overline{D}(x)\right| = \left|\widehat{D}(\Omega \setminus H) - \overline{D}(\Omega \setminus H)\right|.$$

This is because, in the case where $\widehat{D}(H) \geq \overline{D}(H)$, we have $\widehat{D}(x) = \phi \cdot \overline{D}(x) \leq \overline{D}(x)$ for all $x \in \Omega \setminus H$. On the other hand, in the case where $\widehat{D}(H) \leq \overline{D}(H)$ then for all $x \in \Omega \setminus H$, $\widehat{D}(x) \geq \overline{D}(x)$. Thus,

$$\begin{aligned}
\sum_{x\in\Omega\setminus H}\left|\widehat{D}(x)-\overline{D}(x)\right| &= \left|\widehat{D}(\Omega\setminus H)-\overline{D}(\Omega\setminus H)\right| \\
&\leq \left|\widehat{D}(\Omega\setminus H)-\widetilde{D}(\Omega\setminus H)\right| + \left|\overline{D}(\Omega\setminus H)-\widetilde{D}(\Omega\setminus H)\right| \\
&\leq \left|\widehat{D}(\Omega\setminus H)-\widetilde{D}(\Omega\setminus H)\right| + \gamma
\end{aligned}$$

Also note that, from the construction of $\widehat{D}$, we have for all $x \in H$, $\widehat{D}(x) = D_1(x)$ and thus $\widehat{D}(\Omega \setminus H) = D_1(\Omega \setminus H)$. Thus,

$$\begin{aligned}
||\widehat{D}-\overline{D}||_1 &= \sum_{x\in H}|\widehat{D}(x)-\overline{D}(x)| + \sum_{x\in\Omega\setminus H}|\widehat{D}(x)-\overline{D}(x)| \\
&\leq \sum_{x\in H}|\widehat{D}(x)-\overline{D}(x)| + \left|\widehat{D}(\Omega\setminus H)-\widetilde{D}(\Omega\setminus H)\right| + \gamma \\
&= \left(\sum_{x\in H}|D_1(x)-\overline{D}(x)| + \left|D_1(\Omega\setminus H)-\widetilde{D}(\Omega\setminus H)\right|\right) + \gamma \\
&\qquad \text{(From the construction of } \widehat{D}\text{)} \\
&\leq 2\alpha+\beta+2\gamma \quad \text{(By Equation (4.6))}
\end{aligned}$$

Moreover, $\mathsf{High}_{1/q^2}(D_1) \subseteq H$ and by the construction of $\widehat{D}$, we have $\mathsf{High}_{1/q^2}(D_1) = \mathsf{High}_{1/q^2}(\widehat{D})$ and for all $x \in \mathsf{High}_{1/q^2}(D_1)$, $D_1(x) = \widehat{D}(x)$. Since we assumed that $D_1$ is in $\mathcal{P}$, using Lemma 4.4, $\widehat{D}$ is $\varepsilon$-close to $\mathcal{P}$. And since $||\widehat{D} - \overline{D}||_1 \leq 2\alpha + \beta + 2\gamma$, we conclude that $\overline{D}$ is $(\varepsilon + 2\alpha + \beta + 2\gamma)$-close to $\mathcal{P}$, which is a contradiction. □

## 4.3 Computationally efficient tolerant testers

In this section we present a constructive variant of the tolerant tester studied in Section 4.2. Here, for any two vectors $a, b \in \mathbb{R}^N$, we say that $a \leq b$ if $a_i \leq b_i$ holds for every $i \in [N]$. Now let us recall the definitions of *polyhedron* and *projection map*.

**Definition 4.10 (Polyhedron).** Let $A$ be a $M \times N$ real matrix, $b \in \mathbb{R}^M$ be a real vector and $Ax \leq b$ be a system of linear inequalities. The solution set $\{x \in \mathbb{R}^N \mid Ax \leq b\}$ of the system of inequalities is called a polyhedron. The *complexity* of a polyhedron is defined as $MN$.

**Definition 4.11 (Projection map).** Let $n$ be an integer. For all integers $N \geq n$, a *projection map* is denoted as $\pi_n : \mathbb{R}^N \to \mathbb{R}^n$ and is defined as the projection of the points in $\mathbb{R}^N$ on the first $n$ coordinates.

Before directly proceeding to our results, we first define two variants of distribution properties.

**Definition 4.12 (Linear property).** Without loss of generality, let us assume that $\Omega = [n]$. A distribution property $\mathcal{P}$ is said to be a *linear property* if there exists a polyhedron $\mathcal{LP} = \left\{x \in \mathbb{R}^N \ \mid \ Ax \leq b\right\}$, where $A$ is a $M \times N$ real matrix and $b \in \mathbb{R}^M$ be a real vector, and $\pi_n\left(\mathcal{LP}\right)$ [5] is the set of distributions satisfying the property $\mathcal{P}$, that is, for every $z := (z_1, \ldots, z_n, \ldots, z_N) \in \mathcal{LP}$, the distribution $D_z$, defined as

$$D_z(i) = z_i, \quad \forall i \in [n]$$

satisfies the property $\mathcal{P}$, and conversely, for every distribution $D$ that satisfies $\mathcal{P}$, there exists some $z \in \mathcal{LP}$ such that $D = D_z$ as defined above. The *complexity* of $\mathcal{P}$ is defined as $M \times N$.

Similar to linear properties, we can also define properties that are feasible solutions to a system of convex constraints.

**Definition 4.13 (Convex property).** A distribution property $\mathcal{P}$ is said to be a *convex property* if $\mathcal{P}$ is the set of all feasible solutions to a system of convex constraints over $D(i)$ for $i \in \Omega$, where $\Omega$ is the sample space of $D$. In other words, the set $\mathcal{P}$ forms a convex set.

[5] Note that $\pi_n\left(\mathcal{LP}\right)$ will also be a polyhedron in $\mathbb{R}^n$, see, e.g., Corollary 2.5 in Chapter 2 from the book by Bertsimas and Tsitsiklis [BT97]. However, the number of linear inequalities defining the property, which affects the running time of the tester, can sometimes be greatly reduced by using a projection.

Now we show that some well studied label-invariant distribution properties can be represented as linear or convex properties.

**Remark 4.1 (An example of a linear property: Approximate uniformity property).** A distribution $D$ over $[n]$ is said to be uniform if $D(i) = \frac{1}{n}$ for all $i \in [n]$. Let the property $\mathcal{P}_{u,\varepsilon}$ denote the set of all distributions that are $\varepsilon$-close to the uniform distribution, where $\varepsilon \in (0, 1)$ is a parameter. Consider the following polyhedron $\mathcal{LP}_{u,\varepsilon}$ in $\mathbb{R}^{2n}$:

$$\begin{aligned}
&\sum_{i \in [n]} z_{n+i} \leq \varepsilon && \\
&z_i \geq 0 && \forall i \in [2n] \\
&- z_{n+i} \leq z_i - \frac{1}{n} \leq z_{n+i} && \forall i \in [n]
\end{aligned}$$

Now, observe that $\pi_n\left(\mathcal{LP}_{u,\varepsilon}\right)$ will give us the set of distributions that are $\varepsilon$-close to uniform, i.e., the set $\mathcal{P}_{u,\varepsilon}$ (this would serve as the linear transformation mentioned in Definition 4.12). Also, note that approximate uniformity property has complexity $\mathcal{O}(n)$.

Now we present a property that can be expressed as a feasible solution to a system of convex inequalities, but that cannot be expressed as feasible solution to a system of linear inequalities.

**Remark 4.2 (An example of a convex property: Entropy property).** Let $D$ be a distribution supported on $[n]$. Given a parameter $k \in \mathbb{R}$, let $\mathcal{P}_{E,k}$ denote the set of all distributions with entropy at least $k$. $\mathcal{P}_{E,k}$ can be expressed as a convex inequality as follows:

$$\sum_{i \in [n]} D(i) \log \frac{1}{D(i)} \geq k.$$

For a distribution property $\mathcal{P}$, let $\mathcal{CP} \subset \mathbb{R}^n$ denote the *geometric representation* of the set of probability distributions over the set $[n]$ that satisfy $\mathcal{P}$ by considering each distribution over $[n]$ as a point in $\mathbb{R}^n$.

For all $\beta \in [0,1]$, $k \leq n$ and $a \in \mathbb{R}^n$, we define the following convex set:

$$\Delta\left(k, q, a, \beta\right) := \left\{ x \in \mathbb{R}^d : \sum_{i=1}^{k} |x_i - a_i| + \left|\sum_{j>k} x_j - \sum_{j>k} a_j\right| \leq \beta \ \&\ x_i < \frac{1}{q^2}\ \forall i > k \right\}$$

Before proceeding to prove Theorem 3.2, we will first prove a more general result. We will show that if we have access to an emptiness oracle that takes a specific convex set (defined in the following statement) as input, and decides whether the convex set is empty or not, then we can design a tolerant tester for any label-invariant distribution property that takes $\widetilde{O}(\Lambda^2)$ samples and performs a single emptiness query to the oracle. The result is formally stated and proved below.

**Theorem 4.14.** *Let $\mathcal{P}$ be a label-invariant distribution property. If there is a $(0, \varepsilon)$-tester (non-tolerant tester) with sample complexity $\Lambda(n, \varepsilon)$, then for any $\gamma_1$, $\gamma_2$ with $\gamma_1 < \gamma_2$ and $0 < \gamma_1 < \gamma_2 + \varepsilon < 2$, there exists a $(\gamma_1, \gamma_2 + \varepsilon)$-tester (tolerant tester) that takes $s = \widetilde{\mathcal{O}}(\Lambda^2)$ samples and makes a single emptiness query to the set $\mathcal{CP} \cap \Delta(\widetilde{\mathcal{O}}(s), \Lambda, \widetilde{D}, \beta)$, where $\widetilde{D}$ is a known probability distribution and $\beta = \gamma_1 + \frac{\gamma_2 - \gamma_1}{3}$.*

*Proof.* Recall that in Step 5 of the tolerant tester presented in Section 4.2, the tester checks whether there is any distribution $D_1 \in \mathcal{P}$ that satisfies the following two conditions:

$$\sum_{x \in H} \left| D_1(x) - \widetilde{D}(x) \right| + \left| D_1(\Omega \setminus H) - \widetilde{D}(\Omega \setminus H) \right| \leq 26\eta' + \zeta$$

and

$$\mathsf{High}_{1/q^2}(D_1) \subseteq H$$

where $\zeta = \gamma_1$, $\eta = \gamma_2 - \gamma_1$, $\eta = \gamma_2 - \gamma_1$ and $\eta' = \frac{\eta}{64}$. The set $H$ and the distribution $\widetilde{D}$ are defined in the tolerant tester presented in Section 4.2.

Without loss of generality, we can assume that $H = \{1, \ldots, |H|\}$. Therefore, in order to perform Step 5 of the tolerant tester, the following equations are needed to be satisfied:

$$D_1 \in \mathcal{CP} \tag{4.7}$$

$$D_1 \in \Delta\left(|H|, q, \widetilde{D}, 26\eta' + \zeta\right) \tag{4.8}$$

We now present the tolerant $(\gamma_1, \gamma_2 + \varepsilon)$-tester in its entirety, that is, a $(\zeta, \zeta + \varepsilon + \eta)$-tester for the property $\mathcal{P}$, where $\zeta = \gamma_1$, $\eta = \gamma_2 - \gamma_1$, and $\eta' = \frac{\eta}{64}$.

1. Draw $W = \mathcal{O}\left(\frac{q^2}{\eta'} \log q\right)$ samples from the distribution $D$. Let $S \subseteq \Omega$ be the set of (distinct) samples obtained.

2. Draw additional $\mathcal{O}\left(\frac{W}{\eta'^2} \log W\right)$ samples $Z$ to estimate the value of $D(x)$ for all $x \in S$.

3. Construct a set $H$ as the union of $S$ and $q^2$ arbitrary elements from $\Omega \setminus (S \cup Z)$.

4. Define a distribution $\widetilde{D}$ such that, for $x \in H$,

   $$\widetilde{D}(x) = \frac{\#\, x \text{ in the multi-set } Z}{|Z|}.$$

   And for each $x \in \Omega \setminus H$,

   $$\widetilde{D}(x) = \frac{1 - \sum\limits_{x \in H} \widetilde{D}(x)}{|\Omega| - |H|}.$$

5. If there exists a distribution $D_1 \in \mathcal{CP} \cap \Delta\left(|H|, q, \widetilde{D}, 26\eta' + \zeta\right)$, then ACCEPT $D$.

6. If there does not exist any distribution $D_1$ that passes Step $5$, then REJECT $D$.

Observe that the sample complexity of the tester is $\mathcal{O}\left(\frac{q^2}{\eta^2} \log^2 q\right) = \widetilde{\mathcal{O}}(\Lambda^2)$ in addition to a single emptiness query to the set $\mathcal{P} \in \mathcal{CP} \cap \Delta\left(|H|, q, \widetilde{D}, 26\eta' + \zeta\right)$ in Step $5$. The correctness proof of the above tester follows from the correctness argument presented in Section 4.2. □

### 4.3.1 Emptiness checking when $\mathcal{P}$ is a linear property

Now we proceed to analyze the time complexity of the $(\gamma_1, \gamma_2 + \varepsilon)$-tester described in Theorem 4.14 when $\mathcal{P}$ is also a linear property. Recall that as $\mathcal{P}$ is a linear property, there exists a polyhedron $\mathcal{LP} = \left\{x \in \mathbb{R}^N \mid Ax \leq b\right\}$, where $A$ is a $M \times N$ real matrix and $b \in \mathbb{R}^M$ be a real vector, and $\pi_n\left(\mathcal{LP}\right)$ is the set of distributions satisfying the property $\mathcal{P}$. (See Definition 4.12)

Now we show that checking emptiness of $\pi_n(\mathcal{LP}) \cap \Delta\left(|H|, q, \widetilde{D}, 26\eta' + \zeta\right)$ is equivalent to testing the feasibility of a family of inequalities.

**Observation 4.15.** Without loss of generality, assume that $H = \{1, \ldots, |H|\}$ and $\Omega = \{1, \ldots, n\}$. Checking emptiness of $\pi_n(\mathcal{LP}) \cap \Delta(|H|, q, \widetilde{D}, 26\eta' + \zeta)$ is equivalent to testing the feasibility of the following set of inequalities:

$$Az \leq b \tag{4.9}$$

$$\sum_{i=1}^{|H|} \left|z_i - \widetilde{D}(i)\right| + \left|\sum_{i=|H|+1}^{n} z_i - \sum_{i=|H|+1}^{n} \widetilde{D}(i)\right| \leq 26\eta' + \zeta \tag{4.10}$$

$$z_i < \frac{1}{q^2} \qquad \forall i \in [n] \setminus \{1, \ldots, |H|\} \tag{4.11}$$

Note that the inequality in Equation (4.10) can be expressed as the following set of linear inequalities using slack variables $z_{N+i}$ for all $i \in [|H| + 1]$:

$$\begin{aligned}
&\sum_{i=1}^{|H|} z_{N+i} \ + \ z_{N+|H|+1} \leq 26\eta' + \zeta \\
&z_{N+i} \geq 0 && \forall i \in [|H| + 1] \\
&- z_{N+i} \leq z_i - \widetilde{D}(i) \leq z_{N+i} && \forall i \in [|H|] \\
&- z_{N+|H|+1} \leq \sum_{i=|H|+1}^{n} z_i - \sum_{i=|H|+1}^{n} \widetilde{D}(i) \leq z_{N+|H|+1}
\end{aligned}$$

Therefore checking the emptiness of $\pi_n(\mathcal{LP}) \cap \Delta\left(|H|, q, \widetilde{D}, 26\eta' + \zeta\right)$ is equivalent to checking the feasibility of the following set of linear inequalities:

$$\begin{aligned}
&Az \leq b \\
&\sum_{i=1}^{|H|} z_{N+i} + z_{N+|H|+1} \leq 26\eta' + \zeta \\
&z_{N+i} \geq 0 && \forall i \in [|H|+1] \\
&-z_{N+i} \leq z_i - \widetilde{D}(i) \leq z_{N+i} && \forall i \in [|H|] \\
&-z_{N+|H|+1} \leq \sum_{i=|H|+1}^{n} z_i - \sum_{i=|H|+1}^{n} \widetilde{D}(i) \leq z_{N+|H|+1} \\
&z_i < \frac{1}{q^2} && \forall i \in [n] \setminus \{1, \ldots, |H|\}
\end{aligned}$$

The feasibility of the above set of linear inequalities can be solved in a polynomial time in the complexity of the polyhedron, that is, in a polynomial time in $N$ and $M$, using the Ellipsoid Method, where recall that $A$ is a $M \times N$ real matrix (see, e.g., [BT97, GM07]). Thus, we have an efficient $(\gamma_1, \gamma_2 + \varepsilon)$-tester for $\mathcal{P}$, that runs in time polynomial in the complexity of the label-invariant linear property $\mathcal{P}$. This concludes the proof of Theorem 3.2.

# Chapter 5

# Testing of Non Concentrated Properties

## 5.1 Introduction

In this chapter, we present some lower bound results for testing non-concentrated distribution properties (see Definition 3.11). We prove that $\Omega(\sqrt{n})$ samples are required for non-tolerant testing of any non-concentrated properties. Formally, the result is stated as follows:

**Theorem 5.1 (Theorem 3.3 formalized).** *Let $\mathcal{P}$ be any $(\alpha, \beta)$-non-concentrated distribution property for $0 < \alpha < \beta < \frac{1}{2}$. For any $\varepsilon$ with $0 < \varepsilon < \alpha$, any $(0, \varepsilon)$-tester for $\mathcal{P}$ requires $\Omega(\sqrt{n})$ samples, where $n$ is the size of the support of the distribution.*

We will first prove an analogous result for label-invariant non-concentrated distribution properties in Section 5.2. Then in Section 5.3 we generalize the proof to hold for all classes of non-concentrated distribution properties.

Later in Section 5.2.2 we prove that almost linear number of samples are required for tolerant testing of label-invariant non-concentrated distribution properties.

**Theorem 5.2 (Theorem 3.4 formalized).** *Let $\mathcal{P}$ be any $(\alpha, \beta)$-non-concentrated label-invariant distribution property, where $0 < \alpha \leq \beta < \frac{1}{2}$. For any constant $\varepsilon_1$ and $\varepsilon_2$ with $0 < \varepsilon_1 < \varepsilon_2 < \alpha$, any $(\varepsilon_1, \varepsilon_2)$-tester for $\mathcal{P}$ requires $\Omega(n^{1-o(1)})$ samples, where $n$ is the size of the support of the distribution.*

## 5.2 Testing of non-concentrated label-invariant properties

In this section we first prove a lower bound of $\Omega(\sqrt{n})$ on the sample complexity of non-tolerant testing of any non-concentrated label-invariant property. Then we proceed to prove a tolerant lower bound of $\Omega(n^{1-o(1)})$ samples for such properties in Section 5.2.2.

### 5.2.1 Non-tolerant lower bound for label-invariant properties

Here we first prove a lower bound result analogous to Theorem 5.1, where the properties are non-concentrated and label-invariant. In Section 5.3, we discuss why the proof of Theorem 5.3 does not directly work for Theorem 5.1, and then prove Theorem 5.1 using a different argument.

**Theorem 5.3 (Analogous result of Theorem 5.1 for non-concentrated label-invariant properties).** *Let $\mathcal{P}$ be any $(\alpha, \beta)$-non-concentrated label-invariant distribution property, where $0 < \alpha \leq \beta < \frac{1}{2}$. For $\varepsilon$ with $0 < \varepsilon < \alpha$, any $(0, \varepsilon)$-tester for property $\mathcal{P}$ requires $\Omega(\sqrt{n})$ samples, where $n$ is the size of the support of the distribution.*

*Proof.* Let us first consider a distribution $D_{yes}$ that satisfies the property. Since $\mathcal{P}$ is an $(\alpha, \beta)$-non-concentrated property, by Definition 3.11, $D_{yes}$ is an $(\alpha, \beta)$-non-concentrated distribution. From $D_{yes}$, we generate a distribution $D_{no}$ such that the support of $D_{no}$ is a subset of that of $D_{yes}$, and $D_{no}$ is $\varepsilon$-far from $\mathcal{P}$. Hence, if we apply a random permutation over the elements of $\Omega$, we show that $D_{yes}$ and $D_{no}$ are indistinguishable, unless we query for $\Omega(\sqrt{n})$ samples. Below we formally prove this idea.

We will partition the domain $\Omega$ into two parts, depending on the probability mass of $D_{yes}$ on the elements of $\Omega$. Given the distribution $D_{yes}$, let us first order the elements of $\Omega$ according to their probability masses. In this ordering, let $L$ be the smallest $2\beta n$ elements of $\Omega$. We denote $\Omega \setminus L$ by $H$. Before proceeding further, note that the following observation gives an upper bound on the probabilities of the elements in $L$.

**Observation 5.4.** For all $x \in L$, $D_{yes}(x) \leq \frac{1-2\alpha}{1-2\beta}\frac{1}{n}$.

*Proof of Observation 5.4.* By contradiction, assume that there exists $x \in L$ such that $D_{yes}(x) > \frac{1-2\alpha}{1-2\beta}\frac{1}{n}$. This implies, for every $y \in H$, that $D_{yes}(y) > \frac{1-2\alpha}{1-2\beta}\frac{1}{n}$. So,

$$1 = \sum_{x \in \Omega} D_{yes}(x) = \sum_{x \in L} D_{yes}(x) + \sum_{y \in H} D_{yes}(y) > D_{yes}(L) + |H| \frac{1-2\alpha}{1-2\beta}\frac{1}{n}.$$

As $|L| = 2\beta n$ and $D_{yes}$ is an $(\alpha, \beta)$-non-concentrated distribution, $D_{yes}(L) \geq 2\alpha$. Also, $|H| = (1-2\beta)n$. Plugging these into the above inequality, we get a contradiction. $\square$

Note that Observation 5.4 implies that if $S$ is a multi-set of $o\left(\sqrt{\frac{1-2\beta}{1-2\alpha}n}\right)$ samples from $D_{yes}$, then with probability $1 - o(1)$, no element from $L$ appears in $S$ more than once. Now using the distribution $D_{yes}$ and the set $L$, let us define a distribution $D_{no}$ such that $D_{no}$ is $\varepsilon$-far from $\mathcal{P}$. Note that $D_{no}$ is a distribution that comes from a distribution over a set of distributions, all of which are not $(\alpha, \beta)$-non-concentrated. The distribution $D_{no}$ is generated using the following random process:

- We partition $L$ randomly into two equal sets of size $\beta n$: $\{x_1, \ldots, x_{\beta n}\}$ and $\{y_1, \ldots y_{\beta n}\}$. We randomly pair the elements of $L$ into $\beta n$ pairs. Let $(x_1, y_1), \ldots, (x_{\beta n}, y_{\beta n})$ be a random pairing of the elements in $L$, which is represented as $P_L$, that is, $P_L = \{(x_1, y_1), \ldots, (x_{\beta n}, y_{\beta n})\}$.

- The probability mass of $D_{no}$ at $z$ is defined as follows:

  - If $z \notin L$, then $D_{no}(z) = D_{yes}(z)$.
  - For every pair $(x_i, y_i) \in P_L$, $D_{no}(x_i) = D_{yes}(x_i) + D_{yes}(y_i)$, and $D_{no}(y_i) = 0$.

We start by observing that the distribution $D_{no}$ constructed above is supported on a set of at most $(1-\beta)n$ elements. So, any distribution $D_{no}$ constructed using the above procedure is $\varepsilon$-far from satisfying the property $\mathcal{P}$ for any $\varepsilon < \alpha$.

We will now prove that $D_{yes}$ and $D_{no}$ both have similar distributions over the sequences of samples. More formally, we will prove that any algorithm that takes $o(\sqrt{n})$ samples, cannot distinguish between $D_{yes}$ from $D_{no}$ with probability at least $\frac{2}{3}$.

Since any $D_{no}$ produced using the above procedure has exactly the same probability mass on the elements in $H$ as $D_{yes}$, any tester that distinguishes between $D_{yes}$ and $D_{no}$ must rely on samples obtained from $L$. Recall that the algorithm is given a uniformly random permutation of the distribution. Since $\mathrm{supp}(1)(D_{no}) \subset \mathrm{supp}(1)(D_{yes})$ (particularly, $\mathrm{supp}(1)(D_{no}) \cap L \subset \mathrm{supp}(1)(D_{yes}) \cap L$), it is not possible to distinguish between $D_{yes}$ and $D_{no}$, unless an element of $L$ appears at least twice. Otherwise, as in the proof of Lemma 4.4, the elements drawn from $L$ are distributed identically to a uniformly random non-repeating sequence. But observe that $D_{yes}(i) = \mathcal{O}(\frac{1}{n})$ and $D_{no}(i) = \mathcal{O}(\frac{1}{n})$ when $i$ is in $L$. Thus any sequence of $o(\sqrt{n})$ samples will provide only a distance of $o(1)$ between the two distributions, completing the proof. □

### 5.2.2 Tolerant lower bound for label-invariant properties

**Theorem 5.5 (Theorem 5.2 restated).** *Let $\mathcal{P}$ be any $(\alpha, \beta)$-non-concentrated label-invariant distribution property, where $0 < \alpha \leq \beta < \frac{1}{2}$. For any constant $\varepsilon_1$ and $\varepsilon_2$ with $0 < \varepsilon_1 < \varepsilon_2 < \alpha$, any $(\varepsilon_1, \varepsilon_2)$-tester for $\mathcal{P}$ requires $\Omega(n^{1-o(1)})$ samples, where $n$ is the size of the support of the distribution.*

To prove the above theorem, we recall some notions and a theorem from Valiant's paper on a lower bound for the sample complexity of tolerant testing of symmetric properties [Val11]. These definitions refer to invariants of distributions, which are essentially a generalization of properties.

**Definition 5.6.** Let $\Pi : \mathcal{D}_n \to \mathbb{R}$ denote a real-valued function over the set $\mathcal{D}_n$ of all distributions over $[n]$.

1. $\Pi$ is said to be *label-invariant* if for any $D \in \mathcal{D}_n$ the following holds: $\Pi(D) = \Pi(D_\sigma)$ for any permutation $\sigma : [n] \to [n]$.

2. For any $\gamma, \delta$ with $\gamma \geq 0$ and $\delta \in [0, 2]$, $\Pi$ is said to be $(\gamma, \delta)$-*weakly-continuous* if for all distributions $p^+, p^-$ satisfying $||p^+ - p^-||_1 \leq \delta$, we have $|\Pi(p^+) - \Pi(p^-)| \leq \gamma$.

For a property $\mathcal{P}$ of distributions, we define $\Pi_{\mathcal{P}} : \mathcal{D}_n \rightarrow [0, 2]$ with respect to property $\mathcal{P}$ as follows:

$$\text{For } D \in \mathcal{D}_n, \Pi_{\mathcal{P}}(D) := \text{the distance of } D \text{ from } \mathcal{P}.$$

From the triangle inequality property of $\ell_1$ distances, $\Pi_{\mathcal{P}}$ (which refers to the distance function from the property $\mathcal{P}$) is $(\gamma, \gamma)$-weakly continuous, for any $\gamma \in [0, 2]$.

**Theorem 5.7 (Low Frequency Blindness [Val11]).** *Consider a function* $\Pi : \mathcal{D}_n \rightarrow \mathcal{R}$ *that is label-invariant and* $(\gamma, \delta)$*-weakly-continuous, where* $\gamma \geq 0$ *and* $\delta \in [0, 2]$*. Let there exist two distributions* $p^+$ *and* $p^-$ *in* $\mathcal{D}_n$ *with* $n$ *being the size of their supports, such that* $\Pi(p^+) > b$*,* $\Pi(p^-) < a$*, and they are identical for any index occurring with probability at least* $\frac{1}{n}$ *in either distribution, where* $a, b \in \mathbb{R}$*. Then any tester that has sample access to an unknown distribution* $D$ *and distinguishes between* $\Pi(D) > b - \gamma$ *and* $\Pi(D) < a + \gamma$*, requires* $\Omega(n^{1-o_\delta(1)})$ *samples from* $D$ [1]*.*

Note that in Theorem 5.7, we have assumed that $p^+$ and $p^-$ are identical for any index that has probability mass at least $\frac{1}{n}$. We can actually replace this condition to $\mathcal{O}(\frac{1}{n})$ by adding $\mathcal{O}(n)$ "dummy elements" to the support of $p^+$ and $p^-$. Now we are ready to prove Theorem 5.5.

*Proof of Theorem 5.5.* Consider $\Pi_{\mathcal{P}}$ as defined above. As $\mathcal{P}$ is a label-invariant property, the function $\Pi_{\mathcal{P}}$ is also label-invariant. We have already noted that $\Pi_{\mathcal{P}}$ is $(\gamma, \gamma)$-weakly continuous as "*distance from a property*" satisfies the triangle inequality, for any $\gamma \in [0, 2]$. Now recall that the distributions $D_{yes}$ and $D_{no}$ considered in the proof of Theorem 5.3. The probability mass of each element in the support of $D_{yes}$ and $D_{no}$ is $\mathcal{O}(\frac{1}{n})$. Note that $D_{yes}$ is in $\mathcal{P}$ and $D_{no}$ is $\varepsilon$-far from $\mathcal{P}$, for any $\varepsilon < \alpha$, and both of them have a support size of $\Theta(n)$. Here we take $\varepsilon > \varepsilon_2$. Now, we apply Theorem 5.7 with $a = 0$, some $b < \varepsilon$ and $\gamma$ with $\gamma < \min\{\varepsilon_1, \varepsilon - \varepsilon_2\}$. Observe that this completes the proof of Theorem 5.5. □

[1] $o_\delta(\cdot)$ suppresses a term in $\delta$.

## 5.3 Sample complexity of non-concentrated properties

**Theorem 5.8 (Theorem 5.1 restated).** *Let $\mathcal{P}$ be any $(\alpha, \beta)$-non-concentrated distribution property for $0 < \alpha < \beta < \frac{1}{2}$. For any $\varepsilon$ with $0 < \varepsilon < \alpha$, any $(0, \varepsilon)$-tester for $\mathcal{P}$ requires $\Omega(\sqrt{n})$ samples, where $n$ is the size of the support of the distribution.*

**Why does the proof of Theorem 5.3 work only for label-invariant properties?** Note that the proof of Theorem 5.3 crucially uses the fact that the property $\mathcal{P}$ is label-invariant. Recall that, while constructing $D_{no}$ from $D_{yes}$, for each $i \in [\beta n]$, moving the masses of both $x_i$ and $y_i$ in $D_{yes}$ to $x_i$ to produce $D_{no}$ is possible as the property $\mathcal{P}$ is label-invariant. Due to this feature, we can apply a random permutation over $\Omega$, and still the permuted distribution will behave identically with respect to $\mathcal{P}$. After applying the random permutation, the samples coming from $D_{yes}$ and $D_{no}$ are indistinguishable as long as there are no collisions among the elements in $L$, which is the case when we take $o(\sqrt{n})$ samples. However, this technique does not work when the property is not label-invariant, as the value of the distribution with respect to $\mathcal{P}$ may not be invariant under the random permutation over $\Omega$. This requires a new argument; although the proof is similar in spirit to the proof of Theorem 5.3, there are some crucial differences, and we present the proof next. In order to prove Theorem 5.8, instead of moving the masses of both $x_i$ and $y_i$ in $D_{yes}$ to $x_i$ to produce $D_{no}$, we randomly move the sum to either $x_i$ or $y_i$, with probability proportional to the masses of $x_i$ and $y_i$.

### Proof of Theorem 5.8

The proof of Theorem 5.8 starts off identically to the proof of Theorem 5.3, but there is a departure in the construction of $D_{yes}$ and $D_{no}$.

Let us first consider $D_{yes}$, $L$ and $P_L$ as discussed in the proof of Theorem 5.3, only here we cannot and will not pass $D_{yes}$ through a random permutation. The difference starts from the description of the distribution $D_{no}$. In fact, $D_{no}$ will be randomly chosen according to a distribution over a set of distributions, all of which are not $(\alpha, \beta)$-non-concentrated. The distribution $D_{no}$ is generated using the following random process:

- We partition $L$ arbitrarily into two equal sets of size $\beta n$. Let they be $\{x_1, \ldots, x_{\beta n}\}$ and $\{y_1, \ldots, y_{\beta n}\}$. We pair the elements of $L$ arbitrarily into $\beta n$ pairs. Let $(x_1, y_1), \ldots, (x_{\beta n}, y_{\beta n})$ be an arbitrary pairing of the elements in $L$. Let $P_L$ be the set of pairs. So $P_L = \{(x_1, y_1), \ldots, (x_{\beta n}, y_{\beta n})\}$. We refer to $x_i$ and $y_i$ as the elements corresponding to each other with respect to $P_L$, and denote $\pi(x_i) = y_i$ and $\pi(y_i) = x_i$.
- The probability mass of $D_{no}$ at $z$ is defined as follows:
  - If $z \notin L$, then $D_{no}(z) = D_{yes}(z)$.
  - For every pair $(x_i, y_i) \in P_L$, use independent random coins and
    - With probability $\frac{D_{yes}(x_i)}{D_{yes}(x_i)+D_{yes}(y_i)}$, set $D_{no}(x_i) = D_{yes}(x_i) + D_{yes}(y_i)$ and $D_{no}(y_i) = 0$.
    - With the remaining probability, that is, with probability $\frac{D_{yes}(y_i)}{D_{yes}(x_i)+D_{yes}(y_i)}$, set $D_{no}(x_i)$
      $= 0$ and $D_{no}(y_i) = D_{yes}(x_i) + D_{yes}(y_i)$.

Observe that any $D_{no}$ constructed by the above procedure is supported over a set of at most $(1-\beta)n$ elements. So any distribution $D_{no}$ constructed using the above procedure is $\varepsilon$-far from satisfying the property $\mathcal{P}$, for any $\varepsilon < \alpha$. But since any $D_{no}$ produced using the above procedure has exactly the same probability mass on elements in $H$ as $D_{yes}$, any tester that distinguishes between $D_{yes}$ and $D_{no}$ must rely on samples obtained from $L$. However, we can prove that unless we receive two samples from the same pair in $L$ (which occurs with low probability), the sample sequence cannot distinguish $D_{yes}$ from $D_{no}$.

Note that there is an upper bound of $\mathcal{O}(\frac{1}{n})$ on the probability mass of any element in $L$. In fact, for any pair $(x_i, y_i) \in P_L$, the total probability mass of the pair is at most $\mathcal{O}(\frac{1}{n})$.

**Observation 5.9** (Follows from Observation 5.4)**.** For all pairs $(x_i, y_i) \in P_L$, $D_{no}(x_i) + D_{no}(y_i) \leq 2\frac{1-2\alpha}{1-2\beta}\frac{1}{n}$. Also note that $D_{no}(x_i) + D_{no}(y_i) = D_{yes}(x_i) + D_{yes}(y_i)$ with probability $1$ over the construction of $D_{no}$.

From Observation 5.9, observe that if $S$ is a multi-set of $o\left(\sqrt{\frac{1-2\beta}{1-2\alpha}n}\right)$ samples from $D_{yes}$, then with probability $1 - o(1)$, no two elements in $S$ (identical or not) are from the same pair in $P_L$. The same holds for $D_{no}$ as well. Given that no two elements in $S$ are from the same pair in $P_L$, we will now prove that $D_{yes}$ and $D_{no}$ have the same distributions over sample sequences. This implies that, for a sequence of $o\left(\sqrt{\frac{1-2\beta}{1-2\alpha}n}\right)$ samples, $D_{yes}$ and $D_{no}$ induce distributions over samples sequences that have $o(1)$ variation distance from each other.

Note that under the condition that at most one element is drawn from any pair $(x_i, y_i) \in P_L$, the probability that the sample is $x_i$ instead of $y_i$ is equal to $\frac{D_{yes}(x_i)}{D_{yes}(x_i)+D_{yes}(y_i)}$, irrespective of whether the distribution is $D_{yes}$ or $D_{no}$. So, we have the following lemma.

**Lemma 5.10.** *Let $a_1, ..., a_q$ be a sequence of $q$ elements, where no element of $L$ appears twice, additionally containing no two elements from the same pair in $P_L$ (elements of $H$ can appear freely). Then*

$$\Pr_{s_1,\ldots,s_q \sim D_{yes}}[(s_1, \ldots, s_q) = (a_1, \ldots, a_q)] = \Pr_{s_1,\ldots,s_q \sim D_{no}}[(s_1, \ldots, s_q) = (a_1, \ldots, a_q)].$$

*Proof.* Let us begin by defining an event $\mathcal{E}$ as follows:

$\mathcal{E} :=$ no element of L appears twice, and no two elements from the same pair appear.

Observe that we will be done by proving

$$\Pr_{s_1,\ldots,s_q \sim D_{yes}}[s_i = a_i \text{ for each } i \in [q] \mid \mathcal{E}] = \Pr_{s_1,\ldots,s_q \sim D_{no}}[s_i = a_i \text{ for each } i \in [q] \mid \mathcal{E}]. \quad (5.1)$$

We will prove this by using induction over $q$. Let us assume that we have generated samples $s_1 = a_1, \ldots, s_k = a_k$ from the unknown distribution, where $1 \leq k < q$. Let $X_k = \{s_1, \ldots, s_k\} \cap L$ be the samples we have seen until now from $L$, and $X'_k = \{\pi(x) : x \in X_k\}$. By the induction hypothesis, assume that Equation (5.1) holds for each $q$ with $q \leq k$. We will show that Equation (5.1) holds for $q = k + 1$.

To do so, let us now define two distributions $D^{k+1}_{yes}$ and $D^{k+1}_{no}$ as follows, and consider a claim (Claim 5.11) about them.

$$D_{yes}^{k+1}(x) = \Pr_{s_1,\ldots,s_q \sim D_{yes}} \left[s_{k+1} = x \mid \mathcal{E} \text{ and } s_i = a_i \text{ for } i \le k\right].$$

Similarly,

$$D_{no}^{k+1}(x) = \Pr_{s_1,\ldots,s_q \sim D_{no}} [s_{k+1} = x \mid \mathcal{E} \text{ and } s_i = a_i \text{ for } i \le k].$$

**Claim 5.11.** *$D_{yes}^{k+1}(x) = D_{no}^{k+1}(x)$ for every $x \in \Omega$.*

*Proof.* We prove the claim separately when $x \in X_k \cup X'_k \subseteq L$, $x \in L \setminus (X_k \cup X'_k)$, and $x \notin L$.

**(i)** $x \in X_k \cup X'_k$**:** $D_{yes}^{k+1}(x) = D_{no}^{k+1}(x) = 0$. This follows from the condition that no element of $L$ appears twice, additionally containing no two elements of the same pair.

**(ii)** $x \in L \setminus (X_k \cup X'_k)$**:** As $D_{yes}^{k+1}(x) = D_{no}^{k+1}(x) = 0$ for every $x \in X_k \cup X'_k$, we have the followings for each $x \in L \setminus (X_k \cup X'_k)$.

Assume that $x = x_i \in L \setminus (X_k \cup X'_k)$ for some $i \in [\beta n]$ (using the notation defined for the partition of $L$ into pairs while we have described the random process for generating $D_{no}$). The argument for the case where $x = y_j$ for some $j \in [\beta n]$ is analogous to this.

Under $D_{yes}$, a direct calculation gives the probability for obtaining $x = x_i \in L \setminus (X_k \cup X'_k)$ as the $(k+1)$-th sample $s_{k+1}$.

$$\begin{aligned} D_{yes}^{k+1}(x) &= D_{yes}(x \mid x \notin X_k \cup X'_k) \\ &= \frac{D_{yes}(x)}{1 - \sum_{y \in X_k \cup X'_k} D_{yes}(y)} \\ &= \frac{D_{yes}(x)}{1 - \sum\limits_{y \in X_k} (D_{yes}(y) + D_{yes}(\pi(y)))}, \end{aligned}$$

Let us now consider $D_{no}$. Note that $x_i \in L \setminus (X_k \cup X'_k)$, and neither $x_i$ nor $y_i$ is present in the set of first $k$ samples $\{s_1, \ldots, s_k\}$. So, the probability of getting $s_1, \ldots, s_k$ as the sequence of first $k$ samples is completely independent of how $D_{no}(x_i)$ and $D_{no}(y_i)$ are assigned while generating $D_{no}$, that is, whether we chose $D_{no}(x_i)$ to be $D_{yes}(x_i) + D_{yes}(y_i)$, or chose it to be zero (and made $D_{no}(y_i)$ equal to $D_{yes}(x_i) + D_{yes}(y_i)$ instead). That is, even when conditioned on the event that $s_1, \ldots, s_k$ is the sequence of first $k$ samples, the probability that $D_{no}(x_i)$ is $D_{yes}(x_i) + D_{yes}(y_i)$ is $\frac{D_{yes}(x_i)}{D_{yes}(x_i)+D_{yes}(y_i)}$. Note that $D_{no}(x_i)$ is $0$ with probability $\frac{D_{yes}(y_i)}{D_{yes}(x_i)+D_{yes}(y_i)}$.

Now we can calculate the probability of obtaining $x = x_i \in L$ as the $(k+1)$-th sample $s_k$ from the corresponding conditional probabilities.

$$\begin{aligned}
D_{no}^{k+1}(x) &= D_{no}\left(x \mid x \notin X_k \cup X'_k\right) \\
&= \frac{D_{yes}(x_i) + D_{yes}(y_i)}{1 - \sum\limits_{y \in X_k \cup X'_k} D_{no}(y)} \cdot \frac{D_{yes}(x_i)}{D_{yes}(x_i) + D_{yes}(y_i)} \\
&= \frac{D_{yes}(x)}{1 - \sum\limits_{y \in X_k} \left(D_{no}(y) + D_{no}(\pi(y))\right)}.
\end{aligned}$$

From the construction of $D_{yes}$ and $D_{no}$, for each $y \in L$, $D_{yes}(y) + D_{yes}(\pi(y)) = D_{no}(y) + D_{no}(\pi(y))$. As $X_k \subseteq L$,

$$\sum_{y \in X_k} \left(D_{yes}(y) + D_{yes}(\pi(y))\right) = \sum_{y \in X_k} \left(D_{no}(y) + D_{no}(\pi(y))\right).$$

Hence, we have $D_{yes}^{k+1}(x) = D_{no}^{k+1}(x)$.

**(iii)** $x \notin L$**:** Recall that for any $x \notin L$, $D_{yes}(x) = D_{no}(x)$. Proceeding in similar fashion to $D_{yes}^{k+1}(x)$ in Case $(ii)$, we conclude that $D_{yes}^{k+1}(x) = D_{no}^{k+1}$.

□

Now we are ready to prove Equation (5.1) for $q = k + 1$.

$$\begin{aligned}
& \Pr_{s_1,\ldots,s_{k+1}\sim D_{yes}}[s_i = a_i \; \forall i \in [k+1] \mid \mathcal{E}] \\
= \; & \Pr_{s_1,\ldots,s_{k+1}\sim D_{yes}}[s_i = a_i \; \forall i \in [k] \mid \mathcal{E}] \cdot \Pr[s_{k+1} = a_{k+1} \mid \mathcal{E} \; \& \; s_i = a_i \; \forall i \in [k]] \\
= \; & \Pr_{s_1,\ldots,s_k\sim D_{yes}}[s_i = a_i \; \forall i \in [k] \mid \mathcal{E}] \cdot D_{yes}^{k+1}(a_{k+1}) \\
& \hfill \text{(By the definition of } D_{yes}^{k+1}) \\
= \; & \Pr_{s_1,\ldots,s_k\sim D_{no}}[s_i = a_i \; \forall i \in [k] \mid \mathcal{E}] \cdot D_{no}^{k+1}(a_{k+1}) \\
& \hfill \text{(By the induction hypothesis and Claim 5.11, respectively)} \\
= \; & \Pr_{s_1,\ldots,s_k\sim D_{no}}[s_i = a_i \; \forall i \in [k] \mid \mathcal{E}] \cdot \Pr[s_{k+1} = a_{k+1} \mid \mathcal{E} \; \& \; s_i = a_i \; \forall i \in [k]] \\
= \; & \Pr_{s_1,\ldots,s_{k+1}\sim D_{no}}[s_i = a_i \; \forall i \in [k+1] \mid \mathcal{E}].
\end{aligned}$$

□

Following the construction of $D_{yes}$ and $D_{no}$, we know that the two distributions differ only on the elements of $L$. Moreover, following Observation 5.9, we know that if we take $o\left(\sqrt{\frac{1-2\beta}{1-2\alpha}n}\right)$ samples, then with probability $1 - o(1)$, neither any element of $L$ will appear more than once nor two elements of same pair in $P_L$ will appear. Under these two conditions, Lemma 5.10 states that $D_{yes}$ and $D_{no}$ will appear to be the same. Thus we can say that any $(0, \epsilon)$-tester that receives $o\left(\sqrt{\frac{1-2\beta}{1-2\alpha}n}\right)$ samples cannot distinguish between $D_{yes}$ and $D_{no}$, and obtain Theorem 5.8.

# Chapter 6

# Distribution Learning with Unknown Support

## 6.1 Introduction

In this chapter, we prove an upper bound related to the tolerant testing of more general properties. We prove that for concentrated distributions (distributions whose most of the mass is over a subset $S \subseteq [n]$), the distribution can be learnt efficiently, even when the set as well as its size over which the distribution is concentrated are unknown. Formally, we have the following result:

**Theorem 6.1 (Theorem 3.5 formalized).** *Let $D$ denote the unknown distribution over $\Omega = [n]$, and assume that there exists a set $S \subseteq [n]$ with $D(S) \geq 1 - \frac{\eta}{2}$*[1]*, where $\eta \in [0, 2)$ is known but $S$ and $|S|$ are unknown. Then there exists an algorithm that takes $\delta \in (0, 2]$ as input and constructs a distribution $D'$ satisfying $||D - D'||_1 \leq \eta + \delta$ with probability at least $\frac{2}{3}$. Moreover, the algorithm uses, in expectation, $\mathcal{O}\left(\frac{|S|}{\delta^2}\right)$ samples from $D$.*

[1] Recall that the variation distance between two distribution is half than that of the $\ell_1$ distance between them. So, we take $D(S) \geq 1 - \frac{\eta}{2}$ (with $\eta \in [0, 2)$) instead of $D(S) \geq 1 - \eta$ (with $\eta \in [0, 1)$) .

## 6.2 Learning distributions with unknown support

Following a folklore result, when provided with oracle access to an unknown distribution $D$, we can always construct a distribution $D'$, such that the $\ell_1$ distance between $D'$ and $D$ (the unknown distribution) is at most $\varepsilon$, by using $\mathcal{O}(\frac{n}{\varepsilon^2})$ samples from $D$ [2]. In this section, we provide a procedure that can be used for tolerant testing of properties, and in particular hints at how general tolerance gap bounds could be proved in the future. Our algorithm learns an unknown distribution approximately with high probability, adapting to the input, using as few samples as possible. Specifically, we prove that given a distribution $D$, if there exists a subset $S \subseteq [n]$ which holds most of the total probability mass of $D$, then the distribution $D$ can be learnt using $\mathcal{O}(|S|)$ samples, even if the algorithm is unaware of $|S|$ in advance. Our result is formally stated as follows:

**Theorem 6.2 (Theorem 6.1 restated).** *Let $D$ denote the unknown distribution over $\Omega = [n]$, and assume that there exists a set $S \subseteq [n]$ with $D(S) \geq 1 - \frac{\eta}{2}$[3], where $\eta \in [0, 2)$ is known but $S$ and $|S|$ are unknown. Then there exists an algorithm that takes $\delta \in (0, 2]$ as input and constructs a distribution $D'$ satisfying $||D - D'||_1 \leq \eta + \delta$ with probability at least $\frac{2}{3}$. Moreover, the algorithm uses, in expectation, $\mathcal{O}\left(\frac{|S|}{\delta^2}\right)$ samples from $D$.*

Note that in the above theorem, the algorithm has no prior knowledge of $|S|$. Before directly proving the above, we first show that if $|S|$ is known, then $\mathcal{O}(|S|)$ samples are enough to approximately learn the distribution $D$. We would like to point out that a similar question has been studied under the local differential privacy model with communication constraints, by Acharya, Kairouz, Liu and Sun [AKLS21] and by Chen, Kairouz and Özgür [CKÖ20].

**Lemma 6.3 (Theorem 6.2 when $|S|$ is known).** *Let $D$ be the unknown distribution over $\Omega = [n]$ such that there exists a set $S \subseteq [n]$ with $|S| = s$, and $\eta \in [0, 2)$ such that $D(S) \geq 1 - \frac{\eta}{2}$, where $s \in [n]$ and $\eta \in (0, 1)$ are known. Then there exists an*

[2] There is a writeup of this folklore result by Cannone [Can20b].

[3] Recall that the variation distance between two distribution is half than that of the $\ell_1$ distance between them. So, we take $D(S) \geq 1 - \frac{\eta}{2}$ (with $\eta \in [0, 2)$) instead of $D(S) \geq 1 - \eta$ (with $\eta \in [0, 1)$) .

*algorithm that takes $\delta \in (0,2]$ as an input and constructs a distribution $D'$ satisfying $||D - D'||_1 \leq \eta + \delta$ with probability at least $\frac{9}{10}$. Moreover, the algorithm uses $\mathcal{O}\left(\frac{s}{\delta^2}\right)$ samples from $D$.*

We note that Lemma 6.3 can be obtained from the work of Acharya, Diakonikolas, Li and Schmidt [ADLS17, Theorem 2]. For completeness, we give a self-contained proof for this lemma below.

We later adapt the algorithm of Lemma 6.3 to give a proof to the scenario where $|S|$ is unknown, using a guessing technique. The idea is to guess $|S| = s$ starting from $s = 1$, and then to query for $\mathcal{O}(s)$ samples from the unknown distribution $D$. From the samples obtained, we construct a distribution $D_s$, and use Lemma 6.4 presented below to distinguish whether $D_s$ and $D$ are close or far. We argue that, for $s \geq |S|$, $D_s$ will be close to $D$ with probability at least $\frac{9}{10}$. We bound the total probability for the algorithm reporting a $D'$ that is too far from $D$ (for example when terminating before $s \geq |S|$), and also bound the probability of the algorithm not terminating in time when $s$ becomes at least as large as $|S|$.

**Lemma 6.4** ([VV11]). *Let $D_u$ and $D_k$ denote unknown (input) and known (given in advance) distributions respectively over $\Omega = [n]$, such that the support of $D_u$ is a set of $s$ elements of $[n]$. Then there exists an algorithm* TOL-ALG$(D_u, D_k, \varepsilon_1, \varepsilon_2, \kappa)$ *that takes the full description of $D_k$, two proximity parameters $\varepsilon_1, \varepsilon_2$ with $0 \leq \varepsilon_1 < \varepsilon_2 \leq 2$ and $\kappa \in (0,1)$ as inputs, queries $\mathcal{O}\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\frac{s}{\log s}\log\frac{1}{\kappa}\right)$ samples from $D_u$, and distinguishes whether $||D_u - D_k||_1 \leq \varepsilon_1$ or $||D_u - D_k||_1 \geq \varepsilon_2$ with probability at least $1 - \kappa$* [4].

Note that Theorem 6.2 talks about learning a distribution with $\mathcal{O}(|S|)$ samples, where there exists an unknown set $S$ with $D(S) \geq 1 - \eta/2$. To prove Theorem 6.2, we use Lemma 6.4, that crucially uses less than $s$ queries for tolerant identity testing (as opposed to learning).

The bound following the paper of Valiant and Valiant [VV11] is $\mathcal{O}\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\frac{n}{\log n}\right)$, which holds for any general distributions $D_u$ and $D_k$ with constant success probability. When deploying Lemma 6.4, we "contract" the set $\Omega \setminus \mathrm{supp}(1)(D_k)$ to a single

[4] The multiplicative factor $\log\frac{1}{\kappa}$ is for amplifying the success probability from $\frac{2}{3}$ to $1 - \kappa$.

element, which allows us to substitute $s+1$ for $n$. Note that this does not change the distance between $D_k$ and $D_u$. Hence, $\mathcal{O}\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\frac{s}{\log s}\right)$ samples from $D_u$ are enough for constant success probability. Following a recent work of Cannone, Jain, Kamath and Li [CJKL22], the dependence on the proximity parameters can be slightly improved. However we are not using that result since the focus of this work is different.

We first prove Lemma 6.3, and then proceed to prove Theorem 6.2.

*Proof of Lemma 6.3.* Let $Z$ be a multi-set of $\mathcal{O}\left(\frac{s}{\delta^2}\right)$ samples taken from $D$. The algorithm constructs a distribution $D' : [n] \to [0,1]$ such that

$$D'(x) = \frac{\#\text{ times } x \text{ appears in } Z}{|Z|}.$$

Observe that $||D-D'||_1 = 2\max\limits_{E\subseteq[n]}|D(E)-D'(E)|$. So, we will be done by showing the following:

$$\text{With probability at least } \tfrac{9}{10},\ |D(E)-D'(E)| \leq \tfrac{\eta+\delta}{2} \text{ for all } E \subseteq [n] \qquad (6.1)$$

Note that there are $2^n$ possibilities for $E$. So, a direct application of the union bound would require a failure probability of at most $\mathcal{O}(\frac{1}{2^n})$ for each $E$ not satisfying $|D(E)-D'(E)| \leq \frac{\eta+\delta}{2}$, that is, $\mathcal{O}(\frac{n}{\delta^2})$ samples would be needed. Assuming that $D$ is concentrated ($D(S) \geq 1-\frac{\eta}{2}$), we argue below that it is enough to have a failure probability of $\mathcal{O}(\frac{1}{2^s})$ for each $T$ not satisfying $|D(T)-D'(T)| \leq \frac{\delta}{4}$, but first we show that this is indeed the probability that we achieve.

**Observation 6.5.** Consider $T \subseteq [n]$. $|D(T)-D'(T)| \leq \frac{\delta}{4}$ holds with probability at least $1-\frac{1}{100\cdot 2^s}$.

*Proof.* Let $X_i$ denote the binary random variable that takes value $1$ if and only if the $i$-th sample in $Z$ is an element of $T$, where $i \in [|Z|]$. So, $D'(T) = \frac{1}{|Z|}\sum\limits_{i=1}^{|Z|} X_i$.

Observe that the expectation of $D'(T)$ is $\mathbb{E}\left[D'(T)\right] = D(T)$. Applying Chernoff bound (Lemma 2.12), we get the desired result. □

By the above observation for every subset of $S$, applying the union bound over all possible subsets of $S$, we have $|D(T) - D'(T)| \leq \frac{\delta}{4}$ for every $T \subseteq S$ with probability at least $\frac{99}{100}$. Further applying the observation for $T = \Omega \setminus S$, we have $|D(\Omega \setminus S) - D'(\Omega \setminus S)| \leq \frac{\delta}{4}$ with probability at least $1 - \frac{1}{100 \cdot 2^s}$.

Let $\mathcal{E}$ be the event that $|D(T) - D'(T)| \leq \frac{\delta}{4}$ for every $T \subseteq S$, and $|D(\Omega \setminus S) - D'(\Omega \setminus S)| \leq \frac{\delta}{4}$. Note that $\Pr(\mathcal{E}) \geq \frac{9}{10}$. So, to prove Equation (6.1) and conclude the proof of Lemma 6.3, we show that $|D(E) - D'(E)| \leq \frac{\eta+\delta}{2}$ holds, in the conditional probability space when $\mathcal{E}$ occurs, for any $E \subseteq [n]$.

$$\begin{aligned}
|D(E) - D'(E)| &\leq |D(E \cap S) - D'(E \cap S)| + |D(E \cap (\Omega \setminus S)) - D'(E \cap (\Omega \setminus S))| \\
&\leq \frac{\delta}{4} + \max\{D(\Omega \setminus S), D'(\Omega \setminus S)\} \\
&\leq \frac{\delta}{4} + D(\Omega \setminus S) + \frac{\delta}{4} \\
&\leq \frac{\eta+\delta}{2}.
\end{aligned}$$

□

*Proof of Theorem 6.2.* The algorithm is as follows:

1. Set $s = 1$.

2. Query for a multi-set $Z_s$ of $\mathcal{O}\left(\frac{s}{\delta^2}\right)$ samples from $D$.

3. Construct a distribution $D_s : [n] \to [0, 1]$ such that

$$D_s(x) = \frac{\#\text{ times } x \text{ appears in } Z_s}{|Z_s|}$$

4. Call the algorithm TOL-ALG $\left(D_s, D, \eta + \frac{\delta}{2}, \eta + \delta, \frac{1}{100 \log^2 s}\right)$ (corresponding to Lemma 6.4) to distinguish whether $||D - D_s||_1 \leq \eta + \frac{\delta}{2}$ or $||D - D_s||_1 \geq \eta + \delta$. If we get $||D - D_s||_1 \leq \eta + \frac{\delta}{2}$ as the output of TOL-ALG, then we report $D'$ as the output and QUIT. Otherwise, we double the value of $s$. If $s \leq 2n$, go back to

Step 2. Otherwise, report FAILURE [5].

Let $\mathcal{S}$ denote the event that the algorithm quits with the desired output. We first show that $\Pr(S) \geq \frac{2}{3}$. Then we analyze the expected sample complexity of the algorithm.

Observe that the algorithm quits after an iteration with guess $s$ such that ALG-TOL reports $||D - D_s||_1 \leq \eta + \frac{\delta}{2}$. So, in that case, the probability that the algorithm exits with an output not satisfying $||D - D_s||_1 \leq \eta + \delta$ is at most $\frac{1}{100 \log^2 s}$. When summing this up over all possible $s$ (all powers of $k$, even up to infinity), the probability that the algorithm does not produce the desired output, given that it quits, is at most $\sum_{k=1}^{\infty} \frac{1}{100k^2} \leq \frac{1}{10}$. So, denoting $\mathcal{Q}$ as the event that the algorithm quits without reporting FAILURE, $\Pr(\mathcal{S} \mid \mathcal{Q}) \geq \frac{9}{10}$.

For the lower bound on $\Pr(\mathcal{Q})$, consider the case where $s \geq |S|$. In this case, $||D_s - D||_1 \leq \eta + \frac{\delta}{2}$ with probability at least $\frac{9}{10}$, and TOL-ALG quits by reporting $D_s$ as the output with probability at least $1 - \frac{1}{100 \log^2 s}$. So, for any guess $s \geq |S|$, the algorithm quits and reports the desired output with probability at least $\frac{4}{5}$. So, the probability that the algorithm quits without reporting failure is at least the probability that the algorithm quits with a desired output at some iteration with a guess $s \geq |S|$, which is at least $1 - (\frac{1}{5})^{(\log n - \log |S| + 1)}$. That is, $\Pr(\mathcal{Q}) \geq \frac{4}{5}$.

Hence, the success probability of the algorithm can be lower-bounded as

$$\Pr(\mathcal{S}) \geq \Pr(\mathcal{Q}) \cdot \Pr(\mathcal{S} \mid \mathcal{Q}) \geq \frac{9}{10} \cdot \frac{4}{5} > \frac{2}{3}.$$

Now, we analyze the sample complexity of the algorithm. The algorithm queries for $\mathcal{O}(s)$ samples when it runs the iteration whose guess is $s$. The algorithm goes to the iteration with guess $s > |S|$ if all prior iterations which guessed more than $|S|$ failed, which holds with probability at most $\mathcal{O}\left((\frac{1}{5})^{\lfloor \log s/|S| \rfloor}\right)$. Hence the expected sample complexity of the algorithm is at most

$$\sum_{k: s=2^k < |S|} \mathcal{O}(\frac{s}{\delta^2}) + \sum_{k: s=2^k \geq |S|} \mathcal{O}\left(\left(\frac{1}{5}\right)^{\lfloor \log(s/|S|) \rfloor} \cdot \frac{s}{\delta^2}\right) = \mathcal{O}(\frac{|S|}{\delta^2}).$$

[5] By Lemma 6.4, this step uses $\mathcal{O}(\frac{s}{\delta^2})$ samples.

To explain the above equality, note that in the LHS of the above equation, each term of the second sum is bounded by $\mathcal{O}((\frac{1}{5})^{(k-\log|S|)} \cdot 2^{(k-\log|S|)} \cdot \frac{|S|}{\delta^2})$. Thus, substituting $k - \log(|S|)$ by $r$, we see that the second part of the LHS is upper bounded by $\sum\limits_{r \geq 0} \mathcal{O}\left((\frac{2}{5})^r \cdot \frac{|S|}{\delta^2}\right)$ which is clearly $\mathcal{O}(\frac{|S|}{\delta^2})$. Thus we have the above bound. □

# Part II

# Results in the Huge Object Model

# Chapter 7

# Testing in the Huge Object Model

## 7.1 Introduction

The field of distribution testing is currently ubiquitous in property testing, see the books and surveys of [Gol17, BY22, Fis04, Ron08, Ron09, CS10a, RS11, Can20a, Can22] for reference. Distribution testing has also found numerous applications in other areas of research, including topics that have real life applications [CM19, MPC20, CKS20, PM21].

In the original model of distribution testing, a distribution $D$ defined over some set $\Omega$ can be accessed by obtaining independent samples from $D$, and the goal is to approximate various interesting properties of $D$. This model has been studied extensively over the last two decades, and many interesting results and techniques have emerged.

The majority of distribution testing research centers on the goal of minimizing the number of samples required to test for various properties of the underlying distribution. If the domain of the distribution is structured (for example, if the domain is the $n$-dimensional Hamming cube $\{0,1\}^n$), then designing efficient testers brings its own challenges. A number of papers have studied the problem of testing properties of distributions defined over the $n$-dimensional Hamming cube (see [ABR16, CDKS17, BC18, BGMV20, CCK$^+$21, CJLW21, BCY22]). With the rise of big data (translating to $n$ being very large), even reading all the bits in the representation of the samples might be very expensive. To address this issue, recently Goldreich and Ron [GR22] studied distribution testing in a different setting.

In their model, called the *huge object model*, the distribution $D$ is supported over the $n$-dimensional Hamming cube $\{0,1\}^n$, and the tester will obtain $n$-length Boolean strings as samples. However, as reading the sampled strings in their entirety might be infeasible when $n$ is large, the authors in [GR22] considered query access to the samples along with standard sampling access. Note that without loss of generality, the number of samples will be upper-bounded by the number of queries. Thus, a desirable goal in this model is to optimize the number of queries for testing a given property, with respect to the Earth Mover Distance notion that befits this model. [GR22] studied various natural properties like support size estimation, uniformity, identity, equality, and "grainedness" [1] in this model, providing upper and lower bounds on the sample and query complexities for these properties.

In this chapter, we study the sample and query complexities of a very natural class of properties, which we call the *index-invariant properties*, in the huge object model of distribution testing.

**Index-Invariant Distribution Properties:** In general, a distribution property is a collection of distributions over a fixed domain $\Omega$ [2]. Often the property in question has some other "symmetry". For example, a property is called *label-invariant* if any changes in the labels of the domain do not affect whether the distribution is in the property or not. Many of the well studied properties, such as uniformity, entropy estimation, support size estimation, and grainedness, are label-invariant properties. Label-invariant properties have been studied extensively in literature [BDKR05, Pan08, GR11, Val11, DKN14, CDVV14, ADK15, VV17b, BC17, DKS18].

In some cases, the distribution property is not fully label-invariant, but still has a certain amount of symmetry. For illustration, consider the following examples:

[1] A distribution $D$ over $\{0,1\}^n$ is said to be $m$-grained if the probability mass of any element in its support is a multiple of $1/m$, where $m \in \mathbb{N}$.

[2] We use the phrases "a distribution is in the property" and "a distribution has the property" interchangeably to mean the same thing.

1. **Property** MONOTONE**:** Any distribution $D$ over $\{0,1\}^n$ satisfies the MONOTONE property if

$$\mathbf{X} \preceq \mathbf{Y} \text{ implies } D(\mathbf{X}) \leq D(\mathbf{Y}), \text{for any } \mathbf{X}, \mathbf{Y} \in \{0,1\}^n,$$

   where for two vectors $\mathbf{X}, \mathbf{Y} \in \{0,1\}^n$, $\mathbf{X} \preceq \mathbf{Y}$ if $x_i \leq y_i$ holds for every $i \in [n]$.

2. **Property** LOG-SUPER-MODULARITY**:** Any distribution $D$ over $\{0,1\}^n$ satisfies the LOG-SUPER- MODULARITY if

$$D(\mathbf{U})D(\mathbf{V}) \leq D(\mathbf{U} \wedge \mathbf{V})D(\mathbf{U} \vee \mathbf{V}), \text{for any } \mathbf{U}, \mathbf{V} \in \{0,1\}^n,$$

   where the Boolean $\wedge$ and $\vee$ operations over the vectors are performed coordinate-wise.

3. **Property** LOW-AFFINE-DIMENSION**:** A distribution $D$ over $\{0,1\}^n$ is said to satisfy the LOW-AFFINE-DIMENSION property, with parameter $d \in \mathbb{N}$, if the *affine dimension*[3] of the support of $D$ is at most $d$.

Note that for the properties described above, a distribution satisfies the above properties even after the indices $\{1, \ldots, n\}$ of the vectors in $\{0,1\}^n$ are permuted by a permutation $\sigma$ defined over $[n]$. To capture this structure in the properties, we introduce the notion of *index-invariant* properties.

**Definition 7.1 (Index-invariant property).** Let us assume that $D : \{0,1\}^n \to [0,1]$ is a distribution over the $n$-dimensional Hamming cube $\{0,1\}^n$. For any permutation $\sigma : [n] \to [n]$, let $D_\sigma$ be the distribution such that $D(w_1, \ldots, w_n) = D_\sigma(w_{\sigma(1)}, \ldots, w_{\sigma(n)})$ for all $(w_1, \ldots, w_n) \in \{0,1\}^n$. A distribution property $\mathcal{P}$ is said to be *index-invariant* when $D$ is in $\mathcal{P}$ if and only if $D_\sigma$ is in $\mathcal{P}$, for any distribution $D$ and any permutation $\sigma$.

Informally speaking, index-invariant properties refer to those properties that are invariant under the permutations of the indices $\{1, \ldots, n\}$. Note that this set of properties

[3] A set $S \subseteq \mathbb{R}^n$ has *affine dimension* $k$ if the dimension of the smallest *affine set* in $\mathbb{R}^n$ that contains $S$ is $k$.

differs from the more common notion of label-invariant properties, since the total number of possible labels, for distributions over all $n$-length Boolean vectors, is $2^n$. However, we are considering only permutations over $[n]$, thus in total only $n!$ permutations instead of $2^n!$ permutations.

### 7.1.1 Our results

In this part, as already mentioned, we study the sample and query complexities (in the huge object model) of index-invariant properties. We primarily focus on two problems. First, we study the connection between the query complexity for testing an index-invariant property and the VC-dimension of the non-trivial support of the distributions in the property. Secondly, we study the relationship between the query complexities of the adaptive and non-adaptive testers for index-invariant properties, along with their non-index-invariant counterparts.

One important and technical difference between the huge object model and the standard distribution property testing model is the use of *Earth Mover Distance* (EMD) for the notion of "closeness" and "farness", instead of the more prevalent $\ell_1$ or variation distance. Thus, in the rest of the chapter, by an $\varepsilon$-tester for any property $\mathcal{P}$ of distributions over $\{0,1\}^n$, we mean an algorithm that given sample and query access (to the bits of the sampled vectors) to a distribution distinguishes (with probability at least $2/3$) the case where the distribution $D$ is in the property $\mathcal{P}$ from the case where the EMD of $D$ from any distribution in $\mathcal{P}$ is at least $\varepsilon$, where $\varepsilon > 0$ is a proximity parameter.

#### Testing by learning of bounded VC-dimension properties (constant query testable properties):

We prove that a large class of distribution properties are all testable with a number of queries independent of $n$, using the *testing by learning paradigm* [DLM$^+$07, GOS$^+$09, Ser10], where the distributions are supported over the $n$-dimensional Hamming cube $\{0,1\}^n$. More specifically, we prove that every distribution whose support has a bounded VC-dimension can be *efficiently* learnt up to a permutation, leading to efficient testers for index-invariant distribution properties that admit a global VC-dimension bound. Our main result regarding the learning of distributions in the huge object model is the following theorem.

**Theorem 7.2 (Informal).** *For any fixed constant $d \in \mathbb{N}$, given sample and query access to an unknown distribution $D$ over $\{0,1\}^n$ and a proximity parameter $\varepsilon > 0$, there exists an algorithm that makes $\mathrm{poly}(\frac{1}{\varepsilon})$ queries* [4]*, and either outputs the full description of a distribution or* FAIL *satisfying the following conditions:*

**(i)** *If the support of $D$ is of VC-dimension at most $d$, then with probability at least $2/3$, the algorithm outputs a full description of a distribution $D'$ such that $D$ is $\varepsilon$-close to $D'_\sigma$ for some permutation $\sigma : [n] \to [n]$.*

**(ii)** *For any $D$, the algorithm will not output a distribution $D'$ such that $D'_\sigma$ is $\varepsilon$-far from $D$ for all permutations $\sigma : [n] \to [n]$, with probability more than $1/3$. However, if the VC-dimension of the support of $D$ is more than $d$, the algorithm may output* FAIL *with any probability.*

In fact, our result holds for a general class of *clusterable* properties (stated in Theorem 8.3 and Corollary 9.5) that also covers the VC-dimension case as stated in the above theorem. The result for learning clusterable distribution is stated as follows:

**Theorem 7.3 (Informal).** *Given sample and query access to an unknown distribution $D$ over $\{0,1\}^n$, there exists a non-adaptive algorithm that makes a number of queries that is independent of $n$, and either reports a full description of a distribution over $\{0,1\}^n$ or reports* FAIL*, satisfying both of the following conditions:*

**(i)** *If $D$ is clusterable, then with probability at least $\frac{2}{3}$, the algorithm outputs a full description of a distribution $D'$ such that $D$ is $\varepsilon$-close to $D'_\sigma$ for some permutation $\sigma : [n] \to [n]$.*

[4] The degree of the polynomial in $\frac{1}{\varepsilon}$ depends on the parameter $d$.

**(ii)** *For any $D$, the algorithm will not output a distribution $D'$ such that $D'_\sigma$ is $\varepsilon$-far from $D$ for every permutation $\sigma : [n] \to [n]$, with probability more than $\frac{1}{3}$. However, if the distribution $D$ is not clusterable, the algorithm may output* FAIL *with any probability.*

Note that Theorem 7.2 corresponds to the learnability of any distribution when the VC-dimension of its support is bounded. As a corollary, it implies that any index-invariant distribution property admitting a global VC-dimension bound is testable with a constant number of queries, depending only on the proximity parameter $\varepsilon$ and the VC-dimension $d$. The corollary is stated as follows:

**Corollary 7.4 (Informal).** *Let $\mathcal{P}$ be an index-invariant property such that any distribution $D \in \mathcal{P}$ has VC-dimension at most $d$, where $d$ is some constant. There exists an algorithm, that has sample and query access to an unknown distribution $D$ over $\{0,1\}^n$, takes a proximity parameter $\varepsilon > 0$, and distinguishes whether $D \in \mathcal{P}$ or $D$ is $\varepsilon$-far from $\mathcal{P}$ with probability at least $2/3$, by making only $\mathrm{poly}(\frac{1}{\varepsilon})$ queries.*

It turns out that our tester for testing VC-dimension property takes $\exp(d)$ samples, and performs $\exp(\exp(d))$ queries for VC-dimension $d$. We show that this bound is tight, in the sense that there exists an index-invariant property with VC-dimension $d$ such that any tester for the property requires an exponential number of samples and a doubly-exponential number of queries on $d$.

**Theorem 7.5 (Informal).** *Let $d, n \in \mathbb{N}$. There exists an index-invariant property $\mathcal{P}_{\mathsf{vc}}$ with VC-dimension at most $d$ such that any (non-adaptive) tester for $\mathcal{P}_{\mathsf{vc}}$ requires $2^{\Omega(d)}$ samples and $2^{2^{d-\mathcal{O}(1)}}$ queries.*

Note that from a result in [GR22], it follows that there exists an index-invariant property $\mathcal{P}$ such that any distribution $D \in \mathcal{P}$ has VC-dimension $d$ and any algorithm that has sample access to a distribution $D$ over $\{0,1\}^n$ requires $\Omega(2^d/d)$ samples [5], but

[5] Let $\mathcal{P}$ be the distribution property of having support size at most $2^d$. Note that the VC-dimension of any member of $\mathcal{P}$ is at most $d$. By [GR22], for any small enough $\varepsilon$, an $\varepsilon$-test for this property requires at least $\Omega\left(2^d/d\right)$ samples.

Theorem 7.5 proves the lower bound on both sample and query complexities for the same property.

Theorem 7.2 assumes that the properties are index-invariant and have bounded VC-dimension. A natural question in this regard is whether the bounded VC-dimension and index-invariance assumptions are necessary for a property to be constantly testable. We answer this question in the negative. Theorem 7.5 implies that bounded VC-dimension is necessary for a property to be constantly testable even if the property is index-invariant. The following proposition rules out the possibility that only the bounded VC-dimension assumption is good enough for a property to be testable by making a constant number of queries.

**Proposition 7.6 (Necessity of index-invariance (informal)).** *There exists a non-index-invariant property $\mathcal{P}$ such that any distribution $D \in \mathcal{P}$ has VC-dimension $O(1)$ and the following holds. There exists a fixed $\varepsilon > 0$, such that distinguishing whether $D \in \mathcal{P}$ or $D$ is $\varepsilon$-far from $\mathcal{P}$ requires $\Omega(n)$ queries, where the distributions in the property $\mathcal{P}$ are defined over the $n$-dimensional Hamming cube $\{0,1\}^n$.*

The above proposition is formally stated and proved at the end of Subsection 10.3.

Now we study the power of adaptive queries in the huge object model. Till now, our upper bound results are non-adaptive. However, the question how adaptivity helps in designing efficient testers is interesting in its own right. In the standard model of distribution testing, since the model is inherently non-adaptive, there is essentially no gap between adaptive and non-adaptive testers. However, in the related model of conditional sampling of distributions [CFGM16, CRS15], there is a super-exponential separation (constant vs. poly$(\log n)$) between complexities of these two types of testers [ACK18].

In the context of graph testing in the dense graph model, it is known that the gap between the query complexities of adaptive and non-adaptive algorithms is at most quadratic [GT03], which has recently been proved to be tight [GW21]. However, for bounded-degree graphs, the gap between the query complexities for some properties is constant vs. $\Omega(\sqrt{n})$, where $n$ denotes the number of vertices of the graph [GR97]. For testing of functions, there is an exponential separation between the complexity of these two types of testers [RS15].

Thus, a natural question to study in this huge object model is about the gap between the query complexities of non-adaptive and adaptive algorithms. When considering general properties, there can be an exponential gap in the query complexities between non-adaptive and adaptive testers as stated below.

