Sampling Lower Bounds via Information Theory*

Ziv Bar-Yossef†
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120.
Email: ziv@almaden.ibm.com

April 29, 2003

Abstract

We present a novel technique, based on the Jensen-Shannon divergence from information theory, to prove lower bounds on the query complexity of sampling algorithms that approximate functions over arbitrary domain and range. Unlike previous methods, our technique does not use a reduction from a binary decision problem, but rather from a multi-way decision problem. As a result, it gives stronger bounds for functions that possess a large set of inputs, each two of which exhibit a gap in the function value.

We demonstrate the technique with new query complexity lower bounds for three fundamental problems: (1) the "election problem", for which we obtain a quadratic improvement over previous bounds, (2) low rank matrix approximation, for which we prove the first lower bounds, showing that the algorithms given for this problem are almost optimal, and (3) matrix reconstruction.

In addition, we introduce a new method for proving lower bounds on the *expected* query complexity of functions, using the Kullback-Leibler divergence. We demonstrate its use by a simple query complexity lower bound for the mean.

*A preliminary version of this paper is to appear in the 35th ACM Symposium on Theory of Computing (STOC).

†Part of this work was done while the author was a student at UC Berkeley, and a visitor at IBM. Supported by NSF ITR grant CCR-0121555.

1 Introduction

The increased volume of data generated automatically coupled with the dramatic improvements in storage capability have resulted in the emergence of numerous massive data sets, like the web, Internet traffic, stock market transactions, etc. The typically restricted access to these data sets together with their huge size necessitated the invention of new models of computation. Sampling algorithms—algorithms that query only a few (possibly random) locations of the input data—are the most traditional and well suited for this task.

In this paper we study lower bounds on the *query complexity* of sampling algorithms. Specifically, we develop new techniques for proving lower bounds on the number of queries a sampling algorithm is required to perform in order to approximate a given function accurately with low probability of error. Our lower bounds are applicable to general functions over arbitrary domain and range (even infinite or non-metric spaces) and to all types of approximation. The bounds are stated in terms of simple to calculate properties of functions, which are based on two information-theoretic measures of distance between distributions: the *Jensen-Shannon divergence* [Lin91] and the *Kullback-Leibler divergence* [KL51]. We demonstrate the strength of our techniques by deriving new bounds for several natural and practical problems, including the "election problem", low rank matrix approximation, matrix reconstruction, and the mean.

Motivation The standard methodology for proving lower bounds for approximation problems is via a reduction from a (binary) *decision promise* problem. A decision promise problem for a function f is specified by a pair of disjoint subsets of the domain A and B that exhibit a "gap" in the function's value; that is, for all $a \in A$ and $b \in B$, $f(a)$ is "far" from $f(b)$. The decision problem is then: given an input $\mathbf{x}$ which is promised to be in $A \cup B$, decide whether $\mathbf{x} \in A$ or $\mathbf{x} \in B$. Clearly, any algorithm that accurately approximates f can be used to solve the promise problem, and thus a lower bound for the promise problem implies a hardness of approximation result. This methodology has been extensively used in proving lower bounds for sampling algorithms. Usually, one comes up with two inputs or two distributions over inputs that exhibit a gap, and shows that any sampling algorithm that uses a small number of queries cannot distinguish between the two. The "hardness" of such functions arises from the existence of pairs of inputs that on the one hand exhibit a large gap in the function value, but on the other hand are very "similar" and therefore are not distinguishable in a small number of queries.

There are some functions, however, (see examples below) for which *every* two inputs that exhibit a gap are easily distinguishable. For such functions the paradigm mentioned above fails gloriously. The hardness of this type of functions emanates simply from the fact there are *many* inputs, each two of which exhibit a gap in the function's value. (This can happen, of course, only for functions that have a large range). In this paper we develop an information-theoretic lower bound technique, which is able to capture the hardnesses that emanate from both the similarities among "gapped inputs" and the abundance of such inputs.