**Theorem 7.7 (Informal).** *For any non-index-invariant property $\mathcal{P}$, there is at most an exponential gap between the query complexities of adaptive and non-adaptive testers.*

Moreover, we show that this gap is also tight as follows:

**Theorem 7.8 (Informal).** *There exists a non-index-invariant property $\mathcal{P}_{Pal}$ that can be $\varepsilon$-tested by performing $\mathcal{O}(\log n)$ queries adaptively for any $\varepsilon \in (0, 1)$. However, there exists an $\varepsilon \in (0, 1)$ for which $\Omega(\sqrt{n})$ non-adaptive queries are necessary to $\varepsilon$-test $\mathcal{P}_{Pal}$.*

However, for index-invariant properties, this gap can be at most quadratic, as stated in the following theorem.

**Theorem 7.9 (Informal).** *For any index-invariant property, there is at most a quadratic gap between the query complexities of adaptive and non-adaptive testers.*

We also prove that the above gap is almost tight, in the sense that there exists an index-invariant property which can be $\varepsilon$-tested using $\widetilde{\mathcal{O}}(n)$ adaptive queries, while $\widetilde{\Omega}(n^2)$ non-adaptive queries are required to $\varepsilon$-test it.

**Theorem 7.10 (Informal).** *There exists an index-invariant property $\mathcal{P}_{\text{Gap}}$ that can be $\varepsilon$-tested adaptively using $\widetilde{\mathcal{O}}(n)$ queries for any $\varepsilon \in (0, 1)$, while there exists an $\varepsilon \in (0, 1)$ for which $\widetilde{\Omega}(n^2)$ queries are necessary for any non-adaptive $\varepsilon$-tester.*

## Using EMD as the distance metric in conjunction with the notion of index-invariance:

Recall that here we will use the Earth Mover Distance (EMD) as the distance metric defining $\varepsilon$-testing, in contrast to the stronger variation distance, the commonly studied distance measure in distribution testing literature. As discussed in [GR22], this is essential when we restrict ourselves to querying the samples obtained from the distribution. To illustrate this, consider two (say very sparse) distributions $D_1$ and $D_2$ whose supports are disjoint, yet admit a bijection such that every string from $\text{Supp}(D_1)$ is mapped to a string from $\text{Supp}(D_2)$ that is very close to it in terms of the Hamming distance. The variation distance between $D_1$ and $D_2$ would be large, and yet we would not be able to distinguish the two distributions without querying some samples in their entirety, that is, without using $\Theta(n)$ queries per sample. The EMD metric is the one incorporating the Hamming distance between strings (which comes to play when we are not performing many queries to the samples) into the notion of variation distance.

Another question involves what general statements can be said about testers in this model. If we do not restrict ourselves to properties satisfying any sort of invariance, then very little can be proved on testers in general, just as is the case with general string property testing under the Hamming distance (in fact, string testing can be reduced to testing in the huge object model [6]). On the other hand, if we were to restrict ourselves to label-invariant properties only, it would appear that we lose much of the rich structure offered by the ability to define distributions over strings. We believe that index-invariance is a natural middle-of-the-road restriction for the formulation of general statements about testing in the huge object model.

## Organization of the part

We present the related definitions in this part of the thesis in the preliminaries section (Section 7.2). We present the results about learning and testing clusterable distributions in Chapter 8. After that, in Chapter 9, we move on to present algorithms for testing properties with bounded VC-dimension as well as the lower bound results for bounded VC-dimension testing.

Then in Chapter 10, we show the tight exponential separation between the query complexities of adaptive and non-adaptive algorithms for non-index-invariant (general)

[6] We will use this reduction for proving exponential separation between adaptive and non-adaptive testers for non-index-invariant properties (see Subsection 10.3).

properties. Finally in Chapter 11, we prove that for index-invariant properties, there is an almost tight quadratic gap between the query complexities of adaptive and non-adaptive testers, ignoring poly-logarithmic factors.

## 7.2 Preliminaries

We will use the following observation from [ABEF17] which roughly states that given a sequence of non-negative real numbers that sum up to an integer $n$, there is a procedure that by choosing the floor or ceiling of these real numbers, one can obtain another sequence of integers that sum up to $n$. This observation will be used in our proof.

**Observation 7.11 (Restatement of [ABEF17, Lemma 4.8]).** Let $T, n \in \mathbb{N}$. Given $T$ non-negative real numbers $\alpha_1, \ldots, \alpha_T$ such that $\sum\limits_{i=1}^{T} \alpha_i = n$, there exists a procedure of choosing $T$ integers $\beta_1, \ldots, \beta_T$ such that $\beta_i \in \{\lfloor \alpha_i \rfloor, \lceil \alpha_i \rceil\}$ for every $i \in [T]$ and $\sum\limits_{i=1}^{T} \beta_i = n$.

### 7.2.1 Distributions and properties with bounded VC-dimension

Now we move on to define a class of properties using the notion of the VC-dimension of the support of a distribution. Before proceeding to define the class of properties, let us recall the notions of *shattering* and *VC-dimension*.

Let $V$ be a collection of vectors from $\{0,1\}^n$. For a sequence of indices $I = (i_1, \ldots, i_k)$, with $1 \leq i_j \leq n$, let $V \mid_I$ denote the set of *projections* of $V$ onto $I$, that is,

$$V \mid_I = \{(v_{i_1}, \ldots, v_{i_k}) : (v_1, \ldots, v_n) \in V\}.$$

If $V \mid_I = \{0,1\}^k$, then it is said that $V$ *shatters* the index sequence $I$. The *VC-dimension* of $V$ is the size of the largest index sequence $I$ that is shattered by $V$. VC-dimension was introduced by Vapnik and Chervonenkis [VC15] in the context of learning theory, and has found numerous applications in other areas like approximation algorithms, discrete and computational geometry, discrepancy theory, see [Mat99, PA95, Mat02, Cha00].

We now give a natural extension of VC-dimension to distributions.

**Definition 7.12 (Distribution with VC-dimension $d$).** Let $d$, $n \in \mathbb{N}$ and $D$ be a distribution over $\{0,1\}^n$. We say that $D$ has VC-dimension at most $d$ if the support of $D$ has VC-dimension at most $d$. A distribution $D$ is said to be *$\beta$-close to VC-dimension $d$* if there exists a distribution $D_0$ with VC-dimension $d$ such that $d_{EM}(D, D_0) \leq \beta$, where $\beta \in (0,1)$.

Analogously, we can also define the notion of a $(\beta, d)$-VC-dimension property.

**Definition 7.13 ($(\beta, d)$-VC-dimension property).** Let $d, n \in \mathbb{N}$ and $\beta \in (0,1)$. A property $\mathcal{P}$ over $\{0,1\}^n$ is said to be a *$(\beta, d)$-VC-dimension property* if for any distribution $D \in \mathcal{P}$, $D$ is $\beta$-close to VC-dimension $d$. When $\beta = 0$, we say that the VC-dimension of $\mathcal{P}$ is $d$. We also say that a $(0, d)$-VC-dimension property is a *bounded VC-dimension property*.

We now give examples of bounded VC-dimension properties.

**Property** CHAIN**:** For any distribution $D \in$ CHAIN, the support of $D$ can be written as a sequence $\mathbf{X}_1, \ldots, \mathbf{X}_t \in \{0,1\}^n$ such that any two vectors with non-zero probability are comparable, that is,

$$D(\mathbf{X}_i) > 0 \text{ and } D(\mathbf{X}_j) > 0 \text{ implies either } \mathbf{X}_i \preceq \mathbf{X}_j \text{ or } \mathbf{X}_j \preceq \mathbf{X}_i, \forall\, i, j \in [t].$$

**Property** LOW-AFFINE-DIMENSION**:** A distribution $D$ over $\{0,1\}^n$ is said to satisfy the LOW- AFFINE-DIMENSION property, with parameter $d \in \mathbb{N}$, if the *affine dimension*[7] of the support of $D$ is at most $d$.

Observe that the VC-dimension of CHAIN is 1, and the VC-dimension of LOW-AFFINE-DIMENSION is $d$. [8] Moreover, note that both CHAIN and LOW-AFFINE-DIMENSION are examples of index-invariant properties.

[7] A set $S \subseteq \mathbb{R}^n$ has *affine dimension* $k$ if the dimension of the smallest *affine set* in $\mathbb{R}^n$ that contains $S$ is $k$.

[8] In fact, the property LOW-AFFINE-DIMENSION is a sub-property of "support size is at most $2^d$", which has VC-dimension $d$.

### 7.2.2 Yao's lemma for the huge object model

Our lower bound proofs crucially use Yao's lemma [Yao77]. Informally, it states that for any two distributions $D_1$ and $D_2$ such that $D_1$ satisfies some property, and $D_2$ is far from the property, if the variation distance between $D_1$ and $D_2$ with respect to $q$ queries is small, then $D_1$ and $D_2$ remain indistinguishable with respect to $q$ queries. In order to formally state the lemma, we need the following definitions.

**Definition 7.14 (Restriction).** Let $D$ be a distribution over a collection of functions $f : \mathcal{D} \to \{0, 1\}$, and $Q$ be a subset of the domain $\mathcal{D}$ of $D$. The restriction $D \mid_Q$ of $D$ to $Q$ is the distribution over functions of the form $g : Q \to \{0, 1\}$, which is obtained from choosing a random function $f : \mathcal{D} \to \{0, 1\}$ according to the distribution $D$, and then setting $g = f \mid_Q$, where $f \mid_Q$ denotes the restriction of $f$ to $Q$.

The following is the version of Yao's Lemma which is used for non-adaptive testers in the classical setting. The crucial observation that makes this lemma work is the observation that the deterministic version of a non-adaptive tester in the classical setting is characterized by a set of possible responses to a fixed query set $Q \subset \mathcal{D}$.

**Lemma 7.15 (Yao's lemma for non-adaptive testers, see [Fis04]).** *Let $\varepsilon \in (0, 1)$ be a parameter and $q \in \mathbb{N}$ be an integer. Suppose there exists a distribution $D_{yes}$ on inputs over $\mathcal{D}$ that satisfy a given property $\mathcal{P}$, and a distribution $D_{no}$ on inputs that are $\varepsilon$-far from satisfying the property. Moreover, assume that for any set of queries $Q \subset \mathcal{D}$ of size $q$, the variation distance between $D_{yes} \mid_Q$ and $D_{no} \mid_Q$ is less than $\frac{1}{3}$. Then it is not possible for a non-adaptive tester performing $q$ (or less) queries to $\varepsilon$-test $\mathcal{P}$.*

In this chapter, we will prove lower bounds against non-adaptive distribution testers in the huge object model. Hence, $D_{yes}$ and $D_{no}$, rather than being distributions over functions from $\mathcal{D}$ to $\{0, 1\}$, are distributions over distributions over $\{0, 1\}^n$ (since the basic input object is a distribution over $\{0, 1\}^n$).

The deterministic version of a non-adaptive tester in this setting is characterized by a set of possible responses to a sequence of queries $\mathcal{J} = (J_1, \ldots, J_s)$ to the samples. We call $s$ the *length* of $\mathcal{J}$, and call $q = \sum_{i=1}^{s} J_i$, the *size* of $\mathcal{J}$.

Given a distribution $D$ over distributions over $\{0,1\}^n$, we denote by $D\mid_{\mathcal{J}}$ the distribution over $\{0,1\}^q$ that results from first picking a distribution $\widehat{D}$ over $\{0,1\}^n$ according to $D$, then taking $s$ independent samples $\mathbf{X}_1,\ldots,\mathbf{X}_s$ according to $\widehat{D}$, and finally constructing the sequence $\mathbf{X}_1\mid_{J_1},\ldots,\mathbf{X}_s\mid_{J_s}$. The huge object model version of Yao's lemma for non-adaptive testers is the following one.

**Lemma 7.16 (Yao's lemma for non-adaptive testers in the huge object model).** *Let $\varepsilon \in (0,1)$ be a parameter and $q,s \in \mathbb{N}$ be two integers. Suppose there exists a distribution $D_{yes}$ over distributions over $\{0,1\}^n$ that satisfy a given property $\mathcal{P}$, and a distribution $D_{no}$ over distributions over $\{0,1\}^n$ that are $\varepsilon$-far from satisfying the property $\mathcal{P}$. Moreover, assume that for any query sequence $\mathcal{J}$ of length $s$ and size $q$, the variation distance between $D_{yes}\mid_{\mathcal{J}}$ and $D_{no}\mid_{\mathcal{J}}$ is less than $1/3$. Then it is not possible for a non-adaptive tester that takes at most $s$ samples and performs at most $q$ queries to $\varepsilon$-test $\mathcal{P}$.*

## 7.3 Technical overview of our results

In this section, we provide a brief overview of our results. We start by explaining our upper bounds. In our main upper bound result, we prove a learning result for a general class of distributions that covers the case of learning distributions with bounded VC-dimension. We say that a distribution $D$ is $(\zeta,\delta,r)$-clusterable if we can partition the $n$-dimensional Hamming cube $\{0,1\}^n$ into $r+1$ parts $\mathcal{C}_0,\ldots,\mathcal{C}_r$, such that $D(\mathcal{C}_0) \leq \zeta$ and the diameter of $\mathcal{C}_i$ is at most $\delta$ for every $i \in [r]$ (see Definition 8.2). The main upper bound result (Theorem 8.3), that leads to Theorem 7.2, is the design of an algorithm for learning a $(\zeta,\delta,r)$-clusterable distribution up to permutations. That is, given sample and query access to a $(\zeta,\delta,r)$-clusterable distribution, we want to output a distribution $D'$ such that the Earth Mover Distance between $D$ and $D'_\sigma$ is small for some permutation $\sigma : [n] \to [n]$, by performing number of queries independent of $n$.

### 7.3.1 Overview of learning clusterable distributions

The algorithm for learning $(\zeta, \delta, r)$-clusterable distributions is described in Algorithm 8.1 in Section 8.2 as TEST-AND-LEARN. The algorithm starts by taking $t_1 = \mathcal{O}(\frac{r}{\zeta} \log \frac{r}{\zeta})$ samples from the input distribution $D$. Let us denote them as $\mathcal{S} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{t_1}\}$. If $D$ is $(\zeta, \delta, r)$-clusterable, consider its clusters $\mathcal{C}_0, \ldots, \mathcal{C}_r$ as described above. We say that a cluster $C_i$ is *large* if the probability mass of $\mathcal{C}_i$ is more than $\frac{\zeta}{10r}$, that is, $D(\mathcal{C}_i) \geq \frac{\zeta}{10r}$. As the size of $\mathcal{S}$ is sufficiently large, we know that $\mathcal{S}$ intersects every large cluster with probability at least $99/100$ (see Lemma 8.6). In order to estimate the masses of $\mathcal{C}_i$, for each $i \in [t_1]$, we take another set of random samples $\mathcal{T} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_{t_2}\}$ from $D$ where $t_2 = \mathcal{O}(\frac{t_1^2}{\zeta^2} \log t_1)$, and assign each of the vectors in $\mathcal{T}$ to some vector in $\mathcal{S}$ depending on their Hamming distance. However, since computing the exact distances between the vectors in $\mathcal{S}$ and $\mathcal{T}$ requires $\Omega(n)$ queries, we use random sampling.

We take a random set of indices $R \subset [n]$ of suitable size, and project the vectors in $\mathcal{S}$ and $\mathcal{T}$ on $R$ to estimate their pairwise distances up to an additive factor of $\delta$. $R$ not only preserves the distances between all pairs of vectors between $\mathcal{S}$ and $\mathcal{T}$, but also the distances of a large fraction of the vectors in $\{0, 1\}^n$ from all the vectors in $\mathcal{S}$ (see Lemma 8.7). Based on the estimated distances, we assign each vector of $\mathbf{T} \in \mathcal{T}$ to a vector in $\mathbf{S} \in \mathcal{S}$ such that the projected distance between them is at most $2\delta$. If there exists no such vector in $\mathcal{S}$ corresponding to a vector $\mathbf{T} \in \mathcal{T}$, then the vector $\mathbf{T}$ remains unassigned. Let us denote the fraction of vectors in $\mathcal{T}$ that are assigned to $\mathbf{X}_i$ as $w_i$, for every $i \in [t_1]$. Let $w_0$ be the fraction of vectors in $\mathcal{T}$ that are not assigned to any vector in $\mathcal{S}$. If $D$ is $(\zeta, \delta, r)$-clusterable, then $w_0 \leq 3\zeta$ holds with high probability. These $w_i$'s preserve the weights of some approximate clustering (which may not be the original one from which we started, but is close to it in some sense), see Lemma 8.8 for the details.

Consider a distribution $D^*$ supported over $\mathcal{S}$ such that $D(\mathbf{X}_i) \geq w_i$ for every $i \in [t_1]$. Using a number of technical lemmas, we prove that the EMD between $D$ and $D^*$ is *small*. Note that we still can not report $D^*$ as the output distribution, since to do so, we need to know the exact vectors in $\mathcal{S}$, which requires $\Omega(n)$ queries. To bypass this barrier, we use the provision that we are allowed to output any permutation of the distribution.

More specifically, we construct vectors $\mathbf{S}_1, \ldots, \mathbf{S}_{t_1} \in \{0,1\}^n$ such that $d_H(\mathbf{X}_i, \sigma(\mathbf{S}_i))$ is small for every $i \in [t_1]$ and some permutation $\sigma : [n] \to [n]$. This is possible using the projections of the vectors in $\mathcal{S}$ to the random set of indices $R$ for estimating the number of indices of each "type" with respect to $\mathcal{S}$ (see Lemma 8.12). Finally, we output the distribution $D'$ supported over the newly constructed vectors $\mathbf{S}_1, \ldots, \mathbf{S}_{t_1}$ such that $D'(\mathbf{S}_i) = D^*(\mathbf{X}_i)$ for every $i \in [t_1]$. The guarantee on the Hamming distance between $\mathbf{X}_i$ and $\sigma(\mathbf{S}_i)$ provides a bound on the EMD between $D'_\sigma$ and $D^*$, and with the above mentioned EMD bound between $D^*$ and $D$, we are done. To keep the discussion simple, we will not explain here the idea of the proof of Theorem 7.2(ii), which relies on a sort of converse to the above method of approximating cluster weights.

### 7.3.2 Overview of learning index-invariant bounded VC-dimension properties

Now we discuss how learning $(\zeta, \delta, r)$-clusterable distribution implies Theorem 7.2. Let us define a distribution to be $(\alpha, r)$-clusterable if it is $(0, \alpha, r)$-clusterable. The learning of $(\zeta, \delta, r)$-clusterable distribution implies a learning result for any distribution that is close to being $(\alpha, r)$-clusterable (see Corollary 9.5) due to a technical lemma (see Lemma 9.6). If the support of a distribution has bounded VC-dimension, using standard results in VC theory, we can show that it is also $(\alpha, r)$-clusterable, where $r$ is a function of $\alpha$ and $d$. Thus the learning result of $(\alpha, r)$-clusterable distributions implies a result allowing the learning of distributions with bounded VC-dimension.

### 7.3.3 Overview of lower bound for index-invariant bounded VC-dimension properties

To prove Theorem 7.5, let us define the property $\mathcal{P}_{\mathsf{vc}}$. Let $k = 2^d$, $\ell = 2^{2^{d-10}}$ and $\ell' = 2^{2^{d-20}}$. Consider a matrix $A$ of dimension $k \times \ell$ whose column vectors are $1/3$-far from each other. Let $\mathbf{V}_1, \ldots, \mathbf{V}_k \in \{0,1\}^n$ be $k$ vectors that are formed by blowing up the row vectors of $A$ in $\{0,1\}^\ell$ to $\{0,1\}^n$ by repeating each bit of the vectors $n/\ell$ times, and $D_A$ be the uniform distribution over the support $\{\mathbf{V}_1, \ldots, \mathbf{V}_k\}$. Our property $\mathcal{P}_{\mathsf{vc}}$ is the collection of all distribution that can be obtained from $D_A$ by permuting the indices. Let $D_{yes}$ be the distribution obtained from $D_A$ by randomly permuting the indices. Note that $D_{yes} \in \mathcal{P}_{\mathsf{vc}}$. As the support size of any distribution in $\mathcal{P}_{\mathsf{vc}}$ is at most $2^d$, the VC-dimension of $\mathcal{P}_{\mathsf{vc}}$ is at most $d$.

To prove the lower bound on the query complexity, let us define the distribution $D_{no}$. Let us take $\ell'$ columns of $A$ uniformly at random to form a matrix $B$ of dimension $k \times \ell'$, and $\mathbf{W}_1, \ldots, \mathbf{W}_k \in \{0,1\}^n$ be $k$ vectors that are formed by blowing up the row vectors of $B$ in $\{0,1\}^{\ell'}$ to $\{0,1\}^n$ by repeating each bit of the vectors $n/\ell'$ times. Let $D_B$ be the uniform distribution over the support $\{\mathbf{W}'_1, \ldots, \mathbf{W}'_k\}$. $D_{no}$ is the distribution obtained from $D_B$ by randomly permuting the indices. We show that the Earth Mover Distance between $D_{no}$ and any distribution in $\mathcal{P}_{\mathsf{vc}}$ is at least $1/8$ (see Lemma 9.17). Observe that $D_{yes}$ divides the index set $[n]$ into $\ell$ equivalence classes and $D_{no}$ divides the index set into $\ell'$ equivalence classes. The query complexity lower bound follows from the fact that, unless we query $2^{2^{d-\mathcal{O}(1)}}$ indices, we do not hit two indices from the same equivalence class, irrespective of whether the distribution is $D_{yes}$ or $D_{no}$ (see Lemma 9.22).

To prove the lower bound on the sample complexity, let us define another distribution $D'_{no}$. Let us take $k' = 2^{d-20}$ rows of $A$ uniformly at random to form a matrix $B'$ of dimension $k' \times \ell$. Let $\mathbf{W}'_1, \ldots, \mathbf{W}'_{k'} \in \{0,1\}^n$ be $k'$ vectors that are formed by blowing up the row vectors of $B'$ in $\{0,1\}^\ell$ to $\{0,1\}^n$ by repeating each bit of the vectors $n/\ell$ times. Let $D_{B'}$ be the uniform distribution with support $\{\mathbf{W}'_1, \ldots, \mathbf{W}'_{k'}\}$. $D'_{no}$ is the distribution obtained from $D_{B'}$ by randomly permuting the indices. We show that the Earth Mover Distance between $D'_{no}$ and any distribution in $\mathcal{P}_{\mathsf{vc}}$ is at least $1/8$ (see Lemma 9.23). The sample complexity lower bound follows from the fact that, unless we take $2^{\Omega(d)}$ samples, all the samples are distinct with probability $1 - o(1)$, irrespective of whether the distribution is $D_{yes}$ or $D_{no}$ (see Lemma 9.25).

### 7.3.4 Overview of adaptive vs. non-adaptive testers for general properties

Now we explore the relationship between adaptive and non-adaptive testers in the huge object model. It turns out that there is a tight (easy to prove) exponential separation between the query complexities of adaptive and non-adaptive testers for non-index-invariant properties. Roughly, the simulation of an adaptive algorithm by a non-adaptive one follows from unrolling the decision tree of the adaptive algorithm. This is formally proved in Lemma 10.4. Moreover, we show that this separation is tight. For this purpose, we consider a property of strings $\mathcal{P}_{Pal}$, which exhibits an exponential gap between adaptive and non-adaptive testing in the string testing model. We show how to transform a string property $\mathcal{P}$ to a distribution property $1_{\mathcal{P}}$ such that the query bounds on adaptive and non-adaptive testing carry over. Thus, the separation result between adaptive and non-adaptive algorithms for $\mathcal{P}_{Pal}$ carries over to $1_{\mathcal{P}_{Pal}}$ (see Theorem 10.8). This technique, employed for a maximally hard to test string property, is also used for proving Proposition 7.6.

### 7.3.5 Overview of adaptive vs. non-adaptive testers for index invariant properties

In contrast to the non-index-invariant properties, we prove that there can be at most a quadratic gap between the query complexities of adaptive and non-adaptive algorithms for testing index-invariant properties. The proof is very close in spirit to the proof of the quadratic relation between adaptive and non-adaptive testing of graphs in the dense model [GT03]. Given an adaptive algorithm $\mathcal{A}$ with sample complexity $s$ and query complexity $q$, the main idea is to first simulate a *semi-adaptive* algorithm $\mathcal{A}'$ that queries $q$ indices from each of the $s$ samples and decides accordingly. Note that the sample complexity of $\mathcal{A}'$ remains $s$, whereas the query complexity becomes $qs$. Once we have the semi-adaptive algorithm $\mathcal{A}'$, we now simulate a non-adaptive algorithm $\mathcal{A}''$. As the property we are testing is index-invariant, we can first apply a uniformly random permutation $\sigma$ over $[n]$, and then run the semi-adaptive algorithm $\mathcal{A}'$ over $D_\sigma$ instead of $D$, where $D$ is the input distribution to be tested. This makes the tester completely non-adaptive. Its correctness follows from the index-invariance of the property we are testing.

**Quadratic separation between adaptive and non-adaptive testers:** Before proceeding to present an overview of our quadratic separation result, let us first recall the support estimation result of Valiant and Valiant [VV10], which will be crucially used in our proof. Roughly speaking, the result states that in the standard sampling model, given a distribution $D$ over $[2n]$, in order to distinguish whether $D$ has support size at most $n$, or $D$ is far from all distributions with support size at least $n$, $\Theta(n/\log n)$ samples are required.

**Theorem 7.17 (Support Estimation bound, Corollary** $9$ **of Valiant-Valiant [VV10]).** *Given a distribution $D$ over $[2n]$, that can be accessed via independent samples and a proximity parameter $\varepsilon \in (0, 1/8)$, in order to distinguish, with probability at least $\frac{3}{4}$, whether $D$ has support size at most $n$ or $D$ has at least $(1+\varepsilon)n$ elements with non-negligible weights in its support, $\Theta(\frac{n}{\log n})$ samples from $D$ are necessary and sufficient.*

To construct the index-invariant property that provides a quadratic separation between the query complexities of adaptive and non-adaptive testers, we will use the above result. Let $D_{yes}^{\mathrm{Supp}}$ and $D_{no}^{\mathrm{Supp}}$ be the pair of hard distributions corresponding to the support estimation lower bound, from which we define our pair $D_{yes}$ and $D_{no}$ of hard distributions for our property. We will construct a huge object distribution property over a slightly larger domain $[N]$ with $N = \mathcal{O}(n \log n)$, where we will encode the elements of the support of the distributions $D_{yes}^{\mathrm{Supp}}$ and $D_{no}^{\mathrm{Supp}}$. Additionally, we will include a set of "ordering" vectors to both $D_{yes}$ and $D_{no}$ that encode a permutation $\sigma : [n] \rightarrow [n]$. Our property will be defined as a permutation of a non-index-invariant property along with an encoding of the permutation itself.

For the non-index-invariant property, we use an encoding of the elements of $\{0,1\}^n$ that can be decoded only if a sample from a family of special small sets is read in its entirety. For constructing hard distributions, we consider (encodings of) $2n$ special elements of $\{0, 1\}^n$, and use over them the hard distributions corresponding to Theorem 7.17.

The encoding vectors of $D_{yes}$ and $D_{no}$ are designed in such a fashion, that if we can know the index ordering (and thus the identity of the above mentioned small sets), the support size estimation problem becomes relatively easy. However, without knowing the ordering vectors, estimating the size of the support becomes harder. More specifically, if we already know the index ordering, then support size estimation can be done using $\text{poly}(\log n)$ queries from each sample, over the $\widetilde{\mathcal{O}}(n)$ samples that are sufficient for solving the support estimation problem.

On the other hand, an important feature of our property ensures that unless some of the special sets are successfully hit while querying a sampled vector, which is a low probability event without prior knowledge of the encoded index ordering unless we perform $\widetilde{\Omega}(n)$ queries to that vector, then the queries do not provide any useful information about the sampled vector to the tester. This is achieved by the encoding procedure of the vectors, which is motivated from [BFLR20]. However, it is not deployed here the same way as [BFLR20], since the surrounding proofs here are quite different (as well as the end-goal).

Since an adaptive algorithm can first learn the ordering vectors by performing $\widetilde{\mathcal{O}}(n)$ queries (as it takes $\text{poly}(\log n)$ samples to hit all the order encoding vectors), the adaptive tester requires $\widetilde{\mathcal{O}}(n)$ queries in total. However, for non-adaptive testers, since we have to perform all queries simultaneously, the tester would have to make $\widetilde{\Omega}(n)$ queries to each sampled vector to be able to utilize the support estimation procedure (since as explained above, fewer queries would give no useful information about the sample to the tester). As a result, $\widetilde{\Omega}(n^2)$ non-adaptive queries are required following the lower bound result in Theorem 7.17.

Another technical challenge is to construct the property in such a fashion that allows the crafting of "wrong distributions" which remain far from the property, even if we permute the support vectors. This is due to the fact that just replacing the vectors defining the index ordering does not require a change of large Earth Mover Distance. Thus we need the distributions to remain far from the property even if we reorder them. We ensure this by designing the hard distributions such that the support vectors of the distributions are far from each other. This in turn allows us to prove that the distribution $D_{no}$ will remain far from the property, as the size of its support is too large. The arguments involving only the mutual Hamming distance between the vectors in the support and the size of the support are invariant with respect to the index ordering, and are thus not affected by the possibility of "cheaply" changing the index ordering vectors.

# Chapter 8

# Learning Clusterable Distributions

## 8.1 Introduction

In this chapter, we prove a learning result for clusterable distributions in the huge object model. The result is stated as follows:

**Theorem 8.1** (**Restatement of Theorem 1.6**). *Given sample and query access to an unknown distribution $D$ over $\{0,1\}^n$, there exists a non-adaptive algorithm that makes a number of queries that is independent of $n$, and either reports a full description of a distribution over $\{0,1\}^n$ or reports* FAIL, *satisfying both of the following conditions:*

**(i)** *If $D$ is clusterable, then with probability at least $\frac{2}{3}$, the algorithm outputs a full description of a distribution $D'$ such that $D$ is $\varepsilon$-close to $D'_\sigma$ for some permutation $\sigma : [n] \to [n]$.*

**(ii)** *For any $D$, the algorithm will not output a distribution $D'$ such that $D'_\sigma$ is $\varepsilon$-far from $D$ for every permutation $\sigma : [n] \to [n]$, with probability more than $\frac{1}{3}$. However, if the distribution $D$ is not clusterable, the algorithm may output* FAIL *with any probability.*

## 8.2 Learning clusterable distributions

In this section, we define the notion of a $(\zeta, \delta, r)$-clusterable distribution formally (see Definition 8.2), and prove that such distributions can be learnt (up to permutation) efficiently in Theorem 8.3. Intuitively, a distribution $D$ defined over $\{0,1\}^n$ is called $(\zeta, \delta, r)$-clusterable if we can remove a subset of the support vectors of $D$ whose probability mass is at most $\zeta$, and we can partition the remaining vectors in the support of $D$ into at most $r$ parts, each with diameter at most $\delta$. Theorem 8.3 states that, given a distribution $D$ over $\{0,1\}^n$, we can learn it (up to permutation) if it is $(\zeta, \delta, r)$-clusterable, and otherwise, we either report FAIL or learn the input distribution (up to permutation). Note that learning the distribution up to permutation is sufficient to provide testing algorithms for index-invariant properties with bounded VC-dimension, which will be discussed in Section 9.2.

**Definition 8.2** (**$(\zeta, \delta, r)$-clusterable & $(\alpha, r)$-clusterable distribution).** **(i)** Let $\zeta, \delta \in (0,1)$ and $r, n \in \mathbb{N}$. A distribution $D$ over $\{0,1\}^n$ is called $(\zeta, \delta, r)$-*clusterable* if there exists a partition $\mathcal{C}_0, \dots, \mathcal{C}_s$ of $\{0,1\}^n$ such that $D(\mathcal{C}_0) \leq \zeta$, $s \leq r$, and for every $1 \leq i \leq s$, $d_H(\mathbf{U}, \mathbf{V}) \leq \delta$ for any $\mathbf{U}, \mathbf{V} \in \mathcal{C}_i$.

**(ii)** For $\alpha \in (0,1)$ and $r \in \mathbb{N}$, a distribution $D$ over $\{0,1\}^n$ is called $(\alpha, r)$-*clusterable* if it is $(0, \alpha, r)$-*clusterable*. For $\beta \in (0,1)$, a distribution $D$ is called $\beta$-*close to being* $(\alpha, r)$-*clusterable* if there exists an $(\alpha, r)$-clusterable distribution $D_0$ such that $d_{EM}(D, D_0) \leq \beta$.

**Theorem 8.3 (Theorem 8.1 formalized).** *There exists a non-adaptive algorithm* TEST-AND-LEARN, *as described in Algorithm 8.1, that has sample and query access to an unknown distribution $D$ over $\{0,1\}^n$ for $n \in \mathbb{N}$, takes parameters $\zeta, \delta, r$ as inputs such that, $\zeta, \delta \in (0,1)$ and $\varepsilon = 17(\delta + \zeta) < 1$* [1] *and $r \in \mathbb{N}$, makes a number of queries that only depends on $\zeta, \delta$ and $r$, and either reports a full description of a distribution over $\{0,1\}^n$ or reports* FAIL, *satisfying both of the following conditions:*

[1] The constant 17 is arbitrary, and can be improved to a smaller constant. We did not try to optimize.

**(i)** *If $D$ $(\zeta, \delta, r)$-clusterable, then with probability at least $\frac{2}{3}$, the algorithm outputs a full description of a distribution $D'$ over $\{0,1\}^n$ such that $d_{EM}(D, D'_\sigma) \leq \varepsilon$ for some permutation $\sigma : [n] \to [n]$.*

**(ii)** *For any $D$, the algorithm will not output a distribution $D'$ such that $d_{EM}(D, D'_\sigma) > \varepsilon$ for every permutation $\sigma : [n] \to [n]$, with probability more than $\frac{1}{3}$. However, if the distribution $D$ is not $(\zeta, \delta, r)$-clusterable, the algorithm may output* FAIL *with any probability.*

The algorithm for learning $(\zeta, \delta, r)$-clusterable distributions is described in Algorithm 8.1 as TEST-AND-LEARN. It calls a subroutine APPROX-CENTERS, as described in Algorithm 8.2.

**Remark 8.1.** The sample complexity of TEST-AND-LEARN is polynomial in $r$, and the query complexity of TEST-AND-LEARN is exponential in $r$. Moreover, for the case of query complexity, the exponential dependency in $r$ is required. In particular, in Section 9.3, we construct a distribution with support size $r$ that requires $2^{\Omega(r)}$ queries to test for the property of being a permutation thereof.

To prove the correctness of TEST-AND-LEARN (which we will do in Section 8.2.1 and Section 8.2.2), we will need the notion of an $(\eta, \xi)$-clustered distribution around a sequence of vectors $\mathcal{S}$ (see Definition 8.4), and an associated observation (see Observation 8.5).

**Definition 8.4** (**$(\eta, \xi)$-clustered distribution around a sequence).** Let $\eta, \xi \in (0,1)$ and $n \in \mathbb{N}$. Also, for $\mathbf{X} \in \{0,1\}^n$, let $\mathrm{NGB}_\eta(\mathbf{X})$ denote the set of vectors in $\{0,1\}^n$ that are at a distance of at most $\eta$ from $\mathbf{X}$. Let $\mathcal{S} = \{\mathbf{S}_1, \ldots, \mathbf{S}_t\}$ be a sequence of $t$ vectors in $\{0,1\}^n$ and define $\mathrm{NGB}_\eta(\mathcal{S}) = \bigcup\limits_{S \in \mathcal{S}} \mathrm{NGB}_\eta(S)$. Then:

**(i)** A distribution $D$ over $\{0,1\}^n$ is called *$(\eta, \xi)$-clustered around $\mathcal{S}$ with weights* $w_0, \ldots w_{t-1}, w_t \in [0,1]$ satisfying $\sum\limits_{i=0}^{t} w_i = 1$ and $w_0 \leq \xi$, if there exist $t$ pairwise disjoint sets $\mathcal{C}_i$, such that $\mathcal{C}_i \subseteq \mathrm{NGB}_\eta(\mathbf{S}_i)$ and $D(\mathcal{C}_i) \geq w_i$ for every $i \in [t]$.

**Algorithm 8.1:** TEST-AND-LEARN

**Input:** Sample and Query access to a distribution $D$ over $\{0,1\}^n$, and parameters $\zeta, \delta, r$ with $\zeta, \delta \in (0,1)$ and $r \in \mathbb{N}$.

**Output:** Either reports a full description of a distribution over $\{0,1\}^n$ or FAIL, satisfying (i) and (ii) as stated in Theorem 8.3.

**(i)** Take $t_1 = \mathcal{O}(\frac{r}{\zeta} \log \frac{r}{\zeta})$ samples $\mathcal{S} = \mathbf{X}_1, \ldots, \mathbf{X}_{t_1}$ from $D$.

**(ii)** Take $t_2 = \mathcal{O}(\frac{t_1^2}{\zeta^2} \log t_1)$ samples $\mathcal{T} = \mathbf{Y}_1, \ldots, \mathbf{Y}_{t_2}$ from $D$.

**(iii)** Pick a random subset $R \subset [n]$ with $|R| = \mathcal{O}(\frac{4^{t_1}}{\delta^2 \zeta} \log \frac{r}{\delta \zeta})$. Query the indices corresponding to $R$ in each sample of $\mathcal{S}$, to obtain the sequence of vectors $\mathcal{S}_x = \mathbf{x}_1, \ldots, \mathbf{x}_{t_1}$, where $\mathbf{x}_i = \mathbf{X}_i \mid_R$ for each $i \in [t_1]$. Now query the indices corresponding to $R$ in each sample in $\mathcal{T}$, to obtain the sequence of vectors $\mathcal{T}_y = \mathbf{y}_1, \ldots, \mathbf{y}_{t_2}$, where $\mathbf{y}_j = \mathbf{Y}_j \mid_R$ for every $j \in [t_2]$.

**(iv)** For each $j \in \{1, \ldots, t_2\}$, if there exists an $i \in [t_1]$ such that $d_H(\mathbf{y}_j, \mathbf{x}_i) \leq 2\delta$, assign $\mathbf{y}_j$ to $\mathbf{x}_i$, breaking ties by assigning $\mathbf{y}_j$ to the vector in $\mathcal{S}_x$ with the minimum index.

If for some $\mathbf{y}_j$ no suitable $\mathbf{x}_i$ is found, then $\mathbf{y}_j$ remains unassigned.

**(v)** If the total number of unassigned vectors in $\mathcal{T}_y$ is more than $3\zeta t_2$, output FAIL.

**(vi)** For every $i \in \{1, \ldots, t_1\}$, the weight of $\mathbf{x}_i$ is defined as

$$w_i = w(\mathbf{x}_i) = \frac{\text{Number of vectors in } \mathcal{T}_y \text{ assigned to } \mathbf{x}_i}{t_2}.$$

**(vii)** Use APPROX-CENTERS (Algorithm 8.2) with $R$ and $\mathbf{x}_1, \ldots, \mathbf{x}_{t_1}$ to obtain $\mathbf{S}_1, \ldots, \mathbf{S}_{t_1} \in \{0,1\}^n$ (as stated in Lemma 8.17).

**(viii)** Construct and return any distribution $D'$ over $\{0,1\}^n$ such that

- For each $i = 1, \ldots, t_1$, $D'(\mathbf{S}_i) \geq w(\mathbf{x}_i)$.
- $\sum\limits_{i=1}^{t_1} D'(\mathbf{S}_i) = 1$.
- $D'(\mathbf{S}) = 0$ for every $\mathbf{S} \in \{0,1\}^n \setminus \{\mathbf{S}_1, \ldots, \mathbf{S}_{t_1}\}$.

```
Algorithm 8.2: APPROX-CENTERS
Input: A random subset $R \subseteq [n]$ with $|R| = \mathcal{O}(\frac{4^{t_1}}{\delta^2\zeta} \log \frac{r}{\delta\zeta})$, and a sequence of
       vectors $\mathbf{x}_1, \dots, \mathbf{x}_{t_1} \in \{0,1\}^{|R|}$ drawn from the distribution $D \mid_R$.
Output: Sequence of vectors $\mathbf{S}_1, \dots, \mathbf{S}_{t_1}$ such that with probability at least
        $99/100$ over the random choice of $R$, for every $i \in [t_1]$,
        $d_H(\sigma(\mathbf{X}_i), \mathbf{S}_i) \leq \delta/10$, where $\sigma : [n] \to [n]$ is a permutation.
(i) For each $i \in R$, construct the vector $\mathbf{C}_i \in \{0,1\}^{t_1}$ such that $\mathbf{C}_i(j) = \mathbf{x}_j(i)$.
(ii) For any $J \in \{0,1\}^{t_1}$, determine $\gamma_J = \frac{|\{i \in R | \mathbf{C}_i = J\}|}{|R|}$.
(iii) Apply Observation 7.11 to obtain an approximation $\Gamma_J \forall J \in \{0,1\}^{t_1}$, such that
      $\Gamma_J \in \{\lfloor \gamma_J \cdot n \rfloor, \lceil \gamma_J \cdot n \rceil\}$ and $\sum\limits_{J \in \{0,1\}^{t_1}} \Gamma_J = n$.
(v) Construct a matrix $A$ of dimension $t_1 \times n$ by putting $\Gamma_J$ many $J$ column vectors,
    for each $J \in \{0,1\}^{t_1}$.
(vi) Return the row vectors of $A$ as $\mathbf{S}_1, \dots, \mathbf{S}_{t_1}$.
```

**(ii)** A distribution $D$ over $\{0,1\}^n$ is called $(\eta, \xi)$*-clustered around* $\mathcal{S}$ if $D$ is $(\eta, \xi)$-clustered around $\mathcal{S}$ with weights $w_0, \dots, w_t \in [0,1]$, for some $w_0, \dots, w_t$ such that $\sum\limits_{i=0}^{t} w_i = 1$ and $w_0 \leq \xi$.

**Observation 8.5.** Let $D$ be any distribution over $\{0,1\}^n$ and $\mathcal{S}$ be a sequence of vectors in $\{0,1\}^n$ such that $\mathrm{NGB}_\eta(\mathcal{S}) \geq 1 - \xi$. Then $D$ is $(\eta, \xi)$-clustered around $\mathcal{S}$.

*Proof.* Let us partition $\mathrm{NGB}_\eta(\mathcal{S})$ into $t$ parts such that $\mathcal{C}_i = \mathrm{NGB}_\eta(\mathbf{X}_i) \setminus \bigcup\limits_{j=1}^{i-1} \mathrm{NGB}_\eta(\mathbf{X}_j)$ for every $i \in [t]$. For every $i \in [t]$, note that $\mathcal{C}_i \subseteq \mathrm{NGB}_\eta(\mathbf{X}_i)$, and let us define $w_i = D(\mathcal{C}_i)$. Also, set $w_0 = 1 - \sum\limits_{i=1}^{t} w_i$, and observe that $w_0 = 1 - \mathrm{NGB}_\eta(\mathcal{S}) \leq \xi$. This shows that $D$ is $(\eta, \xi)$-clustered around $\mathcal{S}$ with weights $w_0, \dots, w_t$, and we are done. □

The correctness proof of TEST-AND-LEARN is in Subsection 8.2.2. Leading to it, in Subsection 8.2.1, we consider some important lemmas and define a set of events. These lemmas, and the events whose probability they bound from below, provide the infrastructure for the proof of TEST-AND-LEARN in Subsection 8.2.2.

### 8.2.1 Preliminaries to prove the correctness of TEST-AND-LEARN

The central goal of this section is to define an event GOOD and show that $\mathbb{P}(\text{GOOD}) \geq 2/3$. The event GOOD is defined in such a fashion that, if it holds, then the algorithm TEST-AND-LEARN produces the desired output as stated in Theorem 8.3. Note that this bounds the error probability of TEST-AND-LEARN. The event GOOD is formally defined in Definition 8.13. To define the event GOOD, we first consider four lemmas: Lemma 8.6, Lemma 8.7, Lemma 8.8 and Lemma 8.12.

We will first state a lemma (Lemma 8.6) which says that, with high probability, the first set of samples $\mathcal{S}$ (obtained in Step (i) of TEST-AND-LEARN) intersects all the large clusters when $D$ is $(\zeta, \delta, r)$-clusterable.

**Lemma 8.6 (Hitting large clusters).** *Assume that the input distribution $D$ over $\{0,1\}^n$ is $(\zeta, \delta, r)$-clusterable with the clusters $\mathcal{C}_1, \ldots, \mathcal{C}_r$. The cluster $\mathcal{C}_i$ is said to be* large *if $D(\mathcal{C}_i) \geq \frac{\zeta}{10r}$. With probability at least $\frac{99}{100}$, the sequence of vectors $\mathcal{S} = \{\mathbf{X}_1, \ldots \mathbf{X}_{t_1}\}$ (found in Step (i) of* TEST-AND-LEARN*) contains at least one vector from every large cluster.*

*Proof.* Consider any large cluster $\mathcal{C}_i$. As $D(\mathcal{C}_i) \geq \frac{\zeta}{10r}$, the probability that no vector in $\mathcal{S}$ belongs to $\mathcal{C}_i$ is at most $(1 - \frac{\zeta}{10r})^{|\mathcal{S}|} \leq \frac{99}{100r}$. This follows for a suitable choice of the hidden coefficient since $|\mathcal{S}| = t_1 = \mathcal{O}\left(\frac{r}{\zeta} \log \frac{r}{\zeta}\right)$. Since there are at most $r$ large clusters, using the union bound, the lemma follows. □

Recall that TEST-AND-LEARN obtains a second set of sample vectors $\mathcal{T}$ in Step (ii), takes a random set of indices $R \subset [n]$ without replacement in Step (iii), and tries to assign each vector in $\mathcal{T}$ to some vector in $\mathcal{S}$, based on the distance between the vectors when projected to the indices of $R$. Intuitively, the step of assigning vectors performs as desired if $R$ preserves the distances between the vectors in $\mathcal{S}$ and $\mathcal{T}$. For technical reasons, we also need $R$ to preserve most (but not all) distances between $\mathcal{S}$ and the entirety of $\{0,1\}^n$. The following lemma says that indeed $R$ achieves this with high probability.

**Lemma 8.7 (Distance preservation).** *Let us consider the input distribution $D$ over $\{0,1\}^n$, and $\mathcal{S} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{t_1}\}$ and $R \subset [n]$ drawn in Step (i) and (iii) of* TEST-AND-LEARN. *$R$ is said to be* distance preserving *if the following conditions hold:*

**(i)** $|d_H(\mathbf{S}, \mathbf{T}) - d_H(\mathbf{S}\mid_R, \mathbf{T}\mid_R)| \leq \delta$ *for every* $\mathbf{S} \in \mathcal{S}$ *and* $\mathbf{T} \in \mathcal{T}$.

**(ii)** *Let $\mathcal{W} \subseteq \{0,1\}^n$ be such that, for every $\mathbf{W} \in \mathcal{W}$, $|d_H(\mathbf{W}, \mathbf{S}) - d_H(\mathbf{W}\mid_R, \mathbf{S}\mid_R)| \leq \delta$. Then $D(\mathcal{W}) \geq 1 - \frac{\zeta}{t_1}$.*

*The set $R$ chosen in Step (iii) of* TEST-AND-LEARN *is distance preserving with probability at least* $99/100$.

*Proof.* For (i), consider a particular $\mathbf{S} \in \mathcal{S}$ and $\mathbf{T} \in \mathcal{T}$. Applying Observation 2.16 with $K = R$, $\mathbf{U} = \mathbf{S}$ and $\mathbf{V} = \mathbf{T}$, the probability that $|d_H(\mathbf{S}, \mathbf{T}) - d_H(\mathbf{S}\mid_R, \mathbf{T}\mid_R)| \leq \delta$ is at least $1 - \frac{\zeta}{200 t_1^2 t_2}$. Applying the union bound over all possible choices over $(\mathbf{S}, \mathbf{T})$ pairs, we have Part (i) with probability at least $199/200$.

To prove (ii), let us consider an arbitrary vector $\mathbf{V} \in \{0,1\}^n$. Similarly to (i), we know that $|d_H(\mathbf{V}, \mathbf{S}) - d_H(\mathbf{V}\mid_R, \mathbf{S}\mid_R)| \leq \delta$ holds with probability at least $1 - \frac{\zeta}{200 t_1^2 t_2}$. Applying the union bound, we can say that the same holds over all $\mathbf{S} \in \mathcal{S}$ with probability at least $1 - \frac{\zeta}{200 t_1}$. So, the expected value of $D(\{0,1\}^n \setminus \mathcal{W})$ is at most $\frac{\zeta}{200 t_1}$. By Markov's inequality, the probability that Part (ii) holds, that is, $D(\{0,1\}^n \setminus \mathcal{W}) \leq \frac{\zeta}{t_1}$ is at least $199/200$. Putting everything together, we have the result. □

By Lemma 8.6, we know that $\mathcal{S}$ intersects with all large clusters with high probability, and we are trying to assign the vectors in $\mathcal{T}$ to some vectors in $\mathcal{S}$ based on their projected distances on the indices of $R$. To learn the input distribution, we want the second set of sample vectors $\mathcal{T}$ to preserve the mass of all the large clusters, and it is enough for us to approximate it, as well as be able to detect the case where approximation is impossible and we should output FAIL. The following lemma takes care of this.

**Lemma 8.8 (Weight representation).** *Let us consider the input distribution $D$ over $\{0,1\}^n$ to* TEST-AND-LEARN, *$\mathcal{S} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{t_1}\}$ in Step (i), $\mathcal{T} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_{t_2}\}$ in*

*Step (ii), and consider fixed $t_1$ pairwise disjoint subsets $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{t_1}\}$ of $\{0,1\}^n$. $\mathcal{T}$ is said to be* weight preserving *for $\mathcal{S}$ and $\mathcal{C}$ if*

**(i)** $\frac{|\mathcal{T} \cap \text{NGB}_\delta(\mathcal{S})|}{|\mathcal{T}|} \geq D(\text{NGB}_\delta(\mathcal{S})) - \zeta.$

**(ii)** $\frac{|\mathcal{T} \cap \text{NGB}_{3\delta}(\mathcal{S})|}{|\mathcal{T}|} \leq D(\text{NGB}_{3\delta}(\mathcal{S})) + \zeta.$

**(iii)** *for every $i \in [t_1]$, $\frac{|\mathcal{T} \cap \mathcal{C}_i|}{|\mathcal{T}|} \leq D(\mathcal{C}_i) + \frac{\zeta}{t_1}$.*

*Then with probability at least $99/100$, $\mathcal{T}$ is weight reserving for $\mathcal{S}$ and $\mathcal{C}$.*

*Proof.* To prove (i), let $Z_j$ be the indicator random variable such that $Z_j = 1$ if and only if $\mathbf{Y}_j$ is in $\text{NGB}_\delta(\mathcal{S})$, where $j \in [t_2]$. Observe that $|\mathcal{T} \cap NGB_\delta(\mathcal{S})| = \sum\limits_{j=1}^{t_2} Z_j$. As $\mathbb{P}(Z_j = 1) = D(\text{NGB}_\delta(\mathcal{S}))$, the expected value of $\frac{|\mathcal{T} \cap \text{NGB}_\delta(\mathcal{S})|}{|\mathcal{T}|}$ is also $D(\text{NGB}_\delta(\mathcal{S}))$. Applying Hoeffding's inequality (see Lemma 2.14), we conclude that (i) holds with probability at least $299/300$.

Proving (ii) is similar to (i). Again applying Hoeffding's inequality (Lemma 2.14), we can show that (ii) holds with probability at least $299/300$.

In order to prove (iii), we proceed in similar fashion as (i), and after applying Hoeffding's inequality (Lemma 2.14), we apply the union bound over all $j \in [t_1]$ to get the desired result. □

Consider the weights $w_1, \ldots, w_{t_1}$ obtained in Step (vi) of TEST-AND-LEARN. To argue that these weights are good enough to report the desired distribution $D'$ (if we know the vectors in $\mathcal{S}$ exactly), we consider the following observation which says that there exist $t_1$ pairwise disjoint subsets $\mathcal{C}_1^*, \ldots, \mathcal{C}_{t_1}^*$ such that $w_i$ is the fraction of vectors in $\mathcal{T}$ that are in $\mathcal{C}_i^*$ for every $i \in [t_1]$. Also, let us define $\mathcal{C}^* = \{\mathcal{C}_1^*, \ldots, \mathcal{C}_{t_1}^*\}$.

**Observation 8.9.** Let us consider assigning each vector in $\{0,1\}^n$ either to some $S \in \mathcal{S}$ or not assigning to any vector in $\mathcal{S}$, using the same procedure that has been used to assign the set of vectors in $\mathcal{T}$ in Steps (iii) and (iv) of TEST-AND-LEARN. Let $\mathcal{C}_i^* \subseteq \{0,1\}^n$ be the set of all vectors that are assigned to $\mathbf{X}_i$, for every $i \in [t_1]$. Then, for every $i \in [t_1]$, we have $w_i = \frac{|\mathcal{T} \cap \mathcal{C}_i^*|}{|\mathcal{T}|}$.

*Proof.* This follows from the definition of $\mathcal{C}_i^*$. □

Note that $\mathcal{C}^*$ is formed following the procedure that TEST-AND-LEARN performs to assign the vectors of $\mathcal{T}$ to the vectors in $\mathcal{S}$. So, a vector far away from $\mathbf{X}_i \in \mathcal{S}$ might be assigned $\mathbf{X}_i$, and $w_i$ is considered in this case. This is not a problem as the mass on $\mathcal{C}_i^*$ is close to being bounded by the total mass of the vectors in $\mathrm{NGB}_{3\delta}(\mathbf{X}_i)$. This follows from the fact that the set $R$ is distance preserving (see Part (ii) of Lemma 8.7) with high probability. Now let us define $\mathcal{C}^{**} = \{\mathcal{C}_i^* \cap \mathrm{NBG}_{3\delta}(\mathbf{X}_i) : i \in [t_1]\}$. Finally, we will upper bound $w_i$ by $D(\mathcal{C}_i^{**})$ in the following observation. This will be useful for proving the correctness of TEST-AND-LEARN in Section 8.2.2.

**Observation 8.10.** Let us assume that $R$ is distance preserving and $\mathcal{T}$ is weight representative of $\mathcal{S}$ and $\mathcal{C}^*$. Then for every $i \in [t_1]$, $w_i \le D(\mathcal{C}_i^*) + \frac{\zeta}{t_1} \le D(\mathcal{C}_i^{**}) + \frac{2\zeta}{t_1}$, where we define $\mathcal{C}_i^{**} = \mathcal{C}_i^* \cap \mathrm{NBG}_{3\delta}(\mathbf{X}_i)$.

*Proof.* As $R$ is distance preserving, consider $\mathcal{C}^* = \{\mathcal{C}_1^*, \ldots, \mathcal{C}_{t_1}^*\}$ as guaranteed by Observation 8.9. Now, as $\mathcal{T}$ is weight representative of $\mathcal{S}$ and $C^*$ and $w_i = \frac{|\mathcal{T} \cap \mathcal{C}_i^*|}{|T|}$ for every $i \in [t_1]$, by Lemma 8.8 (iii), $w_i \le D(\mathcal{C}_i^*) + \frac{\zeta}{t_1}$. By the definition of $\mathcal{C}_i^*$ and by Lemma 8.7 (ii), $D(\mathcal{C}_i^* \setminus \mathrm{NGB}_{3\delta}(\mathbf{X}_i)) \le \frac{\zeta}{t_1}$, that is, $D(\mathcal{C}_i^*) \le D(\mathcal{C}_i^{**}) + \frac{\zeta}{t_1}$. □

Note that the above observation only gives upper bounds on the set of weights $w_1, \ldots, w_{t_1}$. As Lemma 8.8 provides upper as well as lower bounds on the mass around $\mathcal{S}$, this will not be a problem.

Consider the distribution $D^*$ supported over $\mathcal{S}$ such that $D(\mathbf{X}_i) \ge w_i$ for every $i \in [t_1]$, which we can view as an approximation of $D$. Note that we still can not report $D^*$ as the output distribution, since in order to do so, we need to perform $\Omega(n)$ queries to know the exact vectors of $\mathcal{S}$. Instead we will report a distribution $D'$ such that $D'_\sigma$ is close to $D^*$ for some permutation $\sigma : [n] \to [n]$. The idea is to construct a new set of vectors $\mathbf{S}_1, \ldots, \mathbf{S}_{t_1}$ in Step (vii) such that the Hamming distance between $\mathbf{X}_i$ and $\sigma(\mathbf{S}_i)$ is small for every $i \in [t_1]$ for some permutation $\sigma : [n] \to [n]$. Lemma 8.12 implies that this is possible from the projection of the vectors in $\mathcal{S}$ onto the indices of $R$ (the implication itself will be proved later in Lemma 8.17). Before proceeding to Lemma 8.12, we need the following definition and observation.

**Definition 8.11.** Given any sequence of vectors $\mathcal{S} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{t_1}\} \subseteq \{0,1\}^n$ and $j \in [n]$, we define the vector $C_j^{\mathcal{S}} \in \{0,1\}^{t_1}$ as

$$\text{for every } i \in [t_1],\ C_j^{\mathcal{S}}(i) = \mathbf{X}_i(j)$$

For any $J \in \{0,1\}^{t_1}$, we define

$$\alpha_J = \frac{|\{j \in [n] \mid C_j^{\mathcal{S}} = J\}|}{n}.$$

Intuitively, let us consider a matrix $M$ of order $t_1 \times n$ such that the $i$-th row vector corresponds to the vector $\mathbf{X}_i$. Then observe that $C_j^{\mathcal{S}}$ represents the $j$-th column vector of the matrix $M$ and $\alpha_J$ denotes the fraction of column vectors of $M$ that are identical to $J$.

**Lemma 8.12 (Structure preservation).** *Let us consider the input distribution $D$ over $\{0,1\}^n$, $\mathcal{S} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{t_1}\}$ and $R \subset [n]$ drawn in Step (i) and (iii) of* TEST-AND-LEARN. *Also, let us consider the values of $\Gamma_J$ found in Step (iii) of* APPROX-CENTERS *(called from Step (vii) of* TEST-AND-LEARN*). The set $R$ is said to be* structure preserving *for $\mathcal{S}$ if $\left|\alpha_J - \frac{\Gamma_J}{n}\right| \leq \frac{\delta}{10 \cdot 2^{t_1}}$ holds for every $J \in \{0,1\}^{t_1}$. Then the set $R$ chosen in Step (iii) of* TEST-AND-LEARN *is structure preserving for $\mathcal{S}$ with probability at least* $99/100$.

*Proof.* Consider any particular $J \in \{0,1\}^{t_1}$ and $\gamma_J$ determined by Step (ii) of APPROX-CENTERS. Using Hoeffding's bound for sampling without replacement (Lemma 2.15), we obtain, for any $\eta > 0$,

$$\Pr\left[|\gamma_J - \alpha_J| \geq \frac{\eta}{20}\right] \leq e^{-2\eta^2|R|/400}.$$

By substituting the value of $|R|$ (for a suitable choice of the hidden coefficient) and $\eta = \frac{\delta}{2^{t_1}}$, and using the union bound over all possible $J \in \{0,1\}^{t_1}$, we conclude that with probability at least $99/100$, for all $J \in \{0,1\}^{t_1}$, $|\gamma_J - \alpha_J| \leq \frac{\delta}{20 \cdot 2^{t_1}}$.

Note that APPROX-CENTERS constructs $\Gamma_J$'s from $\gamma_j$'s by applying Observation 7.11.

From the way Observation 7.11 generates $\Gamma_J$'s from $\gamma_j$'s, we conclude that for all $J \in \{0,1\}^{t_1}$, $|\gamma_J - \frac{\Gamma_J}{n}| \leq \frac{1}{n}$, completing the proof, assuming that $n$ is larger than $\frac{20 \cdot 2^{t_1}}{\delta}$. □

Now we are ready to define the event GOOD.

**Definition 8.13 (Definition of the event** GOOD**).** Let us define an event GOOD as $\mathcal{E}_1 \wedge \mathcal{E}_2 \wedge \mathcal{E}_3 \wedge \mathcal{E}_4$, where

**(i)** $\mathcal{E}_1$ : If $D$ is $(\zeta, \delta, r)$-clusterable with the clusters $\mathcal{C}_1, \ldots, \mathcal{C}_r$, then $\mathcal{S} = \{\mathbf{X}_1, \ldots \mathbf{X}_{t_1}\}$ (found in Step (i) of TEST-AND-LEARN) contains at least one vector from every large cluster.

**(ii)** $\mathcal{E}_2$ : $R$ (picked in Step (ii) of TEST-AND-LEARN) is distance preserving.

**(iii)** $\mathcal{E}_3$ : $R$ is structure preserving for $\mathcal{S}$.

**(iv)** $\mathcal{E}_4$: $\mathcal{T}$ is weight preserving for $\mathcal{S}$ and $\mathcal{C}^*$, where $\mathcal{C}^* = \{\mathcal{C}_1^*, \ldots, \mathcal{C}_{t_1}^*\}$ is as defined in Observation 8.9.

Note that the event $\mathcal{E}_1$ follows from Lemma 8.6, $\mathcal{E}_2$ follows from Lemma 8.7, $\mathcal{E}_3$ follows from Lemma 8.12, and $\mathcal{E}_4$ follows from Lemma 8.8. Thus, from the respective guarantees of the aforementioned lemmas, we can say that $\mathbb{P}(\mathcal{E}_1), \mathbb{P}(\mathcal{E}_2), \mathbb{P}(\mathcal{E}_3), \mathbb{P}(\mathcal{E}_4) \geq \frac{99}{100}$. To address a subtle point, note that Lemma 8.7 gives a probability lower bound on $R$ being distance preserving for any choice of $\mathcal{T}$, and hence the lower bound also holds for $\mathcal{T}$ sampled according to the distribution. Similarly, Lemma 8.8 provides a probability lower bound on $\mathcal{T}$ being weight representative for any choice of $R$ (which affects $C^*$) regardless of whether $R$ is distance preserving, and hence the lower bound also holds for the $R$ chosen at random by the algorithm. So, we have the following lemma.

**Lemma 8.14.** $\mathbb{P}\left(\text{GOOD}\right) \geq \frac{2}{3}$.

### 8.2.2 Correctness of TEST-AND-LEARN

In the first three lemmas below (Lemma 8.15, Lemma 8.16 and Lemma 8.17), we prove the correctness of the internal steps of the algorithm. These lemmas are stated under the conditional space that the event GOOD defined in Definition 8.13 occurs. Using these lemmas along with Lemma 8.18, which helps us combine them, allows us to prove Theorem 8.3.

**Lemma 8.15 (Guarantee till Step (v) of** TEST-AND-LEARN**).** *Assume that the event* GOOD *holds.*

**(i)** *If $D$ is $(\zeta, \delta, r)$-clusterable, then $D$ is $(\delta, 2\zeta)$-clustered around $\mathcal{S}$, and the fraction of samples in $\mathcal{T}_y$ that are not assigned to any vector in $\mathcal{S}_x$ will be at most $3\zeta$. That is,* TEST-AND-LEARN *does not output* FAIL *in Step (v) and proceeds to Step (vi).*

**(ii)** *If $D$ is not $(3\delta, 5\zeta)$-clustered around $\mathcal{S}$, then the fraction of samples in $\mathcal{T}_y$ that are not assigned to any vector in $\mathcal{S}_x$ will be at least $3\zeta$. That is,* TEST-AND-LEARN *outputs* FAIL *and does not proceed to Step (vi).*

*Proof.* **(i)** For the first part, as $\mathcal{E}_1$ holds (see Lemma 8.6), the set $\mathcal{S}$ contains at least one vector from every large cluster. Now, if we consider the $\delta$-neighborhood of $\mathcal{S}$, that is, $\text{NGB}_\delta(\mathcal{S})$, we infer that all vectors in large clusters are in $\text{NGB}_\delta(\mathcal{S})$. By the definition of a large cluster, the mass on the vectors that are not in any large cluster is at most $2\zeta$. Hence, we conclude that $D(\text{NGB}_\delta(\mathcal{S})) \geq (1 - 2\zeta)$. Thus, by Observation 8.5, $D$ is $(\delta, 2\zeta)$-clustered around $\mathcal{S}$. For the second part, as the event $\mathcal{E}_4$ holds (see Lemma 8.8(i)), $\mathcal{T}$ is weight representative for $\mathcal{S}$. This follows since $D$ is $(\delta, 2\zeta)$-clustered, and in particular is $(3\delta, 5\zeta)$-clustered around $\mathcal{S}$. Thus, $\frac{|\mathcal{T} \cap \text{NGB}_\delta(\mathcal{S})|}{|\mathcal{T}|} \geq D(\text{NGB}_\delta(\mathcal{S})) - \zeta$. Also, as the event $\mathcal{E}_2$ holds (see Lemma 8.7), $R$ is distance preserving between $\mathcal{S}$ and $\mathcal{T}$, meaning that if $\mathbf{Y}_i$ in $\mathcal{C}_j$, then $\mathbf{y}_i$ is assigned to $\mathbf{x}_j$. Hence,

$$\sum_{i=1}^{t_1} w_i \geq \frac{|\mathcal{T} \cap \text{NGB}_\delta(\mathcal{S})|}{|\mathcal{T}|} \geq D(\text{NGB}_\delta(\mathcal{S})) - \zeta \geq 1 - 3\zeta.$$

That is, $w_0 \leq 3\zeta$, and the algorithm TEST-AND-LEARN does not report FAIL and proceeds to Step (vi).

**(ii)** Since the distribution $D$ is not $(3\delta, 5\zeta)$-clustered around $\mathcal{S}$, by Observation 8.5, $D(\text{NGB}_{3\delta}(\mathcal{S})) < 1 - 5\zeta$. Moreover, following Lemma 8.8 (ii), the event $\mathcal{E}_4$ holds. So, $\frac{|\mathcal{T} \cap \text{NGB}_{3\delta}(\mathcal{S})|}{|\mathcal{T}|} \leq D(\text{NGB}_{3\delta}(\mathcal{S})) + \zeta \leq 1 - 4\zeta$. Also, as the event $\mathcal{E}_2$ holds (see Lemma 8.7), $R$ is distance preserving between $\mathcal{S}$ and $\mathcal{T}$. This implies that

$$\sum_{i=1}^{t_1} w_i \leq \frac{|\mathcal{T} \cap \text{NGB}_{3\delta}(\mathcal{S})|}{|\mathcal{T}|} \leq D(\text{NGB}_{3\delta}(\mathcal{S})) + \zeta < 1 - 3\zeta.$$

That is, $w_0 > 3\zeta$, and the algorithm TEST-AND-LEARN reports FAIL. So, TEST-AND-LEARN does not proceed to Step (vi).

□

**Lemma 8.16 (Guarantee from Step (vi) of TEST-AND-LEARN).** *Assume that the event* GOOD *holds. Moreover, assume that D is $(3\delta, 5\zeta)$-clustered around $\mathcal{S}$ and $w_0 \leq 3\zeta$ holds in Step (vi) of* TEST-AND-LEARN. *Consider the following distribution $D''$ over $\{0,1\}^n$, constructed from the weights obtained from Step (vi) of* TEST-AND-LEARN, *such that*

**(i)** *For each $i \in [t_1]$, $D''(\mathbf{X}_i) = w(\mathbf{x}_i) = w_i$.*

**(ii)** $D''(\mathbf{X}_0) = 1 - \sum_{i=1}^{t_1} w(\mathbf{x}_i)$ *for some arbitrary* $\mathbf{X}_0$.

**(iii)** $D''(\mathbf{X}) = 0$ *for every* $\mathbf{X} \in \{0,1\}^n \setminus \{\mathbf{X}_0, \ldots, \mathbf{X}_{t_1}\}$.

*Then $D''$ is $(5\delta, 5\zeta)$-clustered around $\mathcal{S}$ with weights $w_0, \ldots, w_{t_1}$, where $w_0 = 1 - \sum_{i=1}^{t_1} w_i$, and the EMD between $D$ and $D''$ satisfies $d_{EM}(D, D'') \leq 10\delta + 12\zeta$.*

We will prove Lemma 8.16 in Subsection 8.2.2. Now we proceed to prove the guarantee regarding Step (vii) of TEST-AND-LEARN.

**Lemma 8.17 (Guarantee from Step (vii) of TEST-AND-LEARN).** *Assume that the event* GOOD *holds. Then, in Step (vii), the algorithm* APPROX-CENTERS *(if called* *as described in Algorithm 8.2) outputs a sequence of vectors $\{\mathbf{S}_1, \ldots, \mathbf{S}_{t_1}\}$ in $\{0,1\}^n$, such that there exists a permutation $\sigma : [n] \to [n]$ for which $d_H(\sigma(\mathbf{X}_i), \mathbf{S}_i) \leq \frac{\delta}{10}$ holds for every $i \in [t_1]$.*

*Proof.* Here we assume that the event GOOD holds. In particular, we assume that the event $\mathcal{E}_3$ holds.

Let us consider a matrix $M$ of order $t_1 \times n$ such that the $i$-th row vector corresponds to the vector $\mathbf{X}_i$. Then observe that $C_j^{\mathcal{S}}$ represents the $j$-th column vector of matrix $M$ and $\alpha_J$ denotes the fraction of column vectors of $M$ that are identical to the vector $J$.

Let us consider the matrix $A$ of order $t_1 \times n$ constructed by our algorithm, by putting $\Gamma_J$ many column vectors identical to $J$, for every $J \in \{0,1\}^{t_1}$. Note that $\{\mathbf{S}_1, \ldots, \mathbf{S}_{t_1}\}$ are the row vectors corresponding to $A$. As we are assuming that the event $\mathcal{E}_3$ holds (see Lemma 8.12), $|\alpha_J - \frac{\Gamma_J}{n}| \leq \frac{\delta n}{10 \cdot 2^{t_1}}$ holds for every $J \in \{0,1\}^{t_1}$. Observe that we can permute the columns of the matrix $M$ using a permutation $\sigma : [n] \to [n]$ and create a matrix $M_\sigma$, such that there exists a bad set $I \subset [n]$ of size at most $\frac{\delta \cdot n}{10}$, where after the removal of the columns corresponding to indices of $I$ from both matrices $M_\sigma$ and $A$ become identical. Hence, we infer that $d_H(\sigma(\mathbf{X}_i), \mathbf{S}_i) \leq \frac{\delta}{10}$ for every $i \in [t_1]$, where $\sigma$ is the permutation corresponding to $M_\sigma$. This completes the proof of Lemma 8.17. □

Finally, to prove Theorem 8.3, we need to show that the Earth Mover Distance between two distributions defined over close vectors is bounded when one distribution is clustered around a sequence of vectors and the other distribution has similar weights compared to the first distribution.

**Lemma 8.18 (EMD between distributions having close cluster centers).** *Let $\eta, \kappa, \xi \in (0,1)$ be three parameters such that $\eta + \kappa + \xi < 1$. Suppose that $\mathcal{S} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{t_1}\}$ and $\mathcal{S}' = \{\mathbf{X}'_1, \ldots, \mathbf{X}'_{t_1}\}$ are two sequences of vectors over $\{0,1\}^n$ such that $d_H(\mathbf{X}_i, \mathbf{X}'_i) \leq \kappa$ for every $i \in [t_1]$. Moreover, let $D$ be an $(\eta, \xi)$-clustered distribution around $\mathcal{S}$ with weights $w_0, \ldots, w_{t_1}$ and $D'$ be another distribution such that $D'(\mathbf{X}'_i) \geq w_i$ for every $i \in [t_1]$. Then $d_{EM}(D, D') \leq \eta + \xi + \kappa$.*

*Proof.* Recall that the EMD between $D$ and $D'$ is the solution to the following LP:

$$\begin{array}{ll} \text{Minimize} & \displaystyle\sum_{\mathbf{X},\mathbf{Y}\in\{0,1\}^n} f_{\mathbf{XY}} d_H(\mathbf{X},\mathbf{Y}) \\ \text{Subject to} & \displaystyle\sum_{\mathbf{Y}\in\{0,1\}^n} f_{\mathbf{XY}} = D(\mathbf{X})\ \forall \mathbf{X}\in\{0,1\}^n, \sum_{\mathbf{X}\in\{0,1\}^n} f_{\mathbf{XY}} = D'(\mathbf{Y})\ \forall \mathbf{Y}\in\{0,1\}^n \\ \text{and} & 0 \leq f_{\mathbf{XY}} \leq 1, \quad \forall\, \mathbf{X},\mathbf{Y}\in\{0,1\}^n. \end{array}$$

Here $D$ is $(\eta,\xi)$-clustered around $\mathcal{S}$. Let $\mathcal{C}_1,\ldots,\mathcal{C}_{t_1}$ be the pairwise disjoint subsets of $\{0,1\}^n$ such that $\mathcal{C}_i \subseteq \text{NGB}_\eta(\mathbf{X}_i)$ and $D(\mathcal{C}_i) \geq w_i$ for every $i\in[t_1]$.

Consider a particular solution $\{f^*_{\mathbf{XY}} : \mathbf{X},\mathbf{Y}\in\{0,1\}^n\}$ that also satisfies the constraint

$$\sum_{\mathbf{X}\in\mathcal{C}_i} f_{\mathbf{X}\mathbf{X}'_i} \geq w_i \text{ for every } i\in[t_1].$$

The above constraint is feasible as $D(\mathcal{C}_i)\geq w_i$ and $D'(\mathbf{X}'_i)\geq w_i$, where $i\in[t_1]$.

Now,

$$\begin{aligned} d_{EM}(D,D') &\leq \sum_{\mathbf{X},\mathbf{Y}\in\{0,1\}^n} f^*_{\mathbf{XY}} d_H(\mathbf{X},\mathbf{Y}) \\ &\leq \sum_{i=1}^{t_1}\sum_{\mathbf{X}\in\mathcal{C}_i} f^*_{\mathbf{X}\mathbf{X}'_i} d_H(\mathbf{X},\mathbf{X}'_i) + \sum_{\mathbf{X}\notin\bigcup_{i=1}^{t_1}\mathcal{C}_i,\mathbf{Y}\in\{0,1\}^n} f^*_{\mathbf{XY}} d_H(\mathbf{X},\mathbf{Y}) \\ &\leq \sum_{i=1}^{t_1} w_i\cdot(\eta+\kappa) + w_0\cdot 1 \\ &\leq \eta+\kappa+\xi. \end{aligned}$$

□

## Proof of Theorem 8.3

To prove Theorem 8.3, we need the following lemma.

**Lemma 8.19.** *If $D$ is $(3\delta, 5\zeta)$-clustered around $S$, and* TEST-AND-LEARN *executes Step (vi), then $d_{EM}(D, D'_\sigma) \leq 17(\delta+\zeta)$ for some permutation $\sigma : [n] \to [n]$.*

*Proof.* As $D$ is $(3\delta, 5\zeta)$-clustered around $\mathcal{S}$, by Lemma 8.16, we have that $D''$ is $(5\delta, 5\zeta)$-clustered around $\mathcal{S}$ with weights $w_0, \ldots, w_{t_1}$ and $d_{EM}(D, D'') \leq 10\delta + 12\zeta$.