Summary of contributions Our main result is a novel technique for proving lower bounds on the query complexity of sampling algorithms using the Jensen-Shannon divergence. The technique is applicable to functions over arbitrary domain and range, and forms a "template" for easily deriving lower bounds for specific functions.

Consider a function $f : \mathcal{X}^n \rightarrow \mathcal{Y}$ that possesses ℓ inputs $\mathbf{x}_1, \ldots, \mathbf{x}_\ell \in \mathcal{X}^n$, each two of which exhibit a gap in the value of f (i.e., $f(\mathbf{x}_i)$ is "far" from $f(\mathbf{x}_j)$, for all $i \neq j$). A simple information-theoretic argument can show than any sampling algorithm approximating f would require $\Omega(\log \ell / \log |\mathcal{X}|)$ queries. This bound is useless when $\mathcal{X}$ is large (e.g., $\mathcal{X} = \mathbb{R}$). Moreover, it does not address the hardness arising from similarities among $\mathbf{x}_1, \ldots, \mathbf{x}_\ell$. We prove that any sampling algorithm whose queries on $\mathbf{x}_1, \ldots, \mathbf{x}_\ell$ are i.i.d. (independent and identically distributed) requires $\Omega(\log \ell / JS(\nu_{\mathbf{x}_1}, \ldots, \nu_{\mathbf{x}_\ell}))$ queries to approximate f. Here, $\nu_{\mathbf{x}_j}$ is the joint distribution of queries and their answers when the algorithm runs on input $\mathbf{x}_j$ and $JS(\cdot, \ldots, \cdot)$ is the Jensen-Shannon divergence among these distributions. The $\log \ell$ factor in the lower bound captures the abundance of gapped inputs, and the Jensen-Shannon divergence factor captures the similarities among them.

Lower bounds for sampling algorithms whose queries are i.i.d. are interesting for two reasons. First, in practice many sampling algorithms use i.i.d. (in fact, uniform) queries. Second, and more importantly, for wide classes of functions, sampling algorithm with i.i.d. queries can simulate general, adaptive, sampling with almost no loss of efficiency. For such functions our methods give general query complexity lower bounds. We extend previous known simulations [BKS01, GT01] to more general classes of functions, including g-symmetric functions and g-row- and column-symmetric functions (see definitions in Section 4).

Using our technique, we prove new query complexity lower bounds for three fundamental problems. In the election problem the input is a sequence of n votes to m parties. The goal is to estimate the distribution of votes among the parties up to an additive factor of ϵ w.r.t. to statistical (i.e., L_1) distance. This problem, which can be more generally viewed as estimating the distribution of "types" in a given population, is important in database systems for devising good query optimization plans. Batu *et al.* [BFR+00] gave an $\Omega(m)$ lower bound for the problem, while [BKS01] showed an $O(m/\epsilon^2)$ upper bound and an $\Omega(1/\epsilon^2)$ lower bound. We prove an optimal lower bound of $\Omega(m/\epsilon^2)$, yielding up to a quadratic improvement over the previous bounds.

In the low rank matrix approximation problem, LRM_k, the input is an $m \times n$ real matrix $\mathbf{A}$, and the goal is to output a rank k matrix $\mathbf{B}$, such that $||\mathbf{A} - \mathbf{B}||_F < ||\mathbf{A} - \mathbf{A}_k||_F + \epsilon\, ||\mathbf{A}||_F$. Here, $\mathbf{A}_k$ is the best rank k approximation to $\mathbf{A}$ and $||\cdot||_F$ is the Forbenius norm (L_2 norm of the matrix when viewed as a vector). This problem is central in information retrieval applications, such as web search, collaborative filtering, and Latent Semantic Indexing (cf. [FKV98, AFK+01]). $\mathbf{A}_k$ can be computed exactly using Singular Value Decomposition (SVD) [GV96], however this requires querying the whole input matrix. Frieze, Kannan, and Vempala [FKV98] and Drineas *et al.* [DFK+99] showed sampling algorithms for the problem. (Other algorithms were given also in [AFK+01, AM01, DKR02]). Both algorithms are given some "advice" about their input in the form of distributions which are close to the row and the column weight distributions of the input matrix. The algorithm of [FKV98] chooses s rows and s columns independently according to the advice distribution and queries the resulting $s \times s$ submatrix ($s = O(k^4/\epsilon^3)$). The algorithm of [DFK+99] chooses $O(k/\epsilon^2)$ rows according to the advice distribution and queries all their entries. We prove that: (1) if no advice is given, then $\Omega(m+n)$ queries are needed for the problem; and (2) the above mentioned algorithms are optimal up to polynomial factors with respect to the advice they use.

In the matrix reconstruction problem, the input is an $m \times n$ real matrix $\mathbf{A}$, and the goal is to output any $m \times n$ matrix $\mathbf{B}$ so that $\mathbf{B}$ is close to $\mathbf{A}$. There are two variants of the problem: in the Forbenius variant, MR_F, we need $||\mathbf{A}-\mathbf{B}||_F < \epsilon\,||\mathbf{A}||_F$ and in the L_2 variant, MR_2, we need $||\mathbf{A}-\mathbf{B}||_2 < \epsilon\,||\mathbf{A}||_F$. (Recall that for a matrix $\mathbf{M}$, $||\mathbf{M}||_2 = \max_{||x||_2=1} ||\mathbf{M}x||_2$). The problem is important to collaborative filtering and recommendation systems [DKR02]. Drineas and Kannan [DK02] show an algorithm for MR_2, which uses advice (the same kind of advice as in [FKV98, DFK+99]). Their algorithm selects $O(1/\epsilon^2)$ rows and columns independently according to the advice distribution and queries all their entries. The authors mention that MR_F requires $\Omega(mn)$ queries, and prove that any algorithm solving MR_2 has to output at least $\Omega((m+n)\log(1/\epsilon))$ bits. It is not clear whether this lower bound implies a sampling lower bound, especially when the sampling algorithm is given advice. We prove that indeed MR_F requires $\Omega(mn)$ queries. We then show that the query complexity of MR_2 is $\Omega(m+n)$, even for sampling algorithms that use the kind of advice used by [DK02]. This confirms that their algorithm is optimal.

A secondary result of this paper is a new technique for proving tight lower bounds on the *expected* query complexity of symmetric functions. The Jensen-Shannon technique and the techniques discussed in [BKS01] are tight w.r.t. the worst-case query complexity, but give poor bounds (in terms of the error probability) when applied to the expected query complexity. Our new technique uses the KL divergence [KL51] between distributions. We demonstrate its use by an elementary proof for an optimal lower bound on the expected query complexity of the mean (which was originally proved by Radhakrishnan and Ta-Shma [RT00]).

Methodology Our lower bounds are based on a reduction from statistical classification to sampling algorithms. We prove that any sampling algorithm approximating a function f derives a classifier for the distributions $\nu_{\mathbf{x}_1}, \ldots, \nu_{\mathbf{x}_\ell}$, where $\mathbf{x}_1, \ldots, \mathbf{x}_\ell$ is the set of "gapped inputs". The classifier uses exactly the same number of samples as the algorithm, and therefore a lower bound on its sample complexity yields a lower bound on the query complexity of the algorithm.

The classification lower bound is proved in two steps. First, we use Fano's inequality from information theory to derive a lower bound of $1 - JS(\nu^q_{\mathbf{x}_1}, \ldots, \nu^q_{\mathbf{x}_\ell})/\log \ell$ on the misclassification probability of error (q is the number of samples used by the classifier). (As a side note, to the best of our knowledge, this lower bound gives an exponential improvement of the dependence on the number of classes ℓ over the best previously known bound due to Lin [Lin91].) We then prove a decomposition property of the Jensen-Shannon divergence: $JS(\nu^q_{\mathbf{x}_1}, \ldots, \nu^q_{\mathbf{x}_\ell}) \le q \cdot JS(\nu_{\mathbf{x}_1}, \ldots, \nu_{\mathbf{x}_\ell})$, which allows us to derive a lower bound on q.