Now consider Step (vii) of TEST-AND-LEARN, where we call APPROX-CENTERS with $R$ and $\mathbf{x}_1, \ldots, \mathbf{x}_{t_1}$ to obtain $\mathbf{S}_1, \ldots, \mathbf{S}_{t_1}$. By Lemma 8.17, $d_H(\sigma(\mathbf{X}_i), \mathbf{S}_i) \leq \frac{\delta}{10}$ for every $i \in [t_1]$ for some permutation $\sigma : [n] \to [n]$. Consider the sequence of vectors $\mathbf{X}_1^\sigma \ldots, \mathbf{X}_{t_1}^\sigma$ where $\mathbf{X}_i^\sigma = \sigma(\mathbf{X}_i)$ for every $i \in [t_1]$.

Let us now consider the distribution $D''_\sigma$ over $\{0,1\}^n$ such that $D''_\sigma(\mathbf{X}) = D''(\sigma(\mathbf{X}))$ for every $\mathbf{X} \in \{0,1\}^n$. As $D''$ is $(5\delta, 5\zeta)$-clustered around $\mathcal{S} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{t_1}\}$ with weights $w_0, \ldots, w_{t_1}$, we know that $D''_\sigma$ is $(5\delta, 5\zeta)$-clustered around $\{\mathbf{X}_1^\sigma, \ldots, \mathbf{X}_{t_1}^\sigma\}$ with weights $w_0, \ldots, w_{t_1}$. In the output distribution $D'$, $D'(\mathbf{S}_i) \geq w_i$ for every $i \in [t_1]$. So, by Lemma 8.18, we have $d_{EM}(D', D''_\sigma) \leq 5\delta + \frac{\delta}{10} + 5\zeta$. Combining this with the fact that $d_{EM}(D, D'') \leq 10\delta + 12\zeta$, we conclude that $d_{EM}(D, D'_\sigma) \leq 17(\delta+\zeta)$. □

To prove Theorem 8.3, we first prove that the guarantees of the two parts follow assuming that the event GOOD holds. We will be done since $\mathbb{P}(\text{GOOD}) \geq 2/3$ (see Lemma 8.14). The query complexity of the algorithm follows from the parameters in its description.

*Proof of Part (i):* Here $D$ is $(\zeta, \delta, r)$-clusterable. By Lemma 8.15, $D$ is $(\delta, 2\zeta)$-clustered around $\mathcal{S}$ and the fraction of samples in $\mathcal{T}_y$ that are not assigned to any vector in $\mathcal{S}_x$ is at most $3\zeta$. That is, TEST-AND-LEARN does not output FAIL for $D$ in Step (v). By Lemma 8.19, we conclude that $d_{EM}(D, D'_\sigma) \leq 17(\delta+\zeta)$ for some permutation $\sigma : [n] \to [n]$. This completes the proof of Part (i). □

*Proof of Part (ii):* Recall that we are working under the conditional space that the event GOOD holds. Now consider the following:

- If $D$ is not $(3\delta, 5\zeta)$-clustered around $\mathcal{S}$, then by Lemma 8.15, the algorithm TEST-AND-LEARN reports FAIL.

- If $D$ is $(3\delta, 5\zeta)$-clustered around $\mathcal{S}$, then the algorithm TEST-AND-LEARN either reports FAIL in Step (v) or continues to Step (vi). In case we go to Step (vi), following Lemma 8.19, we again conclude that $d_{EM}(D, D'_\sigma) \leq \varepsilon$.

Observe that the above two statements imply Part (ii). This completes the proof of Theorem 8.3. □

## Proof of Lemma 8.16

Here we assume that the event GOOD holds. In particular, the events $\mathcal{E}_2$ and $\mathcal{E}_4$ hold. To prove Lemma 8.16, we will prove some associated claims and lemmas about the weights $w_0, \dots, w_{t_1}$ obtained in Step (vi) of TEST-AND-LEARN, and the distribution $D''$ defined in Lemma 8.16. Let us start with the following claim.

**Claim 8.20.** *The distribution $D^{''}$ (defined in the statement of Lemma 8.16) is $(5\delta, 5\zeta)$-clustered around $\mathcal{S}$ with weights $w_0, w_1 \dots, w_{t_1}$, where $w_0 = 1 - \sum\limits_{i=1}^{t_1} w_i$.*

*Proof.* This follows from the definition of $D''$, and the fact that $w_0 \leq 3\zeta < 5\zeta$. □

Now we have the following claim.

**Claim 8.21.** *There exists a sequence of weights $w'_0, \dots, w'_{t_1}$ such that $D$ is $(5\delta, 5\zeta)$-clustered around $\mathcal{S}$ with weights $w'_0, \dots, w'_{t_1}$, and $\sum\limits_{i=1}^{t_1} |w_i - w'_i| \leq 2\zeta$.*

*Proof.* As events $\mathcal{E}_2$ and $\mathcal{E}_4$ hold, consider $\mathcal{C}^* = \{\mathcal{C}^*_1, \dots, \mathcal{C}^*_{t_1}\}$ (as guaranteed by Observation 8.9) and $\mathcal{C}^{**} = \{\mathcal{C}^{**}_1, \dots, \mathcal{C}^{**}_{t_1}\}$ such that, for every $i \in [t_1]$, $\mathcal{C}^{**}_i = \mathcal{C}^*_i \cap \mathrm{NGB}_{3\delta}(\mathbf{X}_i)$ and $w_i \leq D(\mathcal{C}^{**}_i) + \frac{2\zeta}{t_1}$ (see Observation 8.10).

Let us define $w'_i = \max\{w_i - \frac{2\zeta}{t_1}, 0\}$ and $w'_0 = 1 - \sum\limits_{i=1}^{t_1} w'_i$. So, $w'_i \leq D\left(\mathcal{C}^{**}_i\right)$.

Now

$$w'_0 = 1 - \sum_{i=1}^{t_1} w'_i \leq 1 - \sum_{i=1}^{t_1} \left(w_i - \frac{2\zeta}{t_1}\right) \leq (w_0 + 2\zeta) \leq 3\zeta + 2\zeta = 5\zeta.$$

Putting everything together, the above $\mathcal{C}^{**}$ satisfies $\mathcal{C}^{**}_i \subseteq \mathrm{NGB}_{3\delta}(\mathbf{X}_i) \subseteq \mathrm{NGB}_{5\delta}(\mathbf{X}_i)$ and has weights $w'_0, \dots, w'_{t_1}$ such that $w'_0 \leq 5\zeta$ and $w'_i \leq D(\mathcal{C}^{**}_i)$ for every $i \in$

$[t_1]$. Hence, $D$ is $(5\delta, 5\zeta)$-clustered around $\mathcal{S}$ with weights $w'_0, \ldots, w'_{t_1}$. Moreover, $\sum\limits_{i=1}^{t_1} |w_i - w'_i| \leq 2\zeta$ holds following the definition of $w'_i$s. □

**Lemma 8.22 (Comparison-by-weights).** *Let $D_1$ and $D_2$ be two distributions defined over $\{0,1\}^n$ that are $(\eta, \xi)$-clustered around a sequence of vectors $\mathcal{S} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{t_1}\}$ with weights $v_0, \ldots, v_{t_1}$ and $w_0, \ldots, w_{t_1}$, respectively. Then the Earth Mover Distance between $D_1$ and $D_2$ is $d_{EM}(D_1, D_2) \leq 2\eta + \sum\limits_{i=1}^{t_1} |v_i - w_i| + 2\xi$.*

*Proof.* Let $\mathbf{U}$ be an arbitrary vector from $\{0,1\}^n$. Let us define a distribution $D'_1$ (supported over $\mathcal{S} \cup \{\mathbf{U}\}$) from the distribution $D_1$ as follows:

$$D'_1(\mathbf{Y}) = \begin{cases} v_i & \mathbf{Y} = \mathbf{X}_i \text{ for every } i \in [t_1] \\ 1 - \sum\limits_{i=1}^{t_1} v_i & \mathbf{Y} = \mathbf{U} \\ 0 & \text{otherwise} \end{cases}$$

Similarly, we define a distribution $D'_2$ from $D_2$. First we have the following claim, which follows from the definitions. From the definitions of $D'_1$ and $D'_2$, we can say that

**(i)** $d_{EM}(D_1, D'_1) \leq \eta + \xi$ and $d_{EM}(D_2, D'_2) \leq \eta + \xi$ (by Lemma 8.18).

**(ii)** $d_{EM}(D'_1, D'_2) \leq \sum\limits_{i=1}^{t_1} |v_i - w_i|$.

Using the triangle inequality, we have

$$\begin{aligned} d_{EM}(D_1, D_2) &\leq d_{EM}(D_1, D'_1) + d_{EM}(D'_1, D'_2) + d_{EM}(D_2, D'_2) \\ &\leq 2\eta + \sum_{i=1}^{t_1} |v_i - w_i| + 2\xi. \end{aligned}$$

This completes the proof of Lemma 8.22. □

Now we proceed to prove Lemma 8.16.

*Proof of Lemma 8.16.* By the description of $D''$ in Lemma 8.16, using Claim 8.20, we know that $D''$ is $(5\delta, 5\zeta)$-clustered around $\mathcal{S}$ with weights $w_1, \ldots, w_{t_1}$. By applying

Claim 8.21, we infer that $D$ is $(5\delta, 5\zeta)$-clustered around $\mathcal{S}$ with weights $w'_0, \ldots, w'_{t_1}$ such that $\sum_{i=1}^{t_1} |w_i - w'_i| \leq 2\zeta$. Now, by applying Lemma 8.22 with $\eta = 5\delta$, $\xi = 5\zeta$, we obtain that the Earth Mover Distance between $D$ and $D''$ is bounded as follows:

$$d_{EM}(D, D'') \leq 10\delta + 2\zeta + 10\zeta \leq 10\delta + 12\zeta.$$

This completes the proof of Lemma 8.16. □

# Chapter 9

# Testing VC dimension properties

## 9.1 Introduction

In this chapter we will prove that distributions over $\{0,1\}^n$ whose support have bounded VC-dimension can be learnt (up to permutations) by performing a number of queries that is independent of the dimension $n$, and depends only on the proximity parameter $\varepsilon$ and the VC-dimension $d$ (Theorem 7.2). In fact, we will prove a generalization, that any distribution $D$ that is $\beta$-close to bounded VC-dimension can be learnt efficiently up to permutations (with a proximity parameter depending on $\beta$) by performing a set of queries whose size is independent of $n$ (Theorem 9.3). The result is formally stated as follows:

**Theorem 9.1 (Learning a distribution $\beta$-close to bounded VC-dimension, Theorem 1.7 generalized).** *Let $d \in \mathbb{N}$ be a constant. There exists a (non-adaptive) algorithm, that given sample and query access to an unknown distribution $D$ over $\{0,1\}^n$, takes $\alpha, \beta \in (0,1)$ with $\beta < \alpha$ as input such that $\varepsilon = 17(3\alpha + \beta/\alpha) < 1$, makes number of queries that depends only on $\alpha, \beta$ and $d$, and either reports a full description of a distribution, or* FAIL*, satisfying both of the following conditions:*

**(i)** *If $D$ is $\beta$-close to VC-dimension $d$, then with probability at least $2/3$, the algorithm outputs a distribution $D'$ such that $d_{EM}(D, D'_\sigma) \leq \varepsilon$ for some permutation $\sigma$ :*

$[n] \to [n]$.

**(ii)** *For any $D$, the algorithm will not output a distribution $D'$ such that $d_{EM}(D, D'_\sigma) > \varepsilon$ for every permutation $\sigma : [n] \to [n]$ with probability more than $\frac{1}{3}$. However, if the distribution $D$ is not $\beta$-close to VC-dimension $d$, the algorithm may output* FAIL *with any probability.*

As a consequence of the learning result of Theorem 9.1, we also obtain a tester for properties having a bounded VC-dimension.

**Corollary 9.2 (Restatement of Corollary 7.4).** *Let $d \in \mathbb{N}$ be a constant, and $\mathcal{P}$ be an index-invariant property with VC-dimension $d$. There exists an algorithm that has sample and query access to an unknown distribution $D$, takes a parameter $\varepsilon \in (0, 1)$, and distinguishes whether $D \in \mathcal{P}$ or $D$ is $\varepsilon$-far from $\mathcal{P}$ with probability at least $2/3$, where the total number of queries made by the algorithm is a function of only $d$ and $\varepsilon$.*

In Subsection 9.2, we connect the notions of $(\zeta, \delta, r)$-clusterablity and being $\beta$-close to $(\alpha, r)$-clusterablity (Definition 8.2) in Lemma 9.6 and prove Corollary 9.5 regarding learning distributions that are $\beta$-close to $(\alpha, r)$-clusterable. Then, we recall some standard results from VC theory to connect the notions of bounded VC-dimension and clusterability, to obtain Corollary 9.14, which is crucially used in Subsection 9.2.1 to prove Theorem 9.3.

## 9.2 Testing properties with bounded VC-dimension

**Theorem 9.3 (Learning a distribution $\beta$-close to bounded VC-dimension, Theorem 9.1 restated).** *Let $d \in \mathbb{N}$ be a constant. There exists a (non-adaptive) algorithm, that given sample and query access to an unknown distribution $D$ over $\{0, 1\}^n$, takes $\alpha, \beta \in (0, 1)$ with $\beta < \alpha$ as input such that $\varepsilon = 17(3\alpha + \beta/\alpha) < 1$, makes number of queries that depends only on $\alpha, \beta$ and $d$, and either reports a full description of a distribution, or* FAIL*, satisfying both of the following conditions:*

**(i)** *If $D$ is $\beta$-close to VC-dimension $d$, then with probability at least $2/3$, the algorithm outputs a distribution $D'$ such that $d_{EM}(D, D'_\sigma) \leq \varepsilon$ for some permutation $\sigma : [n] \rightarrow [n]$.*

**(ii)** *For any $D$, the algorithm will not output a distribution $D'$ such that $d_{EM}(D, D'_\sigma) > \varepsilon$ for every permutation $\sigma : [n] \rightarrow [n]$ with probability more than $\frac{1}{3}$. However, if the distribution $D$ is not $\beta$-close to VC-dimension $d$, the algorithm may output* FAIL *with any probability.*

**Remark 9.1.** Note that $\alpha$ above does not appear anywhere outside the expression for $\varepsilon$, and hence it is tempting to minimize $\varepsilon$ by taking $\alpha = \sqrt{\beta/3}$. However, this is a bad strategy since the number of queries of the algorithm depends on $1/\alpha$. In the common scenario, we would be given $\beta$ and $\varepsilon \geq 34\sqrt{3\beta}$, and solve for $\alpha$.

**Corollary 9.4 (Testing properties with bounded VC-dimension, Corollary 9.2 restated).** *Let $d \in \mathbb{N}$ be a constant, and $\mathcal{P}$ be an index-invariant property with VC-dimension $d$. There exists an algorithm that has sample and query access to an unknown distribution $D$, takes a parameter $\varepsilon \in (0, 1)$, and distinguishes whether $D \in \mathcal{P}$ or $D$ is $\varepsilon$-far from $\mathcal{P}$ with probability at least $2/3$, where the total number of queries made by the algorithm is a function of only $d$ and $\varepsilon$.*

**Remark 9.2.** Note that the algorithm for testing the index-invariant property with constant VC-dimension $d$ takes $\exp(d)$ samples, and performs $\exp(\exp(d))$ queries. It turns out that similarly to the case of TEST-AND-LEARN, the dependencies of the sample and query complexities on $d$ are tight, in the sense that there exists a property of VC-dimension $d$ such that testing it requires $2^{\Omega(d)}$ samples, and $\Omega(2^{2^{d-\mathcal{O}(1)}})$ queries. We will construct such a property and prove its lower bound in Section 9.3.

We will give the proof of Theorem 9.3 in Subsection 9.2.1.

## A corollary of Theorem 8.3 to prove Theorem 9.3:

Here we first connect the notions of $(\zeta, \delta, r)$-clusterablity and being $\beta$-close to $(\alpha, r)$-clusterablity (Definition 8.2) in Lemma 9.6. Then using Lemma 9.6 with our algorithm for learning $(\zeta, \delta, r)$-clusterable distributions (Theorem 8.3), we prove Corollary 9.5 regarding learning distributions that are $\beta$-close to $(\alpha, r)$-clusterable. This corollary will be used later to prove Theorem 9.3.

**Corollary 9.5 (Learning distributions $\beta$-close to $(\alpha, r)$-clusterable).** *Let $n \in \mathbb{N}$. There exists a (non-adaptive) algorithm, that has sample and query access to an unknown distribution $D$ over $\{0,1\}^n$, takes parameters $\alpha, \beta, r$ as inputs such that $\alpha > \beta$ and $\varepsilon = 17(3\alpha + \beta/\alpha) < 1$ and $r \in \mathbb{N}$, makes a number of queries that only depends on $\alpha, \beta$ and $r$, and either reports a full description of a distribution over $\{0,1\}^n$ or reports* FAIL, *satisfying both of the following conditions:*

**(i)** *If $D$ is $\beta$-close to $(\alpha, r)$-clusterable, then with probability at least $2/3$, the algorithm outputs a full description of a distribution $D'$ over $\{0,1\}^n$ such that for some permutation $\sigma : [n] \to [n]$, $d_{EM}(D, D'_\sigma) \leq \varepsilon$.*

**(ii)** *For any $D$, the algorithm will not output a distribution $D'$ such that $d_{EM}(D, D'_\sigma) > \varepsilon$ for every permutation $\sigma : [n] \to [n]$, with probability more than $1/3$. However, if the distribution $D$ is not $\beta$-close to $(\alpha, r)$-clusterable, the algorithm may output* FAIL *with any probability.*

To prove the above corollary, we need the following lemma, that connects the two notions of clusterability: $(\zeta, \delta, r)$-clusterablity and being $\beta$-close to $(\alpha, r)$-clusterability (see Definition 8.2).

**Lemma 9.6.** *Let $\alpha, \beta \in (0,1)$ be such that $\alpha > \beta$, and $D$ be a distribution over $\{0,1\}^n$ that is $\beta$-close to being $(\alpha, r)$-clusterable. Then $D$ is $(3\alpha, r, \beta/\alpha)$-clusterable.*

*Proof.* Let $D_0$ be the distribution such that $D_0$ is $(\alpha, r)$-clusterable and $d_{EM}(D, D_0) \leq \beta$. Let $\mathcal{C}_1, \ldots, \mathcal{C}_s$ be the partition of the support of $D_0$ that realizes the $(\alpha, r)$-clusterability of $D_0$, and let $\{f_{\mathbf{XY}} : \mathbf{X}, \mathbf{Y} \in \{0,1\}^n\}$ be the flow that realizes $d_{EM}(D, D_0) \leq \beta$.

Let $\mathcal{C} = \bigcup_{i=1}^{s} \mathcal{C}_i$, and $\mathcal{C}_{>\alpha}$ be the set of vectors in $\{0,1\}^n$ that have distance of at least $\alpha$ from all the vectors in $\mathcal{C}$. Now we have the following claim.

**Claim 9.7.** $D(\mathcal{C}_{>\alpha}) \leq \frac{\beta}{\alpha}$.

*Proof.* By contradiction, let us assume that $D(\mathcal{C}_{>\alpha}) > \frac{\beta}{\alpha}$. Then we have the following:

$$d_{EM}(D, D_0) \;\; \geq \sum_{\mathbf{X} \in \mathcal{C}_{>\alpha}, \mathbf{Y} \in \mathcal{C}} f_{\mathbf{XY}} d_H(\mathbf{X}, \mathbf{Y}) \geq \alpha \cdot D(\mathcal{C}_{>\alpha}) > \beta.$$

This is a contradiction as we have assumed $d_{EM}(D, D_0) \leq \beta$. □

Now for every $i$, let $\mathcal{C}_i^{\leq \alpha}$ be the vectors that have distance at most $\alpha$ from at least one vector $\mathcal{C}_i$, where $i \in [s]$. Let $\mathcal{C}'_i = \mathcal{C}_i^{\leq \alpha} \setminus \bigcup\limits_{j=1}^{i-1} \mathcal{C}'_j$ for $1 \leq i \leq s$. Now we have the following observation.

**Observation 9.8.** For any $1 \leq i \leq s$, $d_H(\mathbf{U}, \mathbf{V}) \leq 3\alpha$ for any $\mathbf{U}, \mathbf{V} \in \mathcal{C}'_i$.

*Proof.* Since $\mathbf{U}, \mathbf{V} \in \mathcal{C}'_i$, let $\mathbf{U}'$ and $\mathbf{V}'$ be the vectors in $\mathcal{C}_i$ such that $d_H(\mathbf{U}, \mathbf{U}') \leq \alpha$, and $d_H(\mathbf{V}, \mathbf{V}') \leq \alpha$. As $\mathbf{U}', \mathbf{V}' \in \mathcal{C}_i$, and $D_0$ is $(\alpha, r)$-clusterable, using the triangle inequality, we can say that $d_H(\mathbf{U}, \mathbf{V}) \leq d_H(\mathbf{U}, \mathbf{U}') + d_H(\mathbf{U}', \mathbf{V}') + d_H(\mathbf{V}', \mathbf{V}) \leq 3\alpha$. □

Consider $\mathcal{C}'_0 = \mathcal{C}_{>\alpha}$, and by Claim 9.7, note that $D(\mathcal{C}'_0) \leq \beta/\alpha$. The existence of $\mathcal{C}'_0, \mathcal{C}'_1, \ldots, \mathcal{C}'_s$ as above implies that $D$ is $(3\alpha, r, \beta/\alpha)$-clusterable (see Definition 8.2). □

*Proof of Corollary 9.5 using Theorem 8.3 & Lemma 9.6.* The algorithm here (say ALG) calls algorithm TEST-AND-LEARN (as described in Algorithm 8.1) with parameters $\zeta = \beta/\alpha$ and $\delta = 3\alpha$, and reports the output returned by TEST-AND-LEARN as the output of ALG. Now we prove the correctness of ALG.

**Part (i):** Here we consider the case where $D$ is $\beta$-close to $(\alpha, r)$-clusterable. Following Lemma 9.6, $D$ is $(\zeta, \delta, r)$-clusterable. By Theorem 8.3 (i), we get a distribution $D'$ such that $d_{EM}(D, D'_\sigma) \leq 17(\zeta + \delta) = 17(3\alpha + \beta/\alpha) = \varepsilon$ for some permutation $\sigma : [n] \to [n]$, with probability at least 2/3. This completes the proof of Part (i).

**Part (ii):** This follows from Theorem 8.3 (ii) along with our choices of $\delta = 3\alpha$ and $\zeta = \beta/\alpha$.

□

## Result from VC theory

Let us now recall some definitions from VC-dimension theory, and use a well known result of Haussler [Hau95] to obtain Corollary 9.14, which states that if the VC-dimension of a set of vectors $V$ is bounded, then the vectors of $V$ can be covered by bounded number of Hamming balls. This corollary will be crucially used to prove Theorem 9.3 in Subsection 9.2.1.

Let us start by defining the notion of an $\alpha$-separated set.

**Definition 9.9** (**$\alpha$-separated set).** Let $\alpha \in (0,1)$ and $W \subset \{0,1\}^n$ be a set of vectors. $W$ is said to be *$\alpha$-separated* if for any two vectors $\mathbf{X}, \mathbf{Y} \in W$, $d_H(\mathbf{X}, \mathbf{Y}) \geq \alpha$.

Now let us define the notion of the $\alpha$-packing number of a set of vectors.

**Definition 9.10** (**$\alpha$-packing number).** Let $\alpha \in (0,1)$, and $V \subset \{0,1\}^n$ be a set of vectors. The *$\alpha$-packing number* of $V$, denoted by $\mathcal{M}(\alpha, V)$, is defined as the cardinality of the largest $\alpha$-separated subset $W$ of $V$.

Now we define the notion of an $\alpha$-cover of a set of vectors.

**Definition 9.11** (**$\alpha$-cover).** Let $\alpha \in (0,1)$ and $V \subset \{0,1\}^n$ be a set of vectors. A set of vectors $M \subseteq V$ is an *$\alpha$-cover* of $V$ if $V \subseteq \bigcup\limits_{\mathbf{p} \in M} \mathrm{NGB}_\alpha(\mathbf{p})$, where $NGB_\alpha(\mathbf{p}) := \{\mathbf{q} : d_H(\mathbf{p}, \mathbf{q}) \leq \alpha\}$ denotes the set of vectors that are within Hamming distance $\alpha$ from the vector $\mathbf{p}$.

Now let us consider the following theorem from [Hau95], which says that if the VC-dimension of a set of vectors $V$ is $d$, then the size of the $\alpha$-packing number of $V$, that is, $\mathcal{M}(\alpha, V)$, is bounded by a function of $d$ and $\alpha$.

**Theorem 9.12 (Haussler's packing theorem [Hau95, Theorem 1]).** *Let $\alpha \in (0,1)$ be a parameter. If the VC-dimension of a set of vectors $V$ is $d$, then the $\alpha$-packing number* *of $V$ is bounded as follows:*

$$\mathcal{M}(\alpha, V) \leq e(d+1)\left(\frac{2e}{\alpha}\right)^d$$

The following observation is immediate.

**Observation 9.13.** Let $\alpha \in (0,1)$ be a parameter and $M$ be a maximal $\alpha$-packing of a set of vectors $V \subset \{0,1\}^n$. Then $M$ is also an $\alpha$-cover of $V$.

With this observation, along with Theorem 9.12, we get the following bound on the size of a cover of a set of vectors in terms of its VC-dimension.

**Corollary 9.14 (Existence of a small $\alpha$-cover).** *Let $d \in \mathbb{N}$. If the VC-dimension of a set of vectors $V$ is $d$, then for all $\alpha \in (0,1)$, there exists a set $M \subseteq V$ such that $M$ is an $\alpha$-cover of $V$ and $|M| \leq e(d+1)\left(\frac{2e}{\alpha}\right)^d$.*

### 9.2.1 Learning distributions close to having bounded VC-dimension

In this subsection, using Corollary 9.5, we prove that any distribution that is $\beta$-close to bounded VC-dimension can be learnt (up to permutation) by performing a number of queries that depends only on the VC-dimension $d$ and the proximity parameter $\varepsilon$, and is independent of the dimension of the Hamming cube $\{0,1\}^n$ (Theorem 9.3). The crucial ingredient of the proof is Theorem 9.12, through its Corollary 9.14. From Theorem 9.3, we obtain a tester for testing distribution properties with bounded VC-dimension (Corollary 9.4).

*Proof of Theorem 9.3.* We call the algorithm ALG corresponding to Corollary 9.5 with $D$ as the input distribution, the same $\alpha$ and $\beta$ as here, and $r = \lfloor e(d+1)\left(\frac{2e}{\alpha}\right)^d \rfloor$. Note that the output of ALG is either the full description of a distribution $D'$ or FAIL. We output the same output returned by ALG. Now we prove the correctness of this procedure.

**(i)** Here $D$ is $\beta$-close to having VC-dimension $d$. Let $D_0$ be the distribution such that $D_0$ has VC-dimension at most $d$ and $d_{EM}(D, D_0) \leq \beta$. By Corollary 9.14, we can partition the support of $D_0$ into $r$ parts $\mathcal{C}_1, \dots, \mathcal{C}_r$ such that $r \leq e(d+1)\left(\frac{2e}{\alpha}\right)^d$ and the Hamming distance between any pair of vectors in the same cluster $\mathcal{C}_i$ is at most $\alpha$. This means that $D_0$ is $(\alpha, r)$-clusterable. So, with probability at least 2/3, TEST-AND-LEARN outputs a distribution $D'$ such that $d_{EM}(D, D'_\sigma) \leq 17(3\alpha + \beta/\alpha)$ for some permutation $\sigma : [n] \to [n]$, and we are done with the proof.

**(ii)** This follows from the guarantee provided by the subroutine TEST-AND-LEARN, see Corollary 8.3 (ii).

□

## Testing bounded VC-dimension properties

We now present the proof of Corollary 9.4 regarding the testing of properties with bounded VC-dimension.

*Proof of Corollary 9.4.* We call the algorithm (say ALG) corresponding to Theorem 9.3 with the input distribution $D$, $\alpha = \varepsilon/102$, and $\beta = 0$. Let $D'$ be the output of ALG. We check if there exists a distribution $D'' \in \mathcal{P}$ such that $d_{EM}(D', D'') \leq \varepsilon/2$. If yes, we accept $D$. Otherwise, we reject $D$.

Now we argue the correctness. For completeness, let us assume that $D \in \mathcal{P}$, hence $D$ has VC-dimension $d$. By the guarantee for ALG following Theorem 9.3, with probability at least $2/3$, ALG does not report FAIL, and the output distribution $D'$ by ALG satisfies $d_{EM}(D, D'_\sigma) \leq \varepsilon/2$ for some permutation $\sigma : [n] \to [n]$. Since $\mathcal{P}$ is an index-invariant property, $D'$ and $D'_\sigma$ have the same distance from the property $\mathcal{P}$. Also, as $D \in \mathcal{P}$, $D_\sigma \in \mathcal{P}$ as well. Hence, there exists a distribution $D'' \in \mathcal{P}$ (here $D_\sigma$ in particular) such that $d_{EM}(D', D'') \leq \varepsilon/2$, and we accept $D$ with probability at least $2/3$.

For soundness, consider the case where $D$ is $\varepsilon$-far from $\mathcal{P}$. If ALG reports FAIL, we are done. Otherwise, by Theorem 9.3, the output distribution $D'$ is such that for some permutation $\sigma : [n] \to [n]$, $d_{EM}(D, D'_\sigma) \leq \varepsilon/2$. Now we consider any distribution $D''$ with $d_{EM}(D', D'') \leq \varepsilon/2$ and argue that $D''$ is not in $\mathcal{P}$. By contradiction, let us assume that $D'' \in \mathcal{P}$. As $\mathcal{P}$ is index-invariant, $D''_\sigma \in \mathcal{P}$. Note that $d_{EM}(D'_\sigma, D''_\sigma) \leq \varepsilon/2$ as

$d_{EM}(D', D'') \leq \varepsilon/2$. So, $D'_\sigma$ is $\varepsilon/2$-close to property $\mathcal{P}$. As $d_{EM}(D, D'_\sigma) \leq \varepsilon/2$, by the triangle inequality, $D$ is $\varepsilon$-close to $\mathcal{P}$, a contradiction. This completes the proof of Corollary 9.4. □

## 9.3 Lower bounds for testing VC-dimension properties

As mentioned in the introduction, our tester for testing a VC-dimension property takes $\exp(d)$ samples, and performs $\exp(\exp(d))$ queries for VC-dimension $d$. Now we show that there exists an index-invariant property of VC-dimension at most $d$ which requires such sample and query complexities, proving Theorem 7.5.

**Theorem 9.15 (Restatement of Theorem 7.5).** *Let $d, n \in \mathbb{N}$. There exists an index-invariant property $\mathcal{P}_{\mathsf{vc}}$ with VC-dimension at most $d$ such that any (non-adaptive) tester for $\mathcal{P}_{\mathsf{vc}}$ requires $2^{\Omega(d)}$ samples and $2^{2^{d-\mathcal{O}(1)}}$ queries.*

Since the query complexity of non-adaptive testers can be at most quadratic as compared to adaptive ones (Theorem 7.9), arguing only for non-adaptive testers is sufficient for our purpose. We would like to point out that the property of having support size at most $2^d$ is a property with VC-dimension bounded by $d$, for which the authors of [GR22] proved a lower bound of $\Omega(2^{(1-o(1))d})$ samples [GR22, Observation 2.7]. Although the sample lower bound of the property $\mathcal{P}_{\mathsf{vc}}$ of Theorem 9.15 is weaker in comparison to that of the support size property, here we prove both sample and query lower bounds for the same property $\mathcal{P}_{\mathsf{vc}}$. Moreover, $\mathcal{P}_{\mathsf{vc}}$ is defined by being a permutation of a single distribution.

Without loss of generality, in what follows, we assume that $d$ is large enough.

**Property $\mathcal{P}_{\mathsf{vc}}$:** Let $k = 2^d$ and $\ell = 2^{2^{d-10}}$ be two integers and assume that $\ell$ divides $n$. Consider a matrix $A$ of dimension $k \times \ell$ such that the Hamming distance between any pair of column vectors of $A$ is at least $1/3$. [1] Let $D_A$ be a distribution supported over the

[1] One way to construct such a matrix is to select $2^{d-10}$ vectors from $\{0,1\}^{2^d}$ uniformly at random, and let the columns of $A$ be the set of all their linear combinations over the field $\mathbb{Z}_2$.

vectors $\mathbf{V}_1, \ldots, \mathbf{V}_k$ such that, for every $i \in [k]$, the following holds:

- $\mathbf{V}_i$ is the $n/\ell$ times "blow-up" of the $i$-th row of $A$, that is, for $j \in [\ell]$ and $j'$ with $(j-1) \cdot \frac{n}{\ell} < j' \leq j \cdot \frac{n}{\ell}$, $(\mathbf{V}_i)_{j'} = a_{ij}$, where $a_{ij}$ denotes the element of the matrix $A$ present in the $i$-th row and the $j$-th column.
- $D_A(\mathbf{V}_i) = \frac{1}{k} = \frac{1}{2^d}$.

Now we are ready to define the property $\mathcal{P}_{\mathsf{vc}}$.

$$\mathcal{P}_{\mathsf{vc}} = \{D : D = D_A^\sigma \text{ for some permutation } \sigma : [n] \to [n]\}.$$

Now we have the following observation.

**Observation 9.16.** The VC-dimension of $\mathcal{P}_{\mathsf{vc}}$ is at most $d$.

This follows from the fact that the support size of the distribution $D_A$ is $2^d$. We will prove first the query complexity lower bound, and then prove the (easier) sample complexity lower bound.

**Query complexity lower bound:** Let us define the first pair of hard distributions over distributions over $\{0,1\}^n$, that is, $D_{yes}$ and $D_{no}$.

**Distribution $D_{yes}$:** We choose a permutation $\sigma : [n] \to [n]$ uniformly at random, and pick the distribution $D_A^\sigma$ over $\{0,1\}^n$.

The distribution $D_{no}$ is constructed from the matrix $A$ that is used to define $D_{yes}$ as follows:

**Distribution $D_{no}$:** We first choose $\ell' = 2^{2^{d-20}}$ column vectors uniformly at random from $A$ and let $B$ be the resulting matrix of dimension $k \times \ell'$. Let $D_B$ be the distribution supported over the vectors $\mathbf{W}_1, \ldots, \mathbf{W}_k$ such that, for every $i \in [k]$, the following holds:

- $\mathbf{W}_i$ is the $n/\ell'$ times blow-up of the $i$-th row of $B$, that is, for $j \in [\ell']$ and $j'$ with $(j-1) \cdot \frac{n}{\ell'} < j' \leq j \cdot \frac{n}{\ell'}$, $(\mathbf{W}_i)_{j'} = b_{ij}$, where $b_{ij}$ denotes the element of matrix $B$ present in the $i$-th row and the $j$-th column.

- $D_{no}(\mathbf{W}_i) = \frac{1}{k} = \frac{1}{2^d}$.

We choose a permutation $\sigma : [n] \to [n]$ uniformly at random, and pick the distribution $D_B^\sigma$ over $\{0,1\}^n$.

**Lemma 9.17.** *$D_{yes}$ is supported over $\mathcal{P}_{\mathsf{VC}}$ and $D_{no}$ is supported over distributions that are $1/8$-far from $\mathcal{P}_{\mathsf{VC}}$.*

*Proof.* Following the definition of $\mathcal{P}_{\mathsf{VC}}$ and $D_{yes}$, it is clear that $D_{yes}$ is supported over $\mathcal{P}_{\mathsf{VC}}$. To prove the claim about $D_{no}$, consider the following definition and observation.

**Definition 9.18.** Let us consider a distribution $D$ over $\{0,1\}^n$. A matrix $M$ of dimension $s \times n$ is said to be a *corresponding* matrix of $D$ if $D$ is the distribution resulting from picking uniformly at random a row of $M$. [2] For a permutation $\pi : [s] \to [s]$, $M^\pi$ denotes the matrix obtained by permuting the rows of $M$ according to the permutation $\pi$, that is, the $\pi(i)$-th row of $M^\pi$ is same as the $i$-th row of $M$ for every $i \in [s]$.

Note that if $M$ is a corresponding matrix of $D$ with $s$ rows and $s'$ is a multiple of $s$, then the matrix $M'$ constructed by repeating every row of $M$ $s'/s$ many times is also a corresponding matrix of $D$.

Now the following observation connects the Earth Mover Distance between two distributions with the Hamming distance between their corresponding matrices.

**Claim 9.19.** *Let $D_1$ and $D_2$ be two distributions over $\{0,1\}^n$. Also, let $L$ and $M$ be corresponding matrices of $D_1$ and $D_2$, respectively, both of dimension $s \times n$. Then the Earth Mover Distance between $D_1$ and $D_2$ is the same as the minimum Hamming distance between $L$ and $M$ over all row permutations.*

[2] Note that, if $M$ has no duplicate rows, then $D$ is a uniform distribution over its support.

*Formally, let the Hamming distance between $L$ and $M$ be defined as*

$$d_H(L, M) = \frac{|\{(i, j) \in [s] \times [n] \ : \ l_{ij} \neq m_{ij}\}|}{s \cdot n}$$

*Then*

$$d_{EM}(D_1, D_2) = \min_{\pi:[s]\to[s]} d_H(L^{\pi}, M).$$

*Proof.* We first note that any solution $f_{\mathbf{XY}}$ for the EMD between $D_1$ and $D_2$ can be translated to a doubly stochastic matrix $S$ of dimension $s \times s$ as follows:

For every $i$, let $\mathbf{L}_i$ be the $i$-th row of $L$ and $l_i$ be the number of rows of $L$ that are identical to $\mathbf{L}_i$. Similarly, let $\mathbf{M}_i$ be the $i$-th row of $M$ and $m_i$ be the number of rows of $M$ that are identical $\mathbf{M}_i$. To construct the matrix $S$, we set the value of its entry at $i$-th row and $j$-th column as follows:

$$s_{ij} = \frac{f_{\mathbf{L}_i \mathbf{M}_j} \cdot s}{l_i \cdot m_j}$$

Now we claim that the matrix $S$ defined above is a doubly stochastic matrix.

**Observation 9.20.** The matrix $S$ defined above is doubly stochastic.

*Proof.* We will prove that the every row of $S$ sum to 1, and omit the identical proof for the columns of $S$. Note that if we sum the $i$-th row of $S$, we obtain the following:

$$\sum_{j=1}^{s} s_{ij} = \sum_{j=1}^{s} \frac{f_{\mathbf{L}_i \mathbf{M}_j} \cdot s}{l_i \cdot m_j} = \sum_{\mathbf{Y} \in \mathrm{Supp}(D_2)} \frac{f_{\mathbf{L}_i \mathbf{Y}} \cdot s}{l_i} = \frac{D_1(\mathbf{L}_i) \cdot s}{l_i} = 1$$

This completes the proof of the observation. □

Now we will apply the Birkhoff-Newmann theorem [Bir46, VN53], which states that the doubly stochastic matrix $S$ defined above can be expressed as a weighted average of permutation matrices. By translating the EMD expression from $f_{\mathbf{XY}}$ to $S$ and using an averaging argument, we can infer that there exists a permutation $\pi$ (among those in the representation of $S$) such that $d_H(L^{\pi}, M)$ is equal to $d_{EM}(D_1, D_2)$. This completes the proof of the claim. □

Note that $D_{no}$ is supported over the set of distributions $D_B^\sigma$ for any permutation $\sigma$ and any matrix $B$ which consists of $2^{2^{d-20}}$ columns of $A$. We will be done by showing that the Earth Mover Distance between $D$ and $D_B^\sigma$ is at least $1/8$, where $D \in \mathcal{P}_{\mathsf{vc}}$, $\sigma : [n] \to [n]$ is any permutation, and $B$ is any matrix with $2^{2^{d-20}}$ columns.

Note that both $D$ and $D_B^\sigma$ admit respective corresponding matrices $L$ and $M$, respectively, both of dimension $2^d \times n$, where the rows of L are the vectors $\mathbf{V}_i$, and the rows of M are the respective permutations of the vectors $\mathbf{W}_i$. By Claim 9.19, we note that:

$$d_{EM}(D_B^\sigma, D) = \min_{\pi:[2^d]\to[2^d]} d_H(L^\pi, M).$$

The following claim will imply that $d_{EM}(D_B^\sigma, D) \geq 1/8$.

**Claim 9.21.** *For any permutation $\pi : [2^d] \to [2^d]$, $d_H(L^\pi, M)$ is at least $1/8$.*

*Proof.* Let us partition the index set $[n]$ into $\ell'$ equivalence classes $C_1, \ldots, C_{\ell'}$ such that two indices of $[n]$ belong to the same equivalence class if the corresponding column vectors in $L^\pi$ are identical. Observe that

$$d_H(L^\pi, M) = \frac{\sum\limits_{i\in[\ell']}\sum\limits_{j\in C_i} d_H(\mathbf{L}_j^\pi, \mathbf{M}_j)\cdot k}{k\cdot n} = \frac{\sum\limits_{i\in[\ell']}\sum\limits_{j\in C_i} d_H(\mathbf{L}_j^\pi, \mathbf{M}_j)}{n},$$

where $\mathbf{L}_j^\pi$ and $\mathbf{M}_j$ denote the $j$-th column vectors of $L^\pi$ and $M$, respectively.

Hence we will be done by showing $\sum\limits_{j\in C_i} d_H\left(\mathbf{L}_j^\pi, \mathbf{M}_j\right) \geq \frac{n}{8\ell'}$, for every $i \in [\ell']$.

Note that $|C_i| = \frac{n}{\ell'}$. Also, all the columns in $\{\mathbf{L}_j^\pi : j \in C_i\}$ are identical. Consider a column vector $\mathbf{v} \in \{0,1\}^k$. Observe that there can be at most $\frac{n}{\ell}$ columns in $\{\mathbf{M}_j : j \in C_i\}$ that are $1/7$-close to $\mathbf{v}$. This follows from the construction of $\mathcal{P}_{\mathsf{vc}}$, which implies that for every column $\mathbf{M}_j$ of $M$, there are no more than $n/\ell - 1$ other columns of $L^\pi$ whose distance from $\mathbf{M}_j$ is at most $2/7 < 1/3$.

So, in the expression $\sum\limits_{j\in C_i} d_H\left(\mathbf{L}_j^\pi, \mathbf{M}_j\right)$, there are at least $\left(\frac{n}{\ell'} - \frac{n}{\ell}\right)$ terms that are at least $1/7$. Hence, $\sum\limits_{j\in C_i} d_H\left(\mathbf{L}_j^\pi, \mathbf{M}_j\right) \geq \ell' \cdot \frac{1}{7}\left(\frac{n}{\ell'} - \frac{n}{\ell}\right) \geq \frac{n}{8\ell'}$. □

The above two claims conclude the proof of Lemma 9.17. □

**Lemma 9.22 (Query complexity lower bound part of Theorem 9.15).** *Any (non-adaptive) tester, that has sample and query access to either $D_{yes}$ or $D_{no}$ and performs $2^{2^{d-\omega(1)}}$ queries, can not distinguish between $D_{yes}$ and $D_{no}$.*

*Proof.* Let $A'$ and $B'$ be the matrices of dimension $k \times n$ such that the $i$-th row of $A'$ corresponds to the vector $\mathbf{V}_i^\sigma$ (for the permutation $\sigma$ drawn according to $D_{yes}$) and the $i$-th row of $B'$ corresponds to the vector $\mathbf{W}_i^\sigma$ (for the permutation $\sigma$ drawn according to $D_{no}$), where $i \in [k]$.

Let us divide the index set $[n]$ into $\ell$ equivalence classes $C_1, \ldots, C_\ell$ such that two indices belong to the same equivalence class if the corresponding column vectors in $A'$ are identical. Similarly, let us divide the index set $[n]$ into $\ell'$ equivalence classes $C'_1, \ldots, C'_{\ell'}$ such that two indices belong to the same equivalence class if the corresponding column vectors in $B'$ are identical.

Let $Q \subseteq [n]$ be the set of all distinct indices queried by the tester to any sample (that is, the union of the sets $J_1, \ldots, J_s$ as they appear in Definition 2.10). If $|Q| = 2^{2^{d-\omega(1)}}$, then the probability that there exist two indices in $Q$ that belong to the same $C_i$ or the same $C'_i$ is $o(1)$. Observe that, conditioned on the event that $Q$ does not contain two indices from the same equivalence class $C_i$ or $C'_i$, the distributions over the responses to the queries of the tester are identical for both $D_{yes}$ and $D_{no}$. The reason is that in both the cases of $D_{yes}$ and $D_{no}$, the distribution over the responses is identical to the one derived from picking a uniformly random subset of size $|Q|$ of the columns of the matrix $A$, and taking uniformly independent samples of the rows of the resulting matrix. □

Now we will prove the sample complexity lower bound for testing $\mathcal{P}_{\mathsf{vc}}$.

**Sample complexity lower bound:** Let us define the second pair of hard distributions over distributions over $\{0,1\}^n$, $D'_{yes}$ and $D'_{no}$.

**Distribution $D'_{yes}$:** Identically to $D_{yes}$ above, we choose a permutation $\sigma : [n] \rightarrow [n]$ uniformly at random, and pick the distribution $D_A^\sigma$ over $\{0,1\}^n$.

The distribution $D'_{no}$ is constructed from the matrix $A$ used to define $D'_{yes}$ as follows:

**Distribution $D'_{no}$:** We first choose $k' = 2^{d-20}$ row vectors uniformly at random from $A$ and construct a matrix $B'$ of dimension $k' \times \ell$. Let $D_{B'}$ be the distribution supported over the vectors $\mathbf{W}'_1, \ldots, \mathbf{W}'_{k'}$ such that, for every $i \in [k']$, the following hold:

- $\mathbf{W}'_i$ is the $n/\ell$ times blow-up of the $i$-th row of $B'$, that is, for $j \in [\ell]$ and $j'$ with $(j-1) \cdot \frac{n}{\ell} < j' \leq j \cdot \frac{n}{\ell}$, $(\mathbf{W}'_i)_{j'} = b_{ij}$, where $b_{ij}$ denotes the element of matrix $B'$ present in the $i$-th row and the $j$-th column.
- $D_{no}(\mathbf{W}'_i) = \frac{1}{k'} = \frac{1}{2^{d-20}}$.

We choose a permutation $\sigma : [n] \rightarrow [n]$ uniformly at random, and pick the distribution $D^{\sigma}_{B'}$ over $\{0,1\}^n$.

**Lemma 9.23.** *$D'_{yes}$ is supported over $\mathcal{P}_{\mathsf{vc}}$ and $D'_{no}$ is supported over distributions that are $1/8$-far from $\mathcal{P}_{\mathsf{vc}}$.*

*Proof.* Following the definition of $\mathcal{P}_{\mathsf{vc}}$ and $D'_{yes}$, it is clear that $D'_{yes}$ is supported over $\mathcal{P}_{\mathsf{vc}}$. To prove the claim about $D'_{no}$, we will apply Claim 9.19.

Note that $D'_{no}$ is supported over the set of distributions $D^{\sigma}_{B'}$ for any permutation $\sigma$ and any matrix $B'$ which consists of $2^{d-20}$ rows of $A$. We will be done by showing the Earth Mover Distance between $D$ and $D^{\sigma}_{B'}$ is at least $1/8$, where $D \in \mathcal{P}_{\mathsf{vc}}$ and $\sigma : [n] \rightarrow [n]$ be any permutation, and $B'$ is any matrix with $2^{d-20}$ distinct rows.

Let $L$ and $M$ be corresponding matrices of $D$ and $D^{\sigma}_{B'}$, respectively, of dimension $k \times n$, where $k = 2^d$ (where the rows of $L$ are the vectors $\mathbf{V}_i$, and the rows of $M$ are $2^{20}$-fold repetitions of the respective permutations of the vectors $\mathbf{W'}_i$). By Claim 9.19, we know that

$$d_{EM}\left(D^{\sigma}_{B'}, D\right) = \min_{\pi:[k]\rightarrow[k]} d_H(L^{\pi}, M).$$

Thus, the following claim will imply that $d_{EM}(D^{\sigma}_{B'}, D) \geq 1/8$.

**Claim 9.24.** *For any permutation $\pi : [2^d] \rightarrow [2^d]$, $d_H(L^{\pi}, M)$ is at least $1/8$.*

*Proof.* Our proof will follow a similar vain to that of Claim 9.21. Let us first partition the index set $[n]$ into $\ell'$ equivalence classes $C_1, \ldots, C_{\ell'}$ such that two indices of $[n]$ belong to the same equivalence class if the corresponding column vectors in $L^\pi$ are identical. Observe that

$$d_H(L^\pi, M) = \frac{\sum\limits_{i\in[\ell']}\sum\limits_{j\in C_i} d_H(\mathbf{L}_j^\pi, \mathbf{M}_j)\cdot k}{k\cdot n} = \frac{\sum\limits_{i\in[\ell']}\sum\limits_{j\in C_i} d_H(\mathbf{L}_j^\pi, \mathbf{M}_j)}{n},$$

where $\mathbf{L}_j^\pi$ and $\mathbf{M}_j$ denote the $j$-th column vectors of $L^\pi$ and $M$, respectively.

Since $B'$ has only $2^{d-20}$ distinct rows, the number of its equivalence classes is bounded by $\ell' = 2^{2^{d-20}}$. Note that unlike the proof of the query lower bound, the sizes of the equivalence classes here may be different from each other. Also, note that the sizes of the equivalence classes of $L$ are $n/\ell$, as $D \in \mathcal{P}_{\mathsf{vc}}$. Thus we have the following:

$$d_H(L^\pi, M) \geq \frac{1}{7}\cdot\frac{\sum_{i=1}^{\ell'}\max\{0, |C_i| - n/\ell\}}{n} \geq \frac{1}{7}\cdot\left(1 - \frac{1}{2^{10}}\right)\cdot n > \frac{1}{8}n.$$

The inequality follows from the facts that $\ell = 2^{2^{d-10}}$ and $\ell' = 2^{2^{d-20}}$, and the columns of $M$ corresponding to each $C_i$ can be $1/7$-close to at most $n/\ell$ columns of $L$. □

This concludes the proof of Lemma 9.23. □

The sample lower bound for testing $\mathcal{P}_{\mathsf{vc}}$ now follows from the following lemma.

**Lemma 9.25 (Sample complexity lower bound part of Theorem 9.15).** *Any tester that takes at most $2^{o(d)}$ samples from the input distribution can not distinguish between the distributions $D'_{yes}$ and $D'_{no}$.*

*Proof.* Let $\mathcal{S}$ be the set of samples taken by the algorithm. Note that if $|\mathcal{S}| = 2^{o(d)}$, then the probability that $\mathcal{S}$ contains two samples of the same $\mathbf{V}_i$ or the same $\mathbf{W'}_i$ is $o(1)$. Conditioned on the event that $\mathcal{S}$ does not contain two samples from the same vector ($\mathbf{V}_i$ or $\mathbf{W'}_i$), even if the tester queries the samples of $\mathcal{S}$ in their entirety, the distributions over the responses to the queries of the tester are identical for both $D'_{yes}$ and $D'_{no}$. This follows from the fact that the distribution over the responses is identical to a distribution obtained by drawing uniformly without repetitions a sequence of row vectors from $\mathbf{V}_1, \ldots, \mathbf{V}_{2^d}$, and querying the row vectors completely. This completes the proof. $\square$

# Chapter 10

# Role of adaptivity for general properties

## 10.1 Introduction

In this chapter, we prove that for non-index-invariant properties, there can be an exponential gap between the query complexities of adaptive and non-adaptive tester. The result is formally stated as follows:

**Theorem 10.1 (Theorem 1.10 formalized).** *Any property $\mathcal{P}$ that is $\varepsilon$-testable by an adaptive algorithm using $s$ samples and $q$ queries, can be $\varepsilon$-tested by a non-adaptive algorithm that uses $s$ samples and performs at most $2^q - 1$ queries, where $s$ and $q$ are integers.*

We prove this theorem in Section 10.2. The proof follows a simulation-type argument. Later in Section 10.3, we prove the following theorem which states that the exponential gap mentioned in the above theorem is tight.

**Theorem 10.2 (Theorem 1.11 formalized).** *There exists a property of distributions over strings $1_{\mathcal{P}_{Pal}}$ that can be $\varepsilon$-tested adaptively using $\mathcal{O}(\log n)$ queries for any $\varepsilon \in (0, 1)$, but $\Omega(\sqrt{n})$ queries are necessary for any non-adaptive algorithm to $\varepsilon$-test it for some $\varepsilon \in (0, 1)$.*

## 10.2 Exponential gap between adaptive and non-adaptive testers

In this section, we prove that there can be at most an exponential gap between the query complexities of adaptive and non-adaptive algorithms for non-index-invariant properties.

Let $\mathcal{A}$ be the adaptive algorithm that $\varepsilon$-tests $\mathcal{P}$ using $s$ samples $\{\mathbf{V}_1, \ldots, \mathbf{V}_s\}$ and $q$ queries, along with tossing some random coins. Before directly proceeding to the description of the non-adaptive algorithm, let us first consider the following observation.

**Observation 10.3.** For any given outcome sequence of the random coin tosses of $\mathcal{A}$, there are at most $2^q - 1$ possible internal states of $\mathcal{A}$.

*Proof.* Consider the $k$-th step of $\mathcal{A}$, where $\mathcal{A}$ queries the $j_k$-th index of $\mathbf{V}_{i_k}$ for some $i_k \in [s]$, $j_k \in [n]$, and $k \in [q]$. Note that $i_1$ and $j_1$ are functions of only the random coins, and $i_k$ and $j_k$ are functions of the random coins, as well as $\mathbf{V}_{i_1}\mid_{j_1}, \ldots, \mathbf{V}_{i_{k-1}}\mid_{j_{k-1}}$, where $2 \leq k \leq q$. Due to the $2^{k-1}$ possible values of $\mathbf{V}_{i_1}\mid_{j_1}, \ldots, \mathbf{V}_{i_{k-1}}\mid_{j_{k-1}}$, there are $2^k$ possible states of the algorithm $\mathcal{A}$ at Step $k$, for each $1 \leq k \leq q$. Finally, the state of $\mathcal{A}$ depending on the random coins and the values of $\mathbf{V}_{i_1}\mid_{j_1}, \ldots, \mathbf{V}_{i_q}\mid_{j_q}$ will decide the final output. This implies that for any fixed set of outcomes of the random coin tosses used by $\mathcal{A}$, there can be a total of at most $\sum\limits_{i=0}^{q-1} 2^i = 2^q - 1$ internal states, each making one query, as well as $2^q$ final (non-query-making) states. □

Now we proceed to present the non-adaptive algorithm $\mathcal{A}'$ that simulates $\mathcal{A}$ by using $s$ samples and at most $2^q$ queries.

**Theorem 10.4 (Theorem 10.1 restated).** *Let $\mathcal{P}$ be any property that is $\varepsilon$-testable by an adaptive algorithm using $s$ samples and $q$ queries. Then $\mathcal{P}$ can be $\varepsilon$-tested by a non-adaptive algorithm using $s$ samples and at most $2^q - 1$ queries, where $s$ and $q$ are integers.*

*Proof.* Let $\mathcal{A}$ be the adaptive algorithm that $\varepsilon$-tests $\mathcal{P}$ using $s$ samples $\{\mathbf{V}_1, \ldots, \mathbf{V}_s\}$ and $q$ queries. Now we show that a non-adaptive algorithm $\mathcal{A}'$ exists that uses $s$ samples and makes at most $2^q - 1$ queries, such that the output distributions of $\mathcal{A}$ and $\mathcal{A}'$ are identical for any unknown distribution $D$.

The idea of $\mathcal{A}'$ in a high level is to enumerate all possible internal steps of $\mathcal{A}$, and list all possible queries $\mathcal{Q}$ that might be performed by $\mathcal{A}$. Note that $\mathcal{Q}$ depends on the random coins used by $\mathcal{A}$. We then query all the indices of $\mathcal{Q}$ non-adaptively, and finally simulate $\mathcal{A}$ using the full information at hand, with the same random coins that were used to generate $\mathcal{Q}$. As $\mathcal{A}$ has query complexity $q$, the number of possible internal states of $\mathcal{A}$ is at most $2^q - 1$, and the query complexity of $\mathcal{A}'$ follows. Now we formalize the above intuition below.

The algorithm $\mathcal{A}'$ has two phases:

**Phase 1:**

**(i)** $\mathcal{A}'$ first takes $s$ samples $\mathbf{V}_1, \ldots, \mathbf{V}_s$.

**(ii)** $\mathcal{A}'$ now tosses some random coins (same as $\mathcal{A}$) and determines the set of all possible indices $J_i$ of $\mathbf{V}_i$ that might be queried by $\mathcal{A}$, for every $i \in [s]$. The sets of indices $J_i$'s are well defined after we fix the random coins, and follows from Observation 10.3.

Thus at the end of Phase 1, $\mathcal{A}'$ has determined $s$ sets of indices $J_1, \ldots, J_s$ of the vectors $\mathbf{V}_1, \ldots, \mathbf{V}_s$ such that $\sum\limits_{i=1}^{s} |J_i| \leq 2^q - 1$. Now $\mathcal{A}'$ proceeds to the second phase of the algorithm.

**Phase 2:**

**(i)** For every $i \in [s]$ and $j \in J_i$, query the $j$-th index of $\mathbf{V}_i$, where $J_i$ denotes the set of indices of $\mathbf{V}_i$ that might be queried at the internal states of $\mathcal{A}$, determined in Phase 1.

**(ii)** Simulate the algorithm $\mathcal{A}$ using the same random coins used in Phase 1, and report ACCEPT or REJECT according to the output of $\mathcal{A}$.

Note that the set of random coins that are used to determine $J_1, \dots, J_s$ in Step $(ii)$ of Phase 1 of the algorithm are the same random coins that are used to simulate $\mathcal{A}$ in Step $(ii)$ of Phase 2. Thus the correctness of $\mathcal{A}'$ follows from to the correctness of $\mathcal{A}$ along with Observation 10.3. □

## 10.3 Exponential separation between adaptive and non-adaptive testers

Now we prove that the gap of Theorem 10.4 is almost tight, in the sense that there exists a property such that the adaptive and non-adaptive query complexities for testing it are exponentially separated.

Before proceeding to the proof, let us consider any property $\mathcal{P}$ of strings of length $n$ over the alphabet $\{0, 1\}$. Now we describe a related property $1_{\mathcal{P}}$ over distributions as follows:

**Property $1_{\mathcal{P}}$:** For any distribution $D \in 1_{\mathcal{P}}$, the size of the support of $D$ is 1, and the single string in the support of $D$ satisfies $\mathcal{P}$.

Let us first recall the following result from [GR22], which states that $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$ queries are enough to $\varepsilon$-test whether any distribution has support size 1.

**Lemma 10.5 (Restatement of Corollary 2.3.1 of [GR22]).** *There exists a non-adaptive algorithm that $\varepsilon$-tests whether an unknown distribution $D$ has support size 1 and uses $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$ queries, for any $\varepsilon \in (0, 1)$.*

We now prove that the query complexity of $\varepsilon$-testing $1_{\mathcal{P}}$ is at least the query complexity of $\varepsilon$-testing $\mathcal{P}$, and can be at most the query complexity of $\frac{\varepsilon}{2}$-testing of $\mathcal{P}$, along with an additional additive factor of $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$ for testing whether the distribution has support size 1. The result is formally stated as follows:

**Lemma 10.6.** *Let $q_N$ and $q_A$ denote the non-adaptive and adaptive query complexities for $\varepsilon$-testing $\mathcal{P}$, respectively. Similarly, let $Q_N$ and $Q_A$ denote the non-adaptive and adaptive query complexities of $\varepsilon$-testing $1_{\mathcal{P}}$, respectively. Then the following hold:*

1. $q_A(\varepsilon) \leq Q_A(\varepsilon) \leq \widetilde{\mathcal{O}}(\frac{1}{\varepsilon}) + \mathcal{O}\left(q_A(\frac{\varepsilon}{2})\right)$ [1].

2. $q_N(\varepsilon) \leq Q_N(\varepsilon) \leq \widetilde{\mathcal{O}}(\frac{1}{\varepsilon}) + \mathcal{O}\left(q_N(\frac{\varepsilon}{2})\right)$.

*Proof.* We prove here (1), and omit the nearly identical proof of (2).

**Proof of $q_A(\varepsilon) \leq Q_A(\varepsilon)$:** Consider an adaptive algorithm $\mathcal{A}$ that $\varepsilon$-tests $1_{\mathcal{P}}$ by using $Q_A(\varepsilon)$ queries. We construct an algorithm $\mathcal{A}'$ that $\varepsilon$-tests $\mathcal{P}$ using the same number of queries. Let $\mathbf{V}$ be the unknown string of length $n$, where we want to test whether $\mathbf{V} \in \mathcal{P}$ or $\mathbf{V}$ is $\varepsilon$-far from $\mathcal{P}$.

Let us define an unknown distribution $D'$ (over the Hamming cube $\{0,1\}^n$) such that we want to distinguish whether $D' \in 1_{\mathcal{P}}$ or $D'$ is $\varepsilon$-far from $1_{\mathcal{P}}$. The distribution $D'$ is defined as follows:

$$D'(\mathbf{X}) = \begin{cases} 1 & \mathbf{X} = \mathbf{V} \\ 0 & \text{otherwise} \end{cases}$$

Observe that $\mathbf{V} \in \mathcal{P}$ if and only if $D' \in 1_{\mathcal{P}}$. Similarly, it is not hard to see that $\mathbf{V}$ is $\varepsilon$-far from $\mathcal{P}$ if and only if $D'$ is $\varepsilon$-far from $1_{\mathcal{P}}$. We simulate the algorithm $\mathcal{A}$ by $\mathcal{A}'$ as follows: when $\mathcal{A}$ takes a sample, $\mathcal{A}'$ does nothing, and when $\mathcal{A}$ queries an index $i \in [n]$ of any sample, $\mathcal{A}'$ queries the index $i$ of $\mathbf{V}$. Finally, $\mathcal{A}'$ provides the output received from the simulation of $\mathcal{A}$.

From the description, it is clear that $\mathcal{A}'$ performs exactly $Q_{\mathcal{A}}(\varepsilon)$ queries and is indeed simulated by running $\mathcal{A}$ over $D'$.

**Proof of $Q_A(\varepsilon) \leq \widetilde{\mathcal{O}}\left(\frac{1}{\varepsilon}\right) + \mathcal{O}\left(q_A(\frac{\varepsilon}{2})\right)$:** Let us consider an adaptive algorithm $\mathcal{A}_1$ that $\frac{\varepsilon}{2}$-tests $\mathcal{P}$ using $\mathcal{O}\left(q_A(\frac{\varepsilon}{2})\right)$ queries to the unknown string $\mathbf{X} \in \{0,1\}^n$, with success

[1] We are using $\mathcal{O}(\cdot)$ as we are amplifying the success probability of the tester for the property $\mathcal{P}$ to $9/10$ as compared to the usual success probability of $2/3$.

probability at least $\frac{9}{10}$. Now we design an adaptive algorithm $\mathcal{A}_1'$ that $\varepsilon$-tests $1_{\mathcal{P}}$ using $\widetilde{\mathcal{O}}\left(\frac{1}{\varepsilon}\right) + \mathcal{O}\left(q_A(\frac{\varepsilon}{2})\right)$ queries.

**Algorithm $\mathcal{A}_1'$:** Assume that $D$ is the distribution that we want to $\varepsilon$-test for $1_{\mathcal{P}}$. The algorithm $\mathcal{A}_1'$ performs the following steps:

**(i)** Run the tester corresponding to Lemma 10.5 to $\frac{\varepsilon}{20}$-test whether $D$ has support size 1, with success probability at least $\frac{9}{10}$. If the tester decides that $D$ has support size 1, then go to the next step. Otherwise, REJECT.

**(ii)** Take one more sample from $D$ and let it be $\mathbf{U} \in \{0,1\}^n$. Run algorithm $\mathcal{A}_1$ to $\frac{\varepsilon}{2}$-test $\mathcal{P}$ considering $\mathbf{X} = \mathbf{U}$ as the unknown string. If $\mathcal{A}_1$ accepts, ACCEPT. Otherwise REJECT.

Note that the query complexity for performing Step $(i)$ is $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$, which follows from Lemma 10.5. Additionally, the number of queries performed in Step $(ii)$ is $\mathcal{O}\left(q_A(\frac{\varepsilon}{2})\right)$, which follows from the assertion of the lemma. Thus, the algorithm $\mathcal{A}_1'$ performs $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon}) + \mathcal{O}\left(q_A(\frac{\varepsilon}{2})\right)$ queries in total.

Now we will argue the correctness of $\mathcal{A}_1'$. For completeness, assume that $D \in 1_{\mathcal{P}}$. Let $\mathbf{V} \in \{0,1\}^n$ be the string such that $D(\mathbf{V}) = 1$ and $\mathbf{V} \in \mathcal{P}$. Note that, by Lemma 10.5, $\mathcal{A}_1'$ proceeds to Step $(ii)$ with probability at least $\frac{9}{10}$. In Step $(ii)$, $\mathcal{A}'$ sets $\mathbf{U} = \mathbf{V}$, and runs algorithm $\mathcal{A}_1$ to $\frac{\varepsilon}{2}$-test $\mathcal{P}$ considering $\mathbf{X} = \mathbf{V}$ as the unknown string. Since $\mathbf{V} \in \mathcal{P}$, by the assumption on the algorithm $\mathcal{A}_1$, $\mathcal{A}_1'$ accepts with probability at least $\frac{9}{10}$, given that $\mathcal{A}_1'$ does not report REJECT in Step $(i)$. Thus, by the union bound, $\mathcal{A}_1'$ accepts $D$ with probability at least $\frac{4}{5}$.

Now consider the case where $D$ is $\varepsilon$-far from $1_{\mathcal{P}}$. If $D$ is $\frac{\varepsilon}{20}$-far from having support size 1, $\mathcal{A}_1'$ reports REJECT in Step $(i)$ with probability at least $\frac{9}{10}$, and we are done. So, assume that $D$ is $\frac{\varepsilon}{20}$-close to having support size 1. Then there exists a distribution $D'$ with support size 1, and the distance between $D$ and $D'$ is at most $\frac{\varepsilon}{20}$. Let us assume that $D'$ is supported on the string $\mathbf{V}$. By the Markov inequality, this implies that with probability at least $\frac{4}{5}$, a string $\mathbf{U}$ sampled according $D$ will be $\frac{9\varepsilon}{20}$-close to $\mathbf{V}$.

**(i)** If $\mathbf{V}$ is $\frac{19\varepsilon}{20}$-close to $\mathcal{P}$, using the triangle inequality, this implies that $D$ is $\varepsilon$-close to $1_{\mathcal{P}}$, which is a contradiction.

**(ii)** Now consider the case where $\mathbf{V}$ is $\frac{19\varepsilon}{20}$-far from $\mathcal{P}$. Recall that with probability at least $\frac{4}{5}$, the sample $\mathbf{U}$ taken at Step (ii) above is $\frac{9\varepsilon}{20}$-close to $\mathbf{V}$. As we are considering the case where $\mathbf{V}$ is $\frac{19\varepsilon}{20}$-far from $\mathcal{P}$, using the triangle inequality, $\mathbf{U}$ is $\frac{\varepsilon}{2}$-far from $\mathcal{P}$ with the same probability. In this case, the algorithm will REJECT in Step $(ii)$, with probability at least $\frac{9}{10}$. Together, this implies that the algorithm will REJECT the distribution $D$, with probability at least $\frac{7}{10}$.

□

In the following, we will construct the property $\mathcal{P}_{Pal}$ of strings over the alphabet $\{0, 1, 2, 3\}$. It will then be encoded as a property of strings over $\{0, 1\}$ by using two bits per letter.

**Property $\mathcal{P}_{Pal}$:** A string $\mathbf{S}$ of length $n$ is in $\mathcal{P}_{Pal}$ if $\mathbf{S} = \mathbf{XY}$, where $\mathbf{X}$ is a palindrome over the alphabet $\{0, 1\}$, and $\mathbf{Y}$ is a palindrome over the alphabet $\{2, 3\}$.

There is an exponential gap between the query complexities of adaptive and non-adaptive algorithms to $\varepsilon$-test $\mathcal{P}_{Pal}$. The result is stated as follows:

**Lemma 10.7.** *There exists an adaptive algorithm that $\varepsilon$-tests $\mathcal{P}_{Pal}$ by making $\mathcal{O}(\log n)$ queries for any $\varepsilon \in (0, 1)$. However, there exists an $\varepsilon \in (0, 1)$ such that $\Omega(\sqrt{n})$ non-adaptive queries are necessary to $\varepsilon$-test $\mathcal{P}_{Pal}$.*

*Proof.* The lower bound proof (using Yao's lemma), which we omit here, is nearly identical to the one from [AKNS99] (see Theorem 2 therein).

Let us assume that $\mathbf{V}$ is the string that we want to $\varepsilon$-test for $\mathcal{P}_{Pal}$. The adaptive algorithm to $\varepsilon$-test $\mathcal{P}_{Pal}$ uses binary search, and is described below:

**(i)** Use binary search for an index of $\mathbf{V}$ that has "value 1.5" (which is not present in the input). This returns an index $0 \leq i \leq n$, such that (a) $\mathbf{V}_i \in \{0, 1\}$ unless $i = 0$, and (b) $\mathbf{V}_{i+1} \in \{2, 3\}$ unless $i = n$.

**(ii)** Repeat $\mathcal{O}(\frac{1}{\varepsilon})$ times:

- **(a)** Sample an index $j \in [n]$ uniformly at random.
- **(b)** If $j \leq i$, then query $\mathbf{V}_j$ and $\mathbf{V}_{i+1-j}$. REJECT if they are not both equal to the same value in $\{0, 1\}$.
- **(c)** Otherwise query $\mathbf{V}_j$ and $\mathbf{V}_{n+i+1-j}$. REJECT if they are not both equal to the same value in $\{2, 3\}$.

**(iii)** If the input has not been rejected till now, ACCEPT.

We first argue the completeness of the algorithm. Assume that $\mathbf{V}$ is a string such that $\mathbf{V} \in \mathcal{P}_{Pal}$, and $i$ is the index returned by Step $(i)$ of the algorithm. As $\mathbf{V} = \mathbf{XY}$ for some palindrome $\mathbf{X}$ over $\{0, 1\}$ and palindrome $\mathbf{Y}$ over $\{2, 3\}$, the index $i$ will be equal to $|\mathbf{X}|$. This implies that the algorithm will ACCEPT $\mathbf{V}$ with probability $1$.

Now consider the case where $\mathbf{V}$ is $\varepsilon$-far from $\mathcal{P}_{Pal}$. We call an index $j$ *violating* if it does not satisfy the condition appearing either in Step (ii)(b) or Step (ii)(c) above, where $i$ is the index returned in Step $(i)$. The number of violating indices is at least $\varepsilon n$, because otherwise we can change the violating indices such that the modified input is a string of the form $\mathbf{XY}$ following the definition of $\mathcal{P}_{Pal}$, where $|\mathbf{X}| = i$. Since the loop in Step (ii) runs for $\mathcal{O}(\frac{1}{\varepsilon})$ times, we conclude that with probability at least $\frac{2}{3}$ at least one such violating index will be found. So, the algorithm will REJECT $\mathbf{V}$ with probability at least $\frac{2}{3}$. □

Now we are ready to formally state and prove the main result of this section.

**Theorem 10.8 (Theorem 10.2 restated).** *There exists a property of distributions over strings that can be $\varepsilon$-tested adaptively using $\mathcal{O}(\log n)$ queries for any $\varepsilon \in (0, 1)$, but $\Omega(\sqrt{n})$ queries are necessary for any non-adaptive algorithm to $\varepsilon$-test it for some $\varepsilon \in (0, 1)$.*

*Proof.* Consider property $1_{\mathcal{P}_{Pal}}$. From Lemma 10.7, we know that $q_A(\frac{\varepsilon}{2}) = \mathcal{O}(\log n)$, for any fixed $\varepsilon \in (0, 1)$. Using the upper bound of Lemma 10.6, we conclude that $Q_A(\varepsilon) = \mathcal{O}(\log n)$, for any fixed $\varepsilon \in (0, 1)$, ignoring the additive $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$ term.

On the other hand, according to Lemma 10.7, $q_N(\varepsilon) = \Omega(\sqrt{n})$ for some $\varepsilon \in (0,1)$. Thus, following Lemma 10.6, we conclude that $Q_N(\varepsilon) = \Omega(\sqrt{n})$ holds for some $\varepsilon \in (0,1)$. Together, Theorem 10.8 follows. □

Now we present a sketch of a proof of Proposition 7.6, which shows that for a property to be constantly testable, it is not sufficient that the property has constant VC-dimension, unless it is index-invariant as well.

**Proposition 10.9 (Restatement of Proposition 7.6).** *There exists a non-index-invariant property $\mathcal{P}$ such that any distribution $D \in \mathcal{P}$ has VC-dimension $O(1)$ and the following holds. There exists a fixed $\varepsilon > 0$, such that distinguishing whether $D \in \mathcal{P}$ or $D$ is $\varepsilon$-far from $\mathcal{P}$, requires $\Omega(n)$ queries, where the distributions in the property $\mathcal{P}$ are defined over the $n$-dimensional Hamming cube $\{0,1\}^n$.*

*Proof.* Note that the VC-dimension of $1_{\mathcal{P}}$ is $0$, where $1_{\mathcal{P}}$ is the property corresponding to $\mathcal{P}$ as defined before. String properties which are hard to test, for which there is a fixed $\varepsilon > 0$ such that $\varepsilon$-testing them requires $\Omega(n)$ queries, are known to exist. Examples are properties studied in the work of Ben-Eliezer, Fischer, Levi and Rothblum [BFLR20], and in the work of Ben-Sasson, Harsha and Raskhodnikova [BHR05]. Defining $1_{\mathcal{P}}$ for such a property $\mathcal{P}$ provides us the example proving Proposition 10.9. □

# Chapter 11

# Power of adaptivity for index-invariant properties

## 11.1 Introduction

In this chapter, we prove that, unlike the case of non-index-invariant properties, for index-invariant properties, the gap between the query complexities of adaptive and non-adaptive testers can be at most quadratic, as stated in the following theorem.

**Theorem 11.1 (Theorem 1.12 formalized).** *Let $\mathcal{P}$ be any index-invariant property that is $\varepsilon$-testable by an adaptive algorithm using $s$ samples and $q$ queries. Then $\mathcal{P}$ can be $\varepsilon$-tested by a non-adaptive algorithm using $s$ samples and $sq \leq q^2$ queries, where $s$ and $q$ are integers.*

We will prove this theorem in Section 11.2. Later in Section 11.3, we also prove that the above gap is almost tight, in the sense that there exists an index-invariant property which can be $\varepsilon$-tested using $\widetilde{\mathcal{O}}(n)$ adaptive queries, while $\widetilde{\Omega}(n^2)$ non-adaptive queries are required to $\varepsilon$-test it.

**Theorem 11.2 (Theorem 1.13 formalized).** *There exists an index-invariant property $\mathcal{P}_{\mathrm{Gap}}$ that can be $\varepsilon$-tested adaptively using $\widetilde{\mathcal{O}}(n)$ queries for any $\varepsilon \in (0,1)$, while there exists an $\varepsilon \in (0,1)$ for which $\widetilde{\Omega}(n^2)$ queries are necessary for any non-adaptive $\varepsilon$-tester.*

## 11.2 Quadratic relation of adaptive and non-adaptive testers

In this section, we prove Theorem 11.1, that is, there can be at most a quadratic gap between the query complexities of adaptive and non-adaptive algorithms for testing index-invariant properties.