The lower bound via the KL divergence is a direct application of a result from statistical sequential analysis, called *the optimality of the sequential probability ratio test* (cf. [Sie85]).

Related work The work most closely related to ours is a previous paper with Kumar and Sivakumar [BKS01], which gave query complexity lower bounds in terms of the *Hellinger distance*. These bounds give tight results for functions that have few "gapped inputs" and weak bounds for functions that have many gapped inputs. We note that our Jensen-Shannon technique is as good as the Hellinger technique even for functions with few gapped inputs, since the Hellinger distance and the Jensen-Shannon divergence are always at most a constant away of each other [T.S. Jayram, private communication, October 2002]. The Hellinger technique, however, has a tighter dependence on the error probability.

Many ad-hoc sampling lower bounds for function approximations appear in the literature (e.g., [CEG95, DKLR95, SV99, RT00, CCMN00]). All of them are tailored to specific problems, and do not present a general technique. Sampling lower bounds in slightly different settings are given in statistics and learning (e.g., via VC dimension [Vap98, KV94] and the Cramér-Rao inequality [Van68]). Previous lower bounds on the misclassification error (such as Stein's Lemma and Chernoff Bound, and also [Kai67, Tou74, Ziv88, Gut89, Lin91]), are either applicable only to two-class classification, are stated in an asymptotic form, or are weaker than the bound we present in this paper.

The rest of the paper is organized as follows. In Section 2 we review the tools from information theory, statistics, and combinatorics we use. In Section 3 we describe the sampling model in detail. In Section 4 we present the simulations of general sampling by i.i.d. sampling for symmetric functions. In Section 5 we describe the reduction from statistical classification to sampling. In Section 6 we prove the classification lower bound via the Jensen-Shannon divergence. In Section 7 we discuss the applications. Finally, in Section 8 we present the lower bound technique for the expected query complexity.

2 Preliminaries

2.1 Notations and conventions

We denote sets by capital Calligraphic letters (e.g., $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$), elements of sets by lower case letters (e.g., x, y, z), random variables by capital letters (e.g., X, Y, Z), and distributions by lower case Greek letters (e.g., μ, ν, λ). Vectors will be denoted by boldface (e.g., $\mathbf{x}, \mathbf{i}, \mathbf{a}$). $[n]$ stands for $\{1, \ldots, n\}$. All logarithms are to the base of 2. ln is the natural logarithm.

Unless stated otherwise, we deal only with finite discrete probability spaces. $X \sim \mu$ means that μ is the distribution of the random variable X. $\mu(x)$ is the probability of x under μ.

In order to formulate our lower bound technique in a way that is independent of the particular notion of approximation used, we use the following abstraction of approximation [BKS01]: for a function $f : \mathcal{X}^n \to \mathcal{Y}$ and an error parameter $\epsilon > 0$, an approximation notion is a family of subsets $\{A_{f,\epsilon}(\mathbf{x})\}_{\mathbf{x} \in \mathcal{X}^n}$ of $\mathcal{Y}$. $A_{f,\epsilon}(\mathbf{x})$ is the called the "ϵ approximation set" of $\mathbf{x}$ and includes $f(\mathbf{x})$ and all the values that are considered an ϵ-approximation of $f(\mathbf{x})$. For example, in additive approximation, $A_{f,\epsilon}(\mathbf{x}) = (f(\mathbf{x}) - \epsilon, f(\mathbf{x}) + \epsilon)$, and in relative approximation $A_{f,\epsilon}(\mathbf{x}) = ((1-\epsilon)f(\mathbf{x}), f(\mathbf{x})(1+\epsilon))$. An algorithm Δ is said to (ϵ, δ)-approximate f, if for all inputs $\mathbf{x}$, $\Pr(\Delta(\mathbf{x}) \in A_{f,\epsilon}(\mathbf{x})) \geq 1 - \delta$, where the probability is over the internal coin tosses of Δ.

2.2 Statistical distance measures

We use the following measures of distance between distributions:

Definition 2.1. The KL divergence [KL51] and the Jensen-Shannon divergence [Lin91] between two distributions μ_1, μ_2 on $\mathcal{X}$ are:

$$
\begin{aligned}
D_{KL}(\mu_1 \parallel \mu_2) &\stackrel{\text{def}}{=} \textstyle\sum_{x \in \mathcal{X}} \mu_1(x) \log \frac{\mu_1(x)}{\mu_2(x)}. \\
JS(\mu_1, \mu_2) &\stackrel{\text{def}}{=} \tfrac{1}{2}\left(D_{KL}(\mu_1 \parallel \tfrac{\mu_1+\mu_2}{2}) + D_{KL}(\mu_2 \parallel \tfrac{\mu_1+\mu_2}{2})\right).
\end{aligned}
$$

The Jensen-Shannon divergence can be generalized to measure the mutual distance among more than two distributions, and with non-uniform weights:

Definition 2.2. For a distribution λ on $[n]$, and for distributions $\mu_1, \ldots, \mu_n$ on $\mathcal{X}$, let $\mu_\lambda \stackrel{\text{def}}{=} \sum_{i=1}^n \lambda(i)\mu_i$ be the λ-weighted average distribution. The λ-generalized Jensen-Shannon divergence among $\mu_1, \ldots, \mu_n$ is:

$$
JS_\lambda(\mu_1, \ldots, \mu_n) \stackrel{\text{def}}{=} \sum_{i=1}^n \lambda(i) \cdot D_{KL}(\mu_i \parallel \mu_\lambda).
$$

2.3 Information theory

In the following $X \sim \mu_X$, $Y \sim \mu_Y$, $Z \sim \mu_Z$ are random variables on domains $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$, respectively. The entropy of X (or, equivalently, of μ_X) is $H(X) \stackrel{\text{def}}{=} \sum_{x \in \mathcal{X}} \mu_X(x) \log \frac{1}{\mu_X(x)}$. The entropy of a Bernoulli random variable with probability of success p is denoted $H_2(p)$. The joint entropy of X and Y is the entropy of the joint distribution (μ_X, μ_Y). The conditional entropy of X given an event A, denoted $H(X \mid A)$, is the entropy of the conditional distribution of μ_X given A. The conditional entropy of X given Y is $H(X \mid Y) \stackrel{\text{def}}{=} \sum_{y \in \mathcal{Y}} \mu_Y(y) H(X \mid Y = y)$. The mutual information between X and Y is $I(X;\, Y) \stackrel{\text{def}}{=} H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$. The conditional mutual information between X and Y given Z is $I(X;\, Y \mid Z) = H(X \mid Z) - H(X \mid Y, Z) = H(Y \mid Z) - H(Y \mid X, Z)$. Some basic properties of entropy and mutual information we are using in this paper are the following. Proofs can be found in Chapter 2 of [CT91].

Proposition 2.3. *Let X, Y, Z be random variables on domains $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$, respectively.*

1. *$H(X) \leq \log |\mathcal{X}|$. Equality iff X is uniform on $\mathcal{X}$.*

2. *Subadditivity of entropy: $H(X,Y) \leq H(X) + H(Y)$. Equality iff X,Y are independent.*

3. *Data processing inequality: For any function f, $I(X;\, f(Y)) \leq I(X;\, Y)$.*

4. *Chain rule for mutual information: $I(X;\, Y,Z) = I(X;\, Y) + I(X;\, Z \mid Y)$.*

5. *$I(X;\, Y) = 0$ iff X,Y are independent.*

6. *If X,Y are jointly independent of Z, then $I(X;\, Y \mid Z) = I(X;\, Y)$.*

The generalized Jensen-Shannon divergence has an equivalent characterization in terms of mutual information (see a proof in Appendix A.1, generalizing a previous proof [BJKS02]):

Proposition 2.4. *Let $D \sim \lambda$, $(X_1,\ldots,X_n) \sim (\mu_1,\ldots,\mu_n)$, $\pi(d,x_1,\ldots,x_n) = x_d$, and $X_D = \pi(D,X_1,\ldots,X_n)$. That is, X_D is a sample from the distribution μ_D, where D is chosen according to λ. Then, $JS_\lambda(\mu_1,\ldots,\mu_n) = I(D;\, X_D)$.*