**Theorem 11.3** (**Restatement of Theorem 11.1**). *Let $\mathcal{P}$ be any index-invariant property that is $\varepsilon$-testable by an adaptive algorithm using $s$ samples and $q$ queries. Then $\mathcal{P}$ can be $\varepsilon$-tested by a non-adaptive algorithm using $s$ samples and $sq \leq q^2$ queries, where $s$ and $q$ are integers.*

*Proof.* The main idea of the proof is to start with an adaptive algorithm $\mathcal{A}$ as stated above, and then argue for another semi-adaptive algorithm $\mathcal{A}'$ with sample complexity $s$ but query complexity $qs$, such that the output distributions of $\mathcal{A}$ and $\mathcal{A}'$ are the same for any unknown distribution $D$. Finally, we construct a non-adaptive algorithm $\mathcal{A}''$ such that (i) the sample and query complexities of $\mathcal{A}''$ are the same as that of $\mathcal{A}'$, and (ii) the probability bounds of accepting and rejecting distributions depending on their distances to $\mathcal{P}$ are preserved from $\mathcal{A}'$ to $\mathcal{A}''$. Now we proceed to formalize this argument.

Let $\mathcal{A}$ be the adaptive algorithm that $\varepsilon$-tests $\mathcal{P}$ using $s$ samples $\{\mathbf{V}_1, \ldots, \mathbf{V}_s\}$ and $q$ queries. Now we show that a two phase algorithm $\mathcal{A}'$ exists that takes $s$ samples $\{\mathbf{V}_1, \ldots, \mathbf{V}_s\}$ and proceeds as follows:

**Phase 1:** In this phase, $\mathcal{A}'$ queries in an adaptive fashion. If $\mathcal{A}$ queries the $j_k$-th index of $\mathbf{V}_{i_k}$ at its $k$-th step, for some $i_k \in [s]$ and $j_k \in [n]$, then we perform the following steps:

**(i)** If $\mathcal{A}'$ has queried the $j_k$-th index of all the samples before this step, then we reuse the queried value.

**(ii)** Otherwise, we query the $j_k$-th index from all the samples $\{\mathbf{V}_1, \ldots, \mathbf{V}_s\}$.

**Phase 2:** Let $\mathcal{Q} \subset [n]$ be the set of indices queried by $\mathcal{A}'$ while running the $q$ querying steps of $\mathcal{A}$. If $|\mathcal{Q}| < q$, we arbitrarily pick $t = q - |\mathcal{Q}|$ distinct indices $\{j_1', \ldots, j_t'\}$, disjoint from the set of indices $\mathcal{Q}$. We query the set of indices $j_1', \ldots, j_t'$ from the entire set of sampled vectors $\mathbf{V}_1, \ldots, \mathbf{V}_s$.

The output (ACCEPT or REJECT) of $\mathcal{A}'$ is finally set to that of $\mathcal{A}$, and in particular depends only on the answers to the queries made in the first phase.

Now we have the following observation regarding the query complexity of $\mathcal{A}'$, which will be used to argue the query complexity of the non-adaptive algorithm later.

**Observation 11.4.** $\mathcal{A}'$ uses $s$ samples and performs exactly $qs$ queries. Moreover, for any distribution $D$, the output distribution of $\mathcal{A}'$ is the same as that of $\mathcal{A}$.

Let us assume that $\mathcal{A}'$ proceeds in $q$ steps by querying indices $\ell_1, \ldots, \ell_q \in [n]$ in each of the $s$ samples $\mathbf{V}_1, \ldots, \mathbf{V}_s$ (when the unknown distribution is $D$). Equivalently, we can think that the algorithm proceeds in $q$ steps, where in Step $k$ ($k \in [q]$), we query the $\ell_k$-th index of $\{\mathbf{V}_1, \ldots, \mathbf{V}_s\}$, such that $\ell_k$ depends on $\ell_1, \ldots, \ell_{k-1}$, where $2 \leq k \leq q$.

Let us now consider an uniformly random permutation $\sigma : [n] \to [n]$ (unknown to $\mathcal{A}'$). Assume that the unknown distribution is $D_\sigma$ instead of $D$. As $\mathcal{P}$ is index-invariant, we can assume that the algorithm $\mathcal{A}'$ runs on $D_\sigma$ for $q$ steps as follows. In Step $k$, $\mathcal{A}'$ queries the $\sigma(\ell_k)$-th index of each of the $s$ samples, for $k \in [q]$. Now we have the following observation regarding the distribution of the indices queried, which follows from $\sigma$ being uniformly random.

**Observation 11.5.** $\sigma(\ell_1)$ is uniformly distributed over $[n]$, and $\sigma(\ell_k)$ is uniformly distributed over $[n] \setminus \{\sigma(\ell_1), \ldots, \sigma(\ell_{k-1})\}$, where $2 \leq k \leq q$. Moreover, this holds even if we condition on the values $\ell_1, \ldots, \ell_k$ as well as $\sigma(\ell_1), \ldots, \sigma(\ell_{k-1})$.

Now the algorithm $\mathcal{A}''$ works as follows:

- First take a uniformly random permutation $\sigma : [n] \to [n]$.
- Run $\mathcal{A}'$ over $D_\sigma$ instead of $D$.

From the above description, it does not immediately follow that $\mathcal{A}''$ is a non-adaptive algorithm. But from the description along with Observation 11.5, it follows that $\mathcal{A}''$ is the same as the following algorithm:

- First take $s$ samples $\mathbf{V}_1, \ldots, \mathbf{V}_s$, and also pick a uniformly random non-repeating sequence of $q$ indices $r_1, \ldots, r_q \in [n]$.
- Run $\mathcal{A}'$ such that, for every $i \in [q]$, when $\mathcal{A}'$ is about to query $\ell_i$, query $r_i$ from all samples instead. That is, we assume $r_i$ to be the value of $\sigma(\ell_i)$.

The sample complexity and query complexity of algorithm $\mathcal{A}''$ are $s$ and $qs$, respectively, which follows from Observation 11.4 and Observation 11.5. The correctness of the algorithm follows from Observation 11.4 and Observation 11.5 along with the fact that $\mathcal{P}$ is index-invariant. This completes the proof of Theorem 11.3. □

## 11.3 Quadratic separation between adaptive and non-adaptive testers

### Preliminaries towards proving a quadratic separation result

In this subsection, we present some preliminary results required to prove that Theorem 7.9 is almost tight, that is, there exists an index-invariant property for which there is a nearly quadratic gap between the query complexities of adaptive and non-adaptive testers. The result is formally stated as follows.

**Theorem 11.6 (Restatement of Theorem 11.2).** *There exists an index-invariant property $\mathcal{P}_{\text{Gap}}$ that can be $\varepsilon$-tested adaptively using $\widetilde{\mathcal{O}}(n)$ queries for any $\varepsilon \in (0,1)$, while there exists an $\varepsilon \in (0,1)$ for which $\widetilde{\Omega}(n^2)$ queries are necessary for any non-adaptive $\varepsilon$-tester.*

In what follows throughout this section, we assume that the integer $n$ is of the form $n = 2^l$ for some integer $l$, and that $k = \mathcal{O}(l)$ is another integer. We denote vectors in $\{0,1\}^N$ by capital bold letters (for example $\mathbf{X} \in \{0,1\}^N$) and vectors in $\{0,1\}^n$ by small bold letters (for example $\mathbf{x} \in \{0,1\}^n$). For two vectors $\mathbf{X}, \mathbf{Y} \in \{0,1\}^N$, we will use $\delta_H(\mathbf{X}, \mathbf{Y}) = N \cdot d_H(\mathbf{X}, \mathbf{Y})$ to denote the absolute Hamming distance between $\mathbf{X}$ and $\mathbf{Y}$.

To construct the property $\mathcal{P}_{\text{Gap}}$ (as stated in Theorem 11.6), we define two encodings $\text{SE} : \{0,1\}^\ell \to \{0,1\}^k$ and $\text{GE} : [n]^m \to [n]^n$ [1]. The encodings GE and SE follow from the construction of a Probabilistically Checkable Unveiling of a Shared Secret (PCUSS) in [BFLR20]. We can also construct such a function GE using the Reed-Solomon code, where we will assume that $n$ is a prime power and use polynomials of degree $m-1$ over the field $\text{GL}(n)$ for $m = \Theta(n)$ [2].

**Function** SE**:** We will use a function SE of the form $\text{SE} : \{0,1\}^l \times \{0,1\} \to \{0,1\}^k$, where $l$ and $k$ are the integers defined above. In fact, SE takes an integer $i \in [n]$ in its Boolean encoding as an $l$ bit Boolean string and a "secret" bit $a \in \{0,1\}$, and will output a Boolean string of length $k$. SE will have the following properties for some constant $\zeta \in (0, 1/2)$.

**(i)** Let $i, i' \in [n]$ be two integers encoded as binary strings of length $l$ [3], and $a, a' \in \{0,1\}$. If $(i,a) \neq (i',a')$, then $\delta_H(\text{SE}(i,a), \text{SE}(i',a')) \geq \zeta \cdot k$.

**(ii)** Let $a \in \{0,1\}$ be a fixed bit, and suppose that $i$ is an integer chosen uniformly at random from $[n]$. Then for any set of indices $I \subset [k]$ such that $|I| \leq \zeta \cdot k$, the restriction $\text{SE}(i,a) \mid_I$ is uniformly distributed over $\{0,1\}^{|I|}$.

**Function** GE**:** For our construction, we will use another function GE of the form $\text{GE} : [n]^m \to [n]^n$, where $n, m \in \mathbb{N}$ with the following properties for the same constant $\zeta \in (0, 1/2)$ as above.

[1] SE stands for Secret Encoding, and GE stands for General Encoding.

[2] $\text{GL}(n)$ stands for the finite field with $n$ elements.

[3] Binary strings of length $\log n$ can actually encode only integers from $\{0, \ldots, n-1\}$, so we use the encoding of 0 for the value $n$.

**(i)** Let $\mathbf{z}, \mathbf{z}' \in [n]^m$ be two strings such that $\mathbf{z} \neq \mathbf{z}'$. For any two such strings $\mathbf{z}$ and $\mathbf{z}'$, $\delta_H(\mathrm{GE}(\mathbf{z}), \mathrm{GE}(\mathbf{z}')) = |\{i : \mathrm{GE}(\mathbf{z})_i \neq \mathrm{GE}(\mathbf{z}')_i\}| \geq \zeta \cdot n$.

**(ii)** Consider a string $\mathbf{z} \in [n]^m$ chosen uniformly at random. For any set of indices $I \subset [n]$ such that $|I| \leq \zeta \cdot n$, $\mathrm{GE}(\mathbf{z}) \mid_I$ is uniformly distributed over $[n]^{|I|}$.

From now on, we will use the following notation in this subsection: Let $n \in \mathbb{N}$ be such that $n = 2^l$ for some integer $l$, $k = \mathcal{O}(l)$ and $\zeta \in (0, 1/2)$ as above, $b = \lfloor \log(\lceil \log kn \rceil) \rfloor + 1$, $N = 1 + b + kn$ and $\alpha = 1/\log n$. Note that in particular $N = \mathcal{O}(n \log n)$. For a vector $\mathbf{X} \in \{0,1\}^N$ and a permutation $\pi : [N] \to [N]$, $\mathbf{X}_\pi$ denotes the vector obtained from $\mathbf{X}$ by permuting the indices of $\mathbf{X}$ with $\pi$, that is, $\mathbf{X}_\pi = (\mathbf{X}_{\pi(1)}, \ldots, \mathbf{X}_{\pi(N)})$.

Let $B$ be the sequence of integers $B = \{2, \ldots, b+1\}$, and for every $j \in [n]$, let $C_j$ denote the sequence of integers $C_j = \{b+2+k(j-1), \ldots, b+1+kj\}$. For a sequence of integers $A$ and a vector $\mathbf{X}$, we denote by $\mathbf{X} \mid_A$ the vector obtained by projecting $\mathbf{X}$ onto the set of indices of $A$ preserving the sequence order. For a sequence $A \subseteq [N]$ and a permutation $\pi : [N] \to [N]$, we denote by $\pi(A)$ the sequence obtained after permuting every element of $A$ with respect to the permutation $\pi$, that is, if $A = (a_1, \ldots, a_l)$, then $\pi(A) = (\pi(a_1), \ldots, \pi(a_l))$. In particular, we have $\mathbf{X}_\pi \mid_A = \mathbf{X} \mid_{\pi(A)}$. By abuse of notation and for simplicity, for a set of integers $A$ and a vector $\mathbf{X}$, we denote by $\mathbf{X} \mid_A$ the vector obtained by projecting $\mathbf{X}$ onto the set of indices of $A$, whenever the ordering in which we consider the indices in $A$ will be clear from the context [4].

In the following, we use string notation. For example, $\mathbf{1}^k\mathbf{0}^k$ denotes the vector in $\{0,1\}^{2k}$ whose first $k$ coordinates are 1 and whose last $k$ coordinates are 0. Now we formally define the notion of encoding of a vector which will be crucially used to define $\mathcal{P}_{\mathrm{Gap}}$.

**Definition 11.7 (Encoding of a vector).** Let $n, k, b \in \mathbb{N}, N = 1 + b + kn$, and $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \{0,1\}^n$ and $\mathbf{Y} \in \{0,1\}^N$ be two vectors. $\mathbf{Y}$ is said to be an *encoding* of $\mathbf{x}$ with respect to the functions $\mathrm{SE} : \{0,1\}^l \times \{0,1\} \to \{0,1\}^k$ and $\mathrm{GE} : [n]^m \to [n]^n$ if the following hold:

[4] A common scenario is when the indexes of $A$ are considered as a monotone increasing sequence.

**(i)** The first index of $\mathbf{Y}$ is $0$.

**(ii)** $\mathbf{Y}\mid_B$ is the all-1 vector.

**(iii)** $\mathbf{Y}\mid_{[N]\setminus\{1\}\cup B}$ is of the form $\mathrm{SE}(\mathrm{GE}(\mathbf{z})_1, \mathbf{x}_1)\ldots\mathrm{SE}(\mathrm{GE}(\mathbf{z})_n, \mathbf{x}_n)$ for some string $\mathbf{z} \in [n]^m$. In other words, $\mathbf{Y}\mid_{C_j} = \mathrm{SE}(\mathrm{GE}(\mathbf{z})_j, \mathbf{x}_j)$ for every $j \in [n]$.

For simplicity, we will denote this encoding by FE, that is, $\mathrm{FE} : [n]^m \times \{0,1\}^n \to \{0,1\}^N$ is the function [5] such that $\mathrm{FE}(\mathbf{z}, \mathbf{x}) = \mathbf{0}(\mathbf{1}^b)\mathrm{SE}(\mathrm{GE}(\mathbf{z})_1, \mathbf{x}_1)\ldots\mathrm{SE}(\mathrm{GE}(\mathbf{z})_n, \mathbf{x}_n)$ for $\mathbf{z} \in [n]^m$ and $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \{0,1\}^n$. We also say that $\mathbf{X} \in \{0,1\}^N$ is a *valid encoding* of some $\mathbf{x} \in \{0,1\}^n$, if there exists some $\mathbf{z} \in [n]^m$ for which $\mathbf{X} = \mathrm{FE}(\mathbf{z}, \mathbf{x})$. The image of FE will be called the set of all valid encodings.

Now let us infer two properties of the function FE, which will be crucial to our proofs, as stated in the following two claims. These properties of FE are analogous to the properties of SE and GE. As FE is formed by combining SE and GE, the proofs of these observations use their respective properties.

The following observation, particularly Items (i) and (ii), will allow us to prove that certain distributions are indeed far from the property $\mathcal{P}_{\mathrm{Gap}}$ (to be defined later) in the EMD metric. Item (iii) will be useful to prove the soundness of our adaptive algorithm in Subsection 11.3.2, and in particular in Lemma 11.30.

**Observation 11.8 (Distance properties of** FE**).** Let $\mathrm{FE} : [n]^m \times \{0,1\}^n \to \{0,1\}^N$ be the function from Definition 11.7. Then FE has the following properties:

**(i)** Let $\mathbf{x}, \mathbf{x}' \in \{0,1\}^n$ be any two strings and $\mathbf{z}, \mathbf{z}' \in [n]^m$ be two vectors such that $\mathbf{z} \neq \mathbf{z}'$. Then $\delta_H(\mathrm{FE}(\mathbf{z}, \mathbf{x}), \mathrm{FE}(\mathbf{z}', \mathbf{x}')) \geq \zeta^2 \cdot N/2$ holds.

**(ii)** Let $\mathbf{z}, \mathbf{z}' \in [n]^m$ be any two strings, and $\mathbf{x}, \mathbf{x}' \in \{0,1\}^n$ be two other strings such that $\mathbf{x} \neq \mathbf{x}'$. Then $\delta_H(\mathrm{FE}(\mathbf{z}, \mathbf{x}), \mathrm{FE}(\mathbf{z}', \mathbf{x}')) \geq \zeta k \cdot \delta_H(\mathbf{x}, \mathbf{x}')$.

**(iii)** Let $\mathbf{x}, \mathbf{x}' \in \{0,1\}^n$ be two strings and $\mathbf{z} \in [n]^m$ be a vector. Then we have $\delta_H(\mathrm{FE}(\mathbf{z}, \mathbf{x}), \mathrm{FE}(\mathbf{z}, \mathbf{x}')) \leq k \cdot \delta_H(\mathbf{x}, \mathbf{x}')$. Moreover, $d_H(\mathrm{FE}(\mathbf{z}, \mathbf{x}), \mathrm{FE}(\mathbf{z}, \mathbf{x}')) \leq d_H(\mathbf{x}, \mathbf{x}')$ holds.

[5] FE stands for Final Encoding.

*Proof.* We prove each item separately below.

**(i)** Following the properties of GE (Property (i)), for two strings $\mathbf{z}, \mathbf{z}' \in [n]^m$ such that $\mathbf{z} \neq \mathbf{z}'$, we can say that $\delta_H(\mathrm{GE}(\mathbf{z}), \mathrm{GE}(\mathbf{z}')) \geq \zeta \cdot n \geq \zeta N/2k$. That is, the number of indices $j \in [n]$ such that $\mathrm{GE}(\mathbf{z})_j \neq \mathrm{GE}(\mathbf{z}')_j$, is at least $\zeta N/2k$. For every index $j \in [n]$ such that $\mathrm{GE}(\mathbf{z})_j \neq \mathrm{GE}(\mathbf{z}')_j$, $\delta_H(\mathrm{SE}(\mathrm{GE}(\mathbf{z})_j, \mathbf{x}_j), \mathrm{SE}(\mathrm{GE}(\mathbf{z}')_j, \mathbf{x}'_j)) \geq \zeta \cdot k$ holds. This is due to Property (i) of SE. Hence,

$$\begin{aligned} \delta_H(\mathrm{FE}(\mathbf{z}, \mathbf{x}), \mathrm{FE}(\mathbf{z}', \mathbf{x}')) &\geq \sum_{j \in [n] : \mathbf{z}_j \neq \mathbf{z}'_j} \delta_H(\mathrm{SE}(\mathrm{GE}(\mathbf{z})_j, \mathbf{x}_j), \mathrm{SE}(\mathrm{GE}(\mathbf{z})_j, \mathbf{x}'_j)) \\ &\geq \zeta N/2k \cdot \zeta k = \zeta^2 \cdot N/2. \end{aligned}$$

**(ii)** Consider two strings $\mathbf{x}, \mathbf{x}' \in \{0,1\}^n$ such that $\mathbf{x} \neq \mathbf{x}'$. Using Property (i) of SE, we know that $\delta_H(\mathrm{SE}(\mathrm{GE}(\mathbf{z})_j, \mathbf{x}_j), \mathrm{SE}(\mathrm{GE}(\mathbf{z})_j, \mathbf{x}'_j)) \geq \zeta \cdot k$ for every $j$ for which $\mathbf{x}_j \neq \mathbf{x}'_j$. Note that the number of such indices $j$ is $\delta_H(\mathbf{x}, \mathbf{x}')$. Summing over them, we have the result.

**(iii)** Consider any two strings $\mathbf{x}, \mathbf{x}' \in \{0,1\}^n$. Observe that

$$\delta_H(\mathrm{FE}(\mathbf{z}, \mathbf{x}), \mathrm{FE}(\mathbf{z}, \mathbf{x}')) = \sum_{j \in [n]} \delta_H(\mathrm{SE}(\mathrm{GE}(\mathbf{z})_j, \mathbf{x}_j), \mathrm{SE}(\mathrm{GE}(\mathbf{z})_j, \mathbf{x}'_j)).$$

Note that $\delta_H(\mathrm{SE}(\mathrm{GE}(\mathbf{z})_j, \mathbf{x}_j), \mathrm{SE}(\mathrm{GE}(\mathbf{z})_j, \mathbf{x}'_j))$ is at most $k$ for every $j \in [n]$. Moreover, $\delta_H(\mathrm{SE}(\mathrm{GE}(\mathbf{z})_j, \mathbf{x}_j), \mathrm{SE}(\mathrm{GE}(\mathbf{z})_j, \mathbf{x}'_j)) = 0$ for every $j \in [n]$ with $\mathbf{x}_j = \mathbf{x}'_j$. Since the number of indices $j$ such that $\mathbf{x}_j \neq \mathbf{x}'_j$ is $\delta_H(\mathbf{x}, \mathbf{x}')$, we conclude the following:

$$\delta_H(\mathrm{FE}(\mathbf{z}, \mathbf{x}), \mathrm{FE}(\mathbf{z}, \mathbf{x}')) \leq k \cdot \delta_H(\mathbf{x}, \mathbf{x}').$$

Note that this immediately implies $d_H(\mathrm{FE}(\mathbf{z}, \mathbf{x}), \mathrm{FE}(\mathbf{z}, \mathbf{x}')) \leq d_H(\mathbf{x}, \mathbf{x}')$. □

The following lemma will provide us a way to construct distributions that cannot be easily distinguished using non-adaptive queries (following a uniformly random index-permutation which we will deploy).

**Lemma 11.9 (Projection property of** $\mathrm{FE}$**).** *Consider a fixed vector* $\mathbf{x} \in \{0,1\}^n$*, and let* $\mathbf{z} \in [n]^m$ *be a string chosen uniformly at random. For any set of indices* $Q \subseteq [N]$ *such that* $|Q| \leq \zeta \cdot N/2k$ *and* $|Q \cap C_j| \leq \zeta \cdot k$ *for every* $j \in [n]$*, the restriction of* $\mathrm{FE}(\mathbf{z},\mathbf{x}) \mid_{Q \setminus [b+1]}$ *is uniformly distributed over* $\{0,1\}^{|Q \setminus [b+1]|}$ [6].

*Proof.* For the set of indices $Q$, consider the set $J = \{j : Q \cap C_j \neq \emptyset\}$. From the statement of the lemma, we know that $|Q \cap C_j| \leq \zeta \cdot k$ for every $j \in J$. Noting that $|J| \leq |Q| \leq \zeta \cdot n$, if we consider the restriction $\mathrm{GE}(\mathbf{z}) \mid_J$, following Property (ii) of the function GE, we know that $\mathrm{GE}(\mathbf{z}) \mid_J$ is uniformly distributed over $[n]^{|J|}$.

Now when we call $\mathrm{SE}(i_j, \mathbf{x}_j)$ with $i_j \in [n]$ obtained from $\mathrm{GE}(\mathbf{z}) \mid_J$, following the above argument, we can say that $i_j$ has been chosen uniformly at random from $[n]$ (and independently from the other $i_{j'}$). Since $|Q \cap C_j| \leq \zeta \cdot k$, applying Property (ii) of the function SE, we know that the corresponding restriction of $\mathrm{SE}(i_j, \mathbf{x}_j)$ will be uniformly distributed over $\{0,1\}^{|Q \cap C_j|}$. As $\mathrm{FE}(\mathbf{z},\mathbf{x}) = \mathbf{0}(\mathbf{1}^b)\mathrm{SE}(\mathrm{GE}(\mathbf{z})_1, \mathbf{x}_1) \ldots \mathrm{SE}(\mathrm{GE}(\mathbf{z})_n, \mathbf{x}_n)$, combining the above arguments, we conclude that $\mathrm{FE}(\mathbf{z},\mathbf{x}) \mid_{Q \setminus [b+1]}$ is uniformly distributed over $\{0,1\}^{|Q \setminus [b+1]|}$. □

Now we are ready to formally define the property, first constructing a non-index-invariant version to be used in the next index-invariant definition.

**Property** $\mathcal{P}^0_{\mathrm{Gap}}$**:** A distribution $D$ over $\{0,1\}^N$ is in $\mathcal{P}^0_{\mathrm{Gap}}$ if and only if $D$ satisfies the following conditions:

**(i)** $D(\mathbf{U}) = \alpha$, where $\mathbf{U} = \mathbf{1}\mathbf{0}^{N-1}$ is the indicator vector for the index 1.

**(ii)** Consider the set of vectors $\mathcal{S} = \{\mathbf{V}_1, \ldots, \mathbf{V}_b\}$ in $\{0,1\}^N$ such that for every $i \in [b]$, the $i$-th vector $\mathbf{V}_i$ is of the form $\mathbf{1}^{i+1}\mathbf{0}^{N-1-i}$. Note that $\mathbf{V}_i \mid_B = 1^i 0^{b-i}$ for $B = \{2, \ldots, b+1\}$. We require that $D(\mathbf{V}_i) = \alpha/b$ for every $i \in [b]$.

**(iii)** Consider the set of vectors $\mathcal{T} = \{\mathbf{W}_0, \ldots, \mathbf{W}_{\lceil \log kn \rceil - 1}\}$ (disjoint from $\mathcal{S}$) in $\{0,1\}^N$ such that for every $\mathbf{W}_i \in \mathcal{T}$, $\mathbf{W}_i$ is of the form $\mathbf{0}(b(i))(\mathbf{0}^{2^i}\mathbf{1}^{2^i})^{kn/2^{i+1}}$, where $b(i)$

[6] Recall that the restriction $\mathrm{FE}(\mathbf{z},\mathbf{x}) \mid_{[b+1]}$ is always the vector $\mathbf{0}\mathbf{1}^b$.

denotes the length $b$ binary representation of $i$.[7] Note that for $i = b + 2 + j$, with $0 \leq j \leq kn - 1$, the sequence $(\mathbf{W}_0)_i, \ldots, (\mathbf{W}_{(\lceil \log kn \rceil - 1)})_i$ holds the binary representation of $j$. Also, note that there is an one-to-one correspondence between $\mathbf{W}_i \mid_B$ and $\mathbf{W}_i \mid_{[N] \setminus \{B\} \cup \{1\}}$. We require that $D(\mathbf{W}_i) = \alpha / |\mathcal{T}|$ for every $\mathbf{W}_i \in \mathcal{T}$.

**(iv)** Supp$(D) \setminus (\{\mathbf{U}\} \cup \mathcal{S} \cup \mathcal{T})$ consists of valid encodings of at most $n$ vectors from $\{0,1\}^n$ with respect to the functions $\mathrm{SE} : \{0,1\}^l \times \{0,1\} \to \{0,1\}^k$ and $\mathrm{GE} : [n]^m \to [n]^n$, for the integers $l, m, k \in \mathbb{N}$ as defined in Definition 11.7. That is, there exist vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \{0,1\}^n$ for which Supp$(D) \setminus (\{\mathbf{U}\} \cup \mathcal{S} \cup \mathcal{T}) \subseteq \{\mathrm{FE}(\mathbf{z}, \mathbf{x}_i) : \mathbf{z} \in [n]^m, i \in [n]\}$. Note that for $D$ to be a distribution, we must have $D(\text{Supp}(D) \setminus (\{\mathbf{U}\} \cup \mathcal{S} \cup \mathcal{T})) = 1 - 3\alpha$.

**Property $\mathcal{P}_{\text{Gap}}$:** A distribution $D$ over $\{0,1\}^N$ is said to be in the property $\mathcal{P}_{\text{Gap}}$ if $D_\pi$ is in $\mathcal{P}^0_{\text{Gap}}$ for some permutation $\pi : [N] \to [N]$.

**Remark 11.1 (Intuition behind the definition of $\mathcal{P}_{\text{Gap}}$).** If a distribution $D$ is in $\mathcal{P}^0_{\text{Gap}}$, then we can easily check (by querying the indexes in $B$) whether a sample from $D$ would be equal to $\mathrm{FE}(\mathbf{z}, \mathbf{x})$ for some $\mathbf{z} \in [n]^m$ and $\mathbf{x} \in \{0,1\}^n$. In that case, individual bits of $\mathbf{x}$ can be decoded by querying the appropriate $C_j$ and then passed to a tester of distributions over $\{0,1\}^n$.

On the other hand, if we take a uniformly random permutation of such a distribution $D$, which keeps it in $\mathcal{P}_{\text{Gap}}$ (though no longer in $\mathcal{P}^0_{\text{Gap}}$), a non-adaptive algorithm will need many queries to capture sufficiently many bits from any $C_j$, and this will enable us to fully hide the identity of $\mathbf{x}$ if fewer queries are performed.

By contrast, an adaptive tester will use relatively few samples that are queried in their entirety to obtain the (permutations of the) special vectors in Items (i) to (iii) of the definition of $\mathcal{P}^0_{\text{Gap}}$, from which it will be able to fully learn the index-permutation applied to the distribution, and continue to successfully decode individual bits. A few

[7] If $kn/2^{i+1}$ is not an integer, we trim the rightmost copy of $\mathbf{0}^{2^i}\mathbf{1}^{2^i}$ so that the total length of "$(\mathbf{0}^{2^i}\mathbf{1}^{2^i})^{kn/2^{i+1}}$" is exactly $kn$.

further samples queried in their entirety will ensure that there is very little total weight on vectors that are neither special vectors nor equal to $\mathrm{FE}(\mathbf{z}, \mathbf{x})$ for some $\mathbf{z} \in [n]^m$ and $\mathbf{x} \in \{0,1\}^n$.

**Known useful results about support estimation:** Now we state a lemma which will be required later to describe the adaptive tester for $\mathcal{P}_{\mathrm{Gap}}$. Informally, it says that whether a distribution $D$ over $\{0,1\}^n$ has support size $s$ or is $\varepsilon$-far from any such distribution, can be tested by taking $\widetilde{\mathcal{O}}(s)$ samples from $D$, and performing $\widetilde{\mathcal{O}}(s)$ queries on them.

**Lemma 11.10 (Support size estimation, Theorem $1.9$ and Corollary $2.3$ of [GR22] restated).** *There exists an algorithm* $\textsc{Supp-Est}(s, \varepsilon)$ *that uses* $\widetilde{\mathcal{O}}(s/\varepsilon^2)$ *queries to an unknown distribution $D$ defined over $\{0,1\}^n$, and with probability at least $\frac{9}{10}$ distinguishes whether $D$ has at most $s$ elements in its support or $D$ is $\varepsilon$-far from all such distributions with support size at most $s$.*

We will also use a lower bound on the support size estimation problem to prove the lower bound on non-adaptive testers for testing $\mathcal{P}_{\mathrm{Gap}}$. Informally speaking, given a distribution $D$ over $\{1, \ldots, 2n\}$, in order to distinguish in the traditional (non-huge-object) model whether the size of the support of $D$ is $n$, or $D$ is far from all such distributions, $\Omega(\frac{n}{\log n})$ samples are necessary. More formally, we have the following theorem.

**Theorem 11.11 (Support Estimation Lower bound, Corollary $9$ of [VV10] restated).** *There exist two distributions $D_{yes}^{\mathrm{Supp}}$ and $D_{no}^{\mathrm{Supp}}$ over distributions over $\{1, \ldots, 2n\}$, and an $\eta \in (0, 1/8)$ such that the following holds:*

- **(i)** *The probability mass of every element in the support of $D_{yes}^{\mathrm{Supp}}$ as well as $D_{no}^{\mathrm{Supp}}$ is a multiple of $1/2n$.*
- **(ii)** *$D_{yes}^{\mathrm{Supp}}$ is supported over distributions whose support size is $n$.*
- **(iii)** *$D_{no}^{\mathrm{Supp}}$ is supported over distributions whose support size is at least $(1+2\eta)n$, and in particular are $\eta$-far in variation distance from any distribution defined over $\{1, \ldots, 2n\}$ whose support size is $(1+2\eta)n$.*

**(iv)** *If a sequence of* $o(\frac{n}{\log n})$ *samples from a distribution are drawn according to either* $D^{\mathrm{Supp}}_{yes}$ *or* $D^{\mathrm{Supp}}_{no}$*, the resulting distributions over the sample sequences are* $1/4$*-close to each other.*

We present an adaptive algorithm to test $\mathcal{P}_{\mathrm{Gap}}$ in Subsection 11.3.2 and we prove the lower bound for non-adaptive testers in Subsection 11.3.3. In Subsection 11.3.1, we describe a subroutine to determine the unknown permutation that will be used in our adaptive algorithm in Subsection 11.3.2.

### 11.3.1 Determining the permutation $\pi$

Here we design an algorithm that, given a distribution $D \in \mathcal{P}_{\mathrm{Gap}}$, can learn with high probability the permutation $\pi$ for which $D_\pi \in P^0_{\mathrm{Gap}}$.

The crux of the algorithm is that if $D \in \mathcal{P}_{\mathrm{Gap}}$, then there exist $\mathbf{U}' = \mathbf{U}_\pi \in \{0,1\}^n$, $\mathcal{S}' = \mathcal{S}_\pi = \{\mathbf{V}'_i = (\mathbf{V}_i)_\pi : i \in [b]\}$ and $\mathcal{T}' = \mathcal{T}_\pi = \{\mathbf{W}'_j = (\mathbf{W}_j)_\pi : j \in \{0\} \cup [\lceil \log kn \rceil] - 1\}$ in the support of $D$ such that $D(\mathbf{U}') = \alpha$, $D(\mathbf{V}'_i) = \alpha/b$ for every $i \in [b]$ and $D(\mathbf{W}'_j) = \alpha/\lceil \log kn \rceil$. Note that $\mathbf{U}$, $\mathcal{S}$ and $\mathcal{T}$ are as defined in the property $\mathcal{P}^0_{\mathrm{Gap}}$.

The main observation is that, if we are given the set of special vectors $\{\mathbf{U}'\} \cup \mathcal{S}' \cup \mathcal{T}'$, then we can determine the permutation $\pi$. Our algorithm can find $\mathbf{U}', \mathcal{S}'$ and $\mathcal{T}'$ with high probability, if they exist, by taking $\mathcal{O}(\log^2 N/\alpha) = \mathcal{O}(\log^2 n/\alpha)$ samples and reading them in their entirety. This is due to the fact that the probability mass of every vector in the set of special vectors is at least $\Omega(\alpha/\log n)$.

The algorithm is described in the following subroutine FIND-PERMUTATION (see Algorithm 11.1) [8].

Let us start by analyzing the query complexity of FIND-PERMUTATION.

**Lemma 11.12 (Query complexity of** FIND-PERMUTATION**).** *The query complexity of the above defined* FIND-PERMUTATION *is* $\widetilde{\mathcal{O}}(N)$.

[8] This algorithm is not adaptive in itself, but its output is used adaptively in the testing algorithm described later.

**Algorithm 11.1:** FIND-PERMUTATION

**Input:** Sample and Query access to a distribution $D$ over $\{0,1\}^N$.
**Output:** Either a permutation $\pi : [N] \to [N]$, or FAIL.

**(i)** First take a multi-set $\mathcal{X}$ of $\mathcal{O}(\log^2 N/\alpha)$ samples from $D$, and query all the entries of the sampled vectors of $\mathcal{X}$ to know the vectors of $\mathcal{X}$ completely.

**(ii)** Find the set of distinct vectors in $\mathcal{X}$ that have exactly one $1$. If no such vector exists or there is more than one such vector, FAIL. Otherwise, denote by $\mathbf{U}'$ the vector that has exactly one $1$, and denote the corresponding index by $i^*$. Set $\pi(i^*) = 1$, and proceed to the next step.

**(iii)** Find the set of distinct vectors $\mathcal{S}' \subseteq \mathcal{X} \setminus \{\mathbf{U}'\}$ such that every vector in $\mathcal{S}'$ has $1$ at the index $i^*$ and has at least another $1$ among other indices. If no such vector exists, or $|\mathcal{S}'| \neq b$, FAIL. Otherwise, if the vectors of $\mathcal{S}'$ form a chain $\mathbf{V}'_1, \dots, \mathbf{V}'_b$, where $\mathbf{V}'_j$ has exactly $j+1$ many 1, then set $\pi(i_j) = j+1$, where $i_j$ is the index where $\mathbf{V}'_j$ has 1, but $\mathbf{V}'_{j-1}$ has 0 there, for every $j \in [b]$ (denoting $\mathbf{V'}_0 = \mathbf{U}'$ for the purpose here). Also, set $B' = (i_1, \dots, i_b)$. If $\mathcal{S}'$ does not form a chain $\mathbf{V}'_1, \dots, \mathbf{V}'_b$ as mentioned above, FAIL.

**(iv)** Let $\mathcal{T}' \subseteq \mathcal{X}$ be the set of distinct vectors such that every vector in $\mathcal{T}'$ has $0$ at the index $i^*$, and does not have $1$ in all indices of $B'$. If no such vector exists, FAIL. For every $j$, denote by $\mathbf{W}'_j$ the vector in $\mathcal{T}'$ for which $\mathbf{W}'_j \,|_{B'}= b(j)$, where $b(j)$ denotes the binary representation of $j$. For every $j \in \{0\} \cup [\lceil \log kn \rceil - 1]$, if either there are no vectors $\mathbf{W}'_j \in \mathcal{T}'$ or there is more than one distinct vector with $\mathbf{W'}_j \,|_{B'}= b(j)$, FAIL. Also, if there is any vector in $\mathbf{W}'_j \in \mathcal{T}'$ such that $\mathbf{W}'_j \,|_{B'}= b(j)$ for $\log kn \leq j < 2^b - 1$, FAIL.

**(v)** For any $i \in [N] \setminus (\{i^*\} \cup B')$, let $l_i$ be the integer with binary representation $(\mathbf{W'}_0)_i, \dots, (\mathbf{W'}_{\lceil \log kn \rceil - 1})_i$. Set $\pi(i) = b + 2 + l_i$ for every $i \in [N] \setminus (\{i^*\} \cup B')$. If $\pi$ is not a permutation of $[N]$, FAIL.

**(vi)** Take another multi-set $\mathcal{X}'$ of $\mathcal{O}(\log^2 N/\alpha)$ samples from $D$, and query all the entries of the sampled vectors of $\mathcal{X}'$ to know the vectors of $\mathcal{X}'$ completely. Let $\mathcal{Y}$ be a set of vectors in $\mathcal{X}'$ such that $\mathcal{Y} = \{\mathbf{Z} \in \mathcal{X}' : \mathbf{Z} \,|_{\{i^*\} \cup B'} \neq \mathbf{0}\mathbf{1}^b\}$. If $|\mathcal{Y}| \,/\, |\mathcal{X}'| > 4\alpha$, FAIL. Otherwise, output $\pi$.

*Proof.* Note that FIND-PERMUTATION takes a multi-set $\mathcal{X}$ of $\mathcal{O}(\log^2 N/\alpha)$ samples from $D$ in Step (i), and queries them completely. So, FIND-PERMUTATION performs $\mathcal{O}(N \log^2 N/\alpha)$ queries in Step (i). FIND-PERMUTATION does not perform any new queries in Step (ii), Step (iii), Step (iv) and Step (v). Finally, FIND-PERMUTATION takes another multi-set $\mathcal{X}'$ of $\mathcal{O}(\log^2 N/\alpha)$ samples from $D$ and queries them completely, similar to Step (i). Recalling that $\alpha = 1/\log n$, the query complexity of FIND-PERMUTATION is $\widetilde{\mathcal{O}}(N) = \widetilde{\mathcal{O}}(n)$ in total. □

Now we proceed to prove the correctness of FIND-PERMUTATION.

**Lemma 11.13 (Guarantee when $D \in \mathcal{P}_{\text{Gap}}$).** *If $D$ is a distribution defined over $\{0,1\}^N$ such that $D \in \mathcal{P}_{\text{Gap}}$, then with probability at least $9/10$,* FIND-PERMUTATION *reports the permutation $\pi$ such that $D_\pi \in \mathcal{P}^0_{\text{Gap}}$.*

We prove the above lemma by a series of intermediate lemmas. In the following lemmas, we consider $\mathbf{U}$, $\mathcal{S}$ and $\mathcal{T}$ as per the definition of $\mathcal{P}^0_{\text{Gap}}$. Also, consider the permutation $\pi$ such that $D_\pi \in \mathcal{P}_{\text{Gap}}$.

**Lemma 11.14 (Correctly finding $\pi^{-1}(1)$).** *With probability at least $1 - 1/N^3$, $\mathcal{X}$ will contain the vector $\mathbf{U}'$ for which $\mathbf{U}'_\pi = \mathbf{U}$, and $i^* = \pi^{-1}(1)$ will be identified correctly. Moreover,* FIND-PERMUTATION *proceeds to Step (iii).*

*Proof.* By the definition of $\mathcal{P}^0_{\text{Gap}}$, the vector $\mathbf{U}'$ is the only vector in the support of $D$ containing a single 1. Since $D(\mathbf{U}') = D(\mathbf{U}_{\pi^{-1}(1)}) = \alpha$, and we are taking $|\mathcal{X}|$ samples from $D$, the probability that $\mathbf{U}'$ will not appear in $\mathcal{X}$ is at most $(1-\alpha)^{|X|} \leq \frac{1}{N^3}$. Thus, with probability at least $1 - \frac{1}{N^3}$, $\mathbf{U}' \in \mathcal{X}$ and FIND-PERMUTATION in Step (ii) proceeds to the next step. □

**Lemma 11.15 (Correctly finding $B' = \pi^{-1}(B)$).** *With probability at least $1-1/N^3$, the algorithm* FIND-PERMUTATION *will correctly identify $\mathbf{V}'_1, \ldots, \mathbf{V}'_b$ for which $\mathbf{V}'_{i,\pi} = \mathbf{V}_i$, and $B' = \pi^{-1}(2), \ldots, \pi^{-1}(b+1)$ will be identified correctly as well. Moreover,* FIND-PERMUTATION *proceeds to Step (iv).*

*Proof.* Let $\mathbf{V'}_1, \dots, \mathbf{V'}_b$ denote the vectors for which $\mathbf{V'}_{i,\pi} = \mathbf{V}_i$ for every $i$. Note that these are the only vectors outside $\mathbf{U}'$ in the support of $D$ that have $1$ at the index $i^*$. As $D(\mathbf{V}'_i) = \frac{\alpha}{b}$, the probability that $\mathbf{V}'_i$ does not appear in $\mathcal{X}$ is at most $(1 - \frac{\alpha}{b})^{|X|}$. Since $|\mathcal{X}| = \mathcal{O}(\log^2 N/\alpha)$ and $b = \mathcal{O}(\log \log kn)$, the probability that $\mathbf{V}'_i \in \mathcal{X}$ is at least $1 - \frac{1}{N^4}$. Using the union bound over all the vectors of $\mathcal{S}'$, with probability at least $1 - 1/N^3$, we know that all of these vectors are in $\mathcal{X}$, in which case they are identified correctly, so $B'$ is identified correctly as well, and FIND-PERMUTATION in Step (iii) proceeds to the next step. □

**Lemma 11.16 (Identifying $\pi^{-1}(b+2), \dots, \pi^{-1}(N)$).** *Let $\mathbf{W'}_1, \dots, \mathbf{W'}_{\lceil \log kn \rceil - 1}$ denote the vectors for which $\mathbf{W'}_{j,\pi} = \mathbf{W}_j$ for every $j$. With probability at least $1 - 1/N^3$, all these vectors appear in $\mathcal{X}$, in which case they are identified correctly, and so are $\pi^{-1}(b+2), \dots, \pi^{-1}(N)$. Moreover,* FIND-PERMUTATION *proceeds to Step (vi).*

The proof of the above lemma is similar to the proof of Lemma 11.15 and is omitted. Note that from Lemma 11.14, Lemma 11.15 and Lemma 11.16, we know that with probability at least $1 - o(1)$, the algorithm FIND-PERMUTATION has correctly determined the permutation $\pi$ and proceeded to Step (vi). We will finish up the proof of Lemma 11.13 using the following lemma.

**Lemma 11.17.** *The probability that* FIND-PERMUTATION *outputs* FAIL *in Step (vi) (instead of outputting $\pi$) is at most $1/N^3$.*

*Proof.* As $D \in \mathcal{P}_{\text{Gap}}$, from the description of the property, we know that $D(\{\mathbf{U}'\} \cup \mathcal{S}' \cup \mathcal{T}') = 3\alpha$. As $|\mathcal{X}'| = \mathcal{O}(\log^2 N/\alpha)$, using the Chernoff bound (Lemma 2.11), we have the result. □

Combining the above lemmas, we conclude that with probability at least $9/10$, the algorithm FIND-PERMUTATION outputs a correct permutation $\pi$, completing the proof of Lemma 11.13.

To conclude this section, we show that with high probability, we will not output $\pi$ for which too much weight is placed outside the “encoded part” of the distribution.

**Lemma 11.18.** *For any distribution $D$ (regardless of whether $D$ is in $\mathcal{P}_{\text{Gap}}$ or not), the probability that* Find-Permutation *outputs a permutation $\pi$ for which $D(\{\mathbf{X} : \mathbf{X} |_{\{i^*\} \cup B'} = \mathbf{0}\mathbf{1}^b\}) \leq 1 - 5\alpha$ is at most $1/10$.*

*Proof.* Recall the set of vectors $\mathcal{Y}$ as defined in Step (vi) of Find-Permutation: $\mathcal{Y} = \{\mathbf{Z} \in \mathcal{X}' : \mathbf{Z} |_{i^* \cup B'} \neq \mathbf{0}\mathbf{1}^b\}$, where $\mathcal{X}'$ is the multi-set of (new) samples obtained in Step (vi) of Find-Permutation. Consider a distribution $D$ such that $D(\{\mathbf{X} : \mathbf{X}_{\{i^*\} \cup B'} = \mathbf{0}\mathbf{1}^b\}) \leq 1 - 5\alpha$. This implies that $\mathbb{E}\left[|\mathcal{Y}| \,/\, |\mathcal{X}'|\right] \geq 5\alpha$. As $|\mathcal{X}'| = \mathcal{O}(\log^2 N/\alpha)$, using the Chernoff bound (Lemma 2.11), we obtain that with probability at least $9/10$, the algorithm Find-Permutation outputs Fail in Step (vi), and does not output any permutation $\pi$. This completes the proof. □

### 11.3.2 The upper bound on adaptive testing for property $\mathcal{P}_{\text{Gap}}$

In this subsection, we design the adaptive tester for the property $\mathcal{P}_{\text{Gap}}$. Given a distribution $D$ over $\{0,1\}^N$, with high probability, Alg-Adaptive outputs Accept when $D \in \mathcal{P}_{\text{Gap}}$, and outputs Reject when $D$ is far from $\mathcal{P}_{\text{Gap}}$. The formal adaptive algorithm is presented in Alg-Adaptive (see Algorithm 11.2). Note that it has only two adaptive steps.

In the first adaptive step, our tester Alg-Adaptive starts by calling the algorithm Find-Permutation (as described in Subsection 11.3.1) whose query complexity is $\widetilde{\mathcal{O}}(n)$. If $D \in \mathcal{P}_{\text{Gap}}$, with high probability, Find-Permutation returns the permutation $\pi$ such that $D_\pi \in \mathcal{P}^0_{\text{Gap}}$. Once $\pi$ is known, when we obtain a sample $\mathbf{X}$ from $D$, we can consider it as $\mathbf{X}_\pi$ from $D_\pi$. Also, from the structure of the vectors in the support of the distributions in $P^0_{\text{Gap}}$, we can decide whether $X_\pi$ is a special vector, that is, $X_\pi \in \{\mathbf{U}\} \cup \mathcal{S} \cup \mathcal{T}$ or $\mathbf{X}_\pi$ is an encoding vector, that is, $\mathbf{X}_\pi = \text{FE}(\mathbf{z}, \mathbf{x})$ for some $\mathbf{z} \in [n]^m$ and $\mathbf{x} \in \{0,1\}^n$. Observe that, in the later case, we can decode any bit of $\mathbf{x}$ (say $\mathbf{x}_j$) by finding $\mathbf{X}_\pi$ projected into $C_j$, which can be done by performing $\mathcal{O}(\log n)$ queries.

As the second adaptive step, our algorithm asks for a sequence $\mathcal{Y}$ of $\mathcal{O}(n/\varepsilon)$ samples from $D$, that is, from $D_\pi$. Let $\mathcal{Y}' \subseteq \mathcal{Y}$ be the sequence of encoding vectors in $\mathcal{Y}$. We now call Supp-Est$(\mathcal{Y}', \varepsilon/3)$ (from Lemma 11.10), and depending on its output, Alg-

ADAPTIVE either reports ACCEPT or REJECT. Note that we can execute every query by SUPP-EST($\mathcal{Y}', \varepsilon/3$), by performing $\mathcal{O}(\log n)$ queries to the corresponding sample in $\mathcal{Y}'$ as discussed above.

When $D \in \mathcal{P}_{\text{Gap}}$ (that is, $D_\pi \in \mathcal{P}^0_{\text{Gap}}$ for the permutation $\pi$), the set of encoding vectors in $D_\pi$ is the encoding of at most $n$ vectors in $\{0,1\}^n$. So, in that case, ALG-ADAPTIVE reports ACCEPT with high probability. Now consider the case where $D$ is $\varepsilon$-far from $\mathcal{P}_{\text{Gap}}$. If ALG-ADAPTIVE has not rejected $D$ before calling SUPP-EST($\mathcal{Y}', \varepsilon/3$), we will show that the distribution over $\{0,1\}^n$ induced by the vectors decoded from the encoding vectors in $D_\pi$ is $\varepsilon/3$-far from having support size $n$. Then, ALG-ADAPTIVE will still reject $D$ with high probability.

Let us first discuss the query complexity of ALG-ADAPTIVE.

**Lemma 11.19 (Query complexity of ALG-ADAPTIVE).** *The query complexity of the adaptive tester* ALG-ADAPTIVE *for testing the property $\mathcal{P}_{\text{Gap}}$ is $\widetilde{\mathcal{O}}(N) = \widetilde{\mathcal{O}}(n)$.*

*Proof.* Note that ALG-ADAPTIVE calls the algorithm FIND-PERMUTATION in Step (i). Following the query complexity lemma of FIND-PERMUTATION (Lemma 11.12), we know that FIND-PERMUTATION performs $\widetilde{\mathcal{O}}(N)$ queries.

For every sample taken in Step (ii), the sampled vectors of the multi-set $\mathcal{X}$ are queried completely. Since we take $\mathcal{O}(1/\varepsilon)$ samples, this step requires $\mathcal{O}(N/\varepsilon) = \widetilde{\mathcal{O}}(n/\varepsilon)$ queries in total.

Then in Step (iii), ALG-ADAPTIVE takes a multi-set $\mathcal{Y}$ of $\mathcal{O}(n/\varepsilon)$ samples, and queries for the indices in $\{i^*\} \cup B'$ to get the vectors in $\mathcal{Y}'$, which takes $\mathcal{O}(n \log\log kn/\varepsilon)$ queries. Finally, in Step (iv), ALG-ADAPTIVE calls the algorithm SUPP-EST, which performs $\widetilde{\mathcal{O}}(n)$ queries (following Lemma 11.10), each of them simulated by $\mathcal{O}(\log n)$ queries to some $\mathbf{Y}_i \mid_{C'_j}$. Thus, in total, ALG-ADAPTIVE performs $\widetilde{\mathcal{O}}(N) = \widetilde{\mathcal{O}}(n)$ queries. □

Now we prove the correctness of ALG-ADAPTIVE. We will start with the completeness proof.

**Lemma 11.20 (Completeness of ALG-ADAPTIVE).** *Let $D$ be a distribution defined*

**Algorithm 11.2:** ALG-ADAPTIVE

**Input:** Sample and Query access to a distribution $D$ over $\{0,1\}^N$, and a parameter $\varepsilon \in (0,1)$.

**Output:** Either ACCEPT or REJECT.

**(i)** Call FIND-PERMUTATION. If FIND-PERMUTATION returns FAIL, REJECT. Otherwise, let $\pi$ be the permutation returned by FIND-PERMUTATION. Denote for convenience $i^* = \pi^{-1}(1), B' = \pi^{-1}(B)$, and $C'_j = \pi^{-1}(C_j)$ for every $j \in [n]$.

**(ii)** Take a multi-set $\mathcal{X}$ of $\mathcal{O}(1/\varepsilon)$ samples from $D$, and query all the entries of the sampled vectors of $\mathcal{X}$ to know the vectors of $\mathcal{X}$ completely. If there is any vector $\mathbf{X}$ for which $\mathbf{X} \mid_{\{i^*\}\cup B'} = \mathbf{01}^b$ (according to the permutation $\pi$ obtained from Step (i)) for which $\mathbf{X}_\pi$ is not in the image of FE (i.e. it is not a valid encoding of any vector in $\{0,1\}^n$), REJECT. Otherwise, proceed to the next step.

**(iii)** Take a sequence of samples $\mathcal{Y}$ such that $|\mathcal{Y}| = \mathcal{O}(n/\varepsilon)$ from $D$ and construct the sequence of vectors $\mathcal{Y}'$ such that $\mathcal{Y}' = \{\mathbf{Y} \in \mathcal{Y} : \mathbf{Y} \mid_{\{i^*\}\cup B'} = \mathbf{01}^b\}$ by querying the indices corresponding to $\{i^*\} \cup B'$.

**(iv)** Call SUPP-EST$(\mathcal{Y}', \varepsilon/3)$ (from Lemma 11.10), where a query to an index $j$ is simulated by querying the indices of $C'_j$ and decoding the obtained vector with respect to to SE (that is, checking whether the restriction of the queried vector to $C'_j$ is equal to $\mathrm{SE}(i,0)$ for some $i$, or equal to $\mathrm{SE}(i,1)$ for some i). REJECT if any of the following conditions hold:

(a) $|\mathcal{Y}'| \,/\, |\mathcal{Y}| \leq 1/2$ (due to the absence of sufficiently many samples in $\mathcal{Y}'$ to apply SUPP-EST).

(b) SUPP-EST$(\mathcal{Y}', \varepsilon/3)$ queries an index $j$ from some $\mathbf{Y}_i$ corresponding to an invalid encoding of $\mathbf{Y}_i \mid_{C'_j}$ (that is, when $\mathbf{Y}_i \mid_{C'_j}$ is not in the image of SE).

(c) SUPP-EST$(\mathcal{Y}', \varepsilon/3)$ outputs REJECT.

Otherwise, ACCEPT.

*over* $\{0,1\}^N$. *If* $D \in \mathcal{P}_{\text{Gap}}$*, then the algorithm* ALG-ADAPTIVE *will output* ACCEPT *with probability at least* $2/3$*.*

*Proof.* Consider a distribution $D \in \mathcal{P}_{\text{Gap}}$. From the completeness lemma of FIND-PERMUTATION (Lemma 11.13), we infer that FIND-PERMUTATION returns the correct permutation $\pi$ in Step (i), with probability at least $9/10$. Then, by the definition of $\mathcal{P}_{\text{Gap}}$, the algorithm ALG-ADAPTIVE can never encounter any samples with invalid encodings in Step (ii) which could cause it to REJECT. Thus, with probability at least $9/10$, the algorithm proceeds, with the correct permutation $\pi$, to Step (iii) and Step (iv).

As $D \in \mathcal{P}_{\text{Gap}}$, $D(\{\mathbf{U}'\} \cup \mathcal{S}' \cup \mathcal{T}') = 3\alpha$. Since $|\mathcal{Y}| = \mathcal{O}(n/\varepsilon)$, using the Chernoff bound (Lemma 2.12 (ii)), we can say that, with probability at least $9/10$, $|\mathcal{Y}'| / |\mathcal{Y}| \geq 1/2$. Moreover, as the vectors in $\mathcal{Y}'$ are valid encodings with respect to the function FE of at most $n$ vectors from $[2n]$, following the support estimation upper bound lemma (Lemma 11.10), we obtain that SUPP-EST outputs ACCEPT with probability at least $9/10$. Combining these, we conclude that ALG-ADAPTIVE outputs ACCEPT with probability at least $2/3$. □

Now we prove that when $D$ is $\varepsilon$-far from $\mathcal{P}_{\text{Gap}}$, ALG-ADAPTIVE will output REJECT with probability at least $2/3$.

**Lemma 11.21 (Soundness of** ALG-ADAPTIVE**).** *Let* $\varepsilon \in (0,1)$ *be a proximity parameter. Assume that D is a distribution defined over* $\{0,1\}^N$ *such that D is* $\varepsilon$*-far from* $\mathcal{P}_{\text{Gap}}$*. Then* ALG-ADAPTIVE *outputs* REJECT *with probability at least* $2/3$*.*

From the description of ALG-ADAPTIVE (Algorithm 11.2), if the tester reports REJECT before executing all the steps of SUPP-EST$(\mathcal{Y}', \varepsilon/3)$ in Step (iv), then we are done. So, let us assume that ALG-ADAPTIVE executes all the steps of SUPP-EST$(\mathcal{Y}', \varepsilon/3)$. Let $\mathcal{Y}'$ be the set of samples from a distribution $D^{\#}$ over $\{0,1\}^n$ as it is presented to SUPP-EST$(\mathcal{Y}', \varepsilon/3)$. Note that $D^{\#}$ is unknown and we are accessing $D^{\#}$ indirectly via decoding samples from $D$ over $\{0,1\}^N$. From the correctness SUPP-EST$(\mathcal{Y}', \varepsilon/3)$, we will be done with the proof of Lemma 11.21 by proving the following lemma.

**Lemma 11.22 (Property of the decoded distribution).** *$D^{\#}$ is $\varepsilon/3$-far from having support size at most $n$.*

We prove the above lemma using a series of claims. Let $D$ be a distribution which is $\varepsilon$-far from $\mathcal{P}_{\text{Gap}}$, and $\mathcal{V}$ denote the set $\{\mathbf{X} \in \text{Supp}(D) : \mathbf{X} \mid_{\{i^*\} \cup B'} = \mathbf{0}\mathbf{1}^b\}$, and let us define $\mathcal{U} = \text{Supp}(D) \setminus \mathcal{V}$. Let us start with the following observation.

**Observation 11.23.** $D(\mathcal{U}) \leq 5\alpha$, unless the algorithm ALG-ADAPTIVE has rejected with probability at least $1 - 1/N^3$ in Step (i).

*Proof.* Since ALG-ADAPTIVE in Step (i) invokes the algorithm FIND-PERMUTATION, this follows immediately from Lemma 11.18. □

Let $\pi$ be the permutation returned by FIND-PERMUTATION. Now assume $\mathcal{V}^{\text{inv}} \subseteq \mathcal{V}$ denotes the following set of vectors:

$$\mathcal{V}^{\text{inv}} = \{\mathbf{X} \in \mathcal{V} : \mathbf{X}_\pi \neq \text{FE}(\mathbf{z}, \mathbf{x}) \text{ for all } \mathbf{z} \in [n]^m, \mathbf{x} \in \{0,1\}^n\}$$

For every vector $\mathbf{V} \in \mathcal{V}^{\text{inv}}$, let $\Gamma'_{\mathbf{V}} = \{j \in [n] : \mathbf{V} \mid_{C'_j}$ is not in the image of SE$\}$ denotes the set of indices in $[n]$ of chunks of all the "*locally invalid*" encodings in the vector $\mathbf{V}$ [9]. Now we have the following observation.

**Observation 11.24.** $D(\mathcal{V}^{\text{inv}}) \leq \varepsilon/10$.

The above observation holds as otherwise, ALG-ADAPTIVE would have rejected in Step (ii) with probability at least $2/3$.

Let us define a distribution $D_1$ over $\{0,1\}^N$ using the following procedure:

**(i)** Set $D_1(\mathbf{X}) = D(\mathbf{X})$ for every $\mathbf{X} \in \mathcal{U}$.

**(ii)** Recall that $\Gamma'_{\mathbf{V}} = \{j \in [n] : \mathbf{V} \mid_{C'_j}$ is not in the image of SE$\}$ for every vector $\mathbf{V} \in \mathcal{V}^{\text{inv}}$. For every vector $\mathbf{V} \in \mathcal{V}^{\text{inv}}$, we perform the following steps:

[9] Note that it may be the case that $\Gamma'_{\mathbf{V}} = \emptyset$, for example when for every $j \in [n]$, we have $\mathbf{V} \mid_{C'_j} = \text{SE}(i_j, \mathbf{x}_j)$, for some $i_1, \ldots, i_n$ and $\mathbf{x}_1, \ldots, \mathbf{x}_n$ for which $i_1, \ldots, i_n$ are not in the image of GE.

**(a)** For every $j \notin \Gamma'_{\mathbf{V}}$, decode the vector $\mathbf{V} \mid_{C'_j}$ using SE to obtain $\mathbf{x}_j \in \{0, 1\}$.

**(b)** For every $j \in \Gamma'_{\mathbf{V}}$, choose an arbitrary value $\mathbf{x}_j$ from $\{0, 1\}$.

**(c)** Using $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ obtained from (a) and (b), construct a new vector $\mathbf{V}'$ for which $\mathbf{V}'_\pi = \mathrm{FE}(\mathbf{z}, \mathbf{x})$ for an arbitrary $\mathbf{z} \in [n]^m$, where $\pi$ is the permutation obtained from FIND-PERMUTATION in Step (i) of ALG-ADAPTIVE.

**(iii)** For every vector $\mathbf{V} \in \mathcal{V} \setminus \mathcal{V}^{\text{inv}}$, set $\mathbf{V}' = \mathbf{V}$.

**(iv)** Finally define $D_1(\mathbf{W}) = \sum_{\mathbf{V}:\mathbf{V}'=\mathbf{W}} D(\mathbf{V})$ for every $\mathbf{W} \in \mathcal{V}$.

Let $\mathcal{V}'$ be the set of vectors in $\{0, 1\}^N$ that are in the support of $D_1$ but not in $\mathcal{U}$, that is, $\mathcal{V}' = \{\mathbf{X} : \mathbf{X} \in \mathrm{Supp}(D_1) \setminus \mathcal{U}\}$. From the construction of $D_1$, the following observation follows.

**Observation 11.25.** $D_1(\mathcal{U}) = D(\mathcal{U}) \leq 5\alpha$ and $D_1(\mathcal{V}') = D(\mathcal{V}) = 1 - D(\mathcal{U}) \geq 1 - 5\alpha$.

Now we prove that the distributions $D$ and $D_1$ are not far in Earth Mover Distance.

**Lemma 11.26.** *The Earth Mover Distance between $D$ and $D_1$ is at most $\varepsilon/10$.*

*Proof.* Recall that the EMD between $D$ and $D_1$ is the solution to the following LP:

$$\begin{aligned} &\text{Minimize} \quad \sum_{\mathbf{X},\mathbf{Y} \in \{0,1\}^N} f_{\mathbf{XY}} d_H(\mathbf{X}, \mathbf{Y}) \\ &\text{Subject to} \quad \sum_{\mathbf{Y} \in \{0,1\}^N} f_{\mathbf{XY}} = D(\mathbf{X}) \; \forall \, \mathbf{X} \in \{0,1\}^N \\ &\text{and} \quad \sum_{\mathbf{X} \in \{0,1\}^N} f_{\mathbf{XY}} = D_1(\mathbf{Y}) \; \forall \, \mathbf{Y} \in \{0,1\}^N. \end{aligned}$$

Consider the flow $f^*$ such that $f^*_{\mathbf{XX}} = D(\mathbf{X})$ for every $\mathbf{X} \in \mathcal{U} \cup (\mathcal{V} \setminus \mathcal{V}^{\text{inv}})$, $f^*_{\mathbf{VV}'} = D_1(\mathbf{V})$ for every $\mathbf{V} \in \mathcal{V}^{\text{inv}}$, and $f^*_{\mathbf{XY}} = 0$ for all other vectors. Then we have the following:

$$\begin{aligned} d_{EM}(D, D_1) &\leq \sum_{\mathbf{X},\mathbf{Y}\in\{0,1\}^N} f^*_{\mathbf{X}\mathbf{Y}} d_H(\mathbf{X}, \mathbf{Y}) \\ &\leq \sum_{\mathbf{X}\in\{0,1\}^N\setminus\mathcal{V}^{\text{inv}}} f^*_{\mathbf{X}\mathbf{X}} d_H(\mathbf{X}, \mathbf{X}) + \sum_{\mathbf{V}\in\mathcal{V}^{\text{inv}}} f^*_{\mathbf{V}\mathbf{V}'} d_H(\mathbf{V}, \mathbf{V}') \\ &\leq 0 + \sum_{\mathbf{V}\in\mathcal{V}^{\text{inv}}} D(\mathbf{V}) d_H(\mathbf{V}, \mathbf{V}'). \end{aligned}$$

To bound the second term of the last expression, note that

$$\sum_{\mathbf{V}\in\mathcal{V}^{\text{inv}}} D(\mathbf{V}) d_H(\mathbf{V}, \mathbf{V}') \leq D(\mathcal{V}^{\text{inv}}) \leq \varepsilon/10.$$

This follows from Observation 11.24. Thus, we conclude that $d_{EM}(D, D_1) \leq \varepsilon/10$, completing the proof of the lemma. □

Now we have the following observation regarding the rejection probabilities of ALG-ADAPTIVE for the distributions $D$ and $D_1$. This will imply that, as we are executing all steps of SUPP-EST$(\mathcal{Y}', \varepsilon/3)$, the steps of our algorithm are oblivious to both $D$ and $D_1$. That is, we can assume that the input to the algorithm ALG-ADAPTIVE is the distribution $D_1$ instead of $D$.

**Observation 11.27.** The probability that the tester ALG-ADAPTIVE outputs REJECT in Step (iv) where the input distribution is $D$ is at least as large as the probability that ALG-ADAPTIVE outputs REJECT in Step (iv) when the input distribution is $D_1$.

*Proof.* Note that in the distribution $D$, there can be some vectors in Supp$(D)$ that are not valid encodings with respect to the function FE. Thus during its execution, the tester ALG-ADAPTIVE can REJECT $D$ by Condition (ii) and Condition (iv) (b). However, by the construction of $D_1$ from $D$, we have replaced the invalid encoding vectors with valid encoding vectors. Thus, the only difference it makes here is that ALG-ADAPTIVE may eventually accept a sample from $D_1$ when encountering such a place where a sample from $D$ would have been immediately rejected by Condition (ii) or Condition (iv) (b).

Other than this difference, the distributions $D$ and $D_1$ are identical. So, the probability that ALG-ADAPTIVE will REJECT $D$ is at least as large as the probability that it REJECTS $D_1$. $\square$

Now let us return to the proof of Lemma 11.22. Recall that $\mathcal{V}' = \{\mathbf{X} : \mathbf{X} \in \mathrm{Supp}(D_1) \setminus \mathcal{U}\}$. Let us define the distribution $D^{\#}$ over $\{0,1\}^n$ referred in Lemma 11.22. For $\mathbf{x} \in \{0,1\}^n$, we have the following:

$$D^{\#}(\mathbf{x}) = D_1^{dec}(\mathbf{x}) = \frac{1}{D_1(\mathcal{V}')} \sum_{\substack{\mathbf{Y}_\pi = \mathrm{FE}(\mathbf{z},\mathbf{x}) \\ \text{for some } \mathbf{z} \in [n]^m}} D_1(\mathbf{Y}) = \frac{1}{D_1(\mathcal{V}')} \sum_{\mathbf{z} \in [n]^m} D_1(\mathrm{FE}(\mathbf{z},\mathbf{x})_{\pi^{-1}}). \tag{11.1}$$

For the sake of contradiction, assume that $D^{\#} = D_1^{dec}$ is $\varepsilon/3$-close to having support size at most $n$. Let $D_2$ be a distribution over $\{0,1\}^n$ having support size at most $n$ such that the Earth Mover Distance between $D_2$ and $D_1^{dec}$ is at most $\varepsilon/3$.

Given the distribution $D_2$ over $\{0,1\}^n$, and the flow $f'_{\mathbf{xy}}$ from $D_2$ to $D_1^{dec}$ realizing the EMD of at most $\varepsilon/3$ between them, let us consider the distribution $D_2^{enc}$ over $\{0,1\}^N$ as follows:

**(i)** For any $\mathbf{X} \in \mathcal{V}'$, for which $\mathbf{X}_\pi = \mathrm{FE}(\mathbf{z},\mathbf{x})$ for some $\mathbf{z} \in [n]^m$, set:

$$D_2^{enc}(\mathbf{X}) = \sum_{\mathbf{y} \in \{0,1\}^n} f'_{\mathbf{xy}} \frac{D_1(\mathrm{FE}(\mathbf{z},\mathbf{y})_{\pi-1})}{D_1^{dec}(\mathbf{y})}.$$

**(ii)** For every $\mathbf{X} \in \mathcal{U}$, set $D_2^{enc}(\mathbf{X}) = D_1(\mathbf{X})$.

The following observation follows from Observation 11.25 and the construction of $D_2^{enc}$.

**Observation 11.28.** $D_2^{enc}(\mathcal{U}) = D_1(\mathcal{U}) \leq 5\alpha$ and $D_2^{enc}(\mathcal{V}') = D_1(\mathcal{V}') = 1 - D_1(\mathcal{U}) \geq 1 - 5\alpha$.

The following two lemmas bound the distance of $D_2^{enc}$ from $\mathcal{P}_{\mathrm{Gap}}$ and from $D_1$, where $D_1^{dec}$ is $\varepsilon/3$-close to having support size at most $n$. We will prove these two lemmas later.

**Lemma 11.29.** *$D_2^{enc}$ is $6\alpha$-close to $\mathcal{P}_{\text{Gap}}$.*

**Lemma 11.30.** *The Earth Mover Distance between $D_2^{enc}$ and $D_1$ is at most $\varepsilon/3$.*

Assuming Lemma 11.29 and Lemma 11.30 hold, we proceed to prove Lemma 11.22.

*Proof of Lemma 11.22.* From Lemma 11.26, we know that $d_{EM}(D, D_1) \leq \varepsilon/10$. So, the above two lemmas imply that $D$ is $(\varepsilon/3 + \varepsilon/10 + 6\alpha) = 2\varepsilon/3$-close to $\mathcal{P}_{\text{Gap}}$, which contradicts the fact that $D$ is $\varepsilon$-far from $\mathcal{P}_{\text{Gap}}$. This completes the proof of the lemma. □

Now we will prove Lemma 11.29 and Lemma 11.30.

*Proof of Lemma11.29.* We define another distribution $D_3$ over $\{0,1\}^N$ from $D_2^{enc}$ such that $D_3$ is in $\mathcal{P}_{\text{Gap}}$ and $d_{EM}(D_2^{enc}, D_3) \leq 6\alpha$ as follows:

**(i)** $D_3(\mathbf{U}') = \alpha$.

**(ii)** $D_3(\mathbf{X}) = \frac{\alpha}{b}$ for every $\mathbf{X} \in \mathcal{S}'$, $D_3(\mathbf{X}) = \frac{\alpha}{\lceil \log kn \rceil}$ for every $\mathbf{X} \in \mathcal{T}'$.

**(iii)** $D_3(\mathbf{X}) = (1 - 3\alpha) \cdot \frac{D_2^{enc}(\mathbf{X})}{D_2^{enc}(\mathcal{V}')}$ for every $\mathbf{X} \in \mathcal{V}'$.

Recall that $D_2$ is a distribution over $\{0,1\}^n$ that has support size at most $n$. This implies that the set of vectors in $\textsc{Supp}(D_2^{enc}) \setminus \mathcal{U}$ is the encoding of at most $n$ vectors in $\{0,1\}^n$. So, from the definition of $\mathcal{P}_{\text{Gap}}$ and $D_3$, it is clear that $D_3 \in \mathcal{P}_{\text{Gap}}$.

Now we show that the Earth Mover Distance between the distributions $D_3$ and $D_2^{enc}$ is not large.

**Claim 11.31.** *The Earth Mover Distance between $D_2^{enc}$ and $D_3$ is at most $6\alpha$.*

*Proof.* We will bound the Earth Mover Distance between $D_2^{enc}$ and $D_3$ in terms of the variation distance between them as follows:

$$\begin{aligned} d_{EM}(D_2^{enc}, D_3) &\leq \frac{1}{2} \cdot \sum_{\mathbf{X} \in \{0,1\}^N} |D_2^{enc}(\mathbf{X}) - D_3(\mathbf{X})| \\ &= \frac{1}{2} \cdot \sum_{\mathbf{X} \in \mathcal{V}'} |D_2^{enc}(\mathbf{X}) - D_3(\mathbf{X})| + \frac{1}{2} \cdot \sum_{\mathbf{X} \in \{0,1\}^N \setminus \mathcal{V}'} |D_2^{enc}(\mathbf{X}) - D_3(\mathbf{X})| \end{aligned} \tag{11.2}$$

Let us bound the first term as follows:

$$\begin{aligned}
\sum_{\mathbf{X}\in\mathcal{V}'} |D_2^{enc}(\mathbf{X}) - D_3(\mathbf{X})| &= \sum_{\mathbf{X}\in\mathcal{V}'} |(1-3\alpha)\frac{D_2^{enc}(\mathbf{X})}{D_2^{enc}(\mathcal{V}')} - D_2^{enc}(\mathbf{X})| \\
&= \sum_{\mathbf{X}\in\mathcal{V}'} \frac{D_2^{enc}(\mathbf{X})}{D_2^{enc}(\mathcal{V}')}|(1-3\alpha) - D_2^{enc}(\mathcal{V}')| \\
&= \sum_{\mathbf{X}\in\mathcal{V}'} \frac{D_2^{enc}(\mathbf{X})}{D_2^{enc}(\mathcal{V}')}|3\alpha - (1 - D_2^{enc}(\mathcal{V}'))| \\
&\le \sum_{\mathbf{X}\in\mathcal{V}'} 3\alpha\frac{D_2^{enc}(\mathbf{X})}{D_2^{enc}(\mathcal{V}')} \le 3\alpha. \\
&\qquad (\because D_2^{enc}(\mathcal{V}') \ge 1-5\alpha,\ \text{Observation } 11.28)
\end{aligned}$$

From Observation 11.25, $D_2^{enc}(\mathcal{U}) \le 5\alpha$. From the definition of $D_3$, $D_3(\mathcal{U}) = 3\alpha$, we have

$$\sum_{\mathbf{X}\in\{0,1\}^N\setminus\mathcal{V}'} |D_2^{enc}(\mathbf{X}) - D_3(\mathbf{X})| \le 8\alpha.$$

Following Equation 11.2, we conclude that $d_{EM}(D_2^{enc}, D_3) \le 6\alpha$, which completes the proof. □

Since $D_3 \in \mathcal{P}_{\text{Gap}}$, and $d_{EM}(D_2^{enc}, D_3) \le 6\alpha$, we conclude that $D_2^{enc}$ is $6\alpha$-close to $\mathcal{P}_{\text{Gap}}$. □

*Proof of Lemma 11.30.* Recall that the EMD between $D_2^{enc}$ and $D_1$ is the solution to the following LP:

$$\begin{aligned}
\text{Minimize} \quad & \sum_{\mathbf{X},\mathbf{Y}\in\{0,1\}^N} f_{\mathbf{XY}} d_H(\mathbf{X},\mathbf{Y}) \\
\text{Subject to} \quad & \sum_{\mathbf{Y}\in\{0,1\}^N} f_{\mathbf{XY}} = D_2^{enc}(\mathbf{X})\ \forall \mathbf{X}\in\{0,1\}^N \text{ and} \\
& \sum_{\mathbf{X}\in\{0,1\}^N} f_{\mathbf{XY}} = D_1(\mathbf{Y})\ \forall \mathbf{Y}\in\{0,1\}^N.
\end{aligned}$$

Let $f'_{\mathbf{xy}}$ be the flow realizing the EMD between $D_2$ and $D_1^{dec}$. Using $f'$, we now construct a new flow $f^\star$ between $D_2^{enc}$ and $D_1$ as follows:

**(i)** For vectors $\mathbf{X}, \mathbf{Y} \in \mathcal{U}$,

**(a)** If $\mathbf{X} \neq \mathbf{Y}$, then set $f^{\star}_{\mathbf{XY}} = 0$.

**(b)** If $\mathbf{X} = \mathbf{Y}$, then set $f^{\star}_{\mathbf{XY}} = D_2^{enc}(\mathbf{X}) = D_1(\mathbf{Y})$.

**(ii)** For two vectors $\mathbf{X}, \mathbf{Y} \in \mathcal{V}$, we take the vectors $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$ such that $\mathbf{X}, \mathbf{Y} \in \{0,1\}^N$ are their valid encodings (by construction, if $\mathbf{X}$ and $\mathbf{Y}$ are in the support of $D_2$ and $D_1^{enc}$ respectively, such vectors $\mathbf{x}, \mathbf{y}$ exist), and vectors $\mathbf{z}_1, \mathbf{z}_2$ such that $\mathbf{X}_\pi = \text{FE}(\mathbf{z}_1, \mathbf{x})$ and $\mathbf{Y}_\pi = \text{FE}(\mathbf{z}_2, \mathbf{y})$. Now we set the flow as follows:

**(a)** If $\mathbf{z}_1 \neq \mathbf{z}_2$, then set $f^{\star}_{\mathbf{XY}} = 0$.

**(b)** If $\mathbf{z}_1 = \mathbf{z}_2$, then set $f^{\star}_{\mathbf{XY}} = f'_{\mathbf{xy}} \cdot \frac{D_1(\mathbf{Y})}{D_1^{dec}(\mathbf{y})}$.

**(iii)** If one of $\mathbf{X}$ and $\mathbf{Y}$ is in $\mathcal{U}$ and the other one is in $\mathcal{V}$, then $f^{\star}_{\mathbf{XY}} = 0$.

We first argue that the flow $f^{*}_{\mathbf{XY}}$ constructed as above is indeed a valid flow, that is, we have: $\sum_{\mathbf{Y} \in \{0,1\}^N} f^{\star}_{\mathbf{XY}} = D_2^{enc}(\mathbf{X})$ and $\sum_{\mathbf{X} \in \{0,1\}^N} f^{\star}_{\mathbf{XY}} = D_1(\mathbf{Y})$.

To prove $\sum_{\mathbf{Y} \in \{0,1\}^N} f_{\mathbf{XY}} = D_2^{enc}(\mathbf{X})$, first observe that it holds when $\mathbf{X} \in \mathcal{U}$ from (i) and (iii) in the description of $f^{\star}_{\mathbf{XY}}$. Now consider the case where $\mathbf{X} \in \mathcal{V}$. Assume $\mathbf{X}_\pi = \text{FE}(\mathbf{z}, \mathbf{x})$, where $\mathbf{z} \in [n]^m$ and $\mathbf{x} \in \{0,1\}^n$. So, from (ii) in the description of $f^{\star}_{\mathbf{XY}}$, we have

$$\sum_{\mathbf{Y} \in \{0,1\}^N} f^{\star}_{\mathbf{XY}} = \sum_{\mathbf{y} \in \{0,1\}^n} f^{\star}_{\mathbf{X}\text{FE}(\mathbf{z},\mathbf{y})_{\pi^{-1}}} = \sum_{\mathbf{y} \in \{0,1\}^n} f'_{\mathbf{xy}} \frac{D_1(\text{FE}(\mathbf{z},\mathbf{y})_{\pi^{-1}})}{D_1^{dec}(\mathbf{y})} = D_2^{enc}(\mathbf{X}).$$

For $\sum_{\mathbf{X} \in \{0,1\}^N} f^{\star}_{\mathbf{XY}} = D_1(\mathbf{Y})$, consider $\mathbf{Y} \in \mathcal{V}$ for which $\mathbf{Y}_\pi = \text{FE}(\mathbf{z}, \mathbf{y})$ for some $\mathbf{z} \in [n]^m$. Then we have the following:

$$\sum_{\mathbf{X} \in \{0,1\}^N} f^{\star}_{\mathbf{XY}} = \sum_{\mathbf{x} \in \{0,1\}^n} f'_{\mathbf{xy}} \frac{D_1(\text{FE}(\mathbf{z},\mathbf{y})_{\pi^{-1}})}{D_1^{dec}(\mathbf{y})} = D_1^{dec}(\mathbf{y}) \frac{D_1(\text{FE}(\mathbf{z},\mathbf{y})_{\pi^{-1}})}{D_1^{dec}(\mathbf{y})} = D_1(\mathbf{Y}).$$

In the above, we have used the fact that $f'_{\mathbf{xy}}$ is a valid flow from $D_2$ to $D_1^{dec}$.

Now we bound the sum $\sum\limits_{\mathbf{X},\mathbf{Y} \in \{0,1\}^N} f^\star_{\mathbf{X}\mathbf{Y}} d_H(\mathbf{X}, \mathbf{Y})$ below.

$$
\begin{aligned}
& \sum_{\mathbf{X},\mathbf{Y} \in \{0,1\}^N} f^\star_{\mathbf{X}\mathbf{Y}} d_H(\mathbf{X}, \mathbf{Y}) \\
= & \sum_{\mathbf{X},\mathbf{Y} \in \mathcal{V}} f^\star_{\mathbf{X}\mathbf{Y}} d_H(\mathbf{X}, \mathbf{Y}) \quad \text{(From (i) and (iii) in the description of } f^\star\text{)} \\
= & \sum_{\mathbf{x},\mathbf{y} \in \{0,1\}^n} \sum_{\mathbf{z} \in [n]^m} f^\star_{\mathrm{FE}(\mathbf{z},\mathbf{x})_{\pi^{-1}} \mathrm{FE}(\mathbf{z},\mathbf{y})_{\pi^{-1}}} \cdot d_H(\mathrm{FE}(\mathbf{z}, \mathbf{x})_{\pi^{-1}}, \mathrm{FE}(\mathbf{z}, \mathbf{y})_{\pi^{-1}}) \\
& \qquad \text{(From (ii) in the description of } f^\star\text{)} \\
\leq & \sum_{\mathbf{x},\mathbf{y} \in \{0,1\}^n} \sum_{\mathbf{z} \in [n]^m} f^\star_{\mathrm{FE}(\mathbf{z},\mathbf{x})_{\pi^{-1}} \mathrm{FE}(\mathbf{z},\mathbf{y})_{\pi^{-1}}} \cdot d_H(\mathbf{x}, \mathbf{y}) \quad \text{(Observation 11.8 (iii))} \\
= & \sum_{\mathbf{x},\mathbf{y} \in \{0,1\}^n} \sum_{\mathbf{z} \in [n]^m} f'_{\mathbf{x}\mathbf{y}} \frac{D_1(\mathrm{FE}(\mathbf{z}, \mathbf{y})_{\pi^{-1}})}{D_1^{dec}(\mathbf{y})} \cdot d_H(\mathbf{x}, \mathbf{y}) \quad \text{(From (ii) in the description of } f^\star\text{)} \\
= & \sum_{\mathbf{x},\mathbf{y} \in \{0,1\}^n} \left( f'_{\mathbf{x}\mathbf{y}} d_H(\mathbf{x}, \mathbf{y}) \cdot \sum_{\mathbf{z} \in [n]^m} \frac{D_1(\mathrm{FE}(\mathbf{z}, \mathbf{y})_{\pi^{-1}})}{D_1^{dec}(\mathbf{y})} \right) \\
= & D_1(\mathcal{V}') \sum_{\mathbf{x},\mathbf{y} \in \{0,1\}^n} f'_{\mathbf{x}\mathbf{y}} d_H(\mathbf{x}, \mathbf{y}) \quad \text{(By Equation (11.1))} \\
\leq & \sum_{\mathbf{x},\mathbf{y} \in \{0,1\}^n} f'_{\mathbf{x}\mathbf{y}} d_H(\mathbf{x}, \mathbf{y}) \leq \frac{\varepsilon}{3}.
\end{aligned}
$$

The last inequality follows from the fact that $f'$ realizes the assumed EMD between $D_1$ and $D_2^{dec}$. □

### 11.3.3 Near-quadratic lower bound for non-adaptive testing of $\mathcal{P}_{\mathrm{Gap}}$

**Lemma 11.32 (Lower bound on non-adaptive testers).** *Given sample and query access to an unknown distribution $D$, in order to distinguish whether $D$ satisfies $\mathcal{P}_{\mathrm{Gap}}$ or is $\varepsilon$-far from satisfying it, any non-adaptive tester must perform $\widetilde{\Omega}(n^2)$ queries to the samples obtained from $D$, for some $\varepsilon \in (0, 1)$.*

To prove the above lemma, we will construct two hard distributions over distributions, $D_{yes}$ which is supported over $\mathcal{P}_{\mathrm{Gap}}$, and $D_{no}$ which is supported over distributions far from $\mathcal{P}_{\text{Gap}}$, where to distinguish them, any non-adaptive tester must perform $\widetilde{\Omega}(n^2)$ queries. Recall from Theorem 11.11 that $D^{\text{Supp}}_{yes}$ and $D^{\text{Supp}}_{no}$ are two distributions defined over distributions over $\{1, \ldots, 2n\}$, where $D^{\text{Supp}}_{yes}$ provides distributions whose support sizes are $n$, and $D^{\text{Supp}}_{no}$ provides distributions that are $\eta$-far from distributions whose support size is $(1+2\eta)n$, for some constant $\eta \in (0, 1/8)$. We will use these two distributions to construct the hard distributions $D_{yes}$ and $D_{no}$ for the property $\mathcal{P}_{\text{Gap}}$.

**The hard distributions $D_{yes}$ and $D_{no}$:** We describe the distributions $D_{yes}$ and $D_{no}$ over distributions over $\{0,1\}^N$ such that $D_{yes}$ is supported over $\mathcal{P}_{\text{Gap}}$ and $D_{no}$ is supported over distributions that are $\zeta^2 \cdot \eta/5$-far from $\mathcal{P}_{\text{Gap}}$. In what follows, we describe a distribution $D$ ($D = D_{yes}$ or $D = D_{no}$) with $D^{\text{Supp}}$ as parameter, where $D^{\text{Supp}}$ is a distribution defined over distributions over $[2n]$. In particular, $D^{\text{Supp}}$ is either $D^{\text{Supp}}_{yes}$ or $D^{\text{Supp}}_{no}$, where $D = D_{yes}$ when $D^{\text{Supp}} = D^{\text{Supp}}_{yes}$, or $D = D_{no}$ when $D^{\text{Supp}} = D^{\text{Supp}}_{no}$. To generate $D$, we first construct a distribution over distributions $D^0$ as follows. We denote by $\widehat{D}$ the distribution over $\{0,1\}^N$ that we draw according to $D^0$.

**(i)** Set $\widehat{D}(U) = \alpha$, where $\mathbf{U} = \mathbf{1}\mathbf{0}^{N-1}$ is the indicator vector for the index 1.

**(ii)** Take a set of vectors $\mathcal{S} = \{\mathbf{V}_1, \ldots, \mathbf{V}_b\}$ in $\{0,1\}^N$ such that for every $i \in [b]$, the $i$-th vector $\mathbf{V}_i$ is of the form $\mathbf{1}^{i+1}\mathbf{0}^{N-1-i}$. Set $\widehat{D}(\mathbf{V}_i) = \alpha/b$ for every $i \in [b]$.

**(iii)** Take another set of vectors $\mathcal{T} = \{\mathbf{W}_0, \ldots, \mathbf{W}_{\lceil \log kn \rceil - 1}\}$ (disjoint from $\mathcal{S}$) in $\{0,1\}^N$ such that for every $\mathbf{W}_i \in \mathcal{T}$, $\mathbf{W}_i$ is of the form $\mathbf{0}(b(i))(\mathbf{0}^{2^i}\mathbf{1}^{2^i})^{kn/2^{i+1}}$, where $b(i)$ denotes the length $b$ binary representation of $i$. [10] Set $\widehat{D}(\mathbf{W}_i) = \alpha/|\mathcal{T}|$ for every $\mathbf{W}_i \in \mathcal{T}$.

**(iv)** Take a set of vectors $\mathcal{Y} \subseteq \{0,1\}^n$ such that $|\mathcal{Y}| = 2n$, and for any two vectors $\mathbf{y}_i, \mathbf{y}_j \in \mathcal{Y}$, $i \neq j$, $\delta_H(\mathbf{y}_i, \mathbf{y}_j) \geq n/3$. Also, draw a distribution $\widetilde{D}$ over $[2n]$ according to $D^{\text{Supp}}$.

[10] If $kn/2^{i+1}$ is not an integer, we trim the rightmost copy of $\mathbf{0}^{2^i}\mathbf{1}^{2^i}$ so that the total length of "$(\mathbf{0}^{2^i}\mathbf{1}^{2^i})^{kn/2^{i+1}}$" is exactly $kn$.

**(v)** Define $\widehat{D}(\text{FE}(\mathbf{z}, \mathbf{y}_i)) = (1 - 3\alpha)\widetilde{D}(i)/n^m$ for every $i \in [2n]$ and $\mathbf{z} \in [n]^m$, where FE : $[n]^m \times \{0,1\}^n \rightarrow \{0,1\}^N$ is the encoding function from Definition 11.7.

**(vi)** For all other remaining vectors that are not assigned probability mass in the above description, set their probabilities to $0$.

We define $D$ as the process of drawing a distribution $\widehat{D}$ according to $D^0$, and permuting it using a uniformly random permutation $\pi : [N] \rightarrow [N]$.

**Remark 11.2 (Intuition behind the above hard distributions).** Unlike our adaptive algorithm to test $\mathcal{P}_{\text{Gap}}$ (Algorithm 11.2 in Subsection 11.3.2), we can not determine the permutation $\pi$ first, and then perform queries depending on the permutation $\pi$. When the permutation $\pi$ is not known, even if we obtain a sample $\mathbf{X}$ and know that it is equal to FE$(\mathbf{z}, \mathbf{x})_{\pi^{-1}}$ for some $\mathbf{x} \in \{0,1\}^n$ and $\mathbf{z} \in [n]^m$, we can not even decode a single bit of $\mathbf{x}$, unless we query too many of the indices of $\mathbf{X}$. This follows from the properties of our encodings functions SE and GE, used to construct FE (see Lemma 11.9), which "hides" $\mathbf{x}$ inside $\mathbf{X}$. Intuitively, this says that we have to query a quasilinear number of the coordinates of the sample. Since the support estimation problem admits a sample complexity lower bound of $\Omega(n/\log n)$, the non-adaptive query complexity of $\widetilde{\Omega}(n^2)$ follows for non-adaptive algorithms. We will formalize this intuition below.

We will start with the following simple observation.

**Observation 11.33.** The distribution $D_{yes}$ is supported over $\mathcal{P}_{\text{Gap}}$.

*Proof.* From the construction of $D_{yes}$, which is constructed by encoding the elements of the support of the distribution $D_{yes}$ drawn from $D_{yes}^{\text{Supp}}$, it is clear that $D_{yes} \in \mathcal{P}_{\text{Gap}}$. □

Now we show that the distribution $D_{no}$ is supported over distributions that are far from the property $\mathcal{P}_{\text{Gap}}$.

**Lemma 11.34 (Farness lemma).** *$D_{no}$ is supported over distributions that are $\zeta^2 \cdot \eta/5$-far from $\mathcal{P}_{\text{Gap}}$.*

Before directly proceeding to the proof, let us first prove an additional lemma which will be used in the proof of Lemma 11.34.

**Lemma 11.35.** *For any two distinct vectors $\mathbf{X}_1$ and $\mathbf{X}_2$ where $\mathbf{X}_{1,\pi}$, $\mathbf{X}_{2,\pi} \in \text{Supp}(\widehat{D}) \setminus (\{\mathbf{U}\} \cup \mathcal{S} \cup \mathcal{T})$ for $\widehat{D} \in \text{Supp}(D_{no})$, and $\pi$ is the permutation for which $\widehat{D}_\pi \in D^0_{no}$, we have $\delta_H(\mathbf{X}_1, \mathbf{X}_2) \geq \zeta^2 \cdot N/2$.*

*Proof.* We will use the properties of the function FE as mentioned in Observation 11.8. Recall that for a string $\mathbf{z} \in [n]^m$, and a vector $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \{0,1\}^n$, we have $\text{FE}(\mathbf{z}, \mathbf{x}) = \mathbf{0}(\mathbf{1}^b)\text{SE}(\text{GE}(\mathbf{z})_1, \mathbf{x}_1) \ldots \text{SE}(\text{GE}(\mathbf{z})_n, \mathbf{x}_n)$. Now we have the following two cases:

**(a)** Suppose that for some vectors $\mathbf{x} \in \{0,1\}^n$, and $\mathbf{z}_1, \mathbf{z}_2 \in [n]^m$ such that $\mathbf{z}_1 \neq \mathbf{z}_2$, we have $\mathbf{X}_{1,\pi} = \text{FE}(\mathbf{z}_1, \mathbf{x})$ and $\mathbf{X}_{2,\pi} = \text{FE}(\mathbf{z}_2, \mathbf{x})$. Then following Property (i) of FE in Observation 11.8, we know that $\delta_H(\mathbf{X}_1, \mathbf{X}_2) \geq \zeta^2 \cdot N/2$ (noting that permuting the two vectors by the permutation $\pi$ preserves their pairwise distance).

**(b)** Suppose that for some vectors $\mathbf{z} \in [n]^m$, and $\mathbf{x}_1, \mathbf{x}_2 \in \{0,1\}^n$ such that $\mathbf{x}_1 \neq \mathbf{x}_2$, we have $\mathbf{X}_{1,\pi} = \text{FE}(\mathbf{z}, \mathbf{x}_1)$ and $\mathbf{X}_{2,\pi} = \text{FE}(\mathbf{z}, \mathbf{x}_2)$. Then following Property (ii) of FE in Observation 11.8, we know that $\delta_H(\mathbf{X}_1, \mathbf{X}_2) \geq \zeta \cdot \delta_H(\mathbf{x}_1, \mathbf{x}_2)$. From the choice of the vectors $\mathbf{y}_1, \ldots, \mathbf{y}_{2n}$, we know that $\delta_H(\mathbf{x}_1, \mathbf{x}_2) \geq n/3$. Thus, we can say that in this case $\delta_H(\mathbf{X}_1, \mathbf{X}_2) \geq \zeta \cdot nk/3 > \zeta^2 \cdot N/2$ (recalling that $\zeta < 1/2$).

Combining the above, we conclude that $\delta_H(\mathbf{X}_1, \mathbf{X}_2) \geq \zeta^2 \cdot N/2$, for any two distinct vectors $\mathbf{X}_1, \mathbf{X}_2$ as above. □

*Proof of Lemma 11.34.* Suppose that $\widehat{D} \in$ Supp$(D_{no})$, and $\pi$ is the permutation for which $\widehat{D}_\pi \in$ Supp$(D^0_{no})$. We will bound $d_{EM}(\widehat{D}, \mathcal{P}_{\text{Gap}})$. Let us denote the distribution $D_Y \in \mathcal{P}_{\text{Gap}}$ that is closest to $\widehat{D}$, where $\pi_Y$ is the permutation for which $D_{Y,\pi_Y} \in \mathcal{P}^0_{\text{Gap}}$. Let us first define a new distribution $\widetilde{D}_Y$ over $\{0,1\}^N$ as follows:

$$\widetilde{D}_Y(\mathbf{X}) = \begin{cases} \frac{1}{(1-3\alpha)} D_Y(\mathbf{X}) & \mathbf{X}_{\pi_Y} \notin (\{\mathbf{U}\} \cup \mathcal{S} \cup \mathcal{T}) \\ 0 & \text{otherwise} \end{cases}$$

Similarly, we also define another distribution $\widetilde{D}$ from $\widehat{D}$, using $\pi$ instead of $\pi_Y$. Now we have the following claim that bounds the distance between $\widetilde{D}_Y$ and $\widetilde{D}$.

**Claim 11.36.** $d_{EM}(\widetilde{D}, \widetilde{D}_Y) \geq \zeta^2 \cdot \eta/4$.

*Proof.* Following the definition of the property $\mathcal{P}_{\text{Gap}}$, we know that Supp$(\widetilde{D}_Y)$ consists of possible encodings of $n$ distinct vectors from $\{0,1\}^n$, and there are at most $n^m$ valid encodings of every such vector (as per the number of possible vectors $\mathbf{z} \in [n]^m$ that are given as input to GE). This implies that the size of the support of the distribution $\widetilde{D}_Y$ is at most $n^{m+1}$.

Since any distribution in the support of $D_{no}^{\text{Supp}}$ has support size at least $(1+2\eta)n$, following a similar argument as above, we infer that the size of the support of $\widetilde{D}$ is at least $(1+2\eta)n^{m+1}$. Moreover, by Lemma 11.35, we know that any pair of vectors there has distance at least $\zeta^2/2$ (in relative distance). Also, as any vector in the support of any distribution in the support of $D_{no}^{\text{Supp}}$ has probability mass that is multiple of $1/2n$, we infer that every vector in the support of $\widetilde{D}$ has probability mass at least $n^{-m-1}/2$ (as per Item (v) in the definition of $D^0$).

Summing up, we obtain that there are at least $2\eta \cdot n^{m+1}$ vectors in Supp$(\widetilde{D})$ that are $\zeta^2/4$-far (in relative distance) from any vector in Supp$(\widetilde{D}_Y)$, all of whose weights are at least $n^{-m-1}/2$ [11]. Thus, the Earth Mover Distance of $\widetilde{D}$ from $\widetilde{D}_Y$ is at least $\zeta^2 \cdot \eta/4$. □

Recall that we need to bound the distance between $\widehat{D}$ and $D_Y$. From Claim 11.36, we know that $d_{EM}(\widetilde{D}, \widetilde{D}_Y) \geq \zeta^2 \cdot \eta/4$, where the distributions $\widetilde{D}$ and $\widetilde{D}_Y$ are defined over the encoding vectors. From the definition of $\widetilde{D}$ and $\widetilde{D}_Y$ from $\widehat{D}$ and $D_Y$, we conclude that $d_{EM}(D_Y, \widehat{D}) = (1-3\alpha)d_{EM}(\widetilde{D}, \widetilde{D}_Y) \geq \zeta^2 \cdot \eta/5$. □

Now we prove that the distributions $D_{yes}$ and $D_{no}$ remain indistinguishable to any non-adaptive tester, unless it performs $\widetilde{\Omega}(n^2)$ queries. We start with some definitions that will be required for the proof. Recall that $N = \mathcal{O}(n \log n)$.

[11] By the triangle inequality, if we consider a Hamming ball of radius $\zeta^2/4$ around every vector in Supp$(\widetilde{D}_Y)$, there can be at most one vector from Supp$(\widetilde{D})$ inside the ball.

**Definition 11.37 (Large and small query set).** A set of indices $I \subseteq [N]$ is said to be a *large* if $|I| > n/\log^{10} n$. Otherwise, $I$ is said to be a *small*.

Now we show that for a uniformly random permutation $\sigma$, and any $C_j$ as defined in the property $\mathcal{P}_{\text{Gap}}$, with high probability the size of the set of indices $|I \cap \sigma(C_j)|$ will be small, unless $I$ is a large query set.

**Observation 11.38.** Let $\sigma : [N] \to [N]$ be a uniformly random permutation, and $C_j$ correspond to a "bit encoding set" of size $k$ (as per the definition of $\mathcal{P}_{\text{Gap}}$) for an arbitrary $j \in [n]$. For a fixed small query set $I \subseteq [N]$, the probability that $|I \cap \sigma(C_j)|$ is at least $\zeta \cdot k$ is at most $1/n^{10}$.

*Proof.* Let us define a collection of binary random variables $\langle X_i : i \in I \rangle$ such that the following holds:

$$X_i = \begin{cases} 1 & i \in \sigma(C_j) \\ 0 & \text{otherwise} \end{cases}$$

Then as $\sigma$ is a uniformly random permutation, $\mathbb{P}(X_i = 1) = \frac{|\sigma(C_j)|}{N} = \mathcal{O}(\frac{1}{n})$ for any $i \in [n]$. Now let us define another random variable $X = \sum_{i=1}^{n} X_i$. Noting that $X = |I \cap \sigma(C_j)|$, we obtain $\mathbb{E}[X] = \mathcal{O}(1/\log^{10} n)$. By applying Hoeffding's bound for sampling without replacement (Lemma 2.15), we can say that $\mathbb{P}(X \geq \zeta \cdot k) \leq 1/n^{10}$. This completes the proof. □

Now let us define an event $\mathcal{E}_{I,j}$ as follows:

$$\mathcal{E}_{I,j} := \text{The query set } I \text{ satisfies } |I \cap \sigma(C_j)| \leq \zeta \cdot k.$$

Now we are ready to prove that unless $\widetilde{\Omega}(n^2)$ queries are performed, no non-adaptive tester can distinguish $D_{yes}$ from $D_{no}$.

**Lemma 11.39 (Indistinguishibility lemma).** *With probability at least $2/3$, in order to distinguish $D_{yes}$ from $D_{no}$, $\widetilde{\Omega}(n^2)$ queries are necessary for any non-adaptive tester.*

*Proof.* From our result on the adaptive $\varepsilon$-tester for $\mathcal{P}_{\text{Gap}}$, we know that $\widetilde{\mathcal{O}}(n)$ queries are sufficient for adaptively testing $\mathcal{P}_{\text{Gap}}$. Without loss of generality, let us assume that the non-adaptive tester takes at most $n^2$ samples from the unknown distribution $D$ (since we can assume that at least one query is performed in every sample). As per the definition of a non-adaptive tester, assume that the samples taken are $\mathbf{X}_1, \ldots, \mathbf{X}_s$, and their respective query sets are $I_1, \ldots, I_s$ for some integer $s$.

Consider an event $\mathcal{E}$ as follows:

$\mathcal{E}$ := For every $\ell \in [s]$ for which $I_\ell$ is small and every $j \in [n]$, the event $\mathcal{E}_{I_\ell, j}$ occurs.

Since the non-adaptive tester takes at most $n^2$ samples, there can be at most $n^2$ samples for which a small set was queried, that is, $s \leq n^2$. Moreover, there are $n$ possible sets $C_j$ present in a sample. Using the union bound, along with Observation 11.38, we can say that the event $\mathcal{E}$ holds with probability at least $1 - 1/n^7$. Given that the event $\mathcal{E}$ holds, we will now show that the induced distributions of $D_{yes}$ and $D_{no}$ on small query sets are identical and independent of the samples with large query sets.

**Claim 11.40.** *Assume that the event $\mathcal{E}$ holds. Then a non-adaptive tester that uses at most $o(n/\log n)$ large query sets, can not distinguish $D_{yes}$ from $D_{no}$ with probability more than* $1/4$.

*Proof.* Since the distributions produced by $D_{yes}$ and $D_{no}$ are identical over the respective permutations of $(\{\mathbf{U}\} \cup \mathcal{S} \cup \mathcal{T})$, it is sufficient to prove indistinguishability over the restrictions to the valid encodings of $\mathbf{y}_1, \ldots, \mathbf{y}_{2n}$ (as they appear in the definition of $D^0$). Furthermore, we argue that this claim holds even if for every large query set, the tester is provided with the entire vector that was sampled.

Given that the event $\mathcal{E}$ holds, regardless of whether the distribution was produced by $D_{yes}$ or $D_{no}$, the restriction of the samples to the small queried sets are completely uniformly distributed, even when conditioned on the samples with large query sets (which are taken independently of them). Thus we may assume that all samples with small query sets are ignored by the tester, since the answers to these queries can be simulated without taking any samples at all.

Finally, we appeal to the construction of the hard distributions $D_{yes}$ and $D_{no}$ from

$D_{yes}^{\text{Supp}}$ and $D_{no}^{\text{Supp}}$. By Theorem 11.11, the distance between these two distributions over the sample sequence is at most $1/4$, unless there were more than $o(n/\log n)$ samples with large sets. This completes the proof of the claim. □

Combining Claim 11.40 with the above bound on the probability of the event $\mathcal{E}$, we conclude that $\widetilde{\Omega}(n^2)$ queries are necessary for any non-adaptive tester to distinguish $D_{yes}$ from $D_{no}$ with probability at least $2/3$, that is, with a probability difference of at least $1/3$. This concludes the proof of the lemma. □

# Part III

# Results in the Adjacency matrix Model

# Chapter 12

# Testing in the Adjacency matrix Model

## 12.1 Introduction

In this part of the thesis, we study property testing of some graph properties in the dense graph model. Let the unknown graph be $G(V, E)$, with $V$ and $E$ being the set of vertices and edges of $G$ respectively. Recall that in this model, $G$ is represented as an adjacency matrix $M$, and the tester can perform edge-existence query to any entry of $M$. A graph $G$ is said to be $\varepsilon$-far from some property $\mathcal{P}$ if one needs to modify at least $\varepsilon n^2$ entries of $M$. This model was introduced in the seminal work of Goldreich, Goldwasser and Ron [GGR98]. Since then there have been several works in this model.

Here we will study two interesting problems in the dense graph model: (i) the problem of tolerant testing of graph isomorphism, and (ii) the problem of tolerant bipartiteness testing. Below we give introductions to both these problems. Later in Section 12.2, we will state our results, and we will formally prove these results in Chapter 13, Chapter 14 and Chapter 15.

### 12.1.1 Tolerant Graph Isomorphism Testing

Graph isomorphism (GI) has been one of the most celebrated problems in computer science. Roughly speaking, the graph isomorphism problem asks whether two graphs are structure-preserving. Namely, given two graphs $G_u$ and $G_k$, graph isomorphism of $G_u$ and $G_k$ is a bijection $\psi : V(G_u) \to V(G_k)$ such that for all pair of vertices $u, v \in V(G_u)$, the edges $\{u, v\} \in E(G_u)$ if and only if $\{\psi(u), \psi(v)\} \in E(G_k)$ [1]. One central open problem in complexity theory is whether the graph isomorphism problem can be solved in polynomial time. Recently in a breakthrough result, Babai [Bab16] proved that the graph isomorphism problem could be decided in quasi-polynomial time.

For a central problem like the graph isomorphism, naturally, one would like to understand its (and related problems) computational complexity for various models of computation. While most of the focus has been on the standard time complexity in the RAM model for various classes of graphs (and hyper-graphs), other complexity measures like space complexity, parameterized complexity, and query complexity have also been studied over the past few decades (see the Dagstuhl Report [BDST15] and PhD thesis of Sun [Sun16]).

A natural extension of the GI problem is to estimate the "graph isomorphism distance" between two graphs. In other words, given two graphs $G_u$ and $G_k$, what fraction of edges are necessary to add or delete to make the graphs isomorphic.

**Definition 12.1.** Let $G_u = (V_u, E_u)$ and $G_k = (V_k, E_k)$ be two graphs with $|V_u| = |V_k| = n$. Given a bijection $\phi : V_u \to V_k$, the distance between the graphs $G_u$ and $G_k$ with respect to the bijection $\phi$ is

$$d_\phi(G_u, G_k) := \left|\{(u, v) : \text{Exactly one among } (u, v) \in E_u \text{ or } (\phi(u), \phi(v)) \in E_k \text{ holds}\}\right|.$$

The Graph Isomorphism Distance (or GI-distance in short) between graphs $G_u$ and $G_k$ is defined as $\min\limits_{\phi : V_u \to V_k} d_\phi(G_u, G_k)/n^2$, and is denoted by $\delta_{GI}(G_u, G_k)$ (we will use $d(G_u, G_k)$ to mean $n^2 \delta_{GI}(G_u, G_k)$).

The problem of computing GI-distance between two graphs is known to be $\#P$-hard [Lin94]. The next natural question is:

*What is the complexity for approximating (either by a constant additive or*

[1] In a graph $G$, $V(G)$ and $E(G)$ denote the sets of vertices and edges in $G$, respectively.

*multiplicative factor) the graph isomorphism distance between two graphs?*

In [Lin94], it was also proven that the problem of computing GI-distance between two graphs is APX-hard. So, approximating $\delta_{GI}(G_u, G_k)$ up to a constant multiplicative factor is NP-hard. In this chapter, we study this problem of approximating (up to a constant additive factor) the GI-distance between two graphs in the query model (see Section 12.1.1).

## Query Complexity of Graph Isomorphism

Formally speaking, the main problem is: given two graphs $G_u$ and $G_k$ and an approximation parameter $\zeta \in (0, 1)$, the goal is to output an estimate $\alpha$ such that

$$\delta_{GI}(G_u, G_k) - \zeta \leq \alpha \leq \delta_{GI}(G_u, G_k) + \zeta.$$

In the query model, the problem is equivalent (up to a constant factor) to the tolerant property testing of graph isomorphism in the dense graph model (introduced in the work of Parnas, Ron and Rubinfeld [PRR06]). For $0 \leq \gamma < 1$, two graphs $G_u$ and $G_k$, with $n$ vertices, are called $\gamma$*-close* or $\gamma$*-far to isomorphic*[2] if $d(G_u, G_k) \leq \gamma n^2$ or $d(G_u, G_k) \geq \gamma n^2$, respectively. In $(\gamma_1, \gamma_2)$*-tolerant GI testing*, we are given two graphs $G_u$ and $G_k$, and two parameters $0 \leq \gamma_1 < \gamma_2 \leq 1$, with the guarantee that either the graphs are $\gamma_1$-close or $\gamma_2$-far. One of the graphs (usually denoted as $G_u$) is accessed by querying the entries of its adjacency matrix. In contrast, the other graph (usually denoted as $G_k$ [3]) is known to the query algorithm, and no cost for accessing the entries of the adjacency matrix of $G_k$ is incurred. The query complexity is the number of queries (to the adjacency matrix of $G_u$) that are required for testing, (with correctness probability at least $2/3$ [4]), whether $G_u$ and $G_k$ are $\gamma_1$-close or $\gamma_2$-far. The query algorithm is assumed to have unbounded computational power.

[2] As a shorthand, rather than saying $\gamma$-close or $\gamma$-far to isomorphic, we will just say $\gamma$*-close* or $\gamma$*-far* respectively.

[3] $G_u$ and $G_k$ denote the unknown and known graphs, respectively.

[4] The correctness probability can be made any $1 - \delta$ by incurring a multiplicative factor of $O(\log \frac{1}{\delta})$ in the query complexity.

The non-tolerant property testing version of the graph isomorphism problem (that is, when $\gamma_1 = 0$) was first studied by Fischer and Matsliah [FM08] and subsequently, Babai and Chakraborty [BC10] studied the non-tolerant property testing version of the hypergraph isomorphism problem. Recently, the non-tolerant testing of GI has been considered in various other models (like Goldreich [Gol19] studied the problem for the *bounded degree graph model* of property testing and Levi and Medina [LM20] considered the problem in the *distributed* setting). However, the tolerant version of the problem remains elusive and it is surprising that the tolerant version of a fundamental problem like graph isomorphism (in query model) is not addressed in the literature, though the non-tolerant version of GI testing problem has been resolved more than a decade ago in [FM08] (when one graph is unknown). On a different note, there are also studies of non-tolerant version of graph isomorphism testing in the literature when both the graphs are unknown [FM08, OS18]. We will not discuss much about that case as the main focus of this work is different.

Before proceeding further, we want to note that there is a simple algorithm with query complexity $\widetilde{\mathcal{O}}(n)$ for tolerant testing of graph isomorphism (when one of the graphs is known in advance). Basically, one goes over all possible $n!$ bijections $\phi : V_u \to V_k$ and estimates the distance between $G_u$ and $G_k$ with respect to the permutation. The samples may be reused[5], and hence we have the following observation.

**Observation 12.2.** Given a known graph $G_k$ and an unknown graph $G_u$ and any approximation parameter $\zeta \in (0, 1)$, there is a query algorithm that makes $\widetilde{\mathcal{O}}\left(n\right)$ queries and outputs a number $\alpha$ such that, with probability at least $\frac{2}{3}$, the following holds:

$$\delta_{GI}(G_u, G_k) - \zeta \leq \alpha \leq \delta_{GI}(G_u, G_k) + \zeta.$$

But obtaining a lower bound matching (at least up to a polylog factor) the upper bound of Observation 12.2 is not at all obvious. The main contribution here is to show an equivalence between tolerant testing of graph isomorphism and tolerant EMD testing between multi-sets (in the query setting).

[5] If the samples are $\Theta(\log(n!))$, then the error probability can be bounded using the union bound.

Like many other property testing problems, the core difficulty in the testing of GI is understanding certain properties of distributions. In the case of the non-tolerant version of GI, it has been shown in [FM08] that the core problem is testing the variation distance between two distributions. Their upper bound result can be restated as: if there is a property testing algorithm, with query complexity $q(n)$ for testing equivalence between two distributions, on support size $n$ [6], then GI can be tested using $\widetilde{\mathcal{O}}(q(n))$ queries, where the tilde hides a polylogarithmic factor of $n$ (number of vertices). And since the query complexity for testing identity of distributions (from [BFF+01], [Pan08], [ADK15], [VV17a]) is known to be $\mathcal{O}(\sqrt{n}/\varepsilon^2)$, the query complexity for non-tolerant GI-testing is $\widetilde{\mathcal{O}}(\sqrt{n})$.

In the lower bound proof of [FM08], there is no direct reduction of the graph isomorphism problem to the variation distance problem. But it is important to note that lower bound proofs for both of these problems use the tightness of the *birthday paradox*. So, in some sense, one can say that the heart of the non-tolerant testing of GI is in testing variation distance between two distributions. In our work, the main contribution is to show a unified connection between graph isomorphism testing and Earth Mover distance testing which holds across computational models, like query as well as communication models.

### 12.1.2 Tolerant Bipartiteness Testing

In the work of Goldreich, Goldwasser and Ron [GGR98], the authors studied various interesting and important problems in dense graphs and testing bipartiteness was one of them. Given a dense graph $G$ as an input, the problem is to decide if $G$ is bipartite, or we need to modify at least $\varepsilon n^2$ entries of the adjacency matrix of $G$ to make it bipartite, using as few queries to the adjacency matrix of $G$ as possible, where $\varepsilon \in (0, 1)$ is a proximity parameter.

Due to the fundamental nature of the problem, *bipartite testing* has been extensively studied over the past two decades [GGR98]. Though there are several works on non-

[6] Testing identity between two distributions means to test if the unknown distribution (from where the samples are drawn) is identical to the known distribution or if the variation distance between them more than $\epsilon$.

tolerant testing of various graph properties across all models in graph property testing [GGR98, GR99, CMOS19], there are very few works related to their tolerant counterparts (See Goldreich [Gol17], Bhattacharyya and Yoshida [BY22] for an extensive list of various results).

In [GGR98], the authors studied the problem of estimating the size of the maximum cut of a dense graph, and later studied the more general graph partition problem. The authors of [GGR98] presented an algorithm of estimating the size of the maximum cut that uses $poly(1/\varepsilon)$ queries. Note that the maxcut estimation algorithm of [GGR98] provides an algorithm of tolerant bipartiteness testing, with the same number of queries. However, in this work, we improve their result by designing a more efficient algorithm.

Now we formally define the notion of *bipartite distance* and state our main result. Then we discuss our result vis-a-vis the related works.

**Definition 12.3** (Bipartite distance)**.** A *bipartition* of (the vertices of) a graph $G$ is a function $f : V(G) \to \{L, R\}$ [7]. The *bipartite* distance of $G$ with respect to the bipartition $f$ is denoted and defined as

$$d_{bip}(G, f) := \left[ \sum_{v \in V : f(v) = L} \left| N(v) \cap f^{-1}(L) \right| + \sum_{v \in V : f(v) = R} \left| N(v) \cap f^{-1}(R) \right| \right].$$

Here $N(v)$ denotes the neighborhood of $v$ in $G$. Informally, $d_{bip}(G, f)$ measures the distance of the graph $G$ from being bipartite, with respect to the bipartition $f$. The *bipartite distance* of $G$ is defined as the minimum bipartite distance of $G$ over all possible bipartitions $f$ of $G$, that is,

$$d_{bip}(G) := \min_{f} d_{bip}(G, f).$$

Now we are ready to formally state our result.

**Theorem 12.4** (Main result)**.** *Given query access to the adjacency matrix of a dense graph $G$ with $n$ vertices and a proximity parameter $\varepsilon \in (0, 1)$, there exists an algorithm*

[7] $L$ and $R$ denote left and right respectively.

*that, with probability at least $\frac{9}{10}$, decides whether $d_{bip}(G) \leq \varepsilon n^2$ or $d_{bip}(G) \geq (2 + \Omega(1))\varepsilon n^2$, by sampling $\mathcal{O}\left(\frac{1}{\varepsilon^3}\log\frac{1}{\varepsilon}\right)$ vertices in $2^{\mathcal{O}\left(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\right)}$ time, and makes $\mathcal{O}\left(\frac{1}{\varepsilon^3}\log^2\frac{1}{\varepsilon}\right)$ queries.*

## Our result in the context of literature

Non-tolerant bipartite testing refers to the problem where we are given query access to the adjacency matrix of an unknown graph $G$ and a proximity parameter $\varepsilon \in (0, 1)$, and the objective is to decide whether $d_{bip}(G) = 0$ or $d_{bip}(G) \geq \varepsilon n^2$. The problem of non-tolerant bipartite testing in the dense graph model was first studied in the seminal work of Goldreich, Goldwasser and Ron [GGR98], and they showed that it admits an algorithm with query complexity $\widetilde{\mathcal{O}}\left(1/\varepsilon^3\right)$. Later, Alon and Krivelevich [AK02] improved the query complexity of the problem to $\widetilde{\mathcal{O}}\left(1/\varepsilon^2\right)$. They further studied the problem of testing $c$-colorability of dense graph. Note that bipartite testing is a special case of testing $c$-colorability, when $c = 2$. They proved that $c$-colorability can be tested by performing $\widetilde{\mathcal{O}}\left(1/\varepsilon^4\right)$ queries, for $c \geq 3$. This bound was later improved to $\widetilde{\mathcal{O}}\left(1/\varepsilon^2\right)$ by Sohler [Soh12]. On the other hand, for non-tolerant bipartite testing, Bogdanov and Trevisan [BT04] proved that $\Omega(1/\varepsilon^2)$ and $\Omega(1/\varepsilon^{3/2})$ adjacency queries are required by any non-adaptive and adaptive testers, respectively. Later, Gonen and Ron [GR07] further explored the power of adaptive queries for bipartiteness testing. Bogdanov and Li [BL10] showed that bipartiteness can be tested with one-sided error in $\mathcal{O}(1/\varepsilon^c)$ queries, for some constant $c < 2$, assuming a conjecture [8].

Though the non-tolerant variant of bipartite testing is well understood, the query complexity of tolerant version (even for restricted cases such as in Theorem 15.1) is not completely settled. Goldreich, Goldwasser and Ron [GGR98] proved that MAXCUT [9] can be estimated with an additive error of $\varepsilon n^2$ by performing $\widetilde{\mathcal{O}}(1/\varepsilon^7)$ queries and in time $2^{\widetilde{\mathcal{O}}(1/\varepsilon^3)}$. As stated before, this implies that the bipartite distance of a (dense) graph $G$ can be estimated upto an additive error of $\varepsilon n^2$, by performing $\widetilde{\mathcal{O}}(1/\varepsilon^7)$ queries and

[8] The conjecture is stated as follows: if the graph $G$ is $\varepsilon$-far from being bipartite, then the induced subgraph of $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$ vertices of $G$ would be $\widetilde{\Omega}(\varepsilon)$-far from being bipartite.

[9] MAXCUT of a graph $G$ denotes the size of the largest *cut* in $G$.

in time $2^{\widetilde{\mathcal{O}}(1/\varepsilon^3)}$. Later Alon, Vega, Kannan and Karpinski [AdlVKK03] designed an improved algorithm for estimating the size of MAXCUT that performs $\widetilde{\mathcal{O}}\left(1/\varepsilon^6\right)$ queries. Note that this implies an algorithm of estimating the bipartite distance, with similar queries (see Section 15.2 for details, and in particular, see Corollary 15.4). Even for the tolerant version that we consider in Theorem 15.1, their algorithm does not give any bound better than $\widetilde{\mathcal{O}}\left(1/\varepsilon^6\right)$. In this work, we improve the bound for tolerant bipartite testing (for the restricted case as stated in Theorem 15.1) substantially from the work of Alon et al. [AdlVKK03] by designing an algorithm that performs only $\widetilde{\mathcal{O}}\left(1/\varepsilon^3\right)$ queries, and in $2^{\widetilde{\mathcal{O}}(1/\varepsilon)}$ time.

Apart from the dense graph model, this problem has also been studied in other models of property testing. Goldreich and Ron [GR99] studied the problem of bipartiteness testing for bounded degree graphs, where they gave an algorithm of $\widetilde{\mathcal{O}}(\sqrt{n})$ queries, where $n$ denotes the number of vertices of the graph. They also proved a similar lower bound of bipariteness testing of $\Omega(\sqrt{n})$ queries [GR97] in the bounded degree model. Later, Kaufman, Krivelevich and Ron [KKR04] studied the problem in the general graph model and gave an algorithm with query complexity $\widetilde{\mathcal{O}}(\min(\sqrt{n}, n^2/m))$, where $m$ denotes the number of edges of the graph. Few years back, Czumaj, Monemizadeh, Onak and Sohler [CMOS19] studied the problem for planar graphs (more generally, for any minor-free graph), where they employed random walk based techniques, and proved that constant number of queries are enough for the same. Apart from bipartite testing, there have been extensive works related to property testing in the dense graph model and its connection to the regularity lemma [AFNS09, AFKS00, FN07].

## 12.2 Our results

In this section, we present our results of this part of the thesis. Here we will be considering all the distance measures in absolute distance instead of normalized distances. We will start with our result on tolerant graph isomorphism testing.

## Tolerant Isomorphism Testing

One of our main technical result of this part of the thesis is that we prove estimating GI-distance is as hard as tolerant EMD testing over multi-sets with the access of samples **without** replacement over the unknown multi-set $S_u$, ignoring polynomial factors of $\log n$.

**Theorem 12.5.** *Let $G_k$ and $G_u$ denote the known and the unknown graphs on $n$ vertices, respectively, and $Q_{GI}(G_u, G_k)$ denotes the number of adjacency queries to $G_u$, required by the best algorithm that takes two constants $\gamma_1, \gamma_2$ with $0 \leq \gamma_1 < \gamma_2 \leq 1$ and decides whether $d(G_u, G_k) \leq \gamma_1 n^2$ or $d(G_u, G_k) \geq \gamma_2 n^2$ with probability at least $\frac{2}{3}$. Then*

$$Q_{GI}(G_u, G_k) = \widetilde{\Theta}\left(\text{QWoR}_{\textbf{EMD}}(n)\right)$$

*where $\widetilde{\Theta}(\cdot)$ hides polynomial factors in $\frac{1}{\gamma_2 - \gamma_1}$ and $\log n$.*

This gives us a geometric approach for solving the graph isomorphism testing problem. Thus improving the bound of $\text{QWoR}_{\textbf{EMD}}(n)$ would directly provide us a better bound on $Q_{GI}(G_u, G_k)$.

On the other hand, extending the lower bound of $\text{QWR}_{\textbf{EMD}}(n)$ to $\text{QWoR}_{\textbf{EMD}}(n)$ would give us a better lower bound on $Q_{GI}(G_u, G_k)$. However, the difference between sampling **with** and **without** replacement is much more subtle. Freedman [Fre77] has shown the difference when we sample elements **with** replacement from a set and that **without** replacement from the same set. However, when the number of samples is $o(\sqrt{n})$, the distribution of answers to the queries when samples are drawn **with** replacement is very close (in $\ell_1$ distance) to the distribution of answers to the queries when samples are drawn **without** replacement. Thus, following the simulation of samples **with** replacement using samples **without** replacement (stated formally in Proposition 12.15) along with Theorem 12.5, we can get an alternative proof of the following lower bound proved by Fischer and Matsliah [FM08].

**Corollary 12.6** (Fischer and Matsliah [FM08])**.** *There exists a constant $\zeta \in (0, 1)$ such that any query algorithm that decides, with probability at least $2/3$, if a known graph*

*$G_k$ and an unknown graph $G_u$ is isomorphic or $\gamma$-far from isomorphic, with $\gamma \leq \zeta$, must make $\Omega(\sqrt{n})$ queries.*

Our proof of Theorem 12.5 has two parts: for the lower bound, we reduce tolerant testing of EMD of multi-sets over the Hamming cube using samples **without** to tolerant graph isomorphism testing. For the upper bound, we reduce from tolerant graph isomorphism to tolerant testing of EMD of multi-sets over the Hamming cube using samples **without** replacement.

Now we will state our result on tolerant bipartiteness testing below.

### Tolerant Bipartiteness Testing:

**Theorem 12.7.** *There exists an algorithm* TOL-BIP-DIST$(G, \varepsilon)$ *that given adjacency query access to a dense graph $G$ with $n$ vertices and a parameter $\varepsilon \in (0,1)$, decides with probability at least $\frac{9}{10}$, whether $d_{bip}(G) \leq \varepsilon n^2$ or $d_{bip}(G) \geq (2+k)\varepsilon n^2$, by sampling $\mathcal{O}(\frac{1}{k^5\varepsilon^2}\log\frac{1}{k\varepsilon})$ vertices in $2^{\mathcal{O}(\frac{1}{k^3\varepsilon}\log\frac{1}{k\varepsilon})}$ time, using $\mathcal{O}(\frac{1}{k^8\varepsilon^3}\log^2\frac{1}{k\varepsilon})$ queries to the adjacency matrix of $G$.*

### Organization of the part

We prove Theorem 12.5 in two parts. We prove the lower bound part (tolerant graph isomorphism is as hard as tolerant EMD testing) and upper bound part (tolerant EMD testing is as hard as tolerant graph isomorphism) of Theorem 12.5 in Chapter 13 and Chapter 14 respectively. We will prove Theorem 12.7 in Chapter 15.

## 12.3 Preliminaries

All graphs considered here are undirected, unweighted and have no self-loops or parallel edges. For a graph $G(V, E)$, $V(G)$ and $E(G)$ will denote the vertex set and the edge set of $G$, respectively. Since we are considering undirected graphs, we write an edge $(u, v) \in E(G)$ as $\{u, v\}$. $N_G(v)$ denotes the neighborhood of $v$ in $G$, and we will write it as $N(v)$ when the graph $G$ is clear from the context. Finally, $a = (1 \pm \varepsilon)b$ represents $(1-\varepsilon)b \leq a \leq (1+\varepsilon)b$.

### 12.3.1 Notion of distance between two graphs

First let us define the notion of DECIDER of a vertex and then the notion of distance between two graphs, using decider of vertices, that is conceptually same as that of GRAPH ISOMORPHISM DISTANCE defined in Definition 12.1.

**Definition 12.8.** (DECIDER of a vertex) Given two graphs $G_k$ and $G_u$ and a bijection $\phi : V(G_u) \to V(G_k)$, DECIDER of a vertex $x \in V(G_u)$ with respect to $\phi$ is defined as the set of vertices of $G_u$ that create the edge difference in $x$ and $\phi(x)$'s neighbourhood in $G_u$ and $G_k$, respectively. Formally,

$$\textsc{Decider}_{\phi}(x) := \{y \in V(G_u) : \text{one of the edges } \{x,y\} \,\&\, \{\phi(x),\phi(y)\} \text{ is not present}\}$$

**Definition 12.9.** (DISTANCE between two graphs) Let $G_u$ and $G_k$ be two graphs and $\phi : V(G_u) \to V(G_k)$ be a bijection from the vertex set of $G_u$ to that of $G_k$. The *distance* between $G_u$ and $G_k$ under $\phi$ is defined as the sum of the sizes of the deciders of all the vertices in $G_u$, that is,

$$d_{\phi}(G_u, G_k) := \sum_{x \in V(G_u)} \left|\textsc{Decider}_{\phi}(x)\right|.$$

The *distance* between two graphs $G_u$ and $G_k$ is the minimum distance under all possible bijections $\phi$ from $V(G_u)$ to $V(G_k)$, that is, $d(G_u, G_k) := \min_{\phi} d_{\phi}(G_u, G_k)$.

**Remark 12.1.** Recall the definition of $\delta_{GI}(G_u, G_k)$, GRAPH ISOMORPHISM DISTANCE between $G_u$ and $G_k$ (Definition 12.1). Observe that $d(G_u, G_k) = 2\binom{n}{2}\delta_{GI}(G_u, G_k)$. Though, $d(G_u, G_k)$ and $\delta_{GI}(G_u, G_k)$ represent the same thing, conceptually, we will do our calculations by using $d(G_u, G_k)$ for simplicity of presentation.

Next we define the concept of closeness between two graphs.

**Definition 12.10.** (CLOSE and FAR) For $\gamma \in [0, 1)$, two graphs $G_u$ and $G_k$ with $n$ vertices are *$\gamma$-close to isomorphic* if $d(G_u, G_k) \leq \gamma n^2$. Otherwise, we say $G_u$ and $G_k$ are *$\gamma$-far from being isomorphic.* [10]

## Property Testing of Distribution Properties

Understanding different properties of probability distributions have been an active area of research in property testing (For reference, see [Can20c]). The authors studied these problems assuming random sample access from the unknown distributions. Considering the relation between the distributions and their corresponding representative multi-sets, we can say that all these results hold for multi-sets along with access over sampling **with** replacement.

Although it seems that the change of query model from sample **with** replacement to sample **without** replacement does not make much difference, following the work of Freedman [Fre77], we know that the variation distance between probability distributions when accessed via samples **with** and **without** replacement, becomes arbitrary close to $1/2$ when the number of samples is $\Omega(\sqrt{n})$. Because of this reason, many techniques developed for sampling **with** replacement for various problems no longer work anymore. Most importantly, proving any lower bound better than $\Omega(\sqrt{n})$ is often nontrivial.

### 12.3.2 Some results on Earth Mover Distance (EMD)

In this subsection, we study some properties of *Earth Mover* distance (EMD) over probability distributions and multi-sets, which are crucial in the context of both our lower and upper bound. Let $H = \{0, 1\}^n$ be a Hamming cube of dimension $n$, and $p, q$ be two probability distributions on $H$. Recall that the *Earth Mover Distance* between $p$ and $q$ is denoted by $d_{EM}(p, q)$ and defined as the optimum solution to the following linear

[10] By abuse of notation, we will say $G_u$ and $G_k$ are $\gamma$-far when $d(G_u, G_k) \geq \gamma n^2$.

program:

$$\begin{aligned}
\text{Minimize} \sum_{i,j \in H} & f_{ij} d_H(i,j) & \\
\text{Subject to} \sum_{j \in H} & f_{ij} = p(i), & \forall\, i \in H \\
\sum_{i \in H} & f_{ij} = q(j), & \forall\, j \in H \\
0 \leq & f_{ij} \leq 1, & \forall\, i,j \in H
\end{aligned}$$

Earth Mover Distance (EMD) is a fundamental metric over the space of distributions supported on a fixed metric space. Estimating EMD between two distributions, up to a multiplicative factor, has been extensively studied in mathematics and computer science. It is closely related to the embedding of the EMD metric into a $\ell_1$ metric. Even the problem of estimation of EMD between distributions up to an additive factor has been well studied, for reference see [DBNNR11], [SP18]. The hardness of estimating EMD between distributions depends heavily on the structure of the domain on which the distributions are supported. In [DBNNR11], the authors have proved a lower bound of $\Omega((\Delta/\varepsilon)^d)$ on the query complexity for estimating (up to an additive error of $\varepsilon$) EMD between two distributions supported on the real cube $[0,\Delta]^d$. At the same time, it is not hard to see that if the support has certain structures, estimating EMD may be easy. In this chapter, we focus on the estimation of EMD between two distribution when the metric space is the Hamming cube.

A standard way to think of sampling from any probability distribution is to consider it as a multi-set of elements with appropriate multiplicities, and samples are drawn **with** replacement from that multi-set. While estimating EMD between two multi-sets, although the most natural way to access the unknown multi-set is sampling **with** replacement, we introduce the problem of tolerant EMD testing over multi-sets with the access of samples **without** replacement as follows:

**Definition 12.11** (EMD between two multi-sets)**.** Let $S_1, S_2$ be two multi-sets on a Hamming cube $H = \{0,1\}^d$ of dimension $d$ with $|S_1| = |S_2|$. The EMD between $S_1$ and

$S_2$ is denoted by $d_{EM}(S_1, S_2)$ and defined as $d_{EM}(S_1, S_2) = \min\limits_{\phi: S_1 \to S_2} \sum\limits_{x \in S_1} d_H(x, \phi(x))$ where $\phi$ is a bijection from $S_1$ to $S_2$.

Note that an unknown distribution $p$ is accessed by taking samples from $p$. However, a multi-set is accessed as follows:

**Definition 12.12** (Query accesses to multi-sets)**.** A multi-set $S$ of $n$ elements is accessed in one of the following ways:

**Sample Access with replacement:** Each element of $S$ is reported uniformly at random independent of all previous queries.

**Sample Access without replacement:** Let us assume we make $Q$ queries to $S$, where $Q \leq n$. The answer to the first query, say $s_1$, is an element from $S$ chosen uniformly at random. For any $2 \leq i \leq Q$, the answer of the $i$-th query is an element chosen uniformly at random from $S \setminus \{s_1, \ldots, s_{i-1}\}$. Here $s_j, 1 \leq j \leq Q$, denotes the answer to the $j$-th query.

**Example of Sampling with & without replacement:** Consider the following example. Let $S = \{1, 2, 2, 3, 4\}$ be a multi-set, and it corresponds to a distribution $P$. Thus $P(1) = 0.2, P(2) = 0.4, P(3) = 0.2$ and $P(4) = 0.2$. Suppose we have obtained a sample from $P$, and the obtained sample is 2. Now we want to take another sample from $P$. Consider the following two scenarios:

**Sampling with replacement:** The probability of obtaining 2 as the second sample remains same as before, that is, with probability $0.4$, 2 appears as the new sample.

**Sampling without replacement:** Here the underlying multi-set no longer remains the same. After getting 2 as the first sample, the multi-set as well as the distribution have been changed. The new multi-set is $S' = \{1, 2, 3, 4\}$. Thus the probability that 2 appears in the second sample is $0.25$.

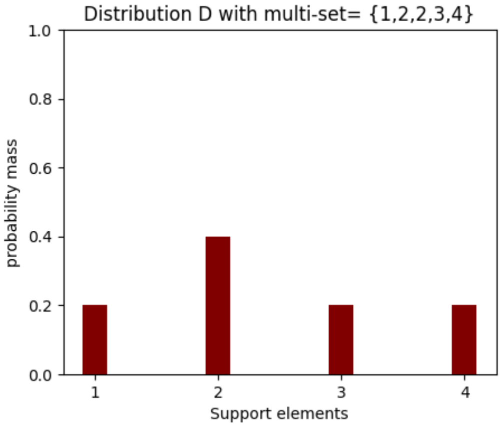

Figure 12.1: Distribution $D$ corresponding to the multi-set $D = \{1, 2, 2, 3, 4\}$

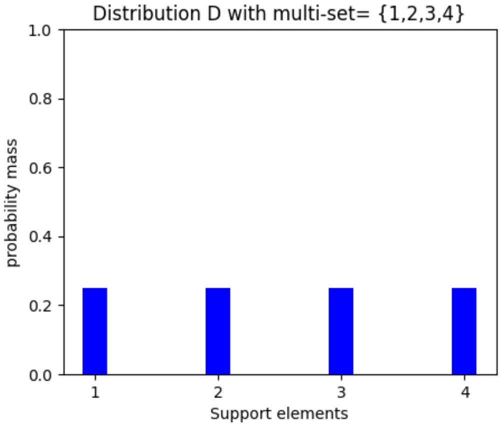

Figure 12.2: Distribution $D$ after $2$ is sampled without replacement

Although sampling **with** replacement is more natural query model, we need sampling **without** replacement for our lower bound proof. We now note that we can simulate samples **with** replacement when we have samples **without** replacement.

**Proposition 12.13** (Simulating samples **with** replacement from samples **without** replacement)**.** *Given $Q$ samples **without** replacement from an unknown multi-set $S_u$ with $n$ elements, we can simulate $Q$ samples **with** replacement from $S_u$ where $Q \leq n$.*

*Proof.* Consider the following procedure to obtain $Q$ samples **with** replacement (say $x_1, \ldots, x_Q$) when we have $Q$ samples **without** replacement $(s_1, \ldots, s_Q)$ from the unknown multi-set $S_u$ with $Q \leq n$.

We first set $x_1 = s_1$. For each $i$ with $2 \leq i \leq Q$, we set $x_i$ as follows: with probability $1 - \frac{i-1}{n}$, we select one of the element from $\{s_1, \ldots, s_{i-1}\}$ uniformly at random as $x_i$; with probability $\frac{i-1}{n}$, we set $x_i = s_i$. From the description of procedure to generate $x_i$'s, we have $\mathbb{P}\left(x_i = s_i\right) = 1/n$.

Thus we can simulate $Q$ samples **with** replacement from $Q$ samples **without** replacement from the unknown multi-set $S_u$. □

The following observation connects the EMD between two probability distributions with that of between two multi-sets.

**Observation 12.14.** Let $p, q$ be two $K$-grained probability distributions [11] on a $n$ dimensional Hamming cube $H = \{0,1\}^n$. Then $p$ and $q$ induces two multi-sets $S_1$ and $S_2$ on $H$, respectively, as follows. $S_1$ ($S_2$) is the multi-set containing $x \in H$ with multiplicity $p(x)K$ ($q(x)K$) for each $x \in H$. Moreover, $d_{EM}(p,q) = \frac{d_{EM}(S_1,S_2)}{K}$.

*Proof.* Recall the definitions of EMD between two distributions and two multi-sets given in Definition 2.2 and 12.11, respectively. We will be done with the proof by showing $d_{EM}(S_1, S_2) \leq K \cdot d_{EM}(p,q)$ and $K \cdot d_{EM}(p,q) \leq d_{EM}(S_1, S_2)$, separately.

For $d_{EM}(S_1, S_2) \leq K \cdot d_{EM}(p,q)$, let $\{f^*_{ij} : i, j \in H\}$ be the set of variables that realizes $d_{EM}(p,q)$, that is, $d_{EM}(p,q) = \sum\limits_{i,j \in H} f^*_{ij} d_H(i,j)$. Consider a bijection $\phi$ from $S_1$

[11] The probability of each element in the sample space is an integer multiple of $\frac{1}{K}$.

to $S_2$ where $\phi(i) = j$ for $g_{ij}$ many $i$'s. Hence, by Definition 12.11,

$$d_{EM}(S_1, S_2) \leq \sum_{x \in S_1} d_H(x, \phi(x)) = \sum_{i,j \in H} K \cdot f^*_{ij} d_H(i,j) = K \cdot d_{EM}(p,q).$$

Now we show $K \cdot d_{EM}(p,q) \leq d_{EM}(S_1, S_2)$. Let $\phi^*$ be a bijection from $S_1$ to $S_2$ that realizes $d_{EM}(S_1, S_2)$, that is, $d_{EM}(S_1, S_2) = \sum\limits_{x \in S_1} d_H(x, \phi^*(x))$. For any $x, y \in H$, let $f_{xy}$ be the number of elements, of the form $(x, y)$ in $S_1 \times S_2$ such that $x$ is mapped to $y$ under $\phi$, divided by $K^2$. Observe that $f_{xy} \geq 0$. Also, $f_{xy} > 0$ if and only if $(x, y) \in S_1 \times S_2$. More over, $\{f_{ij} : i, j \in H\}$ satisfies $\sum\limits_{i \in H} f_{ij} = p(j)\ \forall j \in H$ and $\sum\limits_{j \in H} f_{ij} = q(i)\ \forall i \in H$. Hence, by Definition 2.2,

$$\begin{aligned} K \cdot d_{EM}(p,q) \leq K \sum_{x,y \in H} f_{xy} d_H(x,y) &= \sum_{(x,y) \in S_1 \times S_2} K \cdot f_{xy} d_H(x,y) \\ &= \sum_{x \in S_1} d_H(x, \phi^*(x)) = d_{EM}(S_1, S_2). \end{aligned}$$

□

**Remark 12.2.** Note that sample access from a probability distribution is exactly same as uniform sampling from a multi-set **with** replacement.

**Proposition 12.15.** *Let $\mathcal{D}$ be the set of all multi-sets of size $n$ over a universe $[m]$; let $S_k$ and $S_u$ in $\mathcal{D}$ denote the known and unknown multi-sets over $[n]$; and* $\text{Prop} : \mathcal{D} \times \mathcal{D} \rightarrow \{0, 1\}$ *be a boolean function. Then the following holds:*

*If there exists an algorithm that determines* Prop *by $Q$ samples **without** replacement from $S_u$ with probability at least $2/3$, then there exists an algorithm that determines* Prop *by* $\min\{Q, \sqrt{\min\{n, m\}}\}$ *samples **with** replacement from $S_u$ with probability at least* $2/3 - o(1)$.

This follows from the fact that when $Q = o(\sqrt{n})$ and $D_{WR}$ ($D_{WoR}$) be the probability distribution over all the subsets having $Q$ elements from $[n]$ **with** (**without**) replacement, the $\ell_1$ distance between $D_{WR}$ and $D_{WoR}$ is $o(1)$.

**Definition 12.16 (EMD over multi-sets while sampling with and without replacement).** Let $S_k$ and $S_u$ denote the known and the unknown multi-sets, respectively, over $n$-dimensional Hamming cube $H = \{0,1\}^n$ such that $|S_u| = |S_k| = n$. Consider the two distributions $p_u$ and $p_k$ over the Hamming cube $H$ that are naturally defined by the sets $S_u$ and $S_k$ where for all $x \in H$ probability of $x$ in $p_u$ (and $p_k$) is the number of occurrences of $x$ in $S_u$ (and $S_k$) divided by $n$. We then define the EMD between the multi-sets $S_u$ and $S_k$ as

$$d_{EM}(S_u, S_k) \triangleq n \cdot d_{EM}(p_u, p_k).$$

The problem of estimating the EMD over multi-sets while sampling **with** (or **without**) replacement means designing an algorithm, that given any two constants $\beta_1, \beta_2$ such that $0 \leq \beta_1 < \beta_2 \leq 1$, and access to the unknown set $S_u$ by sampling **with** (or **without**) replacement decides whether $d_{EM}(S_k, S_u) \leq \beta_1 n^2$ or $d_{EM}(S_k, S_u) \geq \beta_2 n^2$ with probability at least $2/3$.

Note that estimating the EMD over multi-sets while sampling **with** replacement is exactly same as estimating EMD between the distributions $p_u$ and $p_k$ with samples drawn according to $p_u$.

Let $\text{QWR}_{\text{EMD}}(n, d, \beta_1, \beta_2)$ (and $\text{QWoR}_{\text{EMD}}(n, d, \beta_1, \beta_2)$) denote the number of samples **with** (and **without**) replacement required to decide the above from the unknown multi-set $S_u$. For ease of presentation, we write $\text{QWoR}_{\text{EMD}}(n, d)$ ($\text{QWR}_{\text{EMD}}(n, d)$) instead of $\text{QWoR}_{\text{EMD}}(n, d)$ ($\text{QWR}_{\text{EMD}}(n, \beta_1, \beta_2)$) when the proximity parameters are clear from the context.

**Proposition 12.17** (Query complexity of EMD increases with number of points as well as dimension)**.** *Let $n, n_1, n_2, d, d_1, d_2 \in \mathbb{N}$ be such that $d_1 \leq d_2$ and $n_1 \leq n_2$. Then*

**(i)** $\text{QWR}_{\text{EMD}}(n_1, d) \leq \text{QWR}_{\text{EMD}}(n_2, d)$*;*

**(ii)** $\text{QWoR}_{\text{EMD}}(n_1, d) \leq \text{QWoR}_{\text{EMD}}(n_2, d)$*;*

**(iii)** $\text{QWR}_{\text{EMD}}(n, d_1) \leq \text{QWR}_{\text{EMD}}(n, d_2)$*; and*

**(iv)** $\text{QWoR}_{\text{EMD}}(n, d_1) \leq \text{QWoR}_{\text{EMD}}(n, d_2)$.

**Remark 12.3.** For $d = n$ (as considered in Definition 12.16), $\text{QWoR}_{\text{EMD}}(n, d)$ (and $\text{QWR}_{\text{EMD}}(n, d)$) are denoted as $\text{QWoR}_{\text{EMD}}(n)$ (and $\text{QWR}_{\text{EMD}}(n)$).

Now let us state the lower bound of $\text{QWR}_{\text{EMD}}(n)$.

**Theorem 12.18.** $\text{QWR}_{\text{EMD}}(n) = \Omega(\frac{n}{\log n})$.

Thus following Proposition 12.15, we have

**Theorem 12.19.** $\text{QWoR}_{\text{EMD}}(n) = \Omega(\sqrt{n})$.

Note that an upper bound of $\text{QWoR}_{\text{EMD}}(n) = \widetilde{\mathcal{O}}(n)$ is trivial. In the rest of the section, we focus on proving Theorem 12.18 that states the lower bound on $\text{QWR}_{\text{EMD}}(n)$. We also provide an upper bound for $\text{QWR}_{\text{EMD}}(n)$ at Lemma 12.24 that shows that $\widetilde{\mathcal{O}}(n)$ samples **with** replacement from $S_u$ to estimate $\text{QWR}_{\text{EMD}}(n)$. Note that by Remark 12.2, it is enough to show the following lemma that states the lower bound for tolerant EMD testing between two distributions.

**Lemma 12.20.** *Let $S$ be a subset of a Hamming cube $H = \{0,1\}^n$ such that the minimum distance between any pair of points in $S$ is at least $n/2$. Also, let $p$ and $q$ be two known and unknown distributions, respectively, supported over a subset of $S$. Then there exists a constant $\varepsilon_{EMD}$ such that the following holds. Given two constants $\beta_1, \beta_2$ with $0 < \beta_1 < \beta_2 < \varepsilon_{EMD}(c)$, $\Omega\left(n/\log n\right)$ samples from the distribution $q$ are necessary in order to decide whether $d_{EM}(p,q) \leq \beta_1 n$ or $d_{EM}(p,q) \geq \beta_2 n$. Moreover, $\varepsilon_{EMD} = \frac{1-\varepsilon_{\ell_1}}{4}$, where $\varepsilon_{\ell_1}$ is the constant that is mentioned in Theorem 12.22.*

To prove the above lower bound, let us first consider the following lower bound for tolerant $\ell_1$ testing between two probability distributions.

**Theorem 12.21** (Valiant and Valiant [VV11])**.** *Let $p$ and $q$ be two known and unknown probability distributions respectively over $[n]$. There is an absolute constant $\varepsilon$ such that in order to decide whether $\|p-q\|_1 \leq \varepsilon$ or $\|p-q\|_1 \geq 1-\varepsilon$, $\Omega(n/\log n)$ samples, from the distribution $q$, are necessary.* [12]

[12] Note that this is rephrasing of the result proved in [VV11]. For reference, see Chapter 5 of the survey by Canonne [Can20c].

Now, we restate the above result for our purpose.

**Theorem 12.22.** *Let $p$ and $q$ be two known and unknown probability distributions, having support size $n$, over a Hamming cube $H = \{0,1\}^n$. There is an absolute constant $\varepsilon_{\ell_1}$ such that in order to decide whether $\|p - q\|_1 \leq \alpha_1$ or $\|p - q\|_1 \geq \alpha_2$ with $0 < \alpha_1 < \alpha_2 \leq 1 - \varepsilon_{\ell_1}$, $\Omega(n/\log n)$ samples, from the distribution $q$, are necessary.*

As noted earlier, we will prove Theorem 12.18 by using Lemma 12.22. However, Theorem 12.18 is regarding EMD between two distributions whereas Lemma 12.22 is regarding $\ell_1$-distance. The following observation (from [DBNNR11]) gives a connection between EMD between two distributions with the $\ell_1$ distance between them, which will be required in lower bound proof.

**Proposition 12.23** ([DBNNR11])**.** *Let $(M, D)$ be a finite metric space and $p$ and $q$ be two probability distributions on $M$. Minimum distance between any two points of $M$ is $\Delta_{\min}$ and diameter of $M$ is $\Delta_{\max}$. Then the following condition holds:*

$$\frac{\|p - q\|_1 \Delta_{\min}}{2} \leq d_{EM}(p, q) \leq \frac{\|p - q\|_1 \Delta_{\max}}{2}.$$

Note that the above proposition gives interesting result when $\frac{\Delta_{\max}}{\Delta_{\min}}$ is bounded by a constant. Note that $S \subset \{0,1\}^n$ satisfies $\frac{\Delta_{\max}}{\Delta_{\min}} \leq 2$.

*Proof of Lemma 12.20.* In $S \subset H = \{0,1\}^n$, the pairwise Hamming distance between any two elements in $S$ is at least $\frac{n}{2}$, to have $\frac{\Delta_{\max}}{\Delta_{\min}} \leq 2$ in our context. It is well known that $|S| = \Omega(n)$. We prove that if there exists an algorithm $\mathcal{A}$ that decides $d_{EM}(p, q) \leq \beta_1 n$ or $d_{EM}(p, q) \geq \beta_2 n$ by using $t$ samples from $q$, then there exists an algorithm $\mathcal{P}$ that decides whether $\|p - q\|_1 \leq \alpha_1$ or $\|p - q\|_1 \geq \alpha_2$ by using $t$ samples from $q$, where $\alpha_1 = 2\beta_1$ and $\alpha_2 = 4\beta_2$. Note that we have $0 < \beta_1 < \beta_2 < \frac{1 - \varepsilon_{\ell_1}}{4}$. So, $0 < \alpha_1 < \alpha_2 < 1 - \varepsilon_{\ell_1}$, which satisfies the requirement of Theorem 12.22.

**Algorithm $\mathcal{P}$:**

**(1)** First run algorithm $\mathcal{A}$.

**(2)** If the output of algorithm $\mathcal{A}$ is $d_{EM}(p,q) \leq \beta_1 n$, algorithm $\mathcal{P}$ returns $\|p-q\|_1 \leq \alpha_1$.