2.4 Set designs

A family of subsets $\mathcal{F}_1,\ldots,\mathcal{F}_\ell \subseteq [m]$ is called a (d,k)-design, if: (1) for all $j \in [\ell]$, $|\mathcal{F}_j| = d$, and (2) for all $j \neq j' \in [\ell]$, $|\mathcal{F}_j \cap \mathcal{F}_{j'}| \leq k$. We will use $(\frac{m}{2}, \Omega(m))$-designs of exponential size, as given by the following proposition. The proof is a slight modification of the probabilistic method argument of [NW94], guaranteeing the sets are of size exactly $m/2$ (and not just at most $m/2$). For completeness, it appears in Appendix A.2.

Proposition 2.5. *For every $m \geq 18$, there exists a $(\frac{m}{2}, \frac{11}{24}m)$-design $\mathcal{F}_1,\ldots,\mathcal{F}_\ell \subseteq [m]$ of size $\ell = 2^{\Omega(m)}$.*

2.5 Sequential distributions

A sequential distribution on a domain $\mathcal{B}$ is a family of distributions $\mu = \{\mu_{\mathbf{b}}\}_{\mathbf{b}}$ on $\mathcal{B}$, indexed by all the finite length sequences $\mathbf{b} = (b_1,\ldots,b_t)$ of elements from $\mathcal{B}$. The family is used to construct an infinite random sequence of elements from $\mathcal{B}$ as follows: the first element is chosen according to $\mu_\emptyset$; if the first t elements chosen are $\mathbf{b} = (b_1,\ldots,b_t)$, the next element is selected according to $\mu_{\mathbf{b}}$.

A sequential distribution induces distributions on all finite length sequences of elements from $\mathcal{B}$. Given a sequential distribution μ and an integer $t > 0$, the t-wise distribution induced by μ, denoted μ_t, is the following distribution on $\mathcal{B}^t$: $\mu_t(b_1,\ldots,b_t) = \mu_\emptyset(b_1) \cdot \mu_{b_1}(b_2) \cdots \mu_{b_1,\ldots,b_{t-1}}(b_t)$.

A sequential distribution μ on $\mathcal{B}$ is called independent and identically distributed (or "i.i.d." in short), if all the distributions in the family are identical to some distribution ν on $\mathcal{B}$. We call ν the base distribution of μ. Note that for i.i.d. distributions, $\mu_t = \nu^t$ for all t.

3 The sampling model

Loosely speaking, a sampling algorithm is a randomized algorithm that has "oracle access" to its input $\mathbf{x} \in \mathcal{X}^n$. In each oracle call, the algorithm queries an index $i \in [n]$, and gets back $\mathbf{x}_i$. The queries are performed sequentially, and can be chosen *adaptively*: that is, based on answers to previous queries the algorithm can choose which index to query next.

Since we are interested in lower bounds for sampling algorithms, we: (1) consider only the number of oracle calls, or queries, the algorithm performs, and ignore other resources like time or space; (2) ignore precision issues and assume the algorithm can query even real numbers at unit cost; and (3) restrict to functions over a fixed input length n.

Formally, a sampling algorithm is a communication protocol Δ among three players: Alice, Bob, and a "referee". Bob gets the input $\mathbf{x} \in \mathcal{X}^n$; Alice and the referee get random strings $r_1 \in \mathcal{R}_1$ and $r_2 \in \mathcal{R}_2$, respectively. The protocol proceeds in rounds as follows. Alice begins each round by sending a query index $i \in [n]$ to Bob. The choice of i may depend on Alice's random string as well as on the answers to her previous queries. Bob replies with $\mathbf{x}_i$. The referee, after seeing i and $\mathbf{x}_i$, applies a "decision function" $D : ([n] \times \mathcal{X})^* \times \mathcal{R}_2 \to \mathcal{Y} \cup \{\textsc{cont}\}$, which determines the decision of the referee: continue to monitor the communication between Alice and Bob or stop and output a value. The decision is based on the query indices sent so far, on their respective answers, and on the referee's random string. Abusing notation we will write $D(i_1, \ldots, i_t, a_1, \ldots, a_t, r_2)$ instead of $D(i_1, a_1, \ldots, i_t, a_t, r_2)$.