**(3)** If the output of algorithm $\mathcal{A}$ is $d_{EM}(p,q) \geq \beta_2 n$, algorithm $\mathcal{P}$ returns $\|p-q\|_1 \geq \alpha_2$.

To complete the proof, we only need to show that $\mathcal{P}$ gives desired output with probability at least $2/3$. The result then follows from Theorem 12.22.

Let us first consider the case $\|p-q\|_1 \leq \alpha_1$. Then by Observation 12.23, we can say that $d_{EM}(p,q) \leq \frac{\alpha_1 n}{2} = \beta_1 n$. Therefore algorithm $\mathcal{A}$ will output that $d_{EM}(p,q) \leq \beta_1 n$. This implies that the algorithm $\mathcal{P}$ will output $\|p-q\|_1 \leq \alpha_1$.

Now, let us consider the case $\|p-q\|_1 \geq \alpha_2$. Using the fact that any pair elements in $S \subset H$ is at least $\frac{n}{2}$ along with Observation 12.23, we get $d_{EM}(p,q) \geq \frac{\alpha_2 n}{4} = \beta_2 n$. This implies $\mathcal{P}$ will output $\|p-q\|_1 \geq \alpha_2$. □

Till now, we were discussing the proof of Lemma 12.20 that states $\text{QWR}_{\text{EMD}}(n) = \Omega(\frac{n}{\log n})$. The lower bound is almost tight, up to a polynomial factor of $\log n$. The upper bound is stated in the following observation.

**Observation 12.24.** $\text{QWR}_{\text{EMD}}(n) = \widetilde{\mathcal{O}}(n)$, where $\widetilde{\mathcal{O}}(\cdot)$ hides a polynomial factor in $\frac{1}{\beta_2-\beta_1}$ and $\log n$.

Instead of proving the above observation, we prove the following lemma that states the upper bound of tolerant EMD testing between two distributions when we know one distribution and have sample access to the unknown distribution. By Remark 12.2, we will be done with the proof of Observation 12.24.

**Lemma 12.25.** *Let $H = \{0,1\}^n$ be a $n$-dimensional Hamming cube, and let $p$ and $q$ denote two known and unknown $n$-grained distribution over $H$. There exists an algorithm that takes two parameters $\beta_1, \beta_2$ with $0 \leq \beta_1 < \beta_2 \leq 1$ and a $\delta \in (0,1)$ as input and decides whether $d_{EM}(p,q) \leq \beta_1 n$ or $d_{EM}(p,q) \geq \beta_2 n$ with probability at least $1-\delta$. Moreover, the algorithm* ALG-EMD *queries for $\widetilde{\mathcal{O}}(n)$ samples from $q$, where $\widetilde{\mathcal{O}}(\cdot)$ hides a polynomial factor in $\frac{1}{\beta_2-\beta_1}$ and $\log n$.*

*Proof.* Let $\varepsilon$ be a constant less than $(\beta_2 - \beta_1)$. We construct a probability distribution $q'$ such that the $\ell_1$ distance between $q$ and $q'$ will be at most $\varepsilon$, that is, $\sum\limits_{i \in [L]} |q(i) - q'(i)| \leq \varepsilon$.

Note that such a $q'$ can be constructed with probability at least $1-\delta$ by querying for $\widetilde{\mathcal{O}}\left(n\right)$ samples of $q$ which follows from [DL12]. Then, we find $d_{EM}(p,q')$. Observe that $|d_{EM}(p,q)-d_{EM}(p,q')| \leq \frac{\varepsilon n}{2}$. This is because

$$\begin{array}{rcl} |d_{EM}(p,q)-d_{EM}(p,q')| & \leq & |d_{EM}(p,q')+d_{EM}(q',q)-d_{EM}(p,q')| \\ & \leq & d_{EM}(q,q') \\ & \leq & \dfrac{\varepsilon d}{2} \quad \text{(By Proposition 12.23)} \end{array}$$

As $d_{EM}(p,q) \leq \beta_1 n$ or $d_{EM}(p,q) \geq \beta_2 n$, by the above observation, we will get either $d_{EM}(p,q') \leq \left(\beta_1+\frac{\varepsilon}{2}\right)n$ or $d_{EM}(p,q') \geq \left(\beta_1+\frac{\varepsilon}{2}\right)n$, respectively. By our choice of $\varepsilon < \beta_2-\beta_1$, we can decide $d_{EM}(p,q) \leq \beta_1 n$ or $d_{EM}(p,q) \geq \beta_2 n$ from the value of $d_{EM}(p,q')$. □

To the best of our knowledge, the sample complexity measure when the distributions are accessed by sampling a multi-set **without** replacement has never been studied before (for testing/estimating *distances* between distributions/multi-sets). However, it is interesting to note that, sampling **without** replacement model has been considered before in a different context by Raskhodnikova, Ron, Shpilka and Smith [RRSS09] for proving a lower bound of distinct elements problem. Also, recently Goldreich [Gol19] considered a similar sampling **without** replacement model while studying the non-tolerant graph isomorphism in the bounded degree model. Note that the main contribution of our work is the introduction of the complexity measure $\text{QWoR}_{\text{EMD}}(n)$ and its connection to graph isomorphism testing in query model.

## 12.4 Overview of our results

### 12.4.1 Overview of our tolerant isomorphism testing result

In this subsection, we give an overview of our result on tolerant isomorphism testing (Theorem 12.5). We will start with the overview of our lower bound as follows:

## Tolerant GI testing is as hard as tolerant EMD testing

In this section, we give an overview of the lower bound part of Theorem 12.5, namely tolerant GI testing is as hard as tolerant EMD testing. In this reduction, we crucially use the fact that the multi-sets are composed of elements from the Hamming cube. The reduction is based upon an involved gadget construction. In fact, we prove the lower bound for a slightly more powerful query model rather than the standard adjacency matrix query model. The most interesting part of our lower bound proof is that thanks to our reduction, we get to observe the importance of the model of accessing the multi-set **without** replacement in the context of EMD testing.

Now, we discuss the overview of our reduction. Let $S_k$ and $S_u$ denote the known and the unknown multi-sets, over a Hamming cube $\{0,1\}^d$ (of dimension $d$) with $d = \Theta(n)$, having $n$ elements each. To start with, let us assume that we know both $S_k$ and $S_u$. We will construct two graphs $G_k$ and $G_u$ on $d+n$ vertices as follows:

- The vertex set of $G_k$ (and $G_u$) are partitioned into two sets $A_k$ and $B_k$ (and $A_u$ and $B_u$) with $|A_k| = |A_u| = n$ and $|B_k| = |B_u| = d$.

- The graph induced by $A_k$ is a clique, and similarly the graph induced by $A_u$ is a clique.

- The graphs induced by $B_k$ and $B_u$ are copies of a special graph with certain nice properties which enable our reduction to work. The existence of such a graph is proved (in Lemma 13.3) using a probabilistic argument.

- Finally, for the cross edges between $A_k$ and $B_k$ (and $A_u$ and $B_u$), we have: there is an edge between the $i$-th vertex of $A_k$ (or $A_u$) and the $j$-th vertex of $B_k$ (or $B_u$) if and only if the $j$-th coordinate of the $i$-th element of $S_k$ (or $S_u$) is 1.

- Finally, a random permutation $\pi$ is applied to the vertices of $G_u$.

The permutation $\pi$ is not known to the GI-tester. Note that we can construct $G_k$ explicitly as $S_k$ is known. However, that is not the same with $G_u$ as $S_u$ is unknown. But since we know the permutation $\pi$, any query to the adjacency matrix of the graph $G_u$

can be answered by a single query to one bit of $S_u$. But unfortunately we don't have query access to $S_u$, and only have sample access to $S_u$. To deal with this problem, it is easier to consider a slightly more powerful query. Say, the GI-tester wants to query the $(i, j)$-th bit of the graph $G_u$. Of course, if both $i$ and $j$ are in $A_u$ or both are in $B_u$, we can answer without even sampling from $S_u$. But if $i$ is in $A_u$ and $j$ is in $B_u$, then what we intend to do is to give the whole neighborhood of $i$ in $B_u$ as the answer to the query. This would be like neighbourhood query in a bipartite graph. But the question remains: how do we intend to answer the query by sampling. The key observation here is that since the GI-tester does not know the permutation $\pi$ that was applied to the vertices in $G_u$, to its eye, all the vertices that have not been touched so far look same. So, every time it queries for $(i, j)$, where $i \in A_u$ and $j \in B_u$, either of the two cases can happen:

- Either, previously a query of the form $(i, j_1)$ was asked where $j_1$ is also in $B_u$, but in that case, it must have already got the answer of $(i, j)$ as we must have given all the neighbors of $i$ in $B_u$. So in that case, we can give back the same answer without sampling.

- Or, previously $i$ did not participate in any query of the form $(i, j_1)$ where $j_1$ is in $B_u$. In this case, to the GI-tester's eye, $i$ is just a new vertex from $A_u$. We can then sample **without** replacement from $S_u$ and whatever sample of the multi-set we have, we can assume that it is the element $i$ and answer accordingly. Note that this is the exact place where sampling **without** replacement is crucial.

To complete our proof, we need to prove how the GI-distance between $G_k$ and $G_u$ is connected to the EMD between $S_k$ and $S_u$. Consider the set $\Phi$ of all SPECIAL bijections from $V(G_k)$ to $V(G_u)$ that maps $A_k$ into $A_u$ and $B_k$ into $B_u$ such that the $i$-th vertex of $B_k$ is mapped to the $i$-th vertex of $B_u$. Observe that $d_\Phi(G_k, G_u) = 2 \cdot d_{EM}(S_k, S_u)$, where $d_\Phi(G_k, G_u) = \min\limits_{\phi \in \Phi} d_\phi(G_k, G_u)$ (See Lemma 13.5 for a formal proof). The factor 2 is because of the way we define $d_\phi(G_k, G_u)$ (See Definition 12.1). This implies that tolerant isomorphism testing between $G_k$ and $G_u$ is at least as hard as tolerant EMD testing between $S_k$ and $S_u$ if we restrict the bijection from $V(G_k)$ to $V(G_u)$ to be a SPECIAL bijection. The reduction works for all possible bijections, because of the careful choice of the subgraph of $G_k$ (and $G_u$) induced by $B_k$ (and $B_u$), thus ensuring $d(G_k, G_u)$ is close to $d_\Phi(G_k, G_u)$ (See Lemma 13.6 for a formal proof).

One might compare our proof technique to the lower bound proof of (non-tolerant) testing of GI from [FM08]. In [FM08], $\Omega\left(\sqrt{n}\right)$ lower bound was proved directly (using Yao's lemma) by constructing two distributions of YES instances and NO instances - the construction of the YES and NO instances were inspired from the tightness of the birthday paradox, which was also the core idea behind the lower bound proof of the equivalence testing of two probability distributions. But, there was no direct reduction from GI testing to equivalence testing of two probability distributions. But in our lower bound proof, we establish a direct reduction to estimating EMD of multi-sets on the Hamming cube with access to samples **without** replacement. This can be of much importance, mainly while considering other models of computation, like in the communication model. From our reduction, we can obtain an alternative proof of $\Omega(\sqrt{n})$ lower bound for the (non-tolerant) GI testing via the $\Omega(\sqrt{n})$ lower bound of the equivalence testing of distributions, as pointed out in Corollary 12.6.

## Tolerant EMD testing is as hard as tolerant GI testing

In this section, we give an overview of the upper bound part of Theorem 12.5, namely tolerant EMD testing is as hard as tolerant GI testing. Given a known graph $G_k$ and query access to an unknown graph $G_u$ (both on $n$ vertices), we present an algorithm for tolerant testing of graph isomorphism between $G_k$ and $G_u$ by using a tolerant EMD tester (for distributions over $H$) as a blackbox. Note that this will prove the upper bound part of Theorem 12.5.

**Algorithm for tolerant GI using tolerant EMD as a black box.** Our testing algorithm is inspired by the algorithm of Fischer and Matsliah [FM08] for non-tolerant GI testing. But our algorithm significantly differs from that of Fischer-Matsliah in some crucial points. As we explain the high level picture of our algorithm, we will point out some of the crucial differences.

We split our algorithm into three phases. In Phase 1, we first choose a $\mathcal{O}\Big(\frac{1}{\gamma_2-\gamma_1}\Big)$ size collection of random subset of vertices, i.e, *coresets* $\mathcal{C}_u$ from the unknown graph $G_u$ where each $C_u \in \mathcal{C}_u$ is of size $\mathcal{O}(\log n)$. Thereafter we find all embeddings of $C_u$ inside the known graph $G_k$. Let the embeddings be $\eta_1, \eta_2, \ldots, \eta_J$ where $C_k^i = \eta_i(C_u)$. Now each $C_u$ (as well as each $C_k^i$) defines a label distribution of the vertices of $G_u$ (as well as $G_k$). Let us denote the set of labels as $X_{C_u}$ (and $Y_{C_k^i}$). Now we test if the EMD between $X_{C_u}$ and $Y_{C_k^i}$ is close or far for each $i \in [J]$ (See Claim 14.2). We keep only those $(C_u, \eta_i)$ for Phase 2 such that $d_{EM}(X_{C_u}, Y_{C_k^i}) \leq \big(\gamma_1 + \frac{\gamma_2-\gamma_1}{2000}\big)\, n\, |C_u|$.

Although Phase 1 of our algorithm is similar to the algorithm of [FM08], there is a striking difference. Since the authors of [FM08] were testing the non-tolerant version of graph isomorphism, they were testing the identity of the label distributions of $X_{C_u}$ and $Y_{C_k^i}$. However, since we are solving the tolerant version of the problem, we need to allow some error among the label distributions. We need to pass only those placements of $C_u$ that under *good bijections* do not produce much error and testing of tolerant EMD fits exactly for this purpose. It is worth noting that Fischer-Matsliah uses an equivalence tester in their algorithm to identify the placements that do not produce "any" error. But, the proof of correctness of the algorithm would not go through even if we use the tolerant testing of the equivalence of distributions. The use of EMD in this phase is crucial for the proof of correctness of our algorithm to hold.

In Phase 2, we choose $\mathcal{O}\Big(\frac{\log^2 n}{(\gamma_2-\gamma_1)^3}\Big)$ many vertices from the unknown graph $G_u$ randomly and call it $W$. We further find the labels of all the vertices of $W$ under $C_u$-labelling by querying the corresponding entries of $G_u$ for each $C_u$ that has passed Phase 1. Then we try to match the vertices of $W$ to the set of all possible labels $\{l_1, l_2, \ldots, l_t\}$ of the vertices of $G_k$ under $C_k^i$-labelling where $C_k^i = \eta_i(C_u)$, for those $\eta_i$ that have passed Phase 1. Ideally, we would like to find a mapping $\psi : W \to \{l_1, l_2, \ldots, l_t\}$ such that the total distance between the labels of the matched vertices is not too large. If no such $\psi$ is possible, we reject the current embedding and try some other embedding that has passed Phase 1.

In Phase 3, we construct a random partial bijection $\widehat{\phi} : W \to V(G_k)$ that maps the vertices of $W$ to the vertices of $G_k$ while preserving the labels according to $\psi$. We achieve this by mapping each $w \in W$ to one vertex of $G_k$ randomly that has same label as determined by $\psi$. Finally, we randomly pair the vertices of $W$ and find the fraction of edge mismatches between the paired up vertices of $W$ and $\widehat{\phi}(W)$. If this fraction is at most $5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)$, we accept and say that $G_u$ and $G_k$ are $\gamma_1$-close. If there is no such embedding of any $C_u \in \mathcal{C}_u$ that achieves this, we report that $G_u$ and $G_k$ are $\gamma_2$-far.

The proofs of completeness and soundness follow kind of similar route as Fischer-Matsliah's proof but the arguments are way more complicated. Many things that were trivial or obvious in the non-tolerant setting become major hurdles in the tolerant setting, and we overcome them with significantly difficult technical arguments, presented in Chapter 14.

### 12.4.2 Overview of our tolerant bipartiteness testing result

In this section, we present an overview of our algorithm. The detailed description of the algorithm is presented in Section 15.3, while its analysis is presented in Section 15.4. We will prove the following theorem, which is our main technical result.

**Theorem 12.26.** *There exists an algorithm* TOL-BIP-DIST$(G, \varepsilon)$ *that given adjacency query access to a dense graph G with $n$ vertices and a proximity parameter $\varepsilon \in (0, 1)$, decides with probability at least $\frac{9}{10}$, whether $d_{bip}(G) \leq \varepsilon n^2$ or $d_{bip}(G) \geq (2 + k)\varepsilon n^2$, by sampling $\mathcal{O}\left(\frac{1}{k^5\varepsilon^2} \log \frac{1}{k\varepsilon}\right)$ vertices in $2^{\mathcal{O}\left(\frac{1}{k^3\varepsilon} \log \frac{1}{k\varepsilon}\right)}$ time, using $\mathcal{O}\left(\frac{1}{k^8\varepsilon^3} \log^2 \frac{1}{k\varepsilon}\right)$ queries to the adjacency matrix of G, where $d_{bip}(G)$ denotes the distance of G from being bipartite.*

Note that Theorem 12.26 implies Theorem 15.1, assuming $k = \Omega(1)$.

### Overview of TOL-BIP-DIST$(G, \varepsilon)$

Assume $C_1, C_2, C_3$ are three suitably chosen large absolute constants. At the beginning of our algorithm, we generate $t$ subsets of vertices $X_1, \ldots, X_t$, each with $\left\lceil \frac{C_2}{k^3\varepsilon} \log \frac{1}{k\varepsilon} \right\rceil$ vertices chosen randomly, where $t = \left\lceil \log \frac{C_1}{k\varepsilon} \right\rceil$. Let $\mathcal{C} = X_1 \cup \ldots \cup X_t$. Apart from the $X_i$'s, we also randomly select a set of pairs of vertices $Z$, with $|Z| = \left\lceil \frac{C_3}{k^5\varepsilon^2} \log \frac{1}{k\varepsilon} \right\rceil$. We find the neighbors of each vertex of $Z$ in $\mathcal{C}$. Then for each vertex pair in $Z$, we check whether it is an edge in the graph or not. Roughly speaking, the set of edges between $\mathcal{C}$ and $V(Z)$ [13] will help us generate partial bipartitions, restricted to $X_i \cup V(Z)$'s, for each $i \in [t]$, and the edges among the pairs of vertices of $Z$ will help us in estimating the bipartite distance of some *specific kind* of bipartitions of $G$. Here we would like to note that no further query will be performed by the algorithm. The set of edges with one vertex in $\mathcal{C}$ and the other in $V(Z)$, and the set of edges among the vertex pairs in $Z$, when treated in a *specific* manner, will give us the desired result. Observe that the number of adjacency queries performed by our algorithm is $\mathcal{O}(\frac{1}{k^8\varepsilon^3}\log^2\frac{1}{k\varepsilon})$.

For each $i \in [t]$, we do the following. We consider all possible bipartitions $\mathcal{F}_i$ of $X_i$. For each bipartition $f_{ij}$ (of $X_i$) in $\mathcal{F}_i$, we extend $f_{ij}$ to a bipartition of $X_i \cup V(Z)$, say $f'_{ij}$, such that both $f_{ij}$ and $f'_{ij}$ are identical with respect to $X_i$. Moreover, we assign $f'_{ij}(z)$ (to either $L$ or $R$), for each $z \in V(Z) \setminus X_i$, based on the neighbors of $z$ in $X_i$. To design a rule of assigning $f'_{ij}(z)$, for each $z \in V(Z) \setminus X_i$ for our purpose, we define the notions of *heavy* and *balanced* vertices, with respect to a bipartition (see Definition 15.8 and Definition 15.9). Heavy and balanced vertices are defined in such a manner that when the bipartite distance of $G$ is at most $\varepsilon n^2$ (that is, $G$ is $\varepsilon$-close), we can infer the following interesting connections. Let $f$ be a bipartition of $V(G)$ such that $d_{bip}(G,f) \leq \varepsilon n^2$. We will prove that the total number of edges, with no endpoints in $X_i$ and whose at least one end point is a balanced vertex with respect to $f$, is bounded (see Claim 15.19). Moreover, if we generate a bipartition $f'$ such that $f$ and $f'$ differ for *large* number of heavy vertices, then the bipartite distance with respect to $f'$ cannot be bounded. To guarantee the correctness of our algorithm, we will prove that a heavy vertex $v$ with respect to $f$, can be detected and $f(v)$ can be determined, with probability at least $1 - o(k\varepsilon)$. Note that the testing of being a heavy vertex will be performed only for the vertices in $V(Z)$. We will see shortly how this will help us to guarantee the completeness of our algorithm.

Finally, our algorithm computes $\zeta_{ij}$, that is, the fraction of vertex pairs in $Z$ that are *monochromatic* [14] edges with respect to $f'_{ij}$. If we find at least one $i$ and $j$ such that

[13] Recall that $V(Z)$ denotes the set of vertices present in at least one pair in $Z$.

[14] An edge is said to be monochromatic with respect to $f'_{ij}$ if both its endpoints have the same $f'_{ij}$ values.

$\zeta_{ij} \leq \left(2 + \frac{k}{20}\right) \varepsilon$, the algorithm decides that $d_{bip}(G) \leq \varepsilon n^2$. Otherwise, it will report that $d_{bip}(G) \geq (2+k)\varepsilon n^2$.

**Overview of completeness:** Let us assume that the bipartite distance of $G$ is at most $\varepsilon n^2$, and let $f$ be a bipartition of $V(G)$ that is optimal. Let us now focus on a particular $i \in [t]$, that is, an $X_i$. Since we are considering all possible bipartitions $\mathcal{F}_i$ of $X_i$, there exists a $f_{ij} \in \mathcal{F}_i$, such that $f_{ij}$ and $f$ are identical with respect to $X_i$. To complete our argument, we introduce (in Definition 15.10) the notion of SPECIAL bipartition $\text{SPL}_i^f : V(G) \rightarrow \{L, R\}$, with respect to $f$ by $f_{ij}$ such that $f(v)$, $f_{ij}(v)$ and $\text{SPL}_i^f(v)$ are identical for each $v \in X_i$, and at least $1 - o(k\varepsilon)$ fraction of heavy vertices, with respect to $f$, are mapped identically both by $f$ and $\text{SPL}_i^f$. We shall prove that the bipartite distance of $G$ with respect to $\text{SPL}_i^f$ is at most $\left(2 + \frac{k}{50}\right) \varepsilon n^2$ (see Lemma 15.13). Now let us think of generating a bipartition $f_{ij}^{''}$ of $V(G)$ such that, for each $v \in V(G) \setminus X_i$, if we determine $f_{ij}^{''}(v)$ by the same rule used by our algorithm to determine $f_{ij}(z)$, for each $z \in V(Z) \setminus X_i$. Note that our algorithm does not find $f_{ij}^{''}$ explicitly, it is used only for the analysis. The number of heavy vertices, with respect to the bipartition $f$, that have different mappings by $f$ and $f_{ij}^{''}$, is at most $o(k\varepsilon n)$ with constant probability. So, with a constant probability, $f_{ij}^{''}$ is a SPECIAL bipartition with respect to $f$ by $f_{ij}$. Note that, if we take $|Z| = \mathcal{O}\left(\frac{1}{k^5\varepsilon^2} \log \frac{1}{k\varepsilon}\right)$ random vertex pairs and determine the fraction $\chi_{ij}^f$ of pairs that form monochromatic edges with respect to the SPECIAL bipartition $f_{ij}^{''}$, we can show that $\chi_{ij}^f \leq \left(2 + \frac{k}{20}\right) \varepsilon$, with probability at least $1 - 2^{-\Omega\left(\frac{1}{k^3\varepsilon} \log \frac{1}{k\varepsilon}\right)} \geq \frac{9}{10}$. However, we are not finding either $f_{ij}^{''}$ or $\chi_{ij}^f$ explicitly. We just find $\zeta_{ij}$, that is, the fraction of vertex pairs in $Z$ that are monochromatic edges with respect to $f_{ij}^{'}$. But the above argument still holds, since $Z$ is chosen randomly and there exists a $f_{ij}^{''}$, such that $f_{ij}^{''}(z) = f_{ij}^{'}(z)$, for each $z \in V(Z)$, and the probability distribution of $\zeta_{ij}$ is identical to that of $\chi_{ij}^f$.

**Overview of soundness:** Let us now consider the case when the bipartite distance of $G$ is at least $(2+k)\varepsilon n^2$, and $f$ be any bipartition of $V(G)$. To prove the soundness of our algorithm, we introduce the notion of DERIVED *bipartition* $\text{DER}_i^f : V(G) \rightarrow \{L, R\}$ with respect to $f$ by $f_{ij}$ (see Definition 15.11), such that $f(v)$, $f_{ij}(v)$ and $\text{DER}_i^f(v)$ are identical for each $v \in X_i$. Observe that the bipartite distance of $G$ with respect to any DERIVED bipartition is at least $(2+k)\varepsilon n^2$ as well. Similar to the discussion of the completeness, if we generate a bipartition $f''_{ij}$ of $V(G)$, $f''_{ij}$ will be a DERIVED bipartition, with respect to $f$ by $f_{ij}$. If we take $|Z| = \mathcal{O}\left(\frac{1}{k^5\varepsilon^2}\log\frac{1}{k\varepsilon}\right)$ random pairs of vertices and determine the fraction $\chi^f_{ij}$ of pairs that form monochromatic edges with respect to the DERIVED bipartition $f''_{ij}$, we can prove that $\chi^f_{ij} \leq \left(2+\frac{k}{20}\right)\varepsilon$ holds, with probability at most $2^{-\Omega\left(\frac{1}{k^3\varepsilon}\log\frac{1}{k\varepsilon}\right)}$. We want to re-emphasize that we are not determining $f''_{ij}$, as well as $\chi^f_{ij}$ explicitly. The argument follows due to the facts that $Z$ is chosen randomly and there exists an $f''_{ij}$ such that $f'_{ij}(z) = f''_{ij}(z)$, for each $z \in V(Z)$, and the probability distribution of $\zeta_{ij}$ is identical to that of $\chi^f_{ij}$. Using the union bound, we can say that the algorithm rejects with probability at least $\frac{9}{10}$.

# Chapter 13

# Tolerant Graph Isomorphism is as hard as tolerant EMD testing

## 13.1 Introduction

In this chapter, we prove that it is necessary to perform $\Omega\left(\text{QWoR}_{\text{EMD}}(n)\right)$ queries to the adjacency matrix of $G_u$ to solve $(\gamma_1, \gamma_2)$-tolerant GI testing of $G_k$ and $G_u$. Namely, we prove the following result:

**Theorem 13.1** (Restatement of the lower bound part of Theorem 12.5)**.** *Let $G_k$ be the known and $G_u$ be the unknown graph on $n$ vertices, where $n \in N$ is sufficiently* large. *There exists a constant $\varepsilon_{ISO} \in (0, 1)$ such that for any given constants $\gamma_1, \gamma_2$ with $0 < \gamma_1 < \gamma_2 < \varepsilon_{ISO}$, any algorithm that decides whether the graphs are $\gamma_1$-close or $\gamma_2$-far, requires* $\text{QWoR}_{\text{EMD}}(n)$ *adjacency queries to the unknown graph $G_u$ where* $\text{QWoR}_{\text{EMD}}$ *is as defined in Definition 12.16.*

In Section 12.4.1, we have discussed an overview of of our idea to prove the above theorem. To prove Theorem 13.1, we show a reduction from tolerant GI testing to tolerant EMD testing over multi-sets when we have samples **without** replacement from the unknown multi-set.

**Lemma 13.2.** *Suppose there is a constant $\varepsilon_0 \in \left(0, \frac{1}{2}\right)$ such that for all constants $\gamma_1, \gamma_2$ with $0 < \gamma_1 < \gamma_2 < \varepsilon_0$ and any constant $T \in \mathbb{N}$, the following holds: There exists a $(\gamma_1, \gamma_2)$-tolerant tester for GI that, given a known graph $G_k$ and an unknown graph $G_u$ with $|V(G_u)| = |V(G_k)| = (T+1)n$, can distinguish whether $d(G_u, G_k) \leq \gamma_1 T n^2$ or $d(G_u, G_k) \geq \gamma_2 T n^2$ by performing $Q$ adjacency queries to $G_u$.*

*Then, for any constants $\beta_1$ and $\beta_2$ with $0 < \beta_1 < \beta_2 < \frac{\varepsilon_0}{2}$, the following holds where $\kappa = \frac{\beta_2 - \beta_1}{8}$ and $T_\kappa = \lceil \frac{30}{\kappa(2-\kappa)} \rceil$. There is a tolerant tester for EMD such that, given a known and an unknown multi-set $S_k$ and $S_u$ respectively, of the Hamming cube $\{0,1\}^{T_\kappa n}$ with $|S_k| = |S_u| = n$, can distinguish whether $d_{EM}(S_k, S_u) \leq \beta_1 T_\kappa n^2$ or $d_{EM}(S_k, S_u) \geq \beta_2 T_\kappa n^2$ with $Q$ samples **without** replacement from $S_u$.*

**Remark 13.1.** Observe that Lemma 13.2 talks about tolerant EMD testing between multi-sets with $n$ elements over a Hamming cube of dimension $T_\kappa n$. But Theorem 13.1 states the lower bound of $\text{QWoR}_{\text{EMD}}(n)$, that is, of tolerant EMD testing of multi-sets with $n$ elements over a Hamming cube of dimension $n$. However, the query complexity of EMD testing increases with dimension of the Hamming cube (See Proposition 12.17). So, we will be done with the proof of Theorem 13.1 by proving Lemma 13.2.

## 13.2 Reduction from Tolerant GI to Tolerant EMD testing

Here we will present the proof of Lemma 13.2. To define the necessary reduction for the proof of Lemma 13.2, we need to show the existence of a graph $G_p$ satisfying some unique properties.

**Lemma 13.3.** *Let $\kappa \in (0,1)$ and $s \geq 3$ be given constants. Then for $C_{\kappa,s} = \lceil \frac{6s}{\kappa(2-\kappa)} \rceil$ and* sufficiently *large $n \in \mathbb{N}$* [1]*, there exists a graph $G_p$ with $C_{\kappa,s} n$ vertices such that the following conditions hold.*

**(i)** *The degree of each vertex in $G_p$ is at least $((1-\kappa)C_{\kappa,s} + 1)n - 1$.*

[1] The lower bound of $n$ is a constant that depends on $\kappa$ and $s$.

**(ii)** *The cardinality of symmetric difference between the sets of neighbors of any two (distinct) vertices in $G_p$ is at least $sn - 2$.*

*Proof.* To prove the claim, we use probabilistic method to show the existence of a graph $G'_p$, with $V(G'_p) = C_{\kappa,s}n$, that can have (possible) self loops and satisfy the followings.

**(i)** The degree of each vertex in $G'_p$ is at least $((1-\kappa)C_{\kappa,s}+1)n$.

**(ii)** The cardinality of symmetric difference between the sets of neighbors of any two (distinct) vertices in $G'_p$ is at least $sn$.

Let us construct a random graph having the vertex set $V(G'_p)$ such that each pair $\{u,v\}$, with $u, v \in V(G'_p)$ , is an edge with probability $1-\frac{\kappa}{2}$ independent of other pairs.

Now we compute the probability that the degree of a vertex $v \in G(V'_p)$, that is $\deg_{G'_p}(v)$, is at most $((1-\kappa)\,C_{\kappa,s}+1)\,n$. For each $v' \in V(G'_p)$, let $X_{v'}$ be the indicator random variable that takes value 1 if and only if $\{v,v'\} \in E(G'_p)$. Note that $\deg_{G'_p}(v) = \sum\limits_{v' \in V(G'_p)} X_{v'}$. Also, $\mathbb{P}(X_{v'}=1) = 1-\frac{\kappa}{2}$. So, the expected value of $\deg_{G'_p}(v)$ is $\left(1-\frac{\kappa}{2}\right)C_{\kappa,s}n$. By using the Chernoff bound (Lemma 2.11), we have

$$\begin{aligned}
&\mathbb{P}\left(\deg_{G'_p}(v) \le ((1-\kappa)\,C_{\kappa,s}+1)\,n\right)\\
&= \mathbb{P}\left(\deg_{G'_p}(v) \le (1-\varepsilon)\left(1-\frac{\kappa}{2}\right)C_{\kappa,s}n\right) \quad \left(\text{where } \varepsilon = \frac{\kappa C_{\kappa,s}-2}{(2-\kappa)C_{\kappa,s}} < 1\right)\\
&\le e^{-\frac{\varepsilon^2(2-\kappa)C_{\kappa,s}n}{6}}
\end{aligned}$$

Let $\mathcal{E}_1$ be the event that there exists a vertex $v \in V(G'_p)$ such that the degree of $v$ in $G'_p$ is at most $((1-\kappa)C_{\kappa,s}+1)n$. Using union bound, we can say that

$$\mathbb{P}(\mathcal{E}_1) \le \left|V(G'_p)\right| e^{-\frac{\varepsilon^2(2-\kappa)C_{\kappa,s}n}{6}} \le C_{\kappa,s}n \cdot e^{-\frac{\varepsilon^2(2-\kappa)C_{\kappa,s}n}{6}}.$$

Let $\mathcal{E}_2$ be the event that there exists two (distinct) vertices $u, v$ with $\left|N_{G'_p}(u)\Delta N_{G'_p}(v)\right| < sn$, where $N_{G'_p}(u)$ denotes the set of neighbors of $u$ in $G'_p$. Our goal is to show that $G'_p$ exists which satisfies the required conditions. Observe that, $G'_p$ satisfies the required conditions if and only if $\mathbb{P}(\mathcal{E}_1^c \cap \mathcal{E}_2^c) > 0$. The rest of the work in this proof is to show $\mathbb{P}(\mathcal{E}_1^c \cap \mathcal{E}_2^c) > 0$.

To bound $\mathbb{P}(\mathcal{E}_2)$, consider two distinct vertices $u$ and $v$. For $w \in V(G'_p)$, let $Y_w$ be the indicator random variable that takes value 1 if and only if $w \in N_{G'_p}(u)\Delta N_{G'_p}(v)$. Note that $\left|N_{G'_p}(u)\Delta N_{G'_p}(v)\right| = \sum\limits_{w\in V(G'_p)} Y_w$ and $\mathbb{P}(Y_w = 1) = 2 \cdot \frac{\kappa}{2}\left(1 - \frac{\kappa}{2}\right)$. So, the expected value of $\left|N_{G'_p}(u)\Delta N_{G'_p}(v)\right|$, that is,

$$\mathbb{E}\left[\left|N_{G'_p}(u)\Delta N_{G'_p}(v)\right|\right] = 2 \cdot \frac{\kappa}{2}\left(1 - \frac{\kappa}{2}\right) C_{\kappa,s}n.$$

As $C_{\kappa,s} = \lceil \frac{6s}{\kappa(2-\kappa)} \rceil$, $\mathbb{E}\left[\left|N_{G'_p}(u)\ \Delta\ N_{G'_p}(v)\right|\right] \geq 3sn$. Now applying the Chernoff bound (Lemma 2.11), we have

$$\mathbb{P}\left(\left|N_{G'_p}(u)\ \Delta\ N_{G'_p}(v)\right| < sn\right) \leq e^{-\frac{4sn}{9}}$$

Now, by using union bound, we can say that $\mathbb{P}(\mathcal{E}_2) \leq \left|V(G'_p)\right|^2 e^{-\frac{4sn}{9}} = C_{\kappa,s}^2 n^2 e^{-\frac{4sn}{9}}$. Finally using union bound one more time and the fact that $n$ is sufficiently large, we have

$$\mathbb{P}(\mathcal{E}_1 \cup \mathcal{E}_2) \leq C_{\kappa,s}n \cdot e^{-\frac{\varepsilon^2(2-\kappa)C_{\kappa,s}n}{6}} + C_{\kappa,s}^2 n^2 e^{-\frac{4sn}{9}} < 1.$$

Hence, $\mathbb{P}(\mathcal{E}_1^c \cap \mathcal{E}_2^c) > 0$. □

Let $ALG(\gamma_1, \gamma_2, T)$ be the algorithm that takes $\gamma_1$ and $\gamma_2$ with $0 < \gamma_1 < \gamma_2 < \varepsilon_0$ as input and decides whether $d(G_k, G_u) \leq \gamma_1 T n^2$ or $d(G_k, G_u) \geq \gamma_2 T n^2$, where $|V(G_k)| = |V(G_u)| = (T + 1)n$. Now we show that for any two constants $\beta_1$ and $\beta_2$ with $0 < \beta_1 < \beta_2 < \frac{\varepsilon_0}{2}$, $\kappa = \frac{\beta_2 - \beta_1}{8}$ and $T_\kappa = \lceil \frac{6s}{\kappa(2-\kappa)} \rceil$, there exists an algorithm $\mathcal{A}(\beta_1, \beta_2, \kappa, T_\kappa)$ that can test whether two multi-sets $S_k$ and $S_u$ over the $T_\kappa n$-dimensional Hamming cube have EMD less than $T_\kappa \beta_1 n^2$ or more than $T_\kappa \beta_2 n^2$ with $Q$ queries to the multi-set $S_u$. To be specific, algorithm $\mathcal{A}(\beta_1, \beta_2, \kappa, T_\kappa)$ for EMD testing will use algorithm $ALG(\gamma_1, \gamma_2, T)$ for $(\gamma_1, \gamma_2)$-tolerant GI such that $\gamma_1 = 2\beta_1$, $\gamma_2 = 2\beta_2 - 2\kappa$ and $T = T_\kappa$. Note that, as $0 < \beta_1 < \beta_2 < \frac{\varepsilon_0}{2}$ and $\kappa = \frac{\beta_2 - \beta_1}{8}$, $0 < \gamma_1 < \gamma_2 < \varepsilon_0$ holds. The details of the reduction, that is, algorithm $\mathcal{A}$ is described below.

**Description of the reduction**

**Input:** A known multi-set $S_k = \{k_1, \ldots, k_n\}$ over $H_{T_\kappa n} = \{0,1\}^{T_\kappa n}$ and query access to an unknown multi-set $S_u = \{u_1, \ldots, u_n\}$ over $H_{T_\kappa n}$.

**Goal:** To decide whether $d_{EM}(S_k, S_u) \leq T_\kappa \beta_1 n^2$ or $d_{EM}(S_k, S_u) \geq T_\kappa \beta_2 n^2$.

**Construction of $G_k$ and $G_u$ from $S_k$ and $S_u$:** Let us first construct the graph $G_k$ from $S_k$. $G_k$ has $(T_\kappa + 1)n$ vertices partitioned into two parts $A_k = \{a_1, \ldots, a_n\}$ and $B_k = \{b_1, \ldots, b_{T_\kappa n}\}$. Now the edges of $G_k$ are described as follows:

- $G_k[A_k]$ is a clique with $n$ vertices.
- $G_k[B_k]$ is a copy of $G_p(V_p, E_p)$ on $T_\kappa n$ vertices stated in Lemma 13.3 with parameters $s = 5$, $\kappa = \frac{\beta_2 - \beta_1}{8}$ and $T_\kappa = C_{\kappa,5}$.
- For the cross edges between the vertices in $A_k$ and $B_k$, we add the edge $(a_i, b_j)$ to $E(G_k)$ if and only if the $j$-th coordinate of $k_i$ is 1 for all $i \in [n]$ and $j \in [T_\kappa n]$.

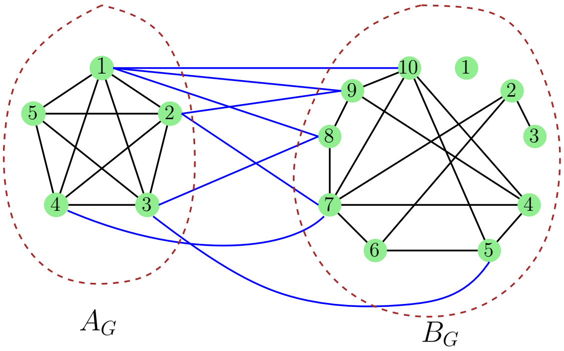

Figure 13.1: Construction of the graph $G \in \{G_k, G_u\}$

Note that the graph $G_k$ constructed above is unique for a given multi-set $S_k$. The graph $G_u$ with the vertex sets $A_u = \{a'_1, \ldots, a'_n\}$ and $B_u = \{b'_1, \ldots, b'_{T_\kappa n}\}$ is constructed from the multi-set $S_u$ in a similar fashion, but at the end, the vertices of $A_u$ are permuted using a random permutation. So,

- $G_u[A_u]$ is a clique with $n$ vertices.

- $G_u[B_u]$ is a copy of the graph $G_p(V_p, E_p)$ on $T_\kappa n$ vertices as stated in Lemma 13.3, with parameters $s = 5$, $\kappa = \frac{\beta_2 - \beta_1}{8}$ and $T_\kappa = C_{\kappa,5}$.

- Let us first pick a random permutation $\pi$ on $[n]$. For the cross edges between the vertices in $A_u$ and $B_u$, we add the edge $(a'_{\pi(i)}, b_j)$ to $E(G_u)$ if and only if the $j$-th coordinate of $u_i$ is 1 for all $i \in [n]$ and $j \in [T_\kappa n]$.

Note that our final objective is to prove a lower bound on the query complexity for tolerant testing of GI, that is, when we have an adjacency query access to $G_u$. We will instead show that the lower bound holds even if we have the following query access, named as *$A_u$-neighborhood-query*: the tester can choose a vertex $a'_i \in A_u$ and in one go obtain the information about the entire neighborhood of $a'_i$ in $B_u$.

Observe that the only part of $G_u$ that is not known to the tester is the cross edges between $A_u$ and $B_u$. So, in this case, the $A_u$-neighborhood query is way more stronger than the standard queries to $G_u$, and a lower bound for the $A_u$-neighborhood query would imply a lower bound on adjacency query.

**Simulating Queries to $G_u$ using samples drawn from $S_u$ without replacement:**

Following above discussion, we only need to show how to simulate $A_u$-neighborhood queries using samples drawn from $S_u$ **without** replacement. So, we can assume that the queries are of the form: *what are the neighbors of $a'_i$ in $B_u$?* And since in each query the entire neighborhood of $a'_i$ is obtained, the tester would pick different $a'_i$ for every query. Note that in $G_u$, by construction, the vertices of $A_u$ were permuted using a random permutation. So, from the point of view of the tester, the $a'_i$ are just randomly drawn from $A_u$ minus the set of $a'_i$ already queried. In other word, the $a'_i$ are just randomly drawn from $A_u$ **without** replacement. Now because of the way the edges between $A_u$ and $B_u$ are constructed, the neighborhood of a random $a'_i$ drawn from $A_u$ **without** replacement is same as obtaining random samples from $S_u$ **without** replacement.

It is also important to note that because of the randomness, the queries made by the tester are actually non-adaptive.

**Description of algorithm $\mathcal{A}$ for testing $d_{EM}(S_k, S_u)$**

Run ALG on $G_k$ and $G_u$ with parameters $\gamma_1 = 2\beta_1$ and $\gamma_2 = 2\beta_2 - 2\kappa$. If ALG reports $d(G_k, G_u) \leq T_\kappa \gamma_1 n^2$, output that $d_{EM}(S_k, S_u) \leq T_\kappa \beta_1 n^2$. Similarly, if ALG reports that $d(G_k, G_u) \geq T_\kappa \gamma_2 n^2$, then output $d_{EM}(S_k, S_u) \geq T_\kappa \beta_2 n^2$.

## 13.3 Correctness of our reduction

To prove the correctness of the above reduction, let us first consider the following definition of SPECIAL bijection and its connection with $d_{EM}(S_k, S_u)$.

**Definition 13.4** (Special bijections)**.** A bijection $\phi$ from $V(G_k)$ to $V(G_u)$ is said to be SPECIAL if $\phi(A_k) = A_u$, $\phi(B_k) = B_u$ and $\phi(b_i) = b'_i$ for all $b_i \in B_k$. The set of all special bijections from $V(G_k)$ to $V(G_u)$ will be denoted by $\Phi$, and $d_\Phi(G_k, G_u) := \min\limits_{\phi \in \Phi} d_\phi(G_k, G_u)$.

**Lemma 13.5.** *Let $S_k, S_u$ be the known and unknown multi-sets, respectively. Then $d_\Phi(G_k, G_u) = 2 \cdot d_{EM}(S_k, S_u)$.*

*Proof.* We will first prove that $d_\Phi(G_k, G_u) \leq 2 \cdot d_{EM}(S_k, S_u)$.

Recall that $S_k = \{k_1, \ldots, k_n\}$ and $S_u = \{u_1, \ldots, u_n\}$ be the known and unknown multi-sets over the Hamming cube $H_{T_\kappa n} = \{0,1\}^{T_\kappa n}$. Also, note that $G_u$ and $G_k$ are the unknown and known graphs with vertex bipartitions $A_u, B_u$ and $A_k, B_k$ respectively as discussed earlier. Let $\psi : S_k \to S_u$ be an optimal bijection that realizes $d_{EM}(S_k, S_u)$. Now, we will construct another bijection $\psi' \in \Phi$ such that $d_{\psi'}(G_k, G_u) = 2 \cdot d_{EM}(S_k, S_u)$.

We construct the bijection $\psi' \in \Phi$ from $V(G_k)$ to $V(G_u)$ as follows: for each $i, j \in [n]$, $\psi'(a_i) = a'_j$ if and only if $\psi(k_i) = u_j$; for each $\ell \in [T_\kappa n]$, $\psi'(b_\ell) = b'_\ell$. From the construction of $\psi'$ and by the definition of $d_{\psi'}(G_k, G_u)$ (See Definition 12.1), it is clear that $d_{\psi'}(G_k, G_u) = 2 \cdot d_{EM}(S_k, S_u)$. Since $d_\Phi(G_k, G_u) = \min\limits_{\phi \in \Phi} d_\phi(G_k, G_u)$, we can say $d_\Phi(G_k, G_u) \leq d_{\psi'}(G_k, G_u) = 2 \cdot d_{EM}(S_k, S_u)$.

Now we will prove the other way around, that is, we will show that $d_{EM}(S_k, S_u) \leq \frac{d_\Phi(G_k, G_u)}{2}$ holds as well. Let $\psi \in \Phi$ be a bijection from $V(G_k) \to V(G_u)$ that realizes

$d_\Phi(G_k, G_u)$. By definition of $\Phi$, we can assume that $\psi(b_i) = b'_i$ for each $i \in [T_\kappa n]$. Now, let us consider a bijection $\psi'$ from the multi-set $S_k$ to $S_u$ defined as follows: $\psi'(k_i) = u_j$ if and only if $\psi(a_i) = a'_j$ for all $i, j \in [n]$. Observe that $\sum\limits_{i \in [n]} d_H(k_i, \psi'(k_i)) = \frac{d_\psi(G_k, G_u)}{2}$. Thus, $d_{EM}(S_k, S_u) \leq \sum\limits_{i \in [n]} d_H(k_i, \psi'(k_i)) = \frac{d_\psi(G_k,G_u)}{2} = \frac{d_\Phi(G_k,G_u)}{2}$.

Putting everything together, we have $d_\Phi(G_k, G_u) = 2 \cdot d_{EM}(S_k, S_u)$. □

Using the following lemma, we will show how $d_\Phi(G_k, G_u)$ is related to $d(G_u, G_k)$, where $\Phi$ is the set of all SPECIAL bijections.

**Lemma 13.6.** *Let $\Phi$ be the set of all* SPECIAL *bijections from $V(G_k)$ to $V(G_u)$ and let $d_\Phi(G_k, G_u) = \min\limits_{\phi \in \Phi} d_\phi(G_k, G_u)$. Then we have $d_\Phi(G_k, G_u) - 2\kappa T_\kappa n^2 \leq d(G_k, G_u) \leq d_\Phi(G_k, G_u)$* [2].

*Proof.* Note that $d(G_k, G_u) \leq d_\Phi(G_k, G_u)$ follows from their definitions.

For the proof of the other side of the inequality, let us consider a bijection $\psi : V(G_k) \to V(G_u)$ that realizes $d(G_k, G_u)$, that is, $d(G_k, G_u) = d_\psi(G_k, G_u)$. If $\psi$ is a bijection such that $\psi \in \Phi$, then $d_\Phi(G_k, G_u) - 2\kappa T_\kappa n^2 \leq d(G_k, G_u)$ holds. So, let us assume that $\psi \notin \Phi$. Then we will show that there exists a bijection $\phi \in \Phi$ such that $d_\phi(G_k, G_u) \leq d_\psi(G_k, G_u) + 2\kappa T_\kappa n^2$, which will imply $d_\Phi(G_k, G_u) \leq d_\psi(G_k, G_u) + 2\kappa T_\kappa n^2$, that is, $d_\Phi(G_k, G_u) - 2\kappa T_\kappa n^2 \leq d(G_k, G_u)$.

We will now present the construction of $\phi \in \Phi$ from $\psi$. Let us first partition the vertices of $B_k$, with respect to $\psi$, into three parts: $B_k = B_{BI} \sqcup B_{BN} \sqcup B_A$; for each $b_i \in B_{BI}$, $\psi(b_i) = b'_i$; for each $b_i \in B_{BN}$, $\psi(b_i) \in B_u$ but $\psi(b_i) \neq b'_i$; for each $b_i \in B_A$, $\psi(b_i) \in A_u$. Also, we partition the vertices of $A_k$ into two parts: $A_k = A_A \sqcup A_B$; for each $a_i \in A_A$, $\psi(a_i) \in A_u$; for each $a_i \in A_B$, $\psi(a_i) \in B_u$. Let $|B_A| = |A_B| = x$ and $|B_{BN}| = y$, where $0 \leq x \leq n$ and $0 \leq x + y \leq T_\kappa n$. Now, we will construct the bijection $\phi \in \Phi$ (from $\psi$) by performing the following three steps in that order. Note that the construction of $\phi$ is not a part of our reduction. This is used for analysis purpose only.

[2] Note that this relation does not hold in general. However this is true for the graphs $G_k$ and $G_u$ constructed in the reduction.

**Step (i)** $\phi(u) = \psi(u)$ for all vertices $u \in B_{BI} \cup A_A$.

**Step (ii)** For each $a_i \in A_B$, $\phi(a_i) \in A_u \setminus \psi(A_A)$. Also, for each $b_i \in B_A$, $\phi(b_i) = b'_i \in B_u \setminus \psi(B_{BI})$.

**Step (iii)** For each $b_i \in B_{BN}$, $\phi(b_i) = b'_i$.

Observe that $\phi(A_k) = A_u$, $\phi(B_k) = B_u$ and $\phi(b_i) = b'_i$ for all $b_i \in B_k$, that is, $\phi$ is a SPECIAL bijection. It remains to show that

$$d_\Phi(G_k, G_u) \leq d_\psi(G_k, G_u) + 2\kappa T_\kappa n^2. \tag{13.1}$$

Recall that the graphs $G_k[B_k]$ and $G_u[B_u]$ are the *same* copies of $G_p(V_p, E_p)$, where $|V_p| = T_\kappa n$. Observe that

- From Lemma 13.3, the graphs $G_k[B_k]$ and $G_u[B_u]$ satisfy the following property[3]: cardinality of symmetric difference between the sets of neighbors of any two distinct vertices is at least $5n - 2$.

- Since $G_k[A_k]$ and $G_u[A_u]$ are cliques, the degree of each vertex in graphs $G_k[A_k]$ and $G_u[A_u]$ is exactly $n - 1$.

To prove $d_\Phi(G_k, G_u) \leq d_\psi(G_k, G_u) + 2\kappa T_\kappa n^2$, it will be sufficient to show that

$$d_\phi(G_u, G_k) \leq d_\psi(G_u, G_k) + 4x(|A_k| + 1) + 2xy + x(x-1) + 2y\,|A_k| - y(5n-2). \tag{13.2}$$

From Equation 13.2, we will be done with the proof of Inequality 13.1 as

$$\begin{aligned}
d_\phi(G_u, G_k) &\leq d_\psi(G_u, G_k) + 4x\,|A_k| + 4x + 2xy + x(x-1) + 2y\,|A_k| - y(5n-2) \\
&= d_\psi(G_k, G_u) + 4xn + 4x + 2xy + n(n-1) + 2ny - y(5n-2) \\
&\leq d_\psi(G_k, G_u) + 4n^2 + 4n + 2ny + n^2 + 2ny - y(5n-2) \\
&\leq d_\psi(G_k, G_u) + 8n^2 \\
&\leq d_\psi(G_k, G_u) + 2\kappa T_\kappa n^2.
\end{aligned}$$

[3] Note that we are using Lemma 13.3 with parameters $s = 5$, $\kappa = \frac{\beta_2 - \beta_1}{8}$ and $T_\kappa = C_{\kappa,5}$.

The last but one inequality follows from the fact that $0 \leq x \leq n$ and the last inequality follows from the fact that $T_\kappa = \lceil \frac{30}{\kappa(2-\kappa)} \rceil$.

Now we present the proof of Inequality 13.2.

*Proof of Inequality* (13.2). Here we prove that

$$d_\phi(G_u, G_k) \leq d_\psi(G_u, G_k) + 4x(|A_k|+1) + 2xy + x(x-1) + 2y\,|A_y| - y(5n-2). \quad (13.3)$$

Instead of directly proving the above inequality, we will prove it in four steps for better exposition. In Step 1, we prove the inequality for $x = 1, y = 0$. Then we generalize it for $x \leq n, y = 0$, followed by $x = 0, y \leq T_\kappa n$. Finally, combining Steps 1, 2 and 3, we prove the inequality for any $0 \leq x \leq n$, and $0 \leq y \leq T_\kappa n$.

**Step** $1$ $(x = 1, y = 0)$**:** So, let us assume that $a_i \in A_k$, $a'_j \in A_u$, $b_s \in B_k$ and $b'_s \in B_u$ be such that the following holds: $\psi(a_i) = b'_s$ and $\psi(b_s) = a'_j$, $\psi(z) \in A_u$ for each $z \in A_k \setminus \{a_i\}$, and $\phi(b_t) = b'_t \in B_u$ for each $b_t \in B_k \setminus \{b_s\}$. By the description of Steps (i), (ii) and (iii) of generating $\phi$ from $\psi$, as discussed in Lemma 13.6, we have the following observation.

**Observation 13.7.** For $x = 1$ and $y = 0$, we have $\psi(a_i) = b'_s$ and $\psi(b_s) = a'_j$; $\phi(a_i) = a'_j$ and $\phi(b_s) = b'_s$; For any $z \in (A_k \cup B_k) \setminus \{a_i, b_s\}$, $\phi(z) = \psi(z)$.

We can think of $\phi$ is generated by performing a *swap* operation, that means, the mappings of $a_i$ and $b_s$ are swapped while generating $\phi$ from $\psi$. Now we show (for the special case of $x = 1$ and $y = 0$) that:

$$d_\phi(G_k, G_u) \leq d_\psi(G_k, G_u) + 4(|A_k| + 1). \quad (13.4)$$

By Observation 13.7, $\phi(x) = \psi(x)$ for all vertices $x \in (A_k \cup B_k) \setminus \{a_i, b_s\}$. So, any pair of vertices in $(A_k \cup B_k) \setminus \{a_i, b_s\}$ has no effect on $d_\phi(G_u, G_k) - d_\psi(G_u, G_k)$. Following

Definition 12.1 and Definition 12.9, we can say that

$$d_\phi(G_u, G_k) - d_\psi(G_u, G_k) \le 2\Big( |\textsc{Decider}_\phi(a_i)| - |\textsc{Decider}_\psi(a_i)| \\ + |\textsc{Decider}_\phi(b_s)| - |\textsc{Decider}_\psi(b_s)| \Big)$$

Note that the first term above can be written as $\textsc{Decider}_\phi(a_i) = (\textsc{Decider}_\phi(a_i) \cap (A_k \cup \{b_s\})) \cup (\textsc{Decider}_\phi(a_i) \cap (B_k \setminus \{b_s\}))$. Breaking other terms in the above expression similarly, we have

$$\begin{aligned} & d_\phi(G_u, G_k) - d_\psi(G_u, G_k) \\ & \le 2\Big[2\big( |A_k| + 1\big) + |\textsc{Decider}_\phi(a_i) \cap (B_k \setminus \{b_s\})| - |\textsc{Decider}_\psi(a_i) \cap (B_k \setminus \{b_s\})| \\ & \qquad + |\textsc{Decider}_\phi(b_s) \cap (B_k \setminus \{b_s\})| - |\textsc{Decider}_\psi(b_s) \cap (B_k \setminus \{b_s\})| \Big] \\ & = 4\,|A_k| + 4 + 2Z, \text{ where} \\ & \qquad Z = |\textsc{Decider}_\phi(a_i) \cap (B_k \setminus \{b_s\})| - |\textsc{Decider}_\psi(a_i) \cap (B_k \setminus \{b_s\})| \\ & \qquad\quad + |\textsc{Decider}_\phi(b_s) \cap (B_k \setminus \{b_s\})| - |\textsc{Decider}_\psi(b_s) \cap (B_k \setminus \{b_s\})| \end{aligned}$$

By showing $Z \le 0$, we will be done with the proof of Inequality (13.4). Observe that we can say $|\textsc{Decider}_\phi(a_i) \cap (B_k \setminus \{b_s\})| = \left|\phi\left(N_{B_k \setminus \{b_s\}}(a_i)\right) \Delta N_{B_u \setminus \{b'_s\}}(\phi(a_i))\right|$. Also, writing the other terms in the expression of $Z$ in the similar fashion, we get

$$\begin{aligned} Z \le & \left|\phi(N_{B_k \setminus \{b_s\}}(a_i)) \Delta \left(N_{B_u \setminus \{b'_s\}}(\phi(a_i))\right)\right| - \left|\psi\left(N_{B_k \setminus \{b_s\}}(a_i)\right) \Delta \left(N_{B_u \setminus \{b'_s\}}(\psi(a_i))\right)\right| \\ & + \left|\phi\left(N_{B_k \setminus \{b_s\}}(b_s)\right) \Delta \left(N_{B_u \setminus \{b'_s\}}(\phi(b_s))\right)\right| - \left|\psi\left(N_{B_k \setminus \{b_s\}}(b_s)\right) \Delta \left(N_{B_u \setminus \{b'_s\}}(\psi(b_s))\right)\right| \end{aligned}$$

Once again, from Observation 13.7,

$$\begin{aligned} \phi(N_{B_k \setminus \{b_s\}}(a_i)) &= \psi(N_{B_k \setminus \{b_s\}}(a_i)) \text{ (Say } I_1) \\ N_{B_u \setminus \{b'_s\}}(\phi(a_i)) &= N_{B_u \setminus \{b'_s\}}(\psi(b_s)) \text{ (Say } I_2) \\ \phi(N_{B_k \setminus \{b_s\}}(b_s)) &= \phi\left(N_{B_k \setminus \{b_s\}}(b_s)\right) \text{ (Say } I_3) \\ N_{B_u \setminus \{b'_s\}}(\psi(a_i)) &= N_{B_u \setminus \{b'_s\}}(\phi(b_s)) \text{ (Say } I_4) \end{aligned}$$

From our above derivation, $|I_3 \Delta I_4| = |\text{DECIDER}_\phi(b_s) \cap (B_k \setminus \{b_s\})|$. Since $y = 0$, we have

$$|\text{DECIDER}_\phi(b_s) \cap (B_k \setminus \{b_s\})| = 0.$$

So, to prove $Z \leq 0$, it is enough to show $Z \leq 2\,|I_3 \Delta T_4|$. Note that

$$Z \leq |I_1 \Delta I_2| - |I_1 \Delta I_4| + |I_3 \Delta I_4| - |I_3 \Delta I_2|\,.$$

By using triangle inequality, $Z$ can be upper bounded as follows:

$$Z \leq |I_2 \Delta I_4|) + |I_3 \Delta I_4| - |I_3 \Delta I_2| \leq |I_3 \Delta I_4| + |I_3 \Delta I_4| = 2\,|I_3 \Delta I_4| = 0.$$

**Step** $2$ ($x \leq n, y = 0$)**:** Let us consider $A_B \subseteq A_k$ and $B_A \subseteq B_k$ such that $\psi(a_i) \in B_u$ for each $a_i \in A_B$, $\psi(b_s) \in A_u$ for each $b_s \in B_A$, $\psi(a_i) \in A_u$ for each $a_i \in A_k \setminus A_B$, and $\psi(b_s) \in B_u$ for each $b_s \in B_k \setminus B_A$. Now let us consider *swapping* (described below) the mapping of $a_i \in A_B$ and $b_s \in B_A$ such that $\psi(a_i) = b_s$. Let $a'_j \in A_u$ be such that $\psi(b_s) = a'_j$. Let us construct $\phi_{x-1} : V(G_k) \to V(G_k)$ from $\phi_x = \psi$ such that the followings hold: $\phi_{x-1}(a_i) = a'_j$, $\phi_1(b_s) = b'_s$, and $\phi_{x-1}(z) = \psi(z)$ for each $z \in (A_k \cup B_k) \setminus \{a_i, b_s\}$. Proceeding in the similar fashion as in the case when $x = 1$ and $y = 0$, we get

$$d_{\phi_{x-1}}(G_u, G_k) - d_\psi(G_u, G_k) \leq 4\,|A_k| + 4 + 2\,|I_3 \Delta I_4|\,,$$

where $|I_3 \Delta I_4| = |\text{DECIDER}_\phi(b_s) \cap (B_k \setminus \{b_s\})| \leq x - 1$. So,

$$d_{\phi_{x-1}}(G_u, G_k) \leq d_\psi(G_u, G_k) + 4\,|A_k| + 4 + 2(x-1).$$

We can proceed in the similar fashion by performing swapping operation of the vertices in $A_B$ and $Y_k$ one by one, and construct $\phi_x = \psi, \phi_{x-1}, \phi_{x-2}, \ldots, \phi_0 = \phi$. Observe that $d_{\phi_{i-1}}(G_u, G_k) \leq d_{\phi_i}(G_u, G_k) + 4\,|A_k| + 4 + 2(i-1)$. Also, note that $\phi$ is a SPECIAL bijection, and moreover

$$d_\phi(G_u, G_k) \leq 4x\,|A_k| + 4x + x(x-1).$$

**Step** 3 **($x = 0, y \leq T_\kappa n$):** Let us consider $B_{BN} \subseteq B_k$ such that $|B_{BN}| = y$. Note that for each $b_s \in B_{BN}$, $\psi(b_s) \neq b'_s$. Consider $b_s \in B_{BN}$ such that $\psi(b_s) = b'_i$, and let $b_j$ be such that $\psi(b_j) = b'_s$. Let us construct $\phi_{y-1} : V(G_u) \to V(G_k)$ from $\phi_y = \psi$ as follows: $\phi_{y-1}(b_s) = b'_s$, $\phi_{y-1}(b_j) = b'_i$, and $\phi_{y-1}(z) = \psi(z)$ for each $z \in (A_k \cup B_k) \setminus \{b_s, b_j\}$. Thus,

$$d_{\phi_{y-1}}(G_u, G_k) \leq d_{\phi_y}(G_u, G_k) + 2\,|A_k| - (5n-2)$$

The term $2\,|A_k|$ corresponds to the fact that any vertex of $B_{BN}$ has at most $|A_k|$ neighbors in $A_k$. The second term comes due to the properties of the probabilistic construction of $B_k$ and $B_u$ following Lemma 13.3.

**Step** 4 **($x \leq n, y \leq T_\kappa n$):** Let us assume $\psi(a_i) = b'_s$. Now there are two possibilities:

(1) $\psi(b_s) = a'_j$.

(2) $\psi(b_s) = b'_t$.

For (1), following the discussion of $x \leq n, y = 0$, we can say that

$$d_{\phi_{x-1,y}} \leq d_\psi(G_u, G_k) + 4(|A_k| + 1) + 2(x + y - 1).$$

For (2), we follow the discussion of $x = 0, y \leq T_\kappa n$, and the following holds:

$$d_{\phi_{x,y-1}}(G_u, G_k) \leq d_\psi(G_u, G_k) + 2\,|A_k| - (5n-2).$$

Putting everything together, we have

$$d_\phi(G_u, G_k) \leq d_\psi(G_u, G_k) + 4x(|A_k| + 1) + 2xy + x(x-1) + 2y\,|A_y| - y(5n-2).$$

□

The following lemma completes the proof of Lemma 13.2.

**Lemma 13.8.** *The described algorithm $\mathcal{A}$ for EMD, that uses Algorithm* ALG *on $G_k$ and $G_u$ with parameters $\gamma_1$ and $\gamma_2$ as a subroutine, determines whether $d_{EM}(S_k, S_u) \leq \beta_1 T_\kappa n^2$ or $d_{EM}(S_k, S_u) \geq \beta_2 T_\kappa n^2$ with probability at least $2/3$, where $\gamma_1 = 2\beta_1$, $\gamma_2 = 2\beta_2 - 2\kappa$.*

*Proof.* By the assumption of the existence of algorithm ALG that decides whether $d(G_k, G_u) \leq T_\kappa \gamma_1 n^2$ or $d(G_k, G_u) \geq T_\kappa \gamma_2 n^2$, we will be done with the proof by showing the followings.

**(i)** If $d_{EM}(S_k, S_u) \leq T_\kappa \beta_1 n^2$, then $d(G_k, G_u) \leq T_\kappa \gamma_1 n^2$,

**(ii)** If $d_{EM}(S_k, S_u) \geq T_\kappa \beta_2 n^2$, then $d(G_k, G_u) \geq T_\kappa \gamma_2 n^2$.

We will first prove (i). From Lemma 13.5, we have $d_\Phi(G_k, G_u) = 2 \cdot d_{EM}(S_k, S_u)$, where $\Phi$ is the set of all SPECIAL bijections from $V(G_k)$ to $V(G_u)$. So, $d_{EM}(S_k, S_u) \leq T_\kappa \beta_1 n^2$ implies $d_\Phi(G_k, G_u) \leq 2T_\kappa \beta_1 n^2 = T_\kappa \gamma_1 n^2$. Now, following the definition of SPECIAL bijections (Definition 13.4) and Lemma 13.6, we can say that $d(G_k, G_u) \leq d_\Phi(G_k, G_u) \leq T_\kappa \gamma_1 n^2$.

Now, for the proof of (ii), considering the fact that $d_\Phi(G_k, G_u) = 2 \cdot d_{EM}(S_k, S_u)$ as above, we can say that $d_{EM}(S_k, S_u) \geq T_\kappa \beta_2 n^2$ implies $d_\Phi(G_k, G_u) \geq 2T_\kappa \beta_2 n^2$. From Lemma 13.6, it follows that $d_\Phi(G_k, G_u) - 2\kappa T_\kappa n^2 \leq d(G_k, G_u)$. Thus, $d(G_k, G_u) \geq T_\kappa(2\beta_2 - 2\kappa)n^2 = T_\kappa \gamma_2 n^2$. □

□

# Chapter 14

# Tolerant EMD testing is as hard as tolerant Graph Isomorphism

## 14.1 Introduction

In this chapter, we prove the following theorem, that discusses about algorithm for tolerant graph isomorphism testing with a blackbox access to tolerant EMD testing over multi-sets.

**Theorem 14.1.** *(Restatement of the upper bound part of Theorem 12.5) Let $G_k$ and $G_u$ be the known and unknown graphs, respectively. There exists an algorithm that takes parameters $\gamma_1$ and $\gamma_2$ as input such that $0 \leq \gamma_1 < \gamma_2 \leq 1$, performs $\widetilde{\mathcal{O}}\left(\text{QWoR}_{\text{EMD}}(n)\right)$ queries to the adjacency matrix of $G_u$ for appropriate $\beta_1$ and $\beta_2$ depending on $\gamma_1$ and $\gamma_2$, and decides whether $d(G_u, G_k) \leq \gamma_1 n^2$ or $d(G_u, G_k) \geq \gamma_2 n^2$, with probability at least $2/3$. Here $\widetilde{\mathcal{O}}(\cdot)$ hides a polynomial factor in $\frac{1}{\beta_2 - \beta_1}$ and $\log n$.*

**Remark 14.1.** The theorem stated above works for any $\gamma_1, \gamma_2$ such that $0 \leq \gamma_1 < \gamma_2 \leq 1$. However, for simplicity of representation, we have assumed $\gamma_2 \geq 11\gamma_1$.

**Remark 14.2.** Note that Theorem 14.1 can also be stated in terms of $\text{QWR}_{\text{EMD}}(n)$ as $\text{QWoR}_{\text{EMD}}(n) \leq \text{QWR}_{\text{EMD}}(n)$ as we can simulate samples **with** replacement when we have query access to samples **without** replacement (See Proposition 12.13).

Our algorithm for tolerant GI testing, as stated in Theorem 14.1, uses a special kind of tolerant EMD tester over multi-sets: we know $t$ multi-sets, one multi-set is unknown and two parameters $\varepsilon_1$ and $\varepsilon_2$ are given; the objective is to test tolerant EMD of each known multi-set with the unknown one. The following theorem gives us the special EMD tester.

**Theorem 14.2.** *Let $H = \{0,1\}^n$ be a $n$-dimensional Hamming cube. Let $\{S_k^i : i \in [t]\} \cup \{S_u\}$ denote the multi-sets with $n$ elements from $H$ where $\{S_k^i : i \in [t]\}$ denote the set of $t$ known multi-sets and $S_u$ denotes the unknown multi-set. There exists an algorithm* ALG-EMD *that takes two proximity parameters $\varepsilon_1, \varepsilon_2$ with $0 \leq \varepsilon_1 < \varepsilon_2 \leq 1$ and a $\delta \in (0,1)$ as input and decides whether $d_{EM}(S_u, S_k^i) \leq \varepsilon_1 n^2$ or $d_{EM}(S_u, S_k^i) \geq \varepsilon_2 n^2$, with probability at least $1-\delta$, for each $i \in [t]$. Moreover,* ALG-EMD *uses* $\text{QWoR}_{\text{EMD}}(n) \cdot \mathcal{O}\left(\log \frac{t}{\delta}\right)$ *samples* ***without*** *replacement from $S_u$.*

The above theorem follows from the definition of $\text{QWoR}_{\text{EMD}}(n)$ (See Definition 12.16) along with union bound and standard argument for amplifying the success probability.

**Remark 14.3.** The algorithm of Theorem 14.1, to be discussed in Section 14.2, formulates a tolerant EMD instance of multi-sets having $n$ elements in $H = \{0,1\}^d$, where $d = \mathcal{O}(\log n/(\gamma_2 - \gamma_1))$. But ALG-EMD is an algorithm for tolerant EMD testing between two multi-sets having $n$ elements in $\{0,1\}^n$. This is not a problem as the query complexity of EMD is an increasing function in dimension (See Proposition 12.17 in Section 12.3.2). Moreover, the algorithm in Section 14.2 calls ALG-EMD with parameters $\varepsilon_1 = (\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000})$, $\varepsilon_2 = \gamma_2/5$, $t = 2^{\mathcal{O}(\log^2 n/(\gamma_2 - \gamma_1))}$ and $\delta$ is a suitable constant depending upon $\gamma_1$ and $\gamma_2$, where $\gamma_1$ and $\gamma_2$ are parameters as stated in Theorem 14.1. So, each call to ALG-EMD, in our context, makes $\widetilde{\mathcal{O}}\left(\text{QWoR}_{\text{EMD}}(n)\right)$ queries.

## 14.2 Algorithm for tolerant GI testing

For our algorithm, we need the following definitions of *label* and *embedding*.

**Definition 14.3.** (*Label* of a vertex) Given a graph $G$ and $C \subset V(G) = \{c_1, \ldots c_{|C|}\}$, the $C$-labelling of $V(G)$ is a function $\mathcal{L}_C : V(G) \to \{0,1\}^{|C|}$ such that the $i$-th entry of $\mathcal{L}_C(v)$ is 1 if and only if $v$ is a neighbor of $c_i \in C$. Also, $\mathcal{L}_C(v)$ is referred as the label of $v$ under $C$-labelling of $V(G)$.

**Definition 14.4.** (*Embedding* of a Vertex Set into another Vertex Set) Let $G_u$ and $G_k$ be two graphs. Consider $A \subseteq V(G_u)$ and $B \subseteq V(G_k)$ such that $|A| \leq |B|$. An injective mapping $\eta$ from $A$ to $B$ is referred as an *embedding* of $A$ into $B$.

Now we present our query algorithm **TolerantGI($G_u$, $G_k$, $\gamma_1$, $\gamma_2$)** that comprises three phases. The technical overview of the algorithm is has been already presented in Section 12.4.1.

## Formal Description of TolerantGI($G_u$, $G_k$, $\gamma_1$, $\gamma_2$):

The three phases of our algorithm are as follows:

**Phase 1:** The first phase of our algorithm consists of the following three steps.

**Step 1** First we sample a collection $\mathcal{C}_u$ of $\mathcal{O}(\log n)$ sized random subsets of $V(G_u)$ with $|\mathcal{C}_u| = \mathcal{O}(\frac{1}{\gamma_2 - \gamma_1})$. We perform **Step 2** and **Step 3** for each $C_u \in \mathcal{C}_u$.

**Step 2** We determine all possible embeddings, that is, $\eta_1, \ldots, \eta_J$, of $C_u$ into $V(G_k)$, where $J = \binom{n}{\mathcal{O}(\log n)} \leq 2^{\mathcal{O}(\log^2 n)}$. For each $i \in [J]$, let $C_k^i$ be the set of images of $C_u$ under the $i$-th embedding of $C_u$ into $V(G_k)$, that is, $C_k^i = \eta_i(C_u)$. For all $i \in [J]$, we construct the multi-set $Y_{C_k^i}$ that contains $C_k^i$-labellings of all the vertices of $G_k$.

**Step 3** Now for each vertex $v \in V(G_u)$, there is a $C_u$-labelling of $v$. Let $X_{C_u}$ be the multi-set of $C_u$-labellings of all the vertices in $V(G_u)$. However, $X_{C_u}$ is unknown to the algorithm. We call ALG-EMD (as stated in Theorem 14.2) by setting parameters as described in Remark 14.3 to decide whether $d_{EM}(X_{C_u}, Y_{C_k^i}) \leq$

$(\gamma_1 + \frac{\gamma_2-\gamma_1}{2000})n\left|C_u\right|$ or $d_{EM}(X_{C_u}, Y_{C_k^i}) \geq \gamma_2 n\left|C_u\right|/5$, for each $i \in [J]$. Let us pair $C_u$'s and their accepted embeddings into $G_k$ and call the set $\Gamma$, that is,

$$\Gamma = \left\{(C_u, \eta_i) \mid \text{ALG-EMD } \textit{decides } d_{EM}(X_{C_u}, Y_{C_k^i}) \leq (\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000})n\left|C_u\right|\right\}.$$

Note that, at the end of the **Phase 1**, we have $\Gamma$ with $|\Gamma| \leq |\mathcal{C}_u| \cdot 2^{\mathcal{O}(\log^2 n)} = \mathcal{O}\left(2^{(\log^2 n)}\right)$. By the description of **Step 3** above, **Phase 1** of our algorithm calls ALG-EMD $\mathcal{O}(|\mathcal{C}_u|)$ times, once for each $C_u \in \mathcal{C}_u$. So, setting $\delta = \frac{1}{9|\Gamma|}$ in Theorem 14.2, we obtain the following observation about $\Gamma$ that will be used to prove the soundness of our algorithm.

**Observation 14.5.** Consider $\Gamma$, the set of accepted embeddings that have passed **Phase 1** paired with corresponding $C_u$, as defined above. Then

$$\mathbb{P}\left(\forall\ (C_u, \eta_i) \in \Gamma, d_{EM}(X_{C_u}, Y_{C_k^i}) \leq \gamma_2 n\left|C_u\right|/5\right) \geq \frac{8}{9}.$$

**Phase 2:** In the second phase, the algorithm performs the following two steps.

**Step 1** We sample a subset $W$ of $\mathcal{O}(\log^2 n/(\gamma_2 - \gamma_1)^3)$ vertices randomly from $G_u$.

**Step 2** For each $(C_u, \eta_i) \in \Gamma$ that has passed **Phase 1**, we perform the following steps:

**(i)** We find the $C_k^i = \eta_i(C_u)$-labelling of the vertices of $G_k$. Let $l_1, \ldots, l_t$ be the labels of the vertices where $t = 2^{|C_k^i|}$ and $V_j \subseteq V(G_k)$ be the set of vertices with label $l_j$.

**(ii)** We define a matrix $M$ of size $|W| \times 2^{|C_k^i|}$ where each row represents the label of a vertex $w \in W$ and each column represents one of the possible $C_k^i$-labelling of $V(G_k)$ [1]. The $(i,j)$-th entry of $M$ is defined as: $M_{ij} = d_H(\mathcal{L}_{C_u}(w_i), l_j)$.

[1] Let $C_u = \left\{x_1, \ldots, x_{\mathcal{O}(\log n/(\gamma_2-\gamma_1))}\right\}$. Note that for each $w_i \in W$, $\mathcal{L}_{C_u}(w_i) \in \{0,1\}^{\mathcal{O}(\log n/(\gamma_2-\gamma_1))}$ such that the $j$-th coordinate is 1 if and only if $w_i$ is a neighbour of $x_j$, where $i \in \left[\mathcal{O}(\log^2 n/(\gamma_2-\gamma_1)^3)\right]$ and $j \in \left[\mathcal{O}\left(\log n/(\gamma_2-\gamma_1)\right)\right]$. Similarly, $l_j \in \{0,1\}^{\mathcal{O}(\log n/(\gamma_2-\gamma_1))}$ such that the $i$-th coordinate of $l_j$ is 1 if and only if $\eta(x_i)$ is a neighbour of $v \in V_j$, where $j \in \left[2^{|C_k^i|}\right]$.

**(iii)** We choose a function $\psi : W \to \{l_1, \ldots l_t\}$ randomly satisfying the following two conditions:

$$\sum_{w \in W} d_H(\mathcal{L}_{C_u}(w), \psi(w)) \leq \frac{2\gamma_2}{5} |C_u| |W| \ \& \ |\{w : \psi(w) = l_j\}| \leq |V_j| \, \forall j \in [t]. \tag{14.1}$$

Let $\Gamma_W$ be the set of tuples such that

$$\Gamma_W = \{(C_u, \eta_i, \psi) : (C_u, \eta_i) \in \Gamma \text{ and } \psi \text{ satisfies Equation (14.1)}\}.$$

Like Observation 14.5, the following observation about the set $\Gamma_W$ will be used to prove the soundness of our algorithm.

**Observation 14.6.** $|\Gamma_W| \leq |\Gamma| \leq 2^{\mathcal{O}(\log^2 n)}$. Moreover, any $(C_u, \eta_i, \psi)$ that has passed this phase satisfies Equation (14.1).

**Phase 3:** The third phase of our algorithm comprises the following four steps.

Step 1 We randomly pair up the vertices of $W$. Let $\{(a_1, b_1), \ldots, (a_p, b_p)\}$ be the pairs of the vertices, where $p = \mathcal{O}(\log^2 n/(\gamma_2 - \gamma_1)^3)$. We now determine which $(a_i, b_i)$ pairs form edges in $G_u$ by querying the corresponding entries of the adjacency matrix of $G_u$.

Step 2 For each $(C_u, \eta_i, \psi) \in \Gamma_W$ that has passed **Phase 2**, we perform **Step 3** and **Step 4** as follows:

Step 3 We choose an embedding $\widehat{\phi} : W \to V(G_k)$ randomly, satisfying $\widehat{\phi}(w) \in V_j$ if and only if $\psi(w) = l_j$ and modulo permutation of the vertices in $V_j$ for all $j \in [t]$. In other words, we map each $w \in W$ to a vertex in $G_k$ randomly having $\psi(w) = l_j$ as its $C_k^i$-labelling in $G_k$.

Step 4 We compute $\zeta(C_u, \eta_i, \psi, \widehat{\phi}) = \left|\{(a_i, b_i) : \mathbb{1}_{(a_i,b_i)} = 1\}\right| / p$, where $\mathbb{1}_{(a_i,b_i)} = 1$ if exactly one among $(a_i, b_i) \in E(G_u)$ and $(\widehat{\phi}(a_i), \widehat{\phi}(b_i)) \in E(G_k)$ holds.

If $\zeta(C_u, \eta_i, \psi, \widehat{\phi}) \leq 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)$, then **HALT and REPORT** that $G_u$ and $G_k$ are $\gamma_1$-close.

While executing **Step 3** and **Step 4** for each tuple in $\Gamma_W$, if we did not **HALT**, then we **HALT** now and **REPORT** that $G_u$ and $G_k$ are $\gamma_2$-far.

**Observation 14.7.** (i) The number of times our algorithm executes **Step 2**, **Step 3** and **Step 4** is at most $|\Gamma_W| \leq 2^{\mathcal{O}(\log^2 n)}$.

(ii) If there exists a $(C_u, \eta_i, \psi)$ such that $\zeta(C_u, \eta_i, \psi, \widehat{\phi}) \leq 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)$, then our algorithm reports that $G_u$ and $G_k$ are $\gamma_1$-close. Otherwise, $G_u$ and $G_k$ are reported to be $\gamma_2$-far.

## 14.3 Proof of correctness

To prove the correctness of our algorithm, we need to show the following three properties:

**Completeness Property** If $G_u$ and $G_k$ are $\gamma_1$-close to isomorphic, then our algorithm reports the same with probability at least $2/3$.

**Soundness Property** If $G_u$ and $G_k$ are $\gamma_2$-far from isomorphic, then the algorithm reports the same with probability at least $2/3$.

**Query Complexity** The query complexity of our algorithm is $\widetilde{\mathcal{O}}(n)$.

### 14.3.1 Proof of completeness

In order to prove the completeness property as described above, we will first prove some claims. Finally, combining the claims, we would conclude the completeness property of our algorithm.

We will first prove that there exists a $C_u \in \mathcal{C}_u$ considered in **Step 1** of **Phase 1** of the algorithm and a corresponding embedding $\eta_i : C_u \to V(G_k)$ in **Step 2** of **Phase 1** such that $d_{EM}(X_{C_u}, Y_{C_k^i}) \leq (\gamma_1 + \frac{\gamma_2-\gamma_1}{2000})n\,|C_u|$ holds with probability at least $20/21$, where $C_k^i = \eta_i(C_u)$.

**Claim 14.8.** *Let $\phi : V(G_u) \to V(G_k)$ be a bijection such that $d_\phi(G_u, G_k) \leq \gamma_1 n^2$. Then there exists a $C_u \in \mathcal{C}_u$ and an embedding $\eta_i : C_u \to V(G_k)$ such that the following hold with probability at least $20/21$.*

- *$\forall v \in C_u$, we have $\eta_i(v) = \phi(v)$, and*
- *$d_{EM}(X_{C_u}, Y_{C_k^i}) \leq (\gamma_1 + \frac{\gamma_2-\gamma_1}{2000})n\,|C_u|$*

*Note that $C_k^i = \eta_i(C_u)$ and $Y_{C_k^i}$ is set of $C_k^i$-labelling of $V(G_k)$.*[2]

*Proof.* Consider a particular $C_u \in \mathcal{C}_u$ and an embedding $\eta_i : C_u \to V(G_k)$ such that $\eta_i(v) = \phi(v)$ for all $v \in C_u$. Note that this embedding $\eta_i$ is considered in **Step 2** of **Phase 1** of the algorithm. Now we will show that $d_{EM}(X_{C_u}, Y_{C_k^i}) \leq (\gamma_1 + \frac{\gamma_2-\gamma_1}{2000})n\,|C_u|$ holds with probability at least a constant, to be specified later, that depends upon $\gamma_1$ and $\gamma_2$, where $C_k^i = \eta_i(C_u)$.

We know that $d_\phi(G_u, G_k) \leq \gamma_1 n^2$ and by Definition 12.9, we have

$$\sum_{x \in V(G_u)} |\textsc{Decider}_\phi(x)| \leq \gamma_1 n^2.$$

Thus,

$$\mathbb{E}\left[\sum_{x \in V(G_u)} |\textsc{Decider}_\phi(x) \cap C_u|\right] \leq \gamma_1 n\,|C_u|\,. \tag{14.2}$$

From Definition 12.9, we can say that

$$\begin{aligned} d_{EM}(X_{C_u}, Y_{C_k^i}) &= \min_{f:V(G_u)\to V(G_k)} \sum_{x \in V(G_u)} |\textsc{Decider}_f(x) \cap C_u| \\ &\leq \sum_{x \in V(G_u)} |\textsc{Decider}_\phi(x) \cap C_u| \end{aligned}$$

[2] $C_k^i$ and $Y_{C_k^i}$ are defined in **Step 2** of **Phase 1**.

Therefore,

$$\begin{aligned}\mathbb{E}\left[d_{EM}(X_{C_u}, Y_{C_k^i})\right] &\leq \mathbb{E}\left[\sum_{x \in V(G_u)} |\text{DECIDER}_{\phi}(x) \cap C_u|\right] \\ &\leq \gamma_1 n \left|C_u\right| \quad \text{(From Equation (14.2))}\end{aligned}$$

Using Markov inequality, we can say that

$$\mathbb{P}\left(d_{EM}(X_{C_u}, Y_{C_k^i}) \leq (\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000})n \left|C_u\right|\right) \geq 1 - \frac{\gamma_1}{\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000}}.$$

Note that $|\mathcal{C}_u| = \mathcal{O}(\frac{1}{\gamma_2 - \gamma_1})$ and we have been arguing for a particular $C_u \in \mathcal{C}_u$. So, taking $|\mathcal{C}_u|$ suitably, we get a $C_u$ and an embedding $\eta_i : C_u \to V(G_k)$ satisfying the properties mentioned in the statement of this claim with probability at least $20/21$. □

The above claim discusses about the existence of a $C_u \in \mathcal{C}_u$ and its embeddings satisfying above mentioned desired properties. Now we discuss how our algorithm determines all $C_u \in \mathcal{C}_u$ that satisfy the properties. Note that **Step 3** of **Phase 1** of our algorithm calls ALG-EMD. Following the correctness of ALG-EMD (Theorem 14.2), we determine all embeddings $\eta_i : C_u \to V(G_k)$ such that $d_{EM}(X_{C_u}, Y_{C_k^i}) \leq (\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000})n \left|C_u\right|$ holds with probability at least $20/21$. The discussion in this paragraph is formalized in the following claim.

**Claim 14.9.** *Let $C_u \in \mathcal{C}_u$ and $\eta_1, \ldots, \eta_J$ be the all possible embeddings of $C_u$ into $V(G_k)$. Then* **Step 3** *of* **Phase 1** *can determine the set $\Gamma = \{(C_u, \eta_i) \mid d_{EM}(X_{C_u}, Y_{C_k^i}) \leq (\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000})n \left|C_u\right|\}$ with probability at least $20/21$. Note that $C_k^i = \eta_i(C_u)$, $X_{C_u}$ is the set of $C_u$-labelling of $V(G_u)$ and $Y_{C_k^i}$ is set of $C_k^i$-labelling of $V(G_k)$.*

As we are considering the case that $G_u$ and $G_k$ are $\gamma_1$-close to being isomorphic, from Claim 14.8, we can assume that there is an appropriate $(C_u, \eta_i) \in \Gamma$ such that $d_{EM}(X_{C_u}, Y_{C_k^i}) \leq (\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000})n \left|C_u\right|$. Now we will prove that there exists a function $\psi : W \to \{l_1, \ldots, l_t\}$ as considered in **Step 2 (iii)** in **Phase 2** of our algorithm such that Equation (14.1) holds with probability at least $20/21$.

**Claim 14.10.** *Let $\phi : V(G_u) \to V(G_k)$ be a bijection such that $d_\phi(G_u, G_k) \le \gamma_1 n^2$ and $(C_u, \eta_i) \in \Gamma$ where $C_u \in \mathcal{C}_u$ and $\eta_i : C_u \to V(G_k)$ be an embedding such that*

**(a)** *$\forall v \in C_u$ we have $\eta_i(v) = \phi(v)$, and*

**(b)** *$d_{EM}(X_{C_u}, Y_{C_k^i}) \le (\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000}) n \left|C_u\right|$ where $C_k^i = \eta_i(C_u)$.*

*Also, let $\{\ell_1, \ldots, \ell_t\}$ be the all possible $C_k^i$-labellings of $V(G_k)$, where $t = \left\lceil 2^{\left|C_k^i\right|} \right\rceil$. Then there exists a mapping $\psi : W \to \{l_1, \ldots, l_t\}$ such that the following hold with probability at least $20/21$.*

**(i)** *$\sum\limits_{w \in W} d_H(\mathcal{L}_{C_u}(w), \psi(w)) \le \frac{2\gamma_2}{5} \left|C_u\right| \left|W\right|$, and*

**(ii)** *$\forall j \in [t]$, we have $\left|\{w : \psi(w) = l_j\}\right| \le \left|V_j\right|$.*

*Proof.* From the conditions given in the statement of the claim, we can say that there exists $f : V(G_u) \to V(G_k)$ such that $f(v) = \eta_i(v) = \phi(v)$ for all $v \in C_u$ and $\sum\limits_{x \in V(G_u)} \left|\textsc{Decider}_f(x) \cap C_u\right| \le (\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000}) n \left|C_u\right|$

Since $\left|\textsc{Decider}_f(x) \cap C_u\right| = d_H(\mathcal{L}_{C_u}(x), \mathcal{L}_{C_k^i}(f(x)))$, we have

$$\sum_{x \in V(G_u)} d_H(\mathcal{L}_{C_u}(x), \mathcal{L}_{C_k^i}(f(x))) \le (\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000}) n \left|C_u\right|$$

Since we are taking the vertices in $W$ uniformly at random from $G_u$, we can say that

$$\mathbb{E}\left[\sum_{w \in W} d_H(\mathcal{L}_{C_u}(w), \mathcal{L}_{C_k^i}(f(w)))\right] \le (\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000}) \left|C_u\right| \left|W\right|$$

Using Hoeffding's inequality, we have

$$\mathbb{P}\left(\sum_{w \in W} d_H(\mathcal{L}_{C_u}(w), \mathcal{L}_{C_k^i}(f(w))) \le \frac{2\gamma_2}{5} \left|C_u\right| \left|W\right|\right) \ge 1 - e^{-\mathcal{O}(|W|)}$$

Now, we define $\psi : W \to \{\ell_1, \ldots, \ell_t\}$ such that $\psi(w) = \mathcal{L}_{C_k^i}(f(w))$. In other words, the $C_k^i$-labelling of $f(w)$ is same as the labelling of $\psi(w)$ for each $w \in W$. Thus, the $\psi$

defined here satisfies the Condition $(i)$ of this claim, that is,

$$\sum_{w \in W} d_H(\mathcal{L}_{C_u}(w), \psi(w)) \leq \frac{2\gamma_2}{5} |C_u| |W| .$$

Observe that

$$\left|\{w \in W : \mathcal{L}_{C_k^i}(f(w)) = l_j\}\right| \leq \left|\{v \in V(G_k) : \mathcal{L}_{\mathcal{C}_k^i}(v) = l_j\}\right| \leq |V_j| .$$

So, by the definition of $\psi$, $|\{w \in W : \psi(w) = l_j\}| \leq |V_j|$. Hence $\psi$ considered above also satisfies Condition $(ii)$ of the claim. □

Now consider the situation when the algorithm is at **Step 1** of **Phase 3**. If $G_u$ and $G_k$ are $\gamma_1$-close, that is, there exists a bijection $\phi$ from $V(G_u)$ to $V(G_k)$ such that $d_\phi(G_u, G_k) \leq \gamma_1 n^2$, then there exists $C_u \in \mathcal{C}_u$, $\eta_i : C_u \to V(G_k)$, and $\psi$ satisfying the conditions given in Claims 14.8 and 14.10. However, we do not know $\phi$. If we construct, though inefficiently, a bijection $\phi'$ that is same as $\phi$ with respect to the same $C_u \in \mathcal{C}_u$, $\eta_i : C_u \to V(G_k)$ and $\psi$ (conditions given in Claims 14.8 and 14.10), then the following claim says that the difference between $d_{\phi'}(G_u, G_k)$ and $d_\phi(G_u, G_k)$ is not too *large*.

**Claim 14.11.** *Let $\phi : V(G_u) \to V(G_k)$ be a bijection such that $d_\phi(G_u, G_k) \leq \gamma_1 n^2$, and $(C_u, \eta_i) \in \Gamma$ where $C_u \in \mathcal{C}_u$ and $\eta_i : C_u \to V(G_k)$ be an embedding such that*

- *$\forall\, v \in C_u$ we have $\eta_i(v) = \phi(v)$, and*
- *$d_{EM}(X_{C_u}, Y_{C_k^i}) \leq (\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000}) n \, |C_u|$ where $C_k^i = \eta_i(C_u)$.*

*Let $\{\ell_1, \ldots, \ell_t\}$ be the all possible $C_k^i$-labellings of the vertices of $G_k$ where $t = \left[2^{|C_k^i|}\right]$, and $W$ be the set of vertices of $G_u$ sampled at random in* **Step 1** *of* **Phase 2** *and $\psi : W \to \{\ell_1, \ldots, \ell_t\}$ be the mapping considered in* **Step 2 (iii)** *in* **Phase 2** *such that*

- *$\sum\limits_{w \in W} d_H(\mathcal{L}_{C_u}(w), \psi(w)) \leq \frac{2\gamma_2}{5} |C_u| |W|$, and*
- *$\forall j \in [t]$, we have $|\{w : \psi(w) = l_j\}| \leq |V_j|$.*

*Then, with probability at least* $18/21$*, there exists a bijection* $\phi' : V(G_u) \to V(G_k)$*, with* $\phi'(x) = \phi(x) = \eta_i(x)$ *for each* $x \in C_u$ *and* $\phi'(w) = \widehat{\phi}(w)$ *for each* $w \in W$ *such that*

$$d_{\phi'}(G_u, G_k) \leq d_{\phi}(G_u, G_k) + (4\gamma_1 + \frac{\gamma_2 - \gamma_1}{2})n^2.$$

*Proof.* We will prove the claim by contradiction. Suppose that

$$d_{\phi'}(G_u, G_k) > d_{\phi}(G_u, G_k) + (4\gamma_1 + \frac{\gamma_2 - \gamma_1}{2})n^2 \tag{14.3}$$

By using Definition 12.9, we write the above equation as

$$\sum_{x \in V(G_u)} |\textsc{Decider}_{\phi'}(x)| > \sum_{x \in V(G_u)} |\textsc{Decider}_{\phi}(x)| + (4\gamma_1 + \frac{\gamma_2 - \gamma_1}{2})n^2$$

So,

$$\sum_{x \in V(G_u)} |\textsc{Decider}_{\phi'}(x) \Delta \textsc{Decider}_{\phi}(x)| > (4\gamma_1 + \frac{\gamma_2 - \gamma_1}{2})n^2$$

Let us denote $\textsc{Decider}_{\phi'}(x) \Delta \textsc{Decider}_{\phi}(x) = \text{Symm}_{\phi\phi'}(x)$. Dividing the sum in the left hand side with respect to the values of $|\textsc{Decider}_{\phi'}(x) \Delta \textsc{Decider}_{\phi}(x)|$'s, that is, $\left|\text{Symm}_{\phi\phi'}(x)\right|$'s, we get

$$\begin{aligned} \sum_{\substack{x \in V(G_u) \\ |\text{Symm}_{\phi\phi'}(x)| \geq \frac{(\gamma_2 - \gamma_1)n}{1000}}} \left|\text{Symm}_{\phi\phi'}(x)\right| \quad &+ \sum_{\substack{x \in V(G_u) \\ |\text{Symm}_{\phi\phi'}(x)| < \frac{(\gamma_2 - \gamma_1)n}{1000}}} \left|\text{Symm}_{\phi\phi'}(x)\right| \\ &> \quad (4\gamma_1 + \frac{\gamma_2 - \gamma_1}{2})n^2 \end{aligned}$$

Note that the second sum of the left hand side is at most $\frac{\gamma_2 - \gamma_1}{1000}n^2$. Therefore,

$$\sum_{\substack{x \in V(G_u): \\ |\text{Symm}_{\phi\phi'}(x)| \geq \frac{(\gamma_2 - \gamma_1)n}{1000}}} \left|\text{Symm}_{\phi\phi'}(x)\right| > (4\gamma_1 + \frac{\gamma_2 - \gamma_1}{2})n^2 - \frac{\gamma_2 - \gamma_1}{1000}n^2 \tag{14.4}$$

Before proceeding further, consider the following observation, which follows from standard Chernoff bound type argument.

**Observation 14.12.** If $\left|\text{Symm}_{\phi\phi'}(x)\right| \geq \frac{(\gamma_2-\gamma_1)n}{1000}$, then

$$\mathbb{P}\left(\left|\text{Symm}_{\phi\phi'}(x) \cap C_u\right| \geq (1-\frac{1}{50})\left|\text{Symm}_{\phi\phi'}(x)\right|\frac{|C_u|}{n}\right) \leq e^{-\mathcal{O}(|C_u|)}.$$

This implies that the following holds with probability at least $1-ne^{-\mathcal{O}(|C_u|)}$:

$$\begin{aligned}\sum_{\substack{x\in V(G_u):\\ \left|\text{Symm}_{\phi\phi'}(x)\right|\geq\\ \frac{(\gamma_2-\gamma_1)n}{1000}}} \left|\text{Symm}_{\phi\phi'}(x)\cap C_u\right| &\geq \left(1-\frac{1}{50}\right)\frac{|C_u|}{n}\sum_{\substack{x\in V(G_u):\\ \left|\text{Symm}_{\phi\phi'}(x)\right|\geq\\ \frac{(\gamma_2-\gamma_1)n}{1000}}}\left|\text{Symm}_{\phi\phi'}(x)\right| \\ &= \frac{49}{50}\left(4\gamma_1+\frac{499(\gamma_2-\gamma_1)}{1000}\right)n\left|C_u\right|\end{aligned}$$

The last line follows from Equation (14.4). Hence, with probability at least $1-ne^{-\mathcal{O}(|C_u|)}$, the following event holds.

$$\sum_{x\in V(G_u)}\left|\text{Symm}_{\phi\phi'}(x)\cap C_u\right| \geq \frac{49}{50}\left(4\gamma_1+\frac{499(\gamma_2-\gamma_1)}{1000}\right)n\left|C_u\right|. \tag{14.5}$$

Assuming Equation (14.5) holds and using the fact that $W \subset V(G_u)$ is taken uniformly at random, we can say that

$$\mathbb{E}\left[\sum_{w\in W}\left|\text{Symm}_{\phi\phi'}(x)\cap C_u\right|\right] > \frac{49}{50}(4\gamma_1+\frac{499(\gamma_2-\gamma_1)}{1000})\left|C_u\right|\left|W\right|.$$

Using the Hoeffding's inequality (Lemma 2.14), we get

$$\begin{aligned}\mathbb{P}\left(\sum_{w\in W}\left|\text{Symm}_{\phi\phi'}(w)\cap C_u)\right| \leq (3\gamma_1+\frac{11(\gamma_2-\gamma_1)}{24})\left|C_u\right|\left|W\right|\right) &\leq e^{-\mathcal{O}(\frac{|C_u|^2|W|^2}{|W||C_u|^2})} \\ &= e^{-\mathcal{O}(|W|)}\end{aligned}$$

As the above equation holds in the conditional space that Equation (14.5) holds, we have:

$$\mathbb{P}\left(\sum_{w\in W}\left|\text{Symm}_{\phi\phi'}\cap C_u)\right| > (3\gamma_1+\frac{11(\gamma_2-\gamma_1)}{24})\left|C_u\right|\left|W\right|\right) \geq 1-\frac{n}{e^{\mathcal{O}(|C_u|)}}-\frac{1}{e^{\mathcal{O}(|W|)}} \tag{14.6}$$

Note that Equation (14.3) implies Equation (14.6). However, till now, we have not used any information given in the statement of Claim 14.11, except that $C_u$ and $W$ are taken uniformly at random. By using the fact that the sum of label differences of the vertices of $W$ under $C_u$-labelling and that of $\psi$ is bounded, we will deduce that

$$\mathbb{P}\left(\sum_{w\in W}\left|\text{Symm}_{\phi\phi'}(w)\cap C_u\right| \leq (2\gamma_1+\frac{9(\gamma_2-\gamma_1)}{20})\left|C_u\right|\left|W\right|\right) \geq 1-\frac{n}{e^{\mathcal{O}(|C_u|)}}-\frac{1}{e^{\mathcal{O}(|W|)}} \tag{14.7}$$

As Equation (14.3) implies Equation (14.6), and Equations (14.6) and (14.7) together implies that Equation (14.3) does not hold with probability at least $1-4ne^{-\mathcal{O}(|C_u|)}-e^{-\mathcal{O}(|W|)}$. Hence, we are done with the proof of Claim 14.11 except that we need to show Equation (14.7).

By the definition of the bijection $\phi$, we have $\sum\limits_{x\in V(G_u)}\left|\text{DECIDER}_\phi(x)\right| \leq \gamma_1 n^2$. This implies

$$\sum_{\substack{x\in V(G_u)\\ \left|\text{DECIDER}_\phi(x)\right|\geq\frac{(\gamma_2-\gamma_1)n}{1000}}}\left|\text{DECIDER}_\phi(x)\right| \leq \gamma_1 n^2 \tag{14.8}$$

To proceed further, we need the following observation, which is a direct application of Chernoff-Hoeffding bound.

**Observation 14.13.** (i) If $\left|\text{DECIDER}_\phi(x)\right| \geq \frac{(\gamma_2-\gamma_1)n}{1000}$, then

$$\mathbb{P}\left(\left|\text{DECIDER}_\phi(x)\cap C_u\right| \geq (1+\frac{1}{50})\left|(\text{DECIDER}_\phi(x)\right|\frac{|C_u|}{n}\right) \leq e^{-\mathcal{O}(|C_u|)}.$$

(ii) If $\left|\text{DECIDER}_\phi(x)\right| < \frac{(\gamma_2-\gamma_1)n}{1000}$, then

$$\mathbb{P}\left(\left|\text{DECIDER}_\phi(x)\cap C_u\right| \geq \frac{\gamma_2-\gamma_1}{750}\left|C_u\right|\right) \leq e^{-\mathcal{O}(|C_u|)}.$$

Note that the above observation implies that the following holds with probability at least $1 - ne^{-\mathcal{O}(|C_u|)}$.

$$
\begin{aligned}
&\sum_{x \in V(G_u)} \left|\textsc{Decider}_{\phi}(x) \cap C_u\right| \\
&= \sum_{\substack{x \in V(G_u): \\ \left|\textsc{Decider}_{\phi}(x)\right| \geq \\ \frac{(\gamma_2-\gamma_1)n}{1000}}} \left|\textsc{Decider}_{\phi}(x) \cap C_u\right| + \sum_{\substack{x \in V(G_u): \\ \left|\textsc{Decider}_{\phi}(x)\right| < \\ \frac{(\gamma_2-\gamma_1)n}{1000}}} \left|\textsc{Decider}_{\phi}(x) \cap C_u\right| \\
&\leq \left(1 + \frac{1}{50}\right) \sum_{\substack{x \in V(G_u): \\ \left|\textsc{Decider}_{\phi}(x)\right| \geq \\ \frac{(\gamma_2-\gamma_1)n}{1000}}} \left|\textsc{Decider}_{\phi}(x)\right| \frac{|C_u|}{n} + \frac{(\gamma_2 - \gamma_1)n\,|C_u|}{750} \\
&\leq \frac{51}{50}\gamma_1 n\,|C_u| + \frac{(\gamma_2 - \gamma_1)n\,|C_u|}{750}
\end{aligned}
$$

Note that the last inequality follows from Equation (14.8). Summarizing the above calculation, we get that the following event occurs with probability at least $1 - ne^{-\mathcal{O}(|C_u|)}$.

$$
\sum_{x \in V(G_u)} \left|\textsc{Decider}_{\phi}(x) \cap C_u\right| \leq \frac{51}{50}\gamma_1 n\,|C_u| + \frac{(\gamma_2 - \gamma_1)n\,|C_u|}{750}. \tag{14.9}
$$

Let us assume Equation (14.9) holds. Since we are taking the vertices of $W$ uniformly at random from $V(G_u)$, we have

$$
\begin{aligned}
\mathbb{E}\left[\sum_{w \in W} \left|\textsc{Decider}_{\phi}(w) \cap C_u\right|\right] &= \mathbb{E}\left[\sum_{w \in W} d_H(\mathcal{L}_{C_u}(w), \mathcal{L}_{C_k^i}(\phi(w))\right] \\
&\leq \frac{51}{50}\gamma_1\,|C_u|\,|W| + \frac{(\gamma_2 - \gamma_1)\,|C_u|\,|W|}{750}.
\end{aligned}
$$

Similarly from **Step 2 (iii)** of **Phase 2**, we have

$$
\begin{aligned}
\sum_{w \in W} \left|\textsc{Decider}_{\phi'}(w) \cap C_u\right| &= \sum_{w \in W} d_H(\mathcal{L}_{C_u}(w), \mathcal{L}_{C_k^i}(\phi'(w))) \\
&\leq \frac{2\gamma_2}{5}\,|C_u|\,|W|
\end{aligned}
$$

Recall that $\text{Symm}_{\phi\phi'}(x) = \text{DECIDER}_{\phi'}(x)\Delta\text{DECIDER}_{\phi}(x)$. Therefore,

$$\begin{aligned}
\mathbb{E}\left[\sum_{w\in W}\left|\text{Symm}_{\phi\phi'}(x)\cap C_u\right|\right] &\leq \mathbb{E}\left[\sum_{w\in W}\left|\text{DECIDER}_{\phi'}(w)\cap C_u\right|\right]\\
&+ \sum_{w\in W}\left|\text{DECIDER}_{\phi}(w)\cap C_u\right|\\
&\leq (\frac{764}{750}\gamma_1+\frac{301(\gamma_2-\gamma_1)}{750})\left|C_u\right|\left|W\right|
\end{aligned}$$

Using Hoeffding's inequality (see Lemma 2.14), we can say that

$$\begin{aligned}
\mathbb{P}\left(\sum_{w\in W}\left|\text{Symm}_{\phi\phi'}(w)\cap C_u\right| > (2\gamma_1+\frac{9(\gamma_2-\gamma_1)}{20})\left|C_u\right|\left|W\right|\right) &\leq e^{-\mathcal{O}(\frac{|C_u|^2|W|^2}{|W||C_u|^2})}\\
&= e^{-\mathcal{O}(|W|)}.
\end{aligned}$$

Note that the above equation holds on the conditional space that Equation (14.9) holds. Hence,

$$\mathbb{P}\left(\sum_{w\in W}\left|\text{Symm}_{\phi\phi'}(w)\cap C_u\right| \leq (2\gamma_1+\frac{9(\gamma_2-\gamma_1)}{20})\left|C_u\right|\left|W\right|\right) \geq 1-\frac{n}{e^{\mathcal{O}(|C_u|)}}-\frac{1}{e^{\mathcal{O}(|W|)}}$$

□

If we had constructed a bijection $\phi'$ as stated in the above claim, we could easily test by sampling *suitable* many random edges from $G_u$ and checking the corresponding edges in $G_k$. It is important to note that, it is not possible to construct $\phi'$ efficiently. However, without constructing the bijection $\phi'$, if we can test for presence of some randomly chosen edges in $G_u$ and their corresponding edges in $G_k$, we are done. In order to achieve this, we choose $W$ randomly in **Step 1** of **Phase 2** and pair up the vertices of $W$ in **Step 1** of **Phase 3**. Using **Step 2 (iii)** of **Phase 2** and **Step 3** of **Phase 3**, we check if $\widehat{\phi}(w) = \phi'(w)$ for each $w \in W$. Note that $\widehat{\phi} : W \to V(G_k)$ is the map constructed in **Step 3** of **Phase 3** and $\phi' : V(G_u) \to V(G_k)$ is the bijection as stated in Claim 14.11. Then we check the edge mismatches between the paired up vertices of

$W$ in $G_u$ and their corresponding mapped vertices in $G_k$ in **Step 4** of **Phase 3**, which is possible as we have constructed the mappings of the vertices in $W$ in **Step 2 (iii)** of **Phase 2**.

The following claim proves that if $G_u$ and $G_k$ are $\gamma_1$-close, then $\zeta(C_u, \eta_i, \psi, \widehat{\phi}) \leq 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)$, as considered in **Step 4** of **Phase 3** holds with probability at least $20/21$.

**Claim 14.14.** *Let $\phi : V(G_u) \to V(G_k)$ be a bijection such that $d_\phi(G_u, G_k) \leq \gamma_1 n^2$, and $(C_u, \eta_i) \in \Gamma$ where $C_u \in \mathcal{C}_u$, and $\eta_i : C_u \to V(G_k)$ be an embedding of $C_u$ such that*

- *$\forall\, v \in C_u$ we have $\eta_i(v) = \phi(v)$, and*
- *$d_{EM}(X_{C_u}, Y_{C_k^i}) \leq (\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000})n\,|C_u|$ where $C_k^i = \eta_i(C_u)$.*

*Let $\{\ell_1, \ldots, \ell_t\}$ be the all possible $C_k^i$-labellings of $G_k$ where $t = \left\lceil 2^{|C_k^i|} \right\rceil$, $W$ be the set of vertices of $G_u$ sampled at random in* **Step 1** *of* **Phase 2***, and $\psi : W \to \{\ell_1, \ldots, \ell_t\}$ be the mapping considered in* **Step 2 (iii)** *of* **Phase 2** *such that*

- *$\sum\limits_{w \in W} d_H(\mathcal{L}_{C_u}(w), \psi(w)) \leq \frac{2\gamma_2}{5}\,|C_u|\,|W|$, and*
- *$\forall j \in [t]$, we have $|\{w : \psi(w) = l_j\}| \leq |V_j|$.*

*If we take an embedding $\widehat{\phi} : W \to V(G_k)$ such that $\widehat{\phi}(w) \in V_j$ if and only if $\psi(w) = \ell_j$, then*

$$\zeta(C_u, \eta_i, \psi, \widehat{\phi}) \leq 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)$$

*holds with probability at least $20/21$, where $\zeta(C_u, \eta_i, \psi, \widehat{\phi})$ is as defined in* **Step 3** *of* **Phase 3**.

*Proof.* Recall that $W$ is a subset of $V(G_u)$ taken uniformly at random in **Step 1** of **Phase 2** and we paired up the vertices of $W$ randomly in **Step 1** of **Phase 3** respectively. Also, we are checking the edge mismatches of the paired up vertices of $W$ and their corresponding mapped vertices in $G_k$ according to the mapping $\widehat{\phi} : W \to V(G_k)$ in **Step 4** of **Phase 3** to compute $\zeta(C_u, \eta_i, \psi, \widehat{\phi})$. Considering the conditions given in the statement of this claim and Claim 14.11, one can think that we are checking the presence of $\frac{|W|}{2}$ randomly chosen edges in $G_u$ and the corresponding edges in $G_k$ according to some bijection $\phi' : V(G_u) \to V(G_k)$, where $\phi'$ is a bijection with $d_{\phi'}(G_u, G_k) \leq (5\gamma_1 + \frac{\gamma_2-\gamma_1}{2})n^2$.

So, $\mathbb{E}\left[\zeta(C_u, \eta_i, \psi, \widehat{\phi})\right] \leq (5\gamma_1 + \frac{\gamma_2-\gamma_1}{2})$. Now, applying Hoeffding's inequality (Lemma 2.14) and taking $|W| = C' \frac{\log^2 n}{(\gamma_2-\gamma_1)^3}$ for suitably large constant $C'$, we have

$$\begin{aligned} & \mathbb{P}\left(\zeta(C_u, \eta_i, \psi, \widehat{\phi}) > 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)\right) \\ = \; & \mathbb{P}\left(\zeta(C_u, \eta_i, \psi, \widehat{\phi})\,|W| > \left(5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)\right)|W|\right) \leq e^{-\mathcal{O}(|W|)} \leq \frac{1}{21} \end{aligned}$$

□

Now we are ready to prove the completeness property using Claims 14.8, 14.10, 14.11, 14.14 and Theorem 14.2.

**Lemma 14.15** (Completeness Lemma). *If $G_u$ and $G_k$ are $\gamma_1$-close to isomorphic, then our algorithm reports the same with probability at least* $2/3$.

*Proof.* Observe that from Claim 14.8, we know that, with probability at least $20/21$, there exists a $C_u \in \mathcal{C}_u$ and an embedding $\eta_i : C_u \to V(G_k)$ such that $d_{EM}(X_{C_u}, Y_{C_k^i}) \leq \left(\gamma_1 + \frac{\gamma_2-\gamma_1}{2000}\right) n\,|C_u|$ where $C_k^i = \eta_i(C_u)$. Similarly, from Theorem 14.2, we can say that, with probability at least $20/21$, the algorithm ALG-EMD returns all embeddings $\eta_i$ such that $d_{EM}(X_{C_u}, Y_{C_k^i}) \leq \left(\gamma_1 + \frac{\gamma_2-\gamma_1}{2000}\right) n\,|C_u|$. Now from Claim 14.10, we know that, with probability at least $20/21$, conditions of Equation (14.1) hold. Again, from Claim 14.11, we can say that constructing partial bijection at **Step 3** of **Phase 3** does not change isomorphism distance by more than $(4\gamma_1 + \frac{\gamma_2-\gamma_1}{2})n^2$ with probability at least $18/21$. Finally, from Claim 14.14, we can say that the algorithm will correctly detect the distance at **Step 4** of **Phase 3** by testing $\zeta(C_u, \eta_i, \psi, \widehat{\phi}) \leq 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)$ with probability at least $20/21$. Thus, using union bound, we can say that when $G_k$ and $G_u$ are $\gamma_1$-close to being isomorphic, **TolerantGI(**$G_u$**,** $G_k$**,** $\gamma_1$**,** $\gamma_2$**)** reports the same with probability at least $2/3$. □

### 14.3.2 Proof of soundness

Similarly for the soundness property of our algorithm, let us consider the case when $G_u$ and $G_k$ are $\gamma_2$-far from being isomorphic. Then we will show that the algorithm will output the correct answer with probability at least $2/3$.

Recall the definition of the set $\Gamma_W$ with which we started **Phase 3** of our algorithm.

$$\Gamma_W = \{(C_u, \eta_i, \psi) : (C_u, \eta_i) \in \Gamma \text{ such that Equation 14.1 holds}\}.$$

By Observation 14.5, we have

$$\Pr\left(\forall\ (C_u, \eta_i, \psi) \in \Gamma_W, d_{EM}(X_{C_u}, Y_{C_k^i}) \leq \frac{\gamma_2}{5}|C_u|n\right) \geq \frac{8}{9}. \tag{14.10}$$

From now on, we work on the conditional space where $d_{EM}(X_{C_u}, Y_{C_k^i}) \leq \frac{\gamma_2}{5}|C_u|n$ for all $(C_u, \eta_i, \psi)$ holds. By Observation 14.7 (i), we know that $|\Gamma_W| \leq 2^{\mathcal{O}(\log^2 n/(\gamma_2-\gamma_1))}$. So, the following claim about any $(C_u, \eta_i, \psi) \in \Gamma_W$ along with union bound over all the elements in $\Gamma_W$, we will be done with the proof of soundness property.

**Claim 14.16.** *Let $(C_u, \eta_i, \psi) \in \Gamma_W$ and $\widehat{\phi}$ be the embedding of $W$ into $G_k$ constructed while executing* **Step 3** *of* **Phase 3** *for $(C_u, \eta_i, \psi)$. Also, let $d_{EM}(X_{C_u}, Y_{C_k^i}) \leq \frac{\gamma_2}{5}|C_u|n$, where $C_k^i = \eta_i(C_u)$. Then the following holds with probability at most $\frac{2}{9|\Gamma_W|}$:*

$$\zeta(C_u, \eta_i, \psi, \widehat{\phi}) \leq 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1).$$

*Proof.* Let $\Phi(C_u, C_k^i)$ be the class of all bijections such that the following hold for each $\phi \in \Phi(C_u, C_k)$.

- $\phi(x) = \eta_i(x)$ for each $x \in C_u$, and
- $\sum\limits_{v \in V(G_u)} |\textsc{Decider}_\phi(v) \cap C_u| \leq \frac{\gamma_2}{5} n\, |C_u|$.

Consider the following observation, about the bijections in $\Phi$, that we will prove later.

**Observation 14.17.** Let $\phi$ be a bijection in $\Phi$. Then with probability at least $1 - \frac{1}{9|\Gamma_W|}$, $\sum\limits_{w \in W} |\textsc{Decider}_\phi(w) \cap C_u| \leq \frac{2\gamma_2}{5} |C_u| \, |W|$ holds.

Our algorithm constructs $\psi : W \to \{\ell_1, \ldots, \ell_t\}$ in **Step 2** of **Phase 2** satisfying

- $\sum\limits_{w \in W} d_H(\mathcal{L}_{C_u}(w), \psi(w)) \leq \frac{2\gamma_2}{5} |C_u| \, |W|$, and
- $\forall j \in [t]$, we have $|\{w : \psi(w) = l_j\}| \leq |V_j|$.

Note that $\sum\limits_{w \in W} d_H(\mathcal{L}_{C_u}(w), \psi(w)) = \sum\limits_{w \in W} |\textsc{Decider}_\phi(w) \cap C_u|$, where $\phi$ is some bijection in $\Phi$. After getting $\psi$, we construct a partial bijection $\widehat{\phi} : W \to V(G_k)$ that satisfies the above two conditions. So, one can think of $W$ is taken uniformly at random from the set of all $W$'s satisfying $\sum\limits_{w \in W} |\textsc{Decider}_\phi(w) \cap C_u| \leq \frac{2\gamma_2}{5} |C_u| \, |W|$. Now, from Observation 14.17, we have the following observation.

**Observation 14.18.** $\widehat{\phi}$ is a *random restriction* of a random bijection $\phi \in \Phi(C_u, C_k)$ by the set $W$ with probability at least $1 - \frac{1}{9|\Gamma_W|}$.

*Proof.* Let us consider a $\phi$ such that $\phi|_W = \widehat{\phi}$. Let $\mathcal{W} = \{\widehat{\phi}_X = \phi|_X : X \subset V(G_u) \text{ and } |X| = |W|\}$, and $\mathcal{W}' \subseteq \mathcal{W}$ is defined as:

$$\mathcal{W}' = \left\{ \widehat{\phi}_X \in \mathcal{W} : \sum_{w \in X} |\textsc{Decider}_\phi(w) \cap C_u| \leq \frac{2\gamma_2}{5} |C_u| \, |W| \right\}$$

Observe that $\widehat{\phi} = \widehat{\phi}_W \in \mathcal{W}$. By Observation 14.17, we know that if we take a set $X \subset V(G_u)$ (i.e, a $\widehat{\phi}_X$ uniformly at random from $\mathcal{W}$), then the probability that $\widehat{\phi}_X \in \mathcal{W}'$, is at least $1 - \frac{1}{9|\Gamma_W|}$. So, $|\mathcal{W}'| \geq \left(1 - \frac{1}{9|\Gamma_W|}\right) |W|$.

Observe that the partial bijection $\widehat{\phi}$, constructed by our algorithm, is same as that of $\widehat{\phi}_W$, and $\widehat{\phi}$ is in $\mathcal{W}'$. Now, using the fact that $|\mathcal{W}'| \geq \left(1 - \frac{1}{9|\Gamma_W|}\right) |W|$, the observation follows. □

Recall that $W$ is a subset of $V(G_u)$ taken uniformly at random in **Step 1** of **Phase 2** and we paired up the vertices of $W$ randomly in **Step 1** of **Phase 3** respectively. Also, we are checking the edge mismatches of the paired up vertices of $W$ and their corresponding mapped vertices in $G_k$ according to the mapping $\widehat{\phi} : W \to V(G_k)$ in **Step 4** of **Phase 3** to compute $\zeta(C_u, \eta_i, \psi, \widehat{\phi})$. Considering the discussion here, one can think of that, we are checking the presence of $\frac{|W|}{2}$ randomly chosen edges in $G_u$ and the corresponding edges in $G_k$ according to some bijection $\phi \in \Phi$.

Note that $d_\phi(G_u, G_k) \geq \gamma_2 n^2$. Thus, $\mathbb{E}\left[\zeta(C_u, \eta_i, \psi, \widehat{\phi})\right] \geq \gamma_2 |W|$. Now we can deduce the following. [3]

$$\begin{aligned} & \mathbb{P}\left(\zeta(C_u, \eta_i, \psi, \widehat{\phi}) \leq 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)\right) \\ = \; & \mathbb{P}\left(\zeta(C_u, \eta_i, \psi, \widehat{\phi})\,|W| \leq (5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1))\,|W|\right) \leq e^{-\mathcal{O}(|W|)} \leq \frac{1}{9\,|\Gamma_W|} \end{aligned}$$

Note that we were deriving above bound on $\mathbb{P}\left(\zeta(C_u, \eta_i, \psi, \widehat{\phi}) \leq 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)\right)$ assuming that $\widehat{\phi}$ is a random restriction of a random $\phi \in \Phi$. Hence, combining Observation 14.18 with the above bound on $\mathbb{P}\left(\zeta(C_u, \eta_i, \psi, \widehat{\phi}) \leq 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)\right)$ (when $\widehat{\phi}$ is a random restriction of a random $\phi \in \Phi$), we get

$$\mathbb{P}\left(\zeta(C_u, \eta_i, \psi, \widehat{\phi}) \leq 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)\right) \leq \frac{2}{9\,|\Gamma_W|}.$$

□

*Proof of Observation 14.17.* Since $W$ is taken uniformly at random,

$$\mathbb{E}\left[\sum_{w \in W} |\textsc{Decider}_\phi(w) \cap C_u|\right] \leq \frac{\gamma_2}{5}\,|C_u|\,|W|$$

Using Hoeffding's inequality, we get

$$\mathbb{P}\left(\sum_{w \in W} |\textsc{Decider}_\phi(w) \cap C_u| \geq \frac{2\gamma_2}{5}\,|C_u|\,|W|\right) \leq e^{-\mathcal{O}(|W|)} \leq \frac{1}{9\,|\Gamma_W|}.$$

□

[3] Here we are assuming $\gamma_2 \geq 11\gamma_1$.

Now we are ready to prove the soundness property of our algorithm.

**Lemma 14.19** (Soundness Lemma)**.** *If $G_u$ and $G_k$ are $\gamma_2$-far from isomorphic, then the algorithm reports the same with probability at least* $2/3$.

*Proof.* From Observation 14.7 (i), it follows that $|\Gamma_W|$ is at most $2^{C_1 \frac{\log^2 n}{\gamma_2 - \gamma_1}}$. In the Claim 14.16, we are proving that $\zeta(C_u, \eta_i, \psi, \widehat{\phi}) \leq 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)$ holds with probability at most $\frac{2}{9|\Gamma_W|}$ for any particular $(C_u, \eta_i, \psi) \in \Gamma_W$ with $d_{EM}(X_{C_u}, Y_{C_k^i}) \leq \frac{\gamma_2}{5}|C_u|n$. So, using the union bound, the probability that there exists a $(C_u, \eta_i, \psi) \in \Gamma_W$ with $d_{EM}(X_{C_u}, Y_{C_k^i}) \leq \frac{\gamma_2}{5}|C_u|n$ such that $\zeta(C_u, \eta_i, \psi, \widehat{\phi}) \leq 5\gamma_1 + \frac{3}{5}(\gamma_2 - \gamma_1)$, is at most $\frac{2}{9}$. Now From Equation 14.10,

$$\Pr\left(\forall\ (C_u, \eta_i, \psi, \widehat{\phi}) \in \Gamma_W, d_{EM}(X_{C_u}, Y_{C_k^i}) \leq \frac{\gamma_2}{5}|C_u|n\right) \geq \frac{8}{9}$$

Putting everything together, the probability that the algorithm reports that $G_u$ and $G_k$ are $\gamma_2$-far, is at least $2/3$. □

Till now we have proved the completeness and soundness property of our algorithm **TolerantGI**. We will prove the query complexity property in the next section when we prove the final theorem.

## 14.4 Proof of upper bound result

*Proof of Theorem 14.1.* From the *Completeness Lemma* (Lemma 14.15) and *Soundness Lemma* (Lemma 14.19), we can say that our algorithm **TolerantGI** correctly decides whether $d(G_u, G_k) \leq \gamma_1 n^2$ or $d(G_u, G_k) \geq \gamma_2 n^2$ with probability at least $2/3$.

Now, we calculate the query complexity of our algorithm. Note that **Step 1** and **Step 2** of **Phase 1**, **Step 1** and **Step 3** of **Phase 2**, **Step 1**, **Step 2** and **Step 3** of **Phase 3**, of the algorithm **TolerantGI**, do not require any query to the adjacency matrix of $G_u$. Let $\textsc{Cost}_{C_u}$ denote the query complexity corresponding to a particular $C_u \in \mathcal{C}_u$. So, the total query complexity of the algorithm **TolerantGI** is $\sum\limits_{C_u \in \mathcal{C}_u} \textsc{Cost}_{C_u}$. Observe that

$$\textsc{Cost}_{C_u} \quad = \quad \text{Query Complexity of algorithm ALG-EMD} \; + \textsc{Cost}_{C_u,W}$$

where $\textsc{Cost}_{C_u,W}$ denotes the query complexity of **Step 1** of **Phase 2** corresponding to $W$ and $C_u \in \mathcal{C}_u$.

Note that ALG-EMD is the algorithm corresponding to Theorem 14.2. In **Step 3** of **Phase 1** of our algorithm, for each $C_u \in \mathcal{C}_u$, we call ALG-EMD with parameters $d = \mathcal{O}\left(\log n\right)$, $t = 2^{\mathcal{O}\left(\log^2 n\right)}$, $\varepsilon_1 = \left(\gamma_1 + \frac{\gamma_2 - \gamma_1}{2000}\right)$, $\varepsilon_2 = \frac{\gamma_2}{5}$ and $\delta = \Theta(1)$. So, the query complexity of each call, to ALG-EMD from our algorithm, is $\widetilde{\mathcal{O}}\left(\min\{n, 2^d\}\right) = \widetilde{\mathcal{O}}(n)$.

Further note that, from the description **Step 1** of **Phase 2**, $\text{COST}_{C_u,W} = \mathcal{O}\left(\log^2 n\right)$. Since $|\mathcal{C}_u| = \mathcal{O}\left(\frac{1}{\gamma_2 - \gamma_1}\right)$, the total query complexity of our algorithm is $\widetilde{\mathcal{O}}(n)$. □

# Chapter 15

# Tolerant Bipartiteness Testing in Dense Graphs

## 15.1 Introduction

In this chapter, we present our result on tolerant bipartiteness testing. We will prove the following theorem.

**Theorem 15.1** (Restatement of Theorem 12.4)**.** *Given query access to the adjacency matrix of a dense graph $G$ with $n$ vertices and a proximity parameter $\varepsilon \in (0,1)$, there exists an algorithm that, with probability at least $\frac{9}{10}$, decides whether $d_{bip}(G) \leq \varepsilon n^2$ or $d_{bip}(G) \geq (2+\Omega(1))\varepsilon n^2$, by sampling $\mathcal{O}\left(\frac{1}{\varepsilon^3}\log\frac{1}{\varepsilon}\right)$ vertices in $2^{\mathcal{O}\left(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\right)}$ time, and makes $\mathcal{O}\left(\frac{1}{\varepsilon^3}\log^2\frac{1}{\varepsilon}\right)$ queries.*

before proceeding to the proof, let us first recall the notion of *bipartite distance* which will be used in our proofs in this chapter.

**Definition 15.2** (Bipartite distance, Restatement of Definition 12.3)**.** A *bipartition* of (the vertices of) a graph $G$ is a function $f : V(G) \to \{L, R\}$ [1]. The *bipartite* distance of

[1] $L$ and $R$ denote left and right respectively.

$G$ with respect to the bipartition $f$ is denoted and defined as

$$d_{bip}(G, f) := \left[ \sum_{v \in V : f(v) = L} \left| N(v) \cap f^{-1}(L) \right| + \sum_{v \in V : f(v) = R} \left| N(v) \cap f^{-1}(R) \right| \right].$$

The *bipartite distance* of $G$ is defined as the minimum bipartite distance of $G$ over all possible bipartitions $f$ of $G$, that is,

$$d_{bip}(G) := \min_{f} d_{bip}(G, f).$$

For a set of pairs of vertices $Z$, we will denote the set of vertices present in at least one pair in $Z$ by $V(Z)$. For a function $f : V(G) \to \{L, R\}$, $f^{-1}(L)$ $(f^{-1}(R))$ represents the set of vertices that are mapped to $L$ $(R)$ by $f$. $\binom{V(G)}{2}$ denotes the set of unordered pairs of the vertices of $G$

In Section 15.2, we present an algorithm of estimating the bipartite distance of a dense graph by applying the result of Alon et al. [AdlVKK03] of estimating the size of MAXCUT of a dense graph (with larger query complexity compared to our final algorithm). In Section 15.3, we formally describe our algorithm, followed by its correctness analysis in Section 15.4.

## 15.2 Estimation of bipartite distance with $\widetilde{\mathcal{O}}\left(\frac{1}{\varepsilon^6}\right)$ queries

Formally, we state the following theorem.

**Theorem 15.3.** *Given an unknown graph G on $n$ vertices and any proximity parameter $\varepsilon \in (0, 1)$, there is an algorithm that performs $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon^6})$ adjacency queries, and outputs a number $\widehat{d_{bip}}(G)$ such that, with probability at least $\frac{9}{10}$, the following holds:*

$$d_{bip}(G) - \varepsilon n^2 \leq \widehat{d_{bip}}(G) \leq d_{bip}(G) + \varepsilon n^2,$$

*where $d_{bip}(G)$ denotes the bipartite distance of G.*

We have the following two corollaries of the above theorem.

**Corollary 15.4.** *There exists an algorithm that given adjacency query access to a graph $G$ with $n$ vertices and a proximity parameter $\varepsilon \in (0,1)$ such that, with probability at least $\frac{9}{10}$, decides whether $d_{bip}(G) \leq \varepsilon n^2$ or $d_{bip}(G) \geq (2+\Omega(1))\varepsilon n^2$ using $\widetilde{\mathcal{O}}\left(\frac{1}{\varepsilon^6}\right)$ queries to the adjacency matrix of G.*

**Corollary 15.5.** *There exists an algorithm that given adjacency query access to a graph $G$ with $n$ vertices and a proximity parameter $\varepsilon \in (0,1)$ such that, with probability at least $\frac{9}{10}$, decides whether $d_{bip}(G) \leq \varepsilon n^2$ or $d_{bip}(G) \geq (1+k)\varepsilon n^2$ using $\widetilde{\mathcal{O}}\left(\frac{1}{k^6\varepsilon^6}\right)$ queries to the adjacency matrix of G.*

To prove Theorem 15.3, we first discuss the connection between MAXCUT and bipartite distance of a graph $G$. Then we use the result for MAXCUT estimation by Alon, Vega, Kannan and Karpinski [AdlVKK03] to obtain Theorem 15.3.

**Connection between MAXCUT and $d_{bip}(G)$:** For a graph $G = (V, E)$ on the *vertex* set $V$ and *edge* set $E$, let $S$ be a subset of $V$. We define

$$\textsc{Cut}(S) := |\ \{\{u,v\} \in E \ \mid\ |\{u,v\} \cap S| = 1\}\ |$$

*Maximum Cut* (henceforth termed as MAXCUT), denoted by $M(G)$, is a partition of the vertex set $V$ of $G$ into two parts such that the number of edges crossing the partition is maximized, that is,

$$M(G) := \max_{S \subseteq V} \textsc{Cut}(S).$$

The following equation connects MAXCUT and the bipartite distance of a graph $G$:

$$d_{bip}(G) = |E(G)| - M(G). \tag{15.1}$$

So, $d_{bip}(G)$ can be estimated by estimating $|E(G)|$ and $M(G)$.

**Result on edge estimation:** Observe that estimating $|E(G)|$ with $\varepsilon n^2$ additive error is equivalent to *parameter estimation problem* in probability theory, see Mitzenmacher and Upfal [MU17, Section 4.2.3].

**Proposition 15.6** (Folklore). *Given any graph $G$ on $n$ vertices and a proximity parameter $\varepsilon \in (0,1)$, the size of the edge set $E(G)$ can be estimated within an additive $\varepsilon n^2$ error, with probability at least $\frac{9}{10}$, using $\mathcal{O}(\frac{1}{\varepsilon^2})$ adjacency queries to $G$.*

## MaxCut estimation by using $\widetilde{\mathcal{O}}\left(\frac{1}{\varepsilon^6}\right)$ queries:

Let $G = (V, E)$ be an $n$ vertex graph. Both Alon et.al [AdlVKK03] and Mathieu and Schudy [MS08] showed that if $S$ is a $t$-sized random subset of $V$, where $t = O\left(\frac{1}{\varepsilon^4}\log\frac{1}{\varepsilon}\right)$, then, with probability at least $\frac{9}{10}$, we have the following:

$$\left|\frac{M(G\mid_S)}{t^2} - \frac{M(G)}{n^2}\right| \leq \frac{\varepsilon}{2}$$

where $G\mid_S$ denotes the induced graph of $G$ on the vertex set $S$. So, the above inequality tells us that if we can get an $\varepsilon t^2/2$ additive error to $M(G\mid_S)$, then we can get an $\varepsilon n^2$ additive estimate for $M(G)$. Observation 15.7 implies that using $O\left(\frac{t}{\varepsilon^2}\right) = O\left(\frac{1}{\varepsilon^6}\log\frac{1}{\varepsilon}\right)$ adjacency queries to $G\mid_S$, we can get an $\frac{\varepsilon t^2}{2}$ additive estimate to $M(G\mid_S)$. Therefore, the query complexity of MaxCut algorithms of Alon, Vega, Kannan and Karpinski [AdlVKK03] and Mathieu and Schudy [MS08] is at most $O\left(\frac{1}{\varepsilon^6}\log\frac{1}{\varepsilon}\right)$.

Now we state and prove the following observation.

**Observation 15.7** (Folklore). For a graph $G$ with $n$ vertices and a proximity parameter $\varepsilon \in (0,1)$, with probability at least $\frac{9}{10}$, $\Theta\left(\frac{n}{\varepsilon^2}\right)$ adjacency queries to $G$ are sufficient to get an $\varepsilon n^2$ additive approximation to MaxCut $M(G)$.

*Proof.* We sample $t$ pairs of vertices $\{a_1, b_1\}, \ldots, \{a_t, b_t\}$ uniformly at random and independent of each other, where $t = \Theta(\frac{n}{\varepsilon^2})$. Thereafter, we perform $t$ adjacency queries to those sampled pairs of vertices. Now fix a subset $S \subset V(G)$ and let us denote $(S, \overline{S})$ to be the set of edges between $S$ and $\overline{S}$.

Let us now define a set of random variables, one for each sampled pair of vertices as follows:

$$X_i = \begin{cases} 1, & \text{if } \{a_i, b_i\} \in (S, \overline{S}) \\ 0, & \text{Otherwise} \end{cases}$$

We will output $\max\limits_{S \subset V(G)} \widehat{M_S}$ as our estimate of $M(G)$, where $\widehat{M_S} = \frac{\binom{n}{2}}{t} \sum\limits_{i=1}^{t} X_i$.

Let us denote $X = \sum\limits_{i=1}^{t} X_i$. Note that

$$\mathbb{E}\left[X_i\right] = \mathbb{P}\left(X_i = 1\right) = \frac{\left|(S, \overline{S})\right|}{\binom{n}{2}},$$

and hence

$$\mathbb{E}\left[\widehat{M_S}\right] = \frac{\binom{n}{2}}{t} \mathbb{E}\left[\sum_{i=1}^{t} X_i\right] = \left|(S, \overline{S})\right|.$$

Using Hoeffding's Inequality (See Lemma 2.14), we can say that

$$\mathbb{P}\left(\left|\left|(S, \overline{S})\right| - \widehat{M_S}\right| \geq \frac{\varepsilon n^2}{10}\right) \leq \mathbb{P}\left(\left|X - \mathbb{E}[X]\right| \geq \frac{\varepsilon t}{10}\right) \leq 2e^{-\Theta(\frac{\varepsilon^2 t^2}{t})} \leq 2e^{-\Theta(n)}.$$

Using union bound over all $S \subset V(G)$, we can show that with probability at least $3/4$, for each $S \subset V(G)$, $\widehat{M_S}$ approximates $\left|(S, \overline{S})\right|$ with $\varepsilon n^2$ additive error. Therefore $\max\limits_{S \subset V(G)} \widehat{M_S}$ estimates $M(G)$ with additive error $\varepsilon n^2$, with probability at least $9/10$. □

## 15.3 Algorithm for Tolerant Bipartite Testing

In this section, we formalize the ideas discussed in Section 12.4.2, and prove Theorem 12.26.

### Formal description of algorithm Tol-Bip-Dist$(G, \varepsilon)$

**Step-1** Let $C_1, C_2, C_3$ be three suitably chosen large constants and $t := \lceil \log \frac{C_1}{k\varepsilon} \rceil$.

**(i)** We start by generating $t$ subset of vertices $X_1, \ldots, X_t \subset V(G)$, each with

$\lceil \frac{C_2}{k^3 \varepsilon} \log \frac{1}{k\varepsilon} \rceil$ vertices, sampled randomly without replacement [2].

**(ii)** We sample $\lceil \frac{C_3}{k^5 \varepsilon^2} \log \frac{1}{k\varepsilon} \rceil$ random pairs of vertices, with replacement, and denote those sampled pairs of vertices as $Z$. Note that $X_1, \ldots, X_t, Z$ are generated independent of each other.

**(iii)** We find all the edges with one endpoint in $\mathcal{C} = X_1 \cup X_2 \cup \ldots X_t$ and the other endpoint in one of the vertices of $V(Z)$ [3], by performing $\mathcal{O}\left(\frac{1}{k^8 \varepsilon^3} \log^2 \frac{1}{k\varepsilon}\right)$ adjacency queries.

**Step-2** **(i)** Let $\{a_1, b_1\}, \ldots \{a_\lambda, b_\lambda\}$ be the pairs of vertices of $Z$, where $\lambda = \lceil \frac{C_3}{k^5 \varepsilon^2} \log \frac{1}{k\varepsilon} \rceil$. Now we find the pairs of $Z$ that are edges in $G$, by performing adjacency queries to all the pairs of vertices of $Z$ (after this step, the algorithm does not make any query further).

**(ii)** For each $i \in [t]$, we do the following:

**(a)** Let $\mathcal{F}_i$ denote the set of all possible bipartitions of $X_i$, that is,

$$\mathcal{F}_i = \left\{ f_{ij} : X_i \to \{L, R\} : j \in \left[2^{|X_i|-2}\right] \right\}.$$

**(b)** For each bipartition $f_{ij}$ (of $X_i$) in $\mathcal{F}_i$, we extend $f_{ij}$ to $f'_{ij} : X_i \cup Z \to \{L, R\}$ to be a bipartition of $X_i \cup Z$, such that the mapping of each

[2] Since we are assuming $n$ is sufficiently large with respect to $\frac{1}{\varepsilon}$, sampling with and without replacement are the same.

[3] Recall that $V(Z)$ denotes the set of vertices present in at least one pair in $Z$.

vertex of $X_i$ are identical in $f_{ij}$ and $f'_{ij}$, and is defined as follows:

$$f'_{ij}(z) = \begin{cases} f_{ij}(z), & z \in X_i \\ L, & z \notin X_i \text{ and } \\ & \left|N(z) \cap f_{ij}^{-1}(R)\right| > \left|N(z) \cap f_{ij}^{-1}(L)\right| + \frac{k\varepsilon|X_i|}{225000} \\ R, & z \notin X_i \text{ and } \\ & \left|N(z) \cap f_{ij}^{-1}(L)\right| > \left|N(z) \cap f_{ij}^{-1}(R)\right| + \frac{k\varepsilon|X_i|}{225000} \\ L \text{ or } R & \\ \text{arbitrarily}, & \text{otherwise} \end{cases}$$

Note that this step can be performed from the adjacency information between the vertices of $\mathcal{C}$ and $Z$, which have already been computed before.

**(c)** We now find the fraction of the vertex pairs of $Z$ that are edges and have the same label with respect to $f'_{ij}$, that is,

$$\zeta_{ij} = 2 \cdot \frac{\left|\{\{a_\ell, b_\ell\} : \ell \in [\lambda], \{a_\ell, b_\ell\} \in E(G) \text{ and } f'_{ij}(a_\ell) = f'_{ij}(b_\ell)\}\right|}{\lambda} \text{[4]}.$$

**(d)** If $\zeta_{ij} \leq \left(2 + \frac{k}{20}\right)\varepsilon$, we ACCEPT $G$ as $\varepsilon$-close to being bipartite, and QUIT the algorithm.

**(iii)** If we arrive at this step, then $\zeta_{ij} > \left(2 + \frac{k}{20}\right)\varepsilon$, for each $i \in [t]$ and $f_{ij} \in \mathcal{F}_i$ in **Step-(ii)**. We REJECT and declare that $G$ is $(2+k)\varepsilon$-far from being bipartite.

We split the analysis of algorithm TOL-BIP-DIST$(G, \varepsilon)$ into five parts:

**Completeness:** If $G$ is $\varepsilon$-close to being bipartite, then TOL-BIP-DIST$(G, \varepsilon)$ reports the same, with probability at least $\frac{9}{10}$.

[4] 2 is multiplied, as in the definition of $d_{bip}(G, f)$, each edge $\{u, v\} \in E(G)$ with $f(u) = f(v)$ is counted twice.

**Soundness:** If $G$ is $(2+k)\varepsilon$-far from being bipartite, then TOL-BIP-DIST$(G,\varepsilon)$ reports the same, with probability at least $\frac{9}{10}$.

**Sample Complexity:** The sample complexity of TOL-BIP-DIST$(G,\varepsilon)$ is $\mathcal{O}(\frac{1}{k^5\varepsilon^2}\log\frac{1}{k\varepsilon})$.

**Query Complexity:** The query complexity of TOL-BIP-DIST$(G,\varepsilon)$ is $\mathcal{O}(\frac{1}{k^8\varepsilon^3}\log^2\frac{1}{k\varepsilon})$.

**Time Complexity:** The time complexity of TOL-BIP-DIST$(G,\varepsilon)$ is $2^{\mathcal{O}(\frac{1}{k^3\varepsilon}\log\frac{1}{k\varepsilon})}$.

Above three quantities follows from the description of TOL-BIP-DIST$(G,\varepsilon)$. In **Step-1(i)** of TOL-BIP-DIST$(G,\varepsilon)$, we sample vertices of $G$ to generate $t = \lceil\log\frac{C_1}{k\varepsilon}\rceil$ subsets, each with $\lceil\frac{C_2}{k^3\varepsilon}\log\frac{1}{k\varepsilon}\rceil$ vertices. Then in **Step-1(ii)** and **Step-1(iii)**, we randomly choose $\lceil\frac{C_3}{k^5\varepsilon^2}\log\frac{1}{k\varepsilon}\rceil$ pairs of vertices and perform adjacency queries for each vertex in any pair of $Z$ to every $X_i$. Thus the sample complexity of TOL-BIP-DIST$(G,\varepsilon)$ is $\mathcal{O}(\frac{1}{k^5\varepsilon^2}\log\frac{1}{k\varepsilon})$ and query complexity is $\mathcal{O}(\frac{1}{k^8\varepsilon^3}\log^2\frac{1}{k\varepsilon})$. The time complexity of the algorithm is $2^{\mathcal{O}(\frac{1}{k^3\varepsilon}\log\frac{1}{k\varepsilon})}$, which follows from **Step-2(ii)**, that dominates the running time.

## 15.4 Correctness of our algorithm

In this section, we present the correctness proof of our algorithm TOL-BIP-DIST$(G,\varepsilon)$. Before proceeding to the proof, we introduce some definitions for classifying the vertices of the graph, with respect to any particular bipartition, into two categories: $(i)$ ***heavy*** vertices, and $(ii)$ ***balanced*** vertices. These definitions will be mostly used in the proof of completeness. Informally speaking, a vertex $v$ is said to be **heavy** with respect to a bipartition $f$, if it has *substantially* large number of neighbors in one side of the bipartition (either $L$ or $R$), as compared to the other side.

**Definition 15.8** (Heavy vertex)**.** A vertex $v \in V$ is said to be $L$**-heavy** *with respect to a bipartition* $f$, if it satisfies two conditions:

**(i)** $|N(v)\cap f^{-1}(L)| \geq |N(v)\cap f^{-1}(R)| + \frac{k\varepsilon n}{150}$;

**(ii)** If $|N(v) \cap f^{-1}(R)| \geq \left(1 + \frac{k}{200}\right)^{-1} \frac{k\varepsilon n}{150}$, then

$$|N(v) \cap f^{-1}(L)| \geq \left(1 + \tfrac{k}{200}\right) |N(v) \cap f^{-1}(R)|;$$

We define $R$-**heavy** vertices analogously. The union of the set of $L$-**heavy** and $R$-**heavy** vertices, with respect to a bipartition $f$, is defined to be the set of heavy vertices (with respect to $f$), and is denoted by $\mathcal{H}_f$.

Similarly, a vertex $v$ is said to be **balanced** if the number of neighbors of $v$ are *similar* in both $L$ and $R$, with respect to a bipartition $f$. We define it formally as follows:

**Definition 15.9** (Balanced vertex)**.** A vertex $v \in V$ is said to be **balanced** *with respect to a bipartition* $f$, if $v \notin \mathcal{H}_f$, that is, it satisfies at least one of the following conditions:

**(i) Type 1:** $||N(v) \cap f^{-1}(R)| - |N(v) \cap f^{-1}(L)|| < \frac{k\varepsilon n}{150}$;

**(ii) Type 2:** Either

$$\left|N(v) \cap f^{-1}(L)\right| \leq \left|N(v) \cap f^{-1}(R)\right| < \left(1 + \frac{k}{200}\right) \left|N(v) \cap f^{-1}(L)\right|,$$

or,

$$\left|N(v) \cap f^{-1}(R)\right| \leq \left|N(v) \cap f^{-1}(L)\right| < \left(1 + \frac{k}{200}\right) \left|N(v) \cap f^{-1}(R)\right|.$$

The set of balanced vertices of **Type 1** with respect to $f$ is denoted as $\mathcal{B}_f^1$, and the set of balanced vertices of **Type 2** with respect to $f$ is denoted as $\mathcal{B}_f^2$. The union of $\mathcal{B}_f^1$ and $\mathcal{B}_f^2$ is denoted by $\mathcal{B}_f$. Note that $\mathcal{B}_f^1$ and $\mathcal{B}_f^2$ may not be disjoint.

In order to prove the completeness (in Section 15.4.1), we also use a notion of SPECIAL *bipartition* to be defined below. The definition of SPECIAL bipartition is based on an optimal bipartition $f$ of $V(G)$, and notions of heavy and balanced vertices. We would also like to note that, later in Lemma 15.13, we show that when $d_{bip}(G) \leq \varepsilon n^2$, the bipartite distance of $G$ with respect to any SPECIAL bipartition is bounded by $(2 + \frac{k}{50})\varepsilon n^2$.

**Definition 15.10** (SPECIAL bipartition)**.** Let $d_{bip}(G) \leq \varepsilon n^2$, and $f : V(G) \to \{L, R\}$ be an optimal bipartition of $V(G)$, that is, $d_{bip}(G, f) \leq \varepsilon n^2$, and there does not exist any bipartition $g$ such that $d_{bip}(G, g) < d_{bip}(G, f)$. For an $X_i$ selected in **Step-1(i)** of the algorithm, let $f_{ij} \in \mathcal{F}_i$ be the bipartition of $X_i$ such that $f \mid_{X_i} = f_{ij}$. Then bipartition $\textsc{Spl}_i^f : V(G) \to \{L, R\}$ is said to be a SPECIAL **bipartition** with respect to $f$ by $f_{ij}$ such that

- $\textsc{Spl}_i^f \mid_{X_i} = f \mid_{X_i} = f_{ij}$;

- There exists a subset $\mathcal{H}_f' \subset \mathcal{H}_f$ such that $\left|\mathcal{H}_f'\right| \geq (1 - o(k\varepsilon)) \left|\mathcal{H}_f\right|$, and for each $v \in \mathcal{H}_f'$, $\textsc{Spl}_i^f(v)$ is defined as follows:

$$\textsc{Spl}_i^f(v) = \begin{cases} R, & v \notin X_i \text{ and } v \text{ is } L - \text{heavy} \\ L, & v \notin X_i \text{ and } v \text{ is } R - \text{heavy} \end{cases}$$

- For each $v \notin (\mathcal{H}_f' \cup X_i)$, $\textsc{Spl}_i^f(v)$ is set to $L$ or $R$ arbitrarily.

In our proof of the soundness theorem (in Section 15.4.2), we apply the notion of DERIVED *bipartition*. Unlike the definition of SPECIAL bipartition, the definition of DERIVED bipartition is more general, in the sense that it is not defined based on either any optimal bipartition, or on heavy or balanced vertices.

**Definition 15.11** (DERIVED bipartition)**.** Let $f : V(G) \to \{L, R\}$ be a bipartition of $V(G)$. For an $X_i$ selected in **Step-1(i)** of the algorithm, let $f_{ij} \in \mathcal{F}_i$ be the bipartition of $X_i$ such that $f \mid_{X_i} = f_{ij}$. A bipartition $\textsc{Der}_i^f : V(G) \to \{L, R\}$ is said to be DERIVED bipartition with respect to $f$ by $f_{ij}$, if $\textsc{Der}_i^f \mid_{X_i} = f \mid_{X_i} = f_{ij}$.

### 15.4.1 Proof of completeness

In this section, we prove the following theorem:

**Theorem 15.12.** *Let G be $\varepsilon$-close to being bipartite. Then* $\textsc{Tol-Bip-Dist}(G, \varepsilon)$ *reports the same, with probability at least* $9/10$.

The proof of Theorem 15.12 will crucially use the following lemma, which says that the bipartite distance of $G$ with respect to any SPECIAL bipartition is bounded by $\left(2 + \frac{k}{50}\right) \varepsilon n^2$.