The round at which the referee decides to stop and output a value is fully determined by the input $\mathbf{x}$ and by the random strings r_1 and r_2. We call this round the stopping time of the protocol on $(\mathbf{x}, r_1, r_2)$, and denote it by $T(\mathbf{x}, r_1, r_2)$. The worst-case query cost of the protocol is $\max_{\mathbf{x}, r_1, r_2} T(\mathbf{x}, r_1, r_2)$. The expected query cost of the protocol is $\max_{\mathbf{x}} \mathrm{E}\,[T(\mathbf{x}, R_1, R_2)]$, where R_1 and R_2 are the random variables corresponding to the random strings of Alice and the referee.

The output of a protocol Δ on $(\mathbf{x}, r_1, r_2)$, denoted $\Delta_{\text{out}}(\mathbf{x}, r_1, r_2)$, is the value the referee outputs at the stopping time $T(\mathbf{x}, r_1, r_2)$. Δ is said to (ϵ, δ)-approximate a function $f : \mathcal{X}^n \to \mathcal{Y}$ with approximation notion $A_{f,\epsilon}$, if for all inputs $\mathbf{x}$, $\Pr(\Delta_{\text{out}}(\mathbf{x}, R_1, R_2) \in A_{f,\epsilon}(\mathbf{x})) \geq 1 - \delta$.

Transcripts The transcript of a protocol at step t is the sequence of t index-answer pairs $(i_1, a_1, \ldots, i_t, a_t)$ communicated between Alice and Bob during the first t rounds. Note that this sequence is fully determined by the input $\mathbf{x}$ and by Alice's random string r_1. We thus denote by $\Delta_t(\mathbf{x}, r_1)$ the transcript of Δ on $(\mathbf{x}, r_1)$ at step t. The subsequence of query indices $i_1, \ldots, i_t$ is called the index transcript of Δ on $(\mathbf{x}, r_1)$ at step t, and is denoted by $A_t(\mathbf{x}, r_1)$. The subsequence of query answers $a_1, \ldots, a_t$ is called the answer transcript of Δ on $(\mathbf{x}, r_1)$ at step t, and is denoted by $A_t(\mathbf{x}, r_1)$.

The index transcript at step t is also fully determined by Alice's random string r_1 and by the answers $\mathbf{a} = (a_1, \ldots, a_{t-1})$ to her first $t-1$ queries. We thus denote it by $A_t(\mathbf{a}, r_1)$.

For the analysis of sampling protocols, it would be convenient to think of Alice and Bob as engaging in an infinite conversation, in which Alice sends queries and Bob replies with answers. The referee may monitor this conversation for a while and stop when he has enough information to make a decision. We can thus talk about transcripts of the protocol at arbitrarily large steps t, even if the referee has made a decision at some step $t' < t$.

Public coins vs. private coins The most general sampling protocols possible are ones in which Alice and the referee share the same random string: $r_1 = r_2 = r$. These are called public-coin protocols. Unfortunately, our lower bound techniques are applicable directly only to private-coin protocols, in which Alice and the referee get independent random strings. Clearly, any private-coin protocol can be simulated by a public-coin one with the same query cost. A natural question is whether a converse efficient simulation exists. We show that efficient simulations exist, if either the domain or the range of the function are small. We do not know of an efficient simulation when the domain and the range are large.