**Lemma 15.13** (SPECIAL bipartition lemma)**.** *Let $f$ be a bipartition such that $d_{bip}(G, f) \leq \varepsilon n^2$ and there does not exist any bipartition $g$ such that $d_{bip}(G, g) < d_{bip}(G, f)$. For any* SPECIAL *bipartition $\text{SPL}_i^f$ with respect to $f$, $d_{bip}(G, \text{SPL}_i^f) \leq \left(2 + \frac{k}{50}\right) \varepsilon n^2$.*

We will prove the above lemma later. For now, we want to establish (in Lemma 15.15) that there exists an $i \in [t]$ and a $f_{ij} \in \mathcal{F}_i$ which can be thought of as a *random restriction* of some SPECIAL bipartition with respect to $f$ by $f_{ij}$. In other words, Lemma 15.15 basically states that if $G$ is $\varepsilon$-close to being bipartite, then the extension according to the rule in **Step-2(ii)(b)** of the mapping obtained by restricting an optimal bipartition to a random $X_i$ is likely to correspond to a SPECIAL bipartition, and therefore, the number of monochromatic edges (with respect to a SPECIAL bipartition) in the randomly picked $Z$ is likely to be low with respect to that bipartition. Thus, $\zeta_{ij}$ must be low for some $i$ and $j$ with high probability.

To prove Lemma 15.15, we need the following lemma (Lemma 15.14) about heavy vertices. In Lemma 15.14, we prove that a heavy vertex with respect to a bipartition $f$ will have significantly more neighbors in the part of $X_i$, that corresponds to the heavy side of that vertex (with respect to $f$). Basically, if a vertex $v$ is L-heavy with respect to $f$, it has more neighbors in the subset of $X_i$ on the L-side as compared to the subset of $X_i$ on the R-side of $f$. Formally, we have the following:

**Lemma 15.14** (Heavy vertex lemma)**.** *Let $f$ be a bipartition of $G$. Consider a vertex $v \in V$. Then we have the following:*

**(i)** *For each L-heavy vertex $v$, $|N(v) \cap f^{-1}(L) \cap X_i| - |N(v) \cap f^{-1}(R) \cap X_i| \geq \frac{k^2 \varepsilon |X_i|}{225000}$ holds with probability at least $1 - o(k\varepsilon)$.*

**(ii)** *For each R-heavy vertex $v$, $|N(v) \cap f^{-1}(L) \cap X_i| - |N(v) \cap f^{-1}(R) \cap X_i| \geq \frac{k^2 \varepsilon |X_i|}{225000}$ holds with probability at least $1 - o(k\varepsilon)$.*

We would like to note that Lemma 15.14 holds for any bipartition. However, we will use it only for completeness with resepct to an optimal bipartition $f$.

**Lemma 15.15.** *If $d_{bip}(G) \leq \varepsilon n^2$, then there exists an $i \in [t]$ and $f_{ij} \in \mathcal{F}_i$ such that $\zeta_{ij} \leq \left(2 + \frac{k}{20}\right)\varepsilon$ holds, with probability at least $1 - o(k\varepsilon)$.*

*Proof.* Let $f$ be an optimal bipartition such that $d_{bip}(G, f) \leq \varepsilon n^2$. First consider a SPECIAL bipartition $\text{SPL}_i^f$, and consider a set of random vertex pairs $Y$ such that $|Y| = |Z|$. Now consider the fraction of monochromatic edges of $Y$, with respect to the bipartition $\text{SPL}_i^f$, that is,

$$\chi_{ij}^f = 2 \cdot \frac{\left|\left\{\{a,b\} \in Y : \{a,b\} \in E(G) \text{ and } \text{SPL}_i^f(a) = \text{SPL}_i^f(b)\right\}\right|}{|Y|}.$$

**Observation 15.16.** With probability at least $\frac{9}{10}$, $\chi_{ij}^f \leq \left(2 + \frac{k}{20}\right)\varepsilon$ holds.

*Proof.* By Lemma 15.13, we know that if $d_{bip}(G) \leq \varepsilon n^2$, $d_{bip}(G, \text{SPL}_i^f) \leq \left(2 + \frac{k}{50}\right)\varepsilon n^2$. So, $\mathbb{E}[\chi_{ij}^f] \leq \left(2 + \frac{k}{50}\right)\varepsilon$. Using Chernoff bound (see Lemma 2.13), we can say that

$$\mathbb{P}\left(\chi_{ij}^f \geq \left(2 + \frac{k}{20}\right)\varepsilon\right) \leq \frac{1}{2^{\Omega\left(\frac{1}{k^3\varepsilon}\log\frac{1}{k\varepsilon}\right)}} \leq \frac{1}{10}.$$

□

Now, we claim that bounding $\chi_{ij}^f$ is equivalent to bounding $\zeta_{ij}$.

**Claim 15.17.** *For any $i \in [t]$, there exists a bipartition $f_{ij} \in \mathcal{F}_i$ such that the probability distribution of $\zeta_{ij}$ is identical to that of $\chi_{ij}^f$, for some* SPECIAL *bipartition $f$ with respect to $f_{ij}$, with probability at least $\frac{1}{2}$.*

As $t = \mathcal{O}(\log\frac{1}{k\varepsilon})$, the above claim implies that there exists an $i \in [t]$ and $f_{ij} \in \mathcal{F}_i$ such that the probability distribution of $\zeta_{ij}$ is identical to that of $\chi_{ij}^f$, with probability at least $1 - o(k\varepsilon)$.

Now we prove Claim 15.17. Recall the procedure of determining $\zeta_{ij}$ as described in **Step 2** of algorithm TOL-BIP-DIST$(G, \varepsilon)$ presented in Section 15.3.

**Fact 1:** For any vertex $v \in \mathcal{H}_f \cap Z$, $\textsc{Spl}_i^f(v) = f'_{ij}(v)$, with probability at least $1 - o(k\varepsilon)$, where $\mathcal{H}_f$ denotes the set of heavy vertices of $X_i$ with respect to the bipartition $f$. This follows according to Claim 15.14, along with the definition of $f'_{ij}(z)$.

**Fact 2:** Consider a bipartition $f_{ij} \in \mathcal{F}_i$ of $X_i$, and its extension $f^{'}_{ij}$ to $X_i \cup Z$, as considered in the algorithm. Assume a bipartition $f^{''}_{ij}$ of $V(G)$, constructed by extending $f'_{ij}$ according to the rule of **Step-2(ii)(b)** of the algorithm. From Heavy vertex lemma (Lemma 15.14), we know that the expected number of vertices in $\mathcal{H}_f$ such that $f^{''}_{ij}(v) \neq f(v)$, is at most $o(k\varepsilon)\left|\mathcal{H}_f\right|$. Using Markov inequality, we can say that, with probability at least $\frac{1}{2}$, the number of vertices in $\mathcal{H}_f$ such that $f^{''}_{ij}(v) \neq f(v)$, is at most $o(k\varepsilon)\left|\mathcal{H}_f\right|$. Thus, with probability at least $\frac{1}{2}$, there exists a set of vertices $\mathcal{H}'_f$ such that $f^{''}_{ij}(v) = f(v)$ holds for at least $(1 - o(k\varepsilon))\left|\mathcal{H}'_f\right|$ vertices. Note that the bipartition $f^{''}_{ij}$ is a SPECIAL bipartition $f$ with respect to $f_{ij}$.

From **Fact 1** and **Fact 2**, we can deduce that, there exists a SPECIAL bipartition $\textsc{Spl}_i^f$ such that $\textsc{Spl}_i^f(v) = f'_{ij}(v)$ for each $z \in Z$. Since we choose $Z$ uniformly at random, Lemma 15.15 follows. □

According to the description of algorithm $\textsc{Tol-Bip-Dist}(G, \varepsilon)$, the algorithm reports that $d_{bip}(G) \leq \varepsilon n^2$, if there exists a $\zeta_{ij}$ such that $\zeta_{ij} \leq \left(2 + \frac{k}{20}\right)\varepsilon$, for some $i \in [t]$ and $j \in [2^{|X_i|-2}]$. Hence, by Lemma 15.15, we are done with the proof of the completeness theorem (Theorem 15.12).

Now we focus on proving SPECIAL bipartition lemma (Lemma 15.13) and Heavy vertex lemma (Lemma 15.14), starting with the proof of SPECIAL bipartition lemma.

## Proof of SPECIAL bipartition lemma (Lemma 15.13)

The idea of the proof relies on decomposing the bipartite distance with respect to a SPECIAL bipartition into a sum of three terms and then carefully bounding the cost of each of those parts individually.

Let us first recall the definition of bipartite distance of $G$ with respect to a special bipartition $\textsc{Spl}_i^f$.

$$d_{bip}(G, \textsc{Spl}_i^f) = \left|\left\{(u,v) \in E(G) : \textsc{Spl}_i^f(u) = \textsc{Spl}_i^f(v)\right\}\right|. \tag{15.2}$$

By abuse of notation, here we are denoting $E(G)$ as the set of ordered edges.

We will upper bound $d_{bip}(G, \textsc{Spl}_i^f)$ as the sum of three terms defined below. Here $\mathcal{H}_f$ and $\mathcal{B}_f$ denote the set of heavy vertices and balanced vertices (with respect to $f$), as defined in Definition 15.8 and Definition 15.9, respectively. Also, $\mathcal{H}'_f \subseteq \mathcal{H}_f$ denotes the set of vertices of $\mathcal{H}_f$ that are mapped according to $f$, as defined in the definition of SPECIAL bipartition in Definition 15.10. The three terms that are used to upper bound $d_{bip}(G, \textsc{Spl}_i^f)$ are as follows:

$$\begin{aligned}
\textbf{(a)}\ D_{\mathcal{H}'_f \cup X_i, \mathcal{H}'_f \cup X_i} &= |\{(u,v) \in E(G) : u \in \mathcal{H}'_f \cup X_i \ \& \\
&\qquad v \in \mathcal{H}'_f \cup X_i, \textsc{Spl}_i^f(u) = \textsc{Spl}_i^f(v)\}|.\\
\textbf{(b)}\ D_{\mathcal{H}_f \setminus (\mathcal{H}'_f \cup X_i), V(G)} &= |\{(u,v) \in E(G) : u \in \mathcal{H}_f \setminus (\mathcal{H}'_f \cup X_i) \ \& \\
&\qquad v \in V(G), \textsc{Spl}_i^f(u) = \textsc{Spl}_i^f(v)\}|.\\
\textbf{(c)}\ D_{\mathcal{B}_f \setminus X_i, V(G)} &= |\{(u,v) \in E(G) : u \in \mathcal{B}_f \setminus X_i \ \& \\
&\qquad v \in V(G), \textsc{Spl}_i^f(u) = \textsc{Spl}_i^f(v)\}|.
\end{aligned}$$

Now from Equation 15.2 along with the above definitions, we can upper bound $d_{bip}(G, \textsc{Spl}_i^f)$ as follows:

$$d_{bip}(G, \textsc{Spl}_i^f) \le D_{\mathcal{H}'_f \cup X_i, \mathcal{H}'_f \cup X_i} + D_{\mathcal{H}_f \setminus (\mathcal{H}'_f \cup X_i), V(G)} + D_{\mathcal{B}_f \setminus X_i, V(G)}. \tag{15.3}$$

We now upper bound $d_{bip}(G, \textsc{Spl}_i^f)$ by bounding each term on the right hand side of the above expression separately, via the two following claims which we will prove later.

**Claim 15.18.** **(i)** $D_{\mathcal{H}'_f \cup X_i, \mathcal{H}'_f \cup X_i} \le d_{bip}(G, f) - \Pi$, *where*

$$\Pi := \left[ \sum_{v \in \mathcal{B}_f \setminus X_i : f(v) = L} |N(v) \cap f^{-1}(L)| + \sum_{v \in \mathcal{B}_f \setminus X_i : f(v) = R} |N(v) \cap f^{-1}(R)| \right];$$

**(ii)** $D_{\mathcal{H}_f \setminus (\mathcal{H}'_f \cup X_i), V(G)} \leq o(k\varepsilon)n^2$*;*

**Claim 15.19.** $D_{\mathcal{B}_f \setminus X_i, V(G)} \leq 2\left(1 + \frac{k}{400}\right)\Pi + \frac{k\varepsilon n^2}{150}$.

Assuming Claim 15.18 and Claim 15.19 hold, along with Equation 15.3, we now upper bound $d_{bip}(G, \text{SPL}_i^f)$ as follows:

$$\begin{aligned} d_{bip}(G, \text{SPL}_i^f) &\leq d_{bip}(G, f) - \Pi + o(k\varepsilon)n^2 + 2\left(1 + \frac{k}{400}\right)\Pi + \frac{k\varepsilon n^2}{150} \\ &\leq d_{bip}(G, f) + \Pi + \frac{k}{200}\Pi + \frac{k\varepsilon n^2}{100}. \end{aligned}$$

Note that $\Pi \leq d_{bip}(G, f)$ and $d_{bip}(G, f) \leq \varepsilon n^2$. Hence, we can say the following:

$$d_{bip}(G, \text{SPL}_i^f) \leq \left(2 + \frac{k}{50}\right)\varepsilon n^2.$$

So, we are done with the proof of the SPECIAL bipartition lemma. We now proceed with the proofs of Claim 15.18 and Claim 15.19.

**Proof of Claim 15.18.** (i) We use the following observation in our proof. The observation follows due to the fact that the bipartition $f$ considered is an optimal bipartition.

**Observation 15.20.** Let $v$ be a $L$-heavy vertex $v$ with respect to $f$. Then $f(v) = R$. Similarly, for every R-heavy vertex $v$ with respect to $f$, $f(v) = L$.

Following the definition of SPECIAL bipartition, we know that there exists a set of vertices $\mathcal{H}'_f \subset \mathcal{H}_f$ such that $\left|\mathcal{H}'_f\right| \geq (1 - o(k\varepsilon))\left|\mathcal{H}_f\right|$, and for each $v \in \mathcal{H}'_f$, the following holds:

$$\text{SPL}_i^f(v) = \begin{cases} R, & v \notin X_i \text{ and } v \text{ is } L-\text{heavy} \\ L, & v \notin X_i \text{ and } v \text{ is } R-\text{heavy} \end{cases}$$

By Observation 15.20, we know that for every $v \in \mathcal{H}'_f$, $\textsc{Spl}_i^f(v) = f(v)$. Moreover, for each $v \in X_i$, $\textsc{Spl}_i^f(v) = f(v)$, following the definition of SPECIAL bipartition $\textsc{Spl}_i^f$. Thus for every $v \in \mathcal{H}'_f \cup X_i$, $\textsc{Spl}_i^f(v) = f(v)$. Hence,

$$\begin{aligned}
&D_{\mathcal{H}'_f \cup X_i, \mathcal{H}'_f \cup X_i} \\
&= \left|\left\{(u,v) \in E(G) : u \in \mathcal{H}'_f \cup X_i \text{ and } v \in \mathcal{H}'_f \cup X_i, \textsc{Spl}_i^f(u) = \textsc{Spl}_i^f(v)\right\}\right| \\
&= \left|\{(u,v) \in E(G) : u \in \mathcal{H}'_f \cup X_i, \text{ and } v \in \mathcal{H}'_f \cup X_i, f(u) = f(v)\}\right| \\
&\qquad\qquad\qquad (\because \text{ for every } v \in \mathcal{H}'_f \cup X_i, \textsc{Spl}_i^f(v) = f(v)) \\
&= d_{bip}(G,f) - \left[\sum_{\substack{v \in V \setminus (\mathcal{H}'_f \cup X_i): \\ f(v)=L}} |N(v) \cap f^{-1}(L)| + \sum_{\substack{v \in V \setminus (\mathcal{H}'_f \cup X_i): \\ f(v)=R}} |N(v) \cap f^{-1}(R)|\right] \\
&\le d_{bip}(G,f) - \left[\sum_{v \in \mathcal{B}_f \setminus X_i : f(v)=L} |N(v) \cap f^{-1}(L)| + \sum_{v \in \mathcal{B}_f \setminus X_i : f(v)=R} |N(v) \cap f^{-1}(R)|\right] \\
&= d_{bip}(G,f) - \Pi.
\end{aligned}$$

(ii) By the definition of $\mathcal{H}'_f$, we know that $\left|\mathcal{H}_f \setminus (\mathcal{H}'_f \cup X_i)\right|$ is upper bounded by $o(k\varepsilon)\left|\mathcal{H}_f\right|$. Following the definition of $D_{\mathcal{H}_f \setminus (\mathcal{H}'_f \cup X_i), V(G)}$, we can say the following:

$$\begin{aligned}
D_{\mathcal{H}_f \setminus (\mathcal{H}'_f \cup X_i), V(G)} &= |\{(u,v) \in E(G) : u \in \mathcal{H}_f \setminus (\mathcal{H}'_f \cup X_i) \;\& \\
&\qquad v \in V(G), \textsc{Spl}_i^f(u) = \textsc{Spl}_i^f(v)\}| \\
&\le \left|\mathcal{H}_f \setminus (\mathcal{H}'_f \cup X_i)\right| \times |V(G)| = o(k\varepsilon)\left|\mathcal{H}_f\right| \times n \le o(k\varepsilon)n^2.
\end{aligned}$$

The last inequality follows as $|\mathcal{H}_f|$ is at most $n$. □

**Proof of Claim 15.19.** Observe that

$$\begin{aligned}
D_{\mathcal{B}_f \setminus X_i, V(G)} &= \left|\left\{(u,v) \in E(G) : u \in \mathcal{B}_f \setminus X_i \;\&\; v \in V(G), \textsc{Spl}_i^f(u) = \textsc{Spl}_i^f(v)\right\}\right| \\
&\le |\{(u,v) \in E(G) : u \in \mathcal{B}_f \setminus X_i \;\&\; v \in V(G)\}| = \sum_{v \in \mathcal{B}_f \setminus X_i} |N(v)|
\end{aligned}$$

As $\mathcal{B}_f = \mathcal{B}_f^1 \cup \mathcal{B}_f^2$, we have

$$D_{\mathcal{B}_f \setminus X_i, V(G)} \leq \sum_{v \in \mathcal{B}_f^1 \setminus X_i} |N(v)| + \sum_{v \in \mathcal{B}_f^2 \setminus X_i} |N(v)| . \tag{15.4}$$

We will bound $D_{\mathcal{B}_f \setminus X_i, V(G)}$ by bounding $\sum_{v \in \mathcal{B}_f^1 \setminus X_i} |N(v)|$ and $\sum_{v \in \mathcal{B}_f^2 \setminus X_i} |N(v)|$ separately, which we prove in the following claim:

**Claim 15.21.** *Let us consider $T_1$ and $T_2$ as follows:*

$$T_1 = 2 \left( \sum_{v \in f^{-1}(L) \cap (\mathcal{B}_f^1 \setminus X_i)} \left| N(v) \cap f^{-1}(L) \right| + \sum_{v \in f^{-1}(R) \cap (\mathcal{B}_f^1 \setminus X_i)} \left| N(v) \cap f^{-1}(R) \right| \right) + \frac{k\varepsilon n^2}{150}$$

$$T_2 = \left( 2 + \frac{k}{200} \right) \left( \sum_{v \in f^{-1}(L) \cap (\mathcal{B}_f^2 \setminus X_i)} \left| N(v) \cap f^{-1}(L) \right| + \sum_{v \in f^{-1}(R) \cap (\mathcal{B}_f^2 \setminus X_i)} \left| N(v) \cap f^{-1}(R) \right| \right)$$

*Then*

**(i)** *For balanced vertices of* **Type 1**, *we have* $\sum\limits_{v \in \mathcal{B}_f^1 \setminus X_i} |N(v)| \leq T_1$.

**(ii)** *For balanced vertices of* **Type 2**, *we have* $\sum_{v \in \mathcal{B}_f^2 \setminus X_i} |N(v)| \leq T_2$.

The proof of the above claim is presented in Section 15.4.2. Using Claim 15.21 and Equation (15.4), we have the following:

$$\begin{aligned} D_{\mathcal{B}_f \setminus X_i, V(G)} &= \sum_{v \in \mathcal{B}_f^1 \setminus X_i} |N(v)| + \sum_{v \in \mathcal{B}_f^2 \setminus X_i} |N(v)| \\ &\leq T_1 + T_2 \\ &\leq 2 \left( 1 + \frac{k}{400} \right) \Pi + \frac{k\varepsilon n^2}{150} && \text{(From the definitions of } T_1, T_2 \text{ and } \Pi) \end{aligned}$$

□

## Proof of Heavy vertex lemma (Lemma 15.14)

Before proceeding to prove the Heavy vertex lemma, we will first prove two intermediate claims that will be crucially used in the proof of the lemma. The first claim states that when we consider a bipartition $f$ of $G$, if a vertex $v \in G$ has a *large* number of neighbors on one side of the partition defined by $f$, the proportion of its neighbors in $X_i$ on the same side of $f$ will be approximately preserved, where $X_i$ is a set of vertices picked at random in **Step-1(i)** of the algorithm $\textsc{Tol-Bip-Dist}(G, \varepsilon)$. The result is formally stated as follows:

**Claim 15.22.** *Let $f$ be a bipartition of $G$. Consider a vertex $v \in V$.*

**(i)** *Suppose $|N(v) \cap f^{-1}(L)| \geq \frac{k\varepsilon n}{150}$. Then, with probability at least $1 - o(k\varepsilon)$, we have*

$$\left|N(v) \cap f^{-1}(L) \cap X_i\right| = \left(1 \pm \frac{k}{500}\right) \left|N(v) \cap f^{-1}(L)\right| \frac{|X_i|}{n}.$$

**(ii)** *Suppose $|N(v) \cap f^{-1}(R)| \geq \frac{k\varepsilon n}{150}$. Then, with probability at least $1 - o(k\varepsilon)$, we have*

$$|N(v) \cap f^{-1}(R) \cap X_i| = \left(1 \pm \tfrac{k}{500}\right) |N(v) \cap f^{-1}(R)| \tfrac{|X_i|}{n}.$$

The next claim is in similar spirit as that of Claim 15.22. Instead of considering vertices with large number of neighbors, it considers the case when a vertex has *small* number of neighbors on one side of a bipartition $f$.

**Claim 15.23.** *Let $f$ be a bipartition of $G$. Consider a vertex $v \in V$.*

**(i)** *Suppose $|N(v) \cap f^{-1}(L)| \leq \left(1 + \frac{k}{200}\right)^{-1} \frac{k\varepsilon n}{150}$. Then, with probability at least $1 - o(k\varepsilon)$, we have*

$$\left|N(v) \cap f^{-1}(L) \cap X_i\right| \leq \left(1 + \frac{k}{300}\right)^{-1} \frac{k\varepsilon \left|X_i\right|}{150}$$

**(ii)** *Suppose $|N(v) \cap f^{-1}(R)| \leq \left(1 + \frac{k}{200}\right)^{-1} \frac{k\varepsilon n}{150}$. Then, with probability at least $1 -$*

*$o(k\varepsilon)$, we have*

$$\left|N(v) \cap f^{-1}(R) \cap X_i\right| \leq \left(1 + \frac{k}{300}\right)^{-1} \frac{k\varepsilon\,|X_i|}{150}$$

Claim 15.22 and Claim 15.23 can be proved by using large deviation inequalities (stated in Section 2.3), and the proofs are presented in Appendix 15.4.2.

Assuming Claim 15.22 and Claim 15.23 hold, we now prove the Heavy vertex lemma (Lemma 15.14).

*Proof of Lemma 15.14.* We will only prove $(i)$ here, which concerns the $L$-heavy vertices. $(ii)$ can be proved in similar fashion. We first characterize $L$-heavy vertices into two categories:

**(a)** Both $|N(v) \cap f^{-1}(L)|$ and $|N(v) \cap f^{-1}(R)|$ are large, that is, $|N(v) \cap f^{-1}(L)| \geq \frac{k\varepsilon n}{150}$ and $|N(v) \cap f^{-1}(R)| \geq \left(1 + \frac{k}{200}\right)^{-1} \frac{k\varepsilon n}{150}$. Moreover, $|N(v) \cap f^{-1}(L)| \geq \left(1 + \frac{k}{200}\right) |N(v) \cap f^{-1}(R)|$.

**(b)** $|N(v) \cap f^{-1}(L)|$ is large and $|N(v) \cap f^{-1}(R)|$ is small, that is, $|N(v) \cap f^{-1}(L)| \geq \frac{k\varepsilon n}{150}$ and $|N(v) \cap f^{-1}(R)| \leq \left(1 + \frac{k}{200}\right)^{-1} \frac{k\varepsilon n}{150}$.

**Case (a):** Here $|N(v) \cap f^{-1}(L)| \geq \left(1 + \frac{k}{200}\right) \frac{k\varepsilon n}{150}$, and $|N(v) \cap f^{-1}(R)| \geq \frac{k\varepsilon n}{150}$. From Claim 15.22, the following hold, with probability at least $1 - o(k\varepsilon)$:

$$\left|N(v) \cap f^{-1}(L) \cap X_i\right| = \left(1 \pm \frac{k}{500}\right) \left|N(v) \cap f^{-1}(L)\right| \frac{|X_i|}{n}$$

and

$$\left|N(v) \cap f^{-1}(R) \cap X_i\right| = \left(1 \pm \frac{k}{500}\right) \left|N(v) \cap f^{-1}(R)\right| \frac{|X_i|}{n}.$$

So, with probability at least $1 - o(k\varepsilon)$, we have the following:

$$
\begin{aligned}
&\left|N(v) \cap f^{-1}(L) \cap X_i\right| - \left|N(v) \cap f^{-1}(R) \cap X_i\right| \\
&\geq \left(1 - \frac{k}{500}\right) \left|N(v) \cap f^{-1}(L)\right| \frac{|X_i|}{n} - \left(1 + \frac{k}{500}\right) \left|N(v) \cap f^{-1}(R)\right| \frac{|X_i|}{n} \\
&\geq \left(1 - \frac{k}{500} - \frac{1 + \frac{k}{500}}{1 + \frac{k}{200}}\right) \frac{|N(v) \cap f^{-1}(L)|\,|X_i|}{n} \\
&\qquad\qquad\qquad \left(\because \left|N(v) \cap f^{-1}(L)\right| \geq \left(1 + \frac{k}{200}\right) \left|N(v) \cap f^{-1}(R)\right|\right) \\
&\geq \frac{k}{1500} \times \frac{k\varepsilon\,|X_i|}{150} \\
&\geq \frac{k^2\varepsilon\,|X_i|}{225000} \qquad\qquad (\because\ k \leq 100)
\end{aligned}
$$

**Case (b):** Here $|N(v) \cap f^{-1}(L)| \geq \frac{k\varepsilon n}{150}$ and $|N(v) \cap f^{-1}(R)| \leq \left(1 + \frac{k}{200}\right)^{-1} \frac{k\varepsilon n}{150}$. From Claim 15.22 and Claim 15.23, the following hold, with probability at least $1 - o(k\varepsilon)$:

$$\left|N(v) \cap f^{-1}(L) \cap X_i\right| = \left(1 \pm \frac{k}{500}\right) \left|N(v) \cap f^{-1}(L)\right| \frac{|X_i|}{n}$$

and $|N(v) \cap f^{-1}(R) \cap X_i| \leq \left(1 + \frac{k}{300}\right)^{-1} \frac{k\varepsilon|X_i|}{150}$. Thus, with probability at least $1 - o(k\varepsilon)$, we have the following:

$$
\begin{aligned}
&\left|N(v) \cap f^{-1}(L) \cap X_i\right| - \left|N(v) \cap f^{-1}(R) \cap X_i\right| \\
&\geq (1 - \frac{k}{500}) \left|N(v) \cap f^{-1}(L)\right| \frac{|X_i|}{n} - \frac{1}{1 + \frac{k}{300}} \frac{k\varepsilon\,|X_i|}{150} \\
&= (1 - \frac{k}{500}) \frac{k\varepsilon\,|X_i|}{150} - \frac{1}{1 + \frac{k}{300}} \frac{k\varepsilon\,|X_i|}{150} \\
&\geq \frac{1}{1500} \left(2k - \frac{k^2}{100}\right) \frac{k\varepsilon\,|X_i|}{150} \\
&\geq \frac{k^2\varepsilon\,|X_i|}{225000} \qquad\qquad (\because\ k \leq 100)
\end{aligned}
$$

This completes the proof of part $(i)$ of Lemma 15.14. □

### 15.4.2 Proof of soundness

In this section, we prove the following theorem:

**Theorem 15.24.** *Let us assume that $G$ is $(2+k)\varepsilon$-far from being bipartite. Then* $\textsc{Tol-Bip-Dist}(G,\varepsilon)$ *reports the same, with probability at least* $9/10$.

Assume $f$ is a bipartition of $V(G)$. Now let us consider a DERIVED bipartition $\textsc{Der}_i^f$ with respect to $f$ by $f_{ij}$, and choose a set of random vertex pairs $Y$ such that $|Y| = |Z|$. Let $\chi_{ij}^f$ denote the fraction of vertex pairs of $Y$ that are monochromatic with respect to the bipartition $\textsc{Der}_i^f$, that is,

$$\chi_{ij}^f = 2 \cdot \frac{\left|\left\{\{a,b\} \in Y : \{a,b\} \in E(G) \text{ and } \textsc{Der}_i^f(a) = \textsc{Der}_i^f(b)\right\}\right|}{|Y|}.$$

**Observation 15.25.** $\chi_{ij}^f \leq \left(2+\frac{k}{20}\right)\varepsilon$ holds with probability at most $\frac{1}{10N}$, where $N = 2^{\mathcal{O}\left(\frac{1}{k^3\varepsilon}\log\frac{1}{k\varepsilon}\right)}$.

*Proof.* Since $G$ is $(2+k)\varepsilon$-far from being bipartite, the same holds for the bipartition $\textsc{Der}_i^f$ as well, that is, $d_{bip}(G, \textsc{Der}_i^f) \geq (2+k)\varepsilon n^2$. So, $\mathbb{E}[\chi_{ij}^f] \geq (2+k)\,\varepsilon$. Using Chernoff bound (see Lemma 2.13), we can say that, $\mathbb{P}\left(\chi_{ij}^f \leq \left(2+\frac{k}{20}\right)\varepsilon\right) \leq \frac{1}{10N}$. Since $|Z| = \mathcal{O}\left(\frac{1}{k^5\varepsilon^2}\log\frac{1}{k\varepsilon}\right)$, the result follows. □

We will be done with the proof by proving the following claim, that says that bounding $\chi_{ij}^f$ is equivalent to bounding $\zeta_{ij}$.

**Claim 15.26.** *For any $i \in [t]$, and any $f_{ij} \in \mathcal{F}_i$, the probability distribution of $\zeta_{ij}$ is identical to that of $\chi_{ij}^f$ for some* DERIVED *bipartition with respect to $f$ by $f_{ij}$.*

*Proof.* Consider a bipartition $f_{ij} \in \mathcal{F}_i$ of $X_i$, and the bipartition $f_{ij}^{'}$ of $X_i \cup Z$, constructed by extending $f_{ij}$, as described in the algorithm. For the sake of the argument, let us construct a new bipartition $f_{ij}^{''}$ of $V(G)$ by extending the bipartition $f'_{ij}$, following the same rule of **Step-2 (ii) (b)** of the algorithm. Observe that $f_{ij}^{''}(v) = f_{ij}(v)$, for each $v \in X_i$. Thus $f_{ij}^{''}$ is a DERIVED bipartition with respect to some $f$ by $f_{ij}$. Hence, the claim follows according to the way we generate $\zeta_{ij}$, along with the fact that $Z$ is chosen uniformly at random by the algorithm in **Step-1 (ii)**. □

Let us now define a pair $(X_i, f_{ij})$, with $i \in [t]$ and $f_{ij} \in \mathcal{F}_i$ as a **configuration**. Now we make the following observation which follows directly from the description of the algorithm.

**Observation 15.27.** Total number of possible configurations is $N = 2^{\mathcal{O}\left(\frac{1}{k^3\varepsilon}\log\frac{1}{k\varepsilon}\right)}$.

Note that Claim 15.26 holds for a particular $f_{ij} \in \mathcal{F}_i$. Recall that in **Step-2(iii)**, our algorithm TOL-BIP-DIST$(G, \varepsilon)$ reports that $G$ is $(2+k)\varepsilon$-far if $\zeta_{ij} > \left(2 + \frac{k}{20}\right)\varepsilon$, for all $i \in [t]$ and $f_{ij} \in \mathcal{F}_i$. So, using the union bound, along with Observation 15.25, Claim 15.26 and Observation 15.27, we are done with the proof of Theorem 15.24.

## Remaining proofs from this section

Here we include proofs of four claims that were not formally proven before in this section.

**Claim 15.28** (Restatement of Claim 15.21 (i)). *Let*

$$T_1 = 2\left(\sum_{v \in f^{-1}(L)\cap(\mathcal{B}_f^1\setminus X_i)} \left|N(v)\cap f^{-1}(L)\right| + \sum_{v \in f^{-1}(R)\cap(\mathcal{B}_f^1\setminus X_i)} \left|N(v)\cap f^{-1}(R)\right|\right) + \frac{k\varepsilon n^2}{150}.$$

*Then for balanced vertices of* **Type 1**, $\sum\limits_{v\in\mathcal{B}_f^1\setminus X_i} |N(v)| \leq T_1$.

*Proof.* Let us consider an optimal bipartition $f$. Then, for any vertex $v \in f^{-1}(L)\cap(\mathcal{B}_f^1\setminus X_i)$, we can show the following:

$$\frac{-k\varepsilon n}{150} \leq \left|N(v)\cap f^{-1}(L)\right| - \left|N(v)\cap f^{-1}(R)\right| \leq 0$$

Thus

$$\begin{aligned}\frac{-k\varepsilon n\left|f^{-1}(L)\cap(\mathcal{B}_f^1\setminus X_i)\right|}{150} &\leq \sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^1\setminus X_i)}\left|N(v)\cap f^{-1}(L)\right| \\ &\quad -\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^1\setminus X_i)}\left|N(v)\cap f^{-1}(R)\right| \\ &\leq 0\end{aligned}$$

Similarly, we can also show that

$$\begin{aligned}\frac{-k\varepsilon n\left|f^{-1}(R)\cap(\mathcal{B}_f^1\setminus X_i)\right|}{150} &\leq \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^1\setminus X_i)}\left|N(v)\cap f^{-1}(R)\right| \\ &\quad -\sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^1\setminus X_i)}\left|N(v)\cap f^{-1}(L)\right| \\ &\leq 0.\end{aligned}$$

Since $f^{-1}(L)\cup f^{-1}(R)=V(G)$, and $f^{-1}(L)\cap f^{-1}(R)=\emptyset$, we have the following four inequalities:

$$\begin{aligned}\frac{-k\varepsilon n\left|\mathcal{B}_f^1\setminus X_i\right|}{150} \leq &\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^1\setminus X_i)}\left|N(v)\cap f^{-1}(L)\right| + \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^1\setminus X_i)}\left|N(v)\cap f^{-1}(R)\right| \\ &-\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^1\setminus X_i)}\left|N(v)\cap f^{-1}(R)\right| + \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^1\setminus X_i)}\left|N(v)\cap f^{-1}(L)\right|\end{aligned}$$

So,

$$\begin{aligned}&\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^1\setminus X_i)}\left|N(v)\cap f^{-1}(R)\right| + \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^1\setminus X_i)}\left|N(v)\cap f^{-1}(L)\right| \\ &\quad\leq \sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^1\setminus X_i)}\left|N(v)\cap f^{-1}(L)\right| + \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^1\setminus X_i)}\left|N(v)\cap f^{-1}(R)\right| \\ &\qquad + \frac{k\varepsilon n\left|\mathcal{B}_f^1\setminus X_i\right|}{150}\end{aligned}$$

Therefore

$$\begin{aligned}
&\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^1\setminus X_i)} |N(v)| + \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^1\setminus X_i)} |N(v)| \\
\leq\ & 2\left(\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^1\setminus X_i)} \left|N(v)\cap f^{-1}(L)\right| + \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^1\setminus X_i)} \left|N(v)\cap f^{-1}(R)\right|\right) + \frac{k\varepsilon n^2}{150}
\end{aligned}$$

So we conclude that $\sum_{v\in\mathcal{B}_f^1\setminus X_i} |N(v)| \ \leq\ T_1$. □

**Claim 15.29** (Restatement of Claim 15.21(ii)). *Let*

$$T_2 = \left(2+\frac{k}{200}\right)\left(\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(L)\right| + \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(R)\right|\right).$$

*Then, for balanced vertices of* **Type 2**, $\sum\limits_{v\in\mathcal{B}_f^2\setminus X_i} |N(v)| \leq T_2$.

*Proof.* Recall the definition of balanced vertices of **Type 2** from Definition 15.9. Summing over all the vertices of $f^{-1}(L)\cap(\mathcal{B}_f^2\setminus X_i)$, we have

$$\begin{aligned}
\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(L)\right| &\leq \sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(R)\right| \\
&\leq \left(1+\frac{k}{200}\right)\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(L)\right|
\end{aligned}$$

Similarly, we can also say that

$$\begin{aligned}
\sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(R)\right| &\leq \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(L)\right| \\
&\leq \left(1+\frac{k}{200}\right)\sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(R)\right|.
\end{aligned}$$

Summing the above two inequalities, we get the following three inequalities:

$$\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(R)\right| + \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(L)\right|$$
$$\leq \left(1+\frac{k}{200}\right)\left(\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(L)\right| + \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(R)\right|\right)$$

So,

$$\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^2\setminus X_i)} |N(v)| + \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^2\setminus X_i)} |N(v)|$$
$$\leq \left(2+\frac{k}{200}\right)\left(\sum_{v\in f^{-1}(L)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(L)\right| + \sum_{v\in f^{-1}(R)\cap(\mathcal{B}_f^2\setminus X_i)} \left|N(v)\cap f^{-1}(R)\right|\right).$$

Thus, we have $\sum_{v\in(\mathcal{B}_f^2\setminus X_i)} |N(v)| \;\leq\; T_2$. □

**Claim 15.30** (Restatement of Claim 15.22)**.** *Let $f$ be a bipartition of $G$. Consider a vertex $v \in V$.*

**(i)** *Suppose $|N(v)\cap f^{-1}(L)| \geq \frac{k\varepsilon n}{150}$. Then with probability at least $1-o(k\varepsilon)$, we have $|N(v)\cap f^{-1}(L)\cap X_i| = \left(1\pm\frac{k}{500}\right)|N(v)\cap f^{-1}(L)|\frac{|X_i|}{n}$ holds.*

**(ii)** *Suppose $|N(v)\cap f^{-1}(R)| \geq \frac{k\varepsilon n}{150}$. Then with probability at least $1-o(k\varepsilon)$, we have $|N(v)\cap f^{-1}(R)\cap X_i| = \left(1\pm\frac{k}{500}\right)|N(v)\cap f^{-1}(R)|\frac{|X_i|}{n}$.*

*Proof.* We prove only part $(i)$ of the claim. Part $(ii)$ can be proven analogously.

From the condition stated in $(i)$, we know that

$$\left|N(v)\cap f^{-1}(L)\right| \geq \frac{k\varepsilon n}{150}.$$

Since $X_i$ is chosen randomly, we can say that

$$\mathbb{E}\left[\left|N(v)\cap f^{-1}(L)\cap X_i\right|\right] \geq \frac{k\varepsilon\,|X_i|}{150}.$$

Using Chernoff bound (see Lemma 2.13), we have

$$\begin{aligned}\mathbb{P}\left(\left|N(v)\cap f^{-1}(L)\cap X_i\right| \neq \left(1\pm\frac{k}{500}\right)\left|N(v)\cap f^{-1}(L)\right|\frac{|X_i|}{n}\right) &\leq 2e^{-\Omega\left(k^3\varepsilon|X_i|\right)}\\ &= o(k\varepsilon)\end{aligned}$$

The last inequality follows from the fact that $|X_i| = \mathcal{O}(\frac{1}{k^3\varepsilon}\log\frac{1}{k\varepsilon})$. □

**Claim 15.31** (Restatement of Claim 15.23). *Let $f$ be a bipartition of $G$. Consider a vertex $v \in V$.*

**(i)** *Suppose $|N(v)\cap f^{-1}(L)| \leq \left(1+\frac{k}{200}\right)^{-1}\frac{k\varepsilon n}{150}$. Then, with probability at least $1-o(k\varepsilon)$, we have $|N(v)\cap f^{-1}(L)\cap X_i| \leq \left(1+\frac{k}{300}\right)^{-1}\frac{k\varepsilon|X_i|}{150}$.*

**(ii)** *Suppose $|N(v)\cap f^{-1}(R)| \leq \left(1+\frac{k}{200}\right)^{-1}\frac{k\varepsilon n}{150}$. Then, with probability at least $1-o(k\varepsilon)$, we have $|N(v)\cap f^{-1}(R)\cap X_i| \leq \left(1+\frac{k}{300}\right)^{-1}\frac{k\varepsilon|X_i|}{150}$.*

*Proof.* We will only prove part $(i)$ here. Part $(ii)$ can be proven in similar manner.

From the condition stated in $(i)$, we know that

$$\left|N(v)\cap f^{-1}(R)\right| \leq \left(1+\frac{k}{200}\right)^{-1}\frac{(1+k)\varepsilon n}{150}.$$

Since $X_i$ is chosen at random, we can say that

$$\mathbb{E}\left[\left|N(v)\cap f^{-1}(R)\cap X_i\right|\right] \leq \left(1+\frac{k}{200}\right)^{-1}\frac{(1+k)\varepsilon\,|X_i|}{150}.$$

Using Chernoff bound (see Lemma 2.13), we have

$$\mathbb{P}\left(\left|N(v)\cap f^{-1}(L)\cap X_i\right| \geq \left(1+\frac{k}{300}\right)^{-1}\frac{(1+k)\varepsilon\,|X_i|}{150}\right) \leq e^{-\Omega\left(k^2\varepsilon|X_i|\right)} \leq o(k\varepsilon).$$

The last inequality follows due to the fact that $|X_i| = \mathcal{O}(\frac{1}{k^3\varepsilon}\log\frac{1}{k\varepsilon})$. □

# Chapter 16

# Conclusion

In this thesis, we considered sample and query complexities of various properties of distributions and graphs. There are several open problems that have come out of these works. We discuss them below:

In Part I (Chapter 4, Chapter 5, and Chapter 6), we studied the relation between the sample complexities of non-tolerant and tolerant testing of label-invariant distribution properties. We proved that this gap is at most quadratic, which is almost tight. We also proved lower bound results of non-tolerant and tolerant testing of non-concentrated properties, where the probability mass of the distributions in the property are sufficiently spread. We further designed an algorithm of learning a concentrated distribution, even for the case when the support of the distribution is unknown apriori. It is interesting to note that our proof technique does not immediately generalize for non-label-invariant properties. So, a natural open question is:

*Can one show a relation between the non-tolerant and tolerant sample complexities of non-label-invariant properties?*

In Part II (Chapter 8, Chapter 9, Chapter 10, and Chapter 11), we studied several properties in the huge object model introduced by Goldreich and Ron [GR22]. In this model, distributions are defined over $n$-dimensional Hamming cube $\{0, 1\}^n$, and the samples obtained from the oracle representing the distribution are $n$-bit strings. We have sampling access to the distribution, along with query access to the sampled strings, and the goal is to optimize the sample and query complexities of the testers. We defined the notion of a new class of properties, namely index-invariant properties, which are properties that are invariant under the permutation of the indices of the strings. In particular, in Chapter 8, we studied the problem of learning distributions that can be clustered, and designed an efficient algorithm for learning such distributions in the huge object model. Then in Chapter 9, we proved that every index-invariant property whose VC-dimension is bounded has a tester with a number of queries independent of $n$, and depends only on the VC-dimension and the proximity parameter. Moreover, the dependencies of the sample and query complexities on the VC-dimension are also tight. Later, in Chapter 10 and Chapter 11, we explored the power of adaptive testers compared to their non-adaptive counterparts in this model. We showed that for index-invariant properties, there is a tight quadratic gap. However, for general non-index-invariant properties, there is a tight exponential gap between the query complexities of adaptive and non-adaptive testers. Since this is a very new model, it would be very interesting to explore how the query complexities of testing various properties depend on different measures other than the VC-dimension.

In Part III of this thesis (Chapter 13, Chapter 14, and Chapter 15), we studied the query complexities of some graph property testing problems in the adjacency matrix model. In this model, the graphs are stored as adjacency matrix, and the tester can ask queries of the form if there is an edge between two vertices, say $u$ and $v$. The oracle storing the adjacency matrix corresponding to the graph will return $1$ if there is an edge between $u$ and $v$, and $0$ otherwise.

In particular, in Chapter 13 and Chapter 14, we studied the problem of tolerant testing of graph isomorphism (GI) between a known graph $G_k$ and an unknown graph $G_u$, each with $n$ vertices. We proved that the query complexity of tolerant graph isomorphism testing between $G_k$ and $G_u$ is the same as tolerant testing of Earth Mover Distance (EMD) between a known multi-set $S_k$ and an unknown multi-set $S_u$ when we have samples without replacement from $S_u$, ignoring polylogarithmic factors. Here the multi-sets $S_k$ and $S_u$ are constructed suitably from the graphs $G_k$ and $G_u$, respectively. We also showed when we are sampling with replacement from $S_u$, $\Omega(n/\log n)$ samples are required for tolerant testing of EMD. However, when we are sampling without replacement from $S_u$, the only known lower bound is $\Omega(\sqrt{n})$. So, a natural open question is:

*What is the tight sample complexity of tolerant EMD testing when we have samples without replacement from the unknown multi-set $S_u$?*

Fischer and Matsliah [FM08] studied graph isomorphism testing for both the cases (i) when one graph is known and the other graph is unknown and (ii) when both the graphs are unknown. They resolved the query complexity of (i), whereas Onak and Sun [OS18] resolved (ii). With this work, we initiate the study of tolerant graph isomorphism problem in the query and communication world. So, another natural open question to look for is:

*What is the query complexity of tolerant graph isomorphism when both the graphs are unknown?*

Finally, in Chapter 15, we studied the query complexity of tolerant bipartiteness testing of dense graphs. Here, given query access to the adjacency matrix of an unknown dense graph $G$, the goal is to distinguish whether $d_{bip}(G) \leq \varepsilon n^2$ or $d_{bip}(G) \geq c\varepsilon n^2$ for any $c > 1$, where $d_{bip}(G)$ denotes the bipartite distance of $G$. For $c \geq 2 + \Omega(1)$, we designed an algorithm that solves this problem by sampling $\mathcal{O}\left(\frac{1}{\varepsilon^3}\log\frac{1}{\varepsilon}\right)$ vertices in $2^{\mathcal{O}\left(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\right)}$ time, and performs $\mathcal{O}\left(\frac{1}{\varepsilon^3}\log^2\frac{1}{\varepsilon}\right)$ queries. For the case of distinguishing $d_{bip}(G) \leq \varepsilon n^2$ from $d_{bip}(G) \geq (1+k)\varepsilon n^2$ for some $k > 0$, there is an algorithm that performs $\widetilde{\mathcal{O}}\left(\frac{1}{k^6\varepsilon^6}\right)$ queries, which can be derived from the work of Alon, Vega, Kannan and Karpinski [AdlVKK03] (see Corollary 15.5 in Section 15.2). So, a natural open question is:

*Is there an algorithm for distinguishing $d_{bip}(G) \leq \varepsilon n^2$ from $d_{bip}(G) \geq (1+k)\varepsilon n^2$ with query complexity $o\left(\frac{1}{k^6\varepsilon^6}\right)$?*

# Bibliography

[ABC$^+$13] Noga Alon, Eric Blais, Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Nearly tight bounds for testing function isomorphism. *SIAM Journal on Computing (SICOMP)*, 2013. 29

[ABEF17] Noga Alon, Omri Ben-Eliezer, and Eldar Fischer. Testing hereditary properties of ordered graphs and matrices. In *Foundations of Computer Science (FOCS)*, 2017. 89

[ABR16] Maryam Aliakbarpour, Eric Blais, and Ronitt Rubinfeld. Learning and testing junta distributions. In *Conference on Learning Theory (COLT)*, 2016. 80

[ACF$^+$21] Jayadev Acharya, Clément L. Canonne, Cody Freitag, Ziteng Sun, and Himanshu Tyagi. Inference under information constraints III: local privacy constraints. *IEEE J. Sel. Areas Inf. Theory*, 2021. 29

[ACK18] Jayadev Acharya, Clément L Canonne, and Gautam Kamath. A chasm between identity and equivalence testing with conditional queries. *Theory of Computing (TOC)*, 2018. 86

[ADK15] Jayadev Acharya, Constantinos Daskalakis, and Gautam Kamath. Optimal testing for properties of distributions. In *Neural Information Processing Systems (NIPS)*, 2015. 2, 8, 32, 81, 188

[ADKR19] Maryam Aliakbarpour, Ilias Diakonikolas, Daniel Kane, and Ronitt Rubinfeld. Private testing of distributions via sample permutations. In *Neural Information Processing Systems (NeurIPS)*, 2019. 29

[ADLS17] Jayadev Acharya, Ilias Diakonikolas, Jerry Li, and Ludwig Schmidt. Sample-optimal density estimation in nearly-linear time. In *Symposium on Discrete Algorithms (SODA)*, 2017. 39, 73

[AdlVKK03] Noga Alon, Wenceslas Fernandez de la Vega, Ravi Kannan, and Marek Karpinski. Random Sampling and Approximation of MAX-CSPs. *Journal of Computer and System Sciences (JCSS)*, 2003. 3, 16, 17, 191, 252, 253, 254, 279

[AFKS00] Noga Alon, Eldar Fischer, Michael Krivelevich, and Mario Szegedy. Efficient testing of large graphs. *Combinatorica*, 2000. 4, 191

[AFNS09] Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. *SIAM Journal on Computing (SICOMP)*, 2009. 4, 191

[AK02] Noga Alon and Michael Krivelevich. Testing k-colorability. *SIAM Journal on Discrete Mathematics (SIDMA)*, 2002. 3, 4, 190

[AKKR08] Noga Alon, Tali Kaufman, Michael Krivelevich, and Dana Ron. Testing triangle-freeness in general graphs. *SIAM Journal on Discrete Mathematics (SIDMA)*, 2008. 4

[AKLS21] Jayadev Acharya, Peter Kairouz, Yuhan Liu, and Ziteng Sun. Estimating sparse discrete distributions under privacy and communication constraints. In *Algorithmic Learning Theory (ALT)*, 2021. 72

[AKNS99] Noga Alon, Michael Krivelevich, Ilan Newman, and Mario Szegedy. Regular languages are testable with a constant number of queries. In *Foundations of Computer Science (FOCS)*, 1999. 145

[AS05] Noga Alon and Asaf Shapira. Every monotone graph property is testable. In *Symposium on Theory of Computing (STOC)*, 2005. 4

[AS08] Noga Alon and Asaf Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM Journal on Computing (SICOMP)*, 2008. 4

[Bab16] László Babai. Graph Isomorphism in Quasipolynomial Time. In *Symposium on Theory of Computing (STOC)*, 2016. 15, 185

[BBC$^+$10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 2010. 29

[BC10] Laszlo Babai and Sourav Chakraborty. Property Testing of Equivalence under a Permutation Group Action. *ACM Transactions on Computation Theory (TOCT)*, 2010. 3, 187

[BC17] Tugkan Batu and Clément L Canonne. Generalized uniformity testing. In *Foundations of Computer Science (FOCS)*, 2017. 29, 32, 81

[BC18] Rishiraj Bhattacharyya and Sourav Chakraborty. Property testing of joint distributions using conditional samples. *ACM Transactions on Computation Theory (TOCT)*, 2018. 80

[BCE$^+$19] Eric Blais, Clément L. Canonne, Talya Eden, Amit Levi, and Dana Ron. Tolerant junta testing and the connection to submodular optimization and function isomorphism. *ACM Transactions on Computation Theory (TOCT)*, 2019. 33

[BCY22] Arnab Bhattacharyya, Clément L. Canonne, and Joy Qiping Yang. Independence testing for bounded degree bayesian network. In *Neural Information Processing Systems (NeurIPS)*, 2022. 80

[BDKR05] Tugkan Batu, Sanjoy Dasgupta, Ravi Kumar, and Ronitt Rubinfeld. The complexity of approximating the entropy. *SIAM Journal on Computing (SICOMP)*, 2005. 81

[BDST15] László Babai, Anuj Dawar, Pascal Schweitzer, and Jacobo Torán. The Graph Isomorphism Problem (Dagstuhl Seminar 15511). *Dagstuhl Reports*, 2015. 15, 185

[BFF$^+$01] Tugkan Batu, Lance Fortnow, Eldar Fischer, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. In *Foundations of Computer Science (FOCS)*, 2001. 2, 29, 32, 188

[BFLR20] Omri Ben-Eliezer, Eldar Fischer, Amit Levi, and Ron D. Rothblum. Hard properties with (very) short pcpps and their applications. In *Innovations in Theoretical Computer Science (ITCS)*, 2020. 98, 147, 153

[BFR$^+$00] Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. Testing that distributions are close. In *Foundations of Computer Science (FOCS)*, 2000. 2, 8, 29, 32

[BGMV20] Arnab Bhattacharyya, Sutanu Gayen, Kuldeep S. Meel, and N. V. Vinodchandran. Efficient distance approximation for structured high-dimensional distributions via learning. In *Neural Information Processing Systems (NeurIPS)*, 2020. 80

[BHR05] Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. Some 3cnf properties are hard to test. *SIAM Journal on Computing (SICOMP)*, 2005. 147

[Bir46] Garrett Birkhoff. Three observations on linear algebra. *Univ. Nac. Tacuman, Rev. Ser. A*, 1946. 132

[BKR04] Tugkan Batu, Ravi Kumar, and Ronitt Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In *Symposium on Theory of Computing (STOC)*, 2004. 2, 8

[BL10] Andrej Bogdanov and Fan Li. A better tester for bipartiteness? *arXiv preprint arXiv:1011.0531*, 2010. 190

[BOT02] Andrej Bogdanov, Kenji Obata, and Luca Trevisan. A lower bound for testing 3-colorability in bounded-degree graphs. In *Symposium on Foundations of Computer Science (FOCS)*, 2002. 4

[BT97] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997. 53, 58

[BT04] Andrej Bogdanov and Luca Trevisan. Lower bounds for testing bipartiteness in dense graphs. In *Conference on Computational Complexity (CCC)*, 2004. 190

[BY22] Arnab Bhattacharyya and Yuichi Yoshida. *Property Testing - Problems and Techniques*. Springer, 2022. 4, 32, 80, 189

[Can20a] Clément L. Canonne. A Survey on Distribution Testing: Your Data is Big. But is it Blue? *Theory of Computing*, (9), 2020. 4, 80

[Can20b] Clément L Canonne. A short note on learning discrete distributions. *https://github.com/ccanonne/probabilitydistributiontoolbox/blob/master/learning.pdf*, 2020. 72

[Can20c] Clément L. Canonne. *A Survey on Distribution Testing: Your Data is Big. But is it Blue?* Theory of Computing Library, 2020. 33, 195, 202

[Can22] Clément L. Canonne. Topics and techniques in distribution testing: A biased but representative sample. *Foundations and Trends® in Communications and Information Theory*, 2022. 4, 33, 80

[CCK$^+$21] Clément L. Canonne, Xi Chen, Gautam Kamath, Amit Levi, and Erik Waingarten. Random restrictions of high dimensional distributions and uniformity testing with subcube conditioning. In *Symposium on Discrete Algorithms (SODA)*, 2021. 80

[CDGR18] Clément L. Canonne, Ilias Diakonikolas, Themis Gouleakis, and Ronitt Rubinfeld. Testing shape restrictions of discrete distributions. *Theory of Computing Systems (TOCS)*, 2018. 29, 33

[CDKS17] Clément L. Canonne, Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Testing bayesian networks. In *Conference on Learning Theory (COLT)*, 2017. 80

[CDVV14] Siu-On Chan, Ilias Diakonikolas, Paul Valiant, and Gregory Valiant. Optimal algorithms for testing closeness of discrete distributions. In *Symposium on Discrete Algorithms (SODA)*, 2014. 81

[CEF$^+$05] Artur Czumaj, Funda Ergün, Lance Fortnow, Avner Magen, Ilan Newman, Ronitt Rubinfeld, and Christian Sohler. Approximating the weight of the euclidean minimum spanning tree in sublinear time. *SIAM Journal on Computing (SICOMP)*, 2005. 4

[CF14] Gregory W Corder and Dale I Foreman. *Nonparametric statistics: A step-by-step approach.* John Wiley & Sons, 2014. 32

[CFG$^+$23] Sourav Chakraborty, Eldar Fischer, Arijit Ghosh, Gopinath Mishra, and Sayantan Sen. Testing of index-invariant properties in the huge object model. In *Conference on Learning Theory (COLT)*, 2023. 11, 13

[CFGM16] Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah. On the power of conditional samples in distribution testing. *SIAM Journal on Computing (SICOMP)*, 2016. 86

[CGMS21] Sourav Chakraborty, Arijit Ghosh, Gopinath Mishra, and Sayantan Sen. Interplay between graph isomorphism and earth mover's distance in the query and communication worlds. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX-/RANDOM)*, 2021. 16

[CGR$^+$14] Artur Czumaj, Oded Goldreich, Dana Ron, C Seshadhri, Asaf Shapira, and Christian Sohler. Finding cycles and trees in sublinear time. *Random Structures & Algorithms (RSA)*, 2014. 4

[Cha00] B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, Cambridge, New York, 2000. 89

[CJKL22] Clément L. Canonne, Ayush Jain, Gautam Kamath, and Jerry Li. The price of tolerance in distribution testing. In *Conference on Learning Theory (COLT)*, 2022. 33, 74

[CJLW21] Xi Chen, Rajesh Jayaram, Amit Levi, and Erik Waingarten. Learning and testing junta distributions with sub cube conditioning. In *Conference on Learning Theory (COLT)*, 2021. 80

[CKÖ20] Wei-Ning Chen, Peter Kairouz, and Ayfer Özgür. Breaking the communication-privacy-accuracy trilemma. In *Neural Information Processing Systems (NeurIPS)*, 2020. 72

[CKS20] Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. In *Neural Information Processing Systems (NeurIPS)*, 2020. 80

[CM19] Sourav Chakraborty and Kuldeep S. Meel. On testing of uniform samplers. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2019. 80

[CMOS19] Artur Czumaj, Morteza Monemizadeh, Krzysztof Onak, and Christian Sohler. Planar graphs: Random walks and bipartiteness testing. *Random Structures & Algorithms (RSA)*, 2019. 189, 191

[CPS15] Artur Czumaj, Pan Peng, and Christian Sohler. Testing cluster structure of graphs. In *Symposium on Theory of Computing (STOC)*, 2015. 4

[CRS15] Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Testing probability distributions using conditional samples. *SIAM Journal on Computing (SICOMP)*, 2015. 86

[CRT05] Bernard Chazelle, Ronitt Rubinfeld, and Luca Trevisan. Approximating the minimum spanning tree weight in sublinear time. *SIAM Journal on computing (SICOMP)*, 2005. 4

[CS09] Artur Czumaj and Christian Sohler. Estimating the weight of metric minimum spanning trees in sublinear time. *SIAM Journal on Computing (SICOMP)*, 2009. 4

[CS10a] Artur Czumaj and Christian Sohler. Sublinear-time algorithms. In *Property Testing - Current Research and Surveys*. 2010. 4, 80

[CS10b] Artur Czumaj and Christian Sohler. Testing expansion in bounded-degree graphs. *Combinatorics, Probability and Computing (CPC)*, 2010. 4

[CSS09] Artur Czumaj, Asaf Shapira, and Christian Sohler. Testing hereditary properties of nonexpanding bounded-degree graphs. *SIAM Journal on Computing (SICOMP)*, 2009. 4

[CT01] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2001. 32

[DBNNR11] Khanh Do Ba, Huy L Nguyen, Huy N Nguyen, and Ronitt Rubinfeld. Sublinear time algorithms for earth mover's distance. *Theory of Computing Systems (TOCS)*, 2011. 196, 203

[DK16] Ilias Diakonikolas and Daniel M. Kane. A new approach for testing properties of discrete distributions. In *Foundations of Computer Science (FOCS)*, 2016. 29

[DKN14] Ilias Diakonikolas, Daniel M Kane, and Vladimir Nikishkin. Testing identity of structured distributions. In *Symposium on Discrete Algorithms (SODA)*, 2014. 81

[DKS17] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *Foundations of Computer Science (FOCS)*, 2017. 29

[DKS18] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Sharp bounds for generalized uniformity testing. In *Neural Information Processing Systems (NeurIPS)*, 2018. 29, 81

[DKW18] Constantinos Daskalakis, Gautam Kamath, and John Wright. Which distribution distances are sublinearly testable? In *Symposium on Discrete Algorithms (SODA)*, 2018. 32

[DL12] Luc Devroye and Gábor Lugosi. *Combinatorial Methods in Density Estimation*. Springer Science & Business Media, 2012. 205

[DLM$^+$07] Ilias Diakonikolas, Homin K Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco A Servedio, and Andrew Wan. Testing for concise representations. In *Foundations of Computer Science (FOCS)*, 2007. 37, 83

[DP09] Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009. 23, 24

[ELRS17] Talya Eden, Amit Levi, Dana Ron, and C Seshadhri. Approximately counting triangles in sublinear time. *SIAM Journal on Computing (SICOMP)*, 2017. 4

[ER18] Talya Eden and Will Rosenbaum. On sampling edges almost uniformly. In *1st Symposium on Simplicity in Algorithms (SOSA 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018. 4

[ERS19] Talya Eden, Dana Ron, and C Seshadhri. Sublinear time estimation of degree distribution moments: The arboricity connection. *SIAM Journal on Discrete Mathematics (SIDMA)*, 2019. 4

[Fei04] Uriel Feige. On sums of independent random variables with unbounded variance, and estimating the average degree in a graph. In *Symposium on Theory of computing (STOC)*, 2004. 4

[Fis04] Eldar Fischer. The art of uninformed decisions. *Current Trends in Theoretical Computer Science: The Challenge of the New Century*, 2004. 4, 80, 91

[FLV15] Eldar Fischer, Oded Lachish, and Yadu Vasudev. Trading query complexity for sample-based testing and multi-testing scalability. In *Foundations of Computer Science (FOCS)*, 2015. 33

[FLV17] Eldar Fischer, Oded Lachish, and Yadu Vasudev. Improving and extending the testing of distributions for shape-restricted properties. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, 2017. 29, 33

[FM08] Eldar Fischer and Arie Matsliah. Testing graph isomorphism. *SIAM Journal on Computing (SICOMP)*, 2008. 3, 15, 29, 187, 188, 192, 208, 209, 279

[FN07] Eldar Fischer and Ilan Newman. Testing versus estimation of graph properties. *SIAM Journal on Computing (SICOMP)*, 2007. 4, 191

[Fre77] David Freedman. A Remark on the Difference between Sampling with and without Replacement. *Journal of the American Statistical Association*, 1977. 192, 195

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property Testing and its Connection to Learning and Approximation. *Journal of the ACM (JACM)*, 1998. 2, 3, 6, 15, 16, 184, 188, 189, 190

[GKK$^+$20] Sivakanth Gopi, Gautam Kamath, Janardhan Kulkarni, Aleksandar Nikolov, Zhiwei Steven Wu, and Huanyu Zhang. Locally private hypothesis selection. In *Conference on Learning Theory (COLT)*, 2020. 29

[GM07] Bernd Gärtner and Jirí Matousek. *Understanding and using linear programming*. Springer, 2007. 58

[GMRS22] Arijit Ghosh, Gopinath Mishra, Rahul Raychaudhury, and Sayantan Sen. Tolerant bipartiteness testing in dense graphs. In *International Colloquium on Automata, Languages and Programming (ICALP)*, 2022. 17

[Gol17] Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017. 4, 32, 80, 189

[Gol19] Oded Goldreich. Testing isomorphism in the bounded-degree graph model. *Electron. Colloquium Comput. Complex.*, 2019. 29, 187, 205

[GOS$^+$09] Parikshit Gopalan, Ryan O'Donnell, Rocco A. Servedio, Amir Shpilka, and Karl Wimmer. Testing fourier dimensionality and sparsity. In *International Colloquium on Automata, Languages and Programming (ICALP)*, 2009. 83

[GR97] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. In *Symposium on Theory of Computing (STOC)*, 1997. 2, 4, 13, 86, 191

[GR99] Oded Goldreich and Dana Ron. A sublinear bipartiteness tester for bounded degree graphs. *Combinatorica*, 1999. 4, 189, 191

[GR00] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. *Electronic Colloquium of Computational Complexity (ECCC)*, 2000. 2, 32

[GR07] Mira Gonen and Dana Ron. On the benefits of adaptivity in property testing of dense graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*. 2007. 190

[GR08] Oded Goldreich and Dana Ron. Approximating average parameters of graphs. *Random Structures & Algorithms (RSA)*, 2008. 4

[GR11] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*. Springer, 2011. 2, 4, 81

[GR16] Oded Goldreich and Dana Ron. On sample-based testers. *ACM Transactions on Computation Theory (TOCT)*, 2016. 33

[GR22] Oded Goldreich and Dana Ron. Testing distributions of huge objects. In *Innovations in Theoretical Computer Science (ITCS)*, 2022. 5, 11, 13, 81, 85, 88, 129, 142, 159, 277

[GT03] Oded Goldreich and Luca Trevisan. Three theorems regarding testing graph properties. *Random Structures & Algorithms*, 2003. 13, 86, 96

[GW21] Oded Goldreich and Avi Wigderson. Non-adaptive vs adaptive queries in the dense graph testing model. In *Foundations of Computer Science (FOCS)*, 2021. 13, 86

[Hau95] David Haussler. Sphere packing numbers for subsets of the boolean n-cube with bounded vapnik-chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 1995. 126

[Hoe94] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The collected works of Wassily Hoeffding*. Springer, 1994. 24

[ILR12] Piotr Indyk, Reut Levi, and Ronitt Rubinfeld. Approximating and testing k-histogram distributions in sub-linear time. In *Symposium on Principles of Database Systems (PODS)*, 2012. 2, 8

[Jan04] Svante Janson. Large Deviations for Sums of Partly Dependent Random Variables. *Random Structures & Algorithms (RSA)*, 2004. 25

[Kin97] Terry King. *A guide to chi-squared testing*. Taylor & Francis, 1997. 32

[KKR04] Tali Kaufman, Michael Krivelevich, and Dana Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM Journal on Computing (SICOMP)*, 2004. 4, 191

[KSS18] Akash Kumar, C Seshadhri, and Andrew Stolman. Finding forbidden minors in sublinear time: A $n^{1/2+o(1)}$-query one-sided tester for minor closed properties on bounded degree graphs. In *Symposium on Foundations of Computer Science (FOCS)*, 2018. 4

[Lev21] Reut Levi. Testing triangle freeness in the general model in graphs with arboricity $o(\sqrt{n})$. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021. 4

[Lin94] Chih-Long Lin. Hardness of Approximating Graph Transformation Problem. In *International Symposium on Algorithms and Computation (ISAAC)*, 1994. 185, 186

[LM20] Reut Levi and Moti Medina. Distributed testing of graph isomorphism in the CONGEST model. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2020. 187

[Mac03] David J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003. 32

[Mat99] J. Matoušek. *Geometric Discrepancy: An Illustrated Guide*. Algorithms and Combinatorics. Springer, Berlin, New York, 1999. 89

[Mat02] Jirí Matousek. *Lectures on Discrete Geometry*, volume 212 of *Graduate texts in mathematics*. Springer, 2002. 89

[MPC20] Kuldeep S. Meel, Yash Pote, and Sourav Chakraborty. On testing of samplers. In *Neural Information Processing Systems (NeurIPS)*, 2020. 80

[MR09] Sharon Marko and Dana Ron. Approximating the distance to properties in bounded-degree and general sparse graphs. *ACM Transactions on Algorithms (TALG)*, 2009. 4

[MS08] Claire Mathieu and Warren Schudy. Yet Another Algorithm for Dense Max Cut: Go Greedy. In *Symposium on Discrete Algorithms (SODA)*, 2008. 254

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, 2017. 254

[NO08] Huy N Nguyen and Krzysztof Onak. Constant-time approximation algorithms via local improvements. In *Symposium on Foundations of Computer Science (FOCS)*, 2008. 4

[ORRR12] Krzysztof Onak, Dana Ron, Michal Rosen, and Ronitt Rubinfeld. A near-optimal sublinear-time algorithm for approximating the minimum vertex cover size. In *Symposium on Discrete Algorithms (SODA)*, 2012. 4

[OS18] Krzysztof Onak and Xiaorui Sun. The Query Complexity of Graph Isomorphism: Bypassing Distribution Testing Lower Bounds. In *Symposium on Theory of Computing (STOC)*, 2018. 187, 279

[PA95] J. Pach and P. K. Agarwal. *Combinatorial Geometry*. John Wiley & Sons, New York, NY, 1995. 89

[Pan08] Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Transactions on Information Theory*, 2008. 2, 3, 8, 29, 32, 81, 188

[PM21] Yash Pote and Kuldeep S. Meel. Testing probabilistic circuits. In *Neural Information Processing Systems (NeurIPS)*, 2021. 80

[PR02] Michal Parnas and Dana Ron. Testing the diameter of graphs. *Random Structures & Algorithms (RSA)*, 2002. 4

[PR07] Michal Parnas and Dana Ron. Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms. *Theoretical Computer Science (TCS)*, 2007. 4

[PRR06] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences (JCSS)*, 2006. 33, 186

[Ron08] Dana Ron. Property testing: A learning theory perspective. *Found. Trends Mach. Learn.*, 2008. 4, 80

[Ron09] Dana Ron. Algorithmic and analysis techniques in property testing. *Found. Trends Theor. Comput. Sci.*, 2009. 4, 80

[RRSS09] Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM Journal on Computing (SICOMP)*, 2009. 205

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust Characterizations of Polynomials with Applications to Program Testing. *SIAM Journal on Computing (SICOMP)*, 1996. 2

[RS11] Ronitt Rubinfeld and Asaf Shapira. Sublinear time algorithms. *SIAM Journal on Discrete Mathematics (SIDMA)*, 2011. 4, 80

[RS15] Dana Ron and Rocco A Servedio. Exponentially improved algorithms and lower bounds for testing signed majorities. *Algorithmica*, 2015. 87

[Rub12] Ronitt Rubinfeld. Taming big probability distributions. *XRDS: Crossroads, The ACM Magazine for Students*, 2012. 4

[Ser10] Rocco A Servedio. Testing by implicit learning: a brief survey. *Property Testing*, 2010. 37, 83

[Soh12] Christian Sohler. Almost Optimal Canonical Property Testers for Satisfiability. In *Foundations of Computer Science (FOCS)*, 2012. 3, 4, 190

[SP18] Shashank Singh and Barnabás Póczos. Minimax distribution estimation in wasserstein distance. *arXiv preprint arXiv:1802.08855*, 2018. 196

[Sun16] Xiaorui Sun. *On the Isomorphism Testing of Graphs*. PhD thesis, Columbia University, 2016. 185

[Val11] Paul Valiant. Testing Symmetric Properties of Distributions. *SIAM Journal on Computing (SICOMP)*, 2011. 29, 33, 38, 62, 63, 81

[VC15] Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*. Springer, 2015. 89

[VN53] John Von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. *Contributions to the Theory of Games*, 1953. 132

[VV10] Gregory Valiant and Paul Valiant. A clt and tight lower bounds for estimating entropy. In *Electron. Colloquium Comput. Complex.*, 2010. 3, 8, 97, 159

[VV11] Gregory Valiant and Paul Valiant. The power of linear estimators. In *Foundations of Computer Science (FOCS)*, 2011. 3, 8, 29, 32, 33, 39, 73, 202

[VV17a] Gregory Valiant and Paul Valiant. An automatic inequality prover and instance optimal identity testing. *SIAM Journal on Computing (SICOMP)*, 2017. 29, 32, 188

[VV17b] Gregory Valiant and Paul Valiant. Estimating the unseen: Improved estimators for entropy and other properties. *Journal of the ACM (JACM)*, 2017. 30, 81

[Yao77] Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *Foundations of Computer Science (FOCS)*, 1977. 91

[YYI09] Yuichi Yoshida, Masaki Yamamoto, and Hiro Ito. An improved constant-time approximation algorithm for maximum matchings. In *Symposium on Theory of computing (STOC)*, 2009. 4

[Zha21] Huanyu Zhang. Statistical inference in the differential privacy model. *CoRR*, abs/2108.05000, 2021. 29

# List of Publications (Based on content of the thesis)

1. **Interplay between Graph Isomorphism and Earth Mover's Distance in the Query and Communication Worlds**

   Joint work with Sourav Chakraborty, Arijit Ghosh & Gopinath Mishra.

   In the proceedings of the $25^{th}$ International Conference on Randomization and Computation (RANDOM), 2021, Volume 207, 34:1-34:23, doi: 10.4230/LIPIcs.APPROX/RANDOM.2021.34. Presented in Highlights of Algorithms (HALG), 2022.

   Submitted to the journal ACM Transactions on Computation Theory (TOCT).

2. **Tolerant Bipartiteness Testing in Dense Graphs**

   Joint work with Arijit Ghosh, Gopinath Mishra & Rahul Raychaudhury.

   In the proceedings of the $49^{th}$ International Colloquium on Automata, Languages and Programming (ICALP), 2022, Volume 229, 69:1-69:19, doi: 10.4230/LIPIcs.ICALP.2022.69. Presented in Highlights of Algorithms (HALG), 2023.

   Submitted to the journal Combinatorics, Probability and Computing (CPC).

3. **Exploring the Gap between Tolerant and Non-tolerant Distribution Testing**

   Joint work with Sourav Chakraborty, Eldar Fischer, Arijit Ghosh & Gopinath Mishra.

   In the proceedings of the $26^{th}$ International Conference on Randomization and Computation (RANDOM), Volume 245, 27:1-27:23, 2022, doi: 10.4230/LIPIcs.APPROX/RANDOM.2022.27. Presented in Highlights of Algorithms (HALG), 2023.

   Submitted to the journal IEEE Transactions on Information Theory.

4. **Testing of Index-Invariant Properties in the Huge Object Model**

   Joint work with Sourav Chakraborty, Eldar Fischer, Arijit Ghosh & Gopinath Mishra.

In the proceedings of the $36^{th}$ Conference on Learning Theory (COLT) 2023, Volume 195, pages 3065–3136, url: https://proceedings.mlr.press/v195/chakraborty23a.html.

Featured in Oded Goldreich's Choices (https://www.wisdom.weizmann.ac.il/õded/MC/335.html).

# List of publications by the author (other than thesis)

1. **Testing of Horn Samplers**

   Joint work with Ansuman Banerjee, Shayak Chakraborty, Sourav Chakraborty, Kuldeep S. Meel & Uddalok Sarkar.

   In the proceedings of the $26^{th}$ International Conference on Artificial Intelligence and Statistics (AISTATS), 2023.

2. **A (simple) classical algorithm for estimating Betti numbers**

   Joint work with Simon Apers, Sander Gribling & Dániel Szabó.

   Quantum Computing Theory in Practice (QCTiP), 2023.

   Submitted to the journal Quantum.

3. **Sampling Triangles Almost Uniformly Over Data Streams**

   Joint work with Arijit Bishnu, Arijit Ghosh & Gopinath Mishra.

   Submitted to the conference Symposium on Simplicity in Algorithms (SOSA), 2024.