Proposition 3.1. *Let Δ be a public-coin sampling protocol that (ϵ, δ)-approximates a function $f : \mathcal{X}^n \to \mathcal{Y}$ and whose worst-case query cost is q. Then,*

1. *There exists a private-coin sampling protocol Δ' that $(\epsilon, \delta + \delta')$-approximates f and whose worst-case query cost is at most $q + O((\log\log|\mathcal{X}| + \log(1/\delta'))/\log n)$.*
2. *There exists a private-coin sampling protocol Δ'' that (ϵ, δ)-approximates f and whose worst-case query cost is at most $2q + O(\log|\mathcal{Y}|/\log n)$.*

The first simulation is an adaptation of a result by Newman [New91]. The second is the following simple simulation: Alice of the private-coin protocol simulates both Alice and the referee of the public-coin protocol. She communicates to the referee when to stop and what value to output via "dummy" queries.

Index-oblivious sampling A protocol is called index-oblivious, if the referee's decision function is of the form $D : \mathcal{X}^* \times \mathcal{R}_2 \to \{0, 1\}$, meaning that the referee uses only his random string and the answer transcript (but not the index transcript) to make his decision. Public-coin protocols are always, WLOG, index-oblivious, since the referee can recover the query indices just from the answers to the queries and from his random string (which is the same as Alice's random string).

Query distributions Any fixing of the first $t-1$ query answers $\mathbf{a} = (a_1, \ldots, a_{t-1})$ induces a probability distribution over Alice's possible choices for the t-th query index. We denote this distribution by $\xi_{\mathbf{a}}$. Note that the sequence $\mathbf{a}$ is fully determined by the sequence $\mathbf{i} = (i_1, \ldots, i_{t-1})$ of indices Alice queried in the first $t-1$ rounds and by the input $\mathbf{x}$. We can thus define the distribution $\mu_{\mathbf{x},\mathbf{i}} \stackrel{\text{def}}{=} \xi_{\mathbf{x}_\mathbf{i}}$. The family of distributions $\mu_{\mathbf{x}} = \{\mu_{\mathbf{x},\mathbf{i}}\}_{\mathbf{i}}$ is called the index distribution of the protocol on $\mathbf{x}$. $\mu_{\mathbf{x}}$ is a "sequential distribution" on $[n]$ (recall the definition from Section 2.5). Two index distributions we consider in this paper are the following:

1. Uniform sampling without replacement: For $t = 0, \ldots, n$, and for any t distinct indices $i_1, \ldots, i_t \in [n]$, $\mu_{\mathbf{x},i_1,\ldots,i_t}$ is the uniform distribution on $[n] \setminus \{i_1, \ldots, i_t\}$.

2. Uniform sampling with replacement: For any $\mathbf{i}$, $\mu_{\mathbf{x},\mathbf{i}}$ is uniform on $[n]$.

Note that uniform sampling with replacement is an i.i.d. sequential distribution, while uniform sampling without replacement is not.

Any input $\mathbf{x}$ and any sequence $\mathbf{a}$ of answers to the first $t-1$ queries induces a probability distribution $\alpha_{\mathbf{x},\mathbf{a}}$ over Bob's answers in the t-th round (which is $\mathbf{x}_I$, where I is a random index chosen from $\xi_{\mathbf{a}}$). The family of distributions $\alpha_{\mathbf{x}} = \{\alpha_{\mathbf{x},\mathbf{a}}\}_{\mathbf{a}}$ is called the answer distribution of the protocol on $\mathbf{x}$. Any input $\mathbf{x}$ and any sequence $\mathbf{i} = (i_1, \ldots, i_{t-1})$ of $t-1$ indices induces a probability distribution $\psi_{\mathbf{x},\mathbf{i}}$ on the possible index-answer pairs that can be produced at the t-th round of the protocol. The family of distributions $\psi_{\mathbf{x}} = \{\psi_{\mathbf{x},\mathbf{i}}\}_{\mathbf{i}}$ is called the index-answer distribution of the protocol on $\mathbf{x}$. It is easy to verify that if the index distribution of a protocol on $\mathbf{x}$ is i.i.d., then so are the answer distribution and the index-answer distribution.

Sampling with advice A sampling algorithm with "advice" is one in which Alice is given some prior information about the input $\mathbf{x}$. This information is conveyed to Alice via her random string: each input $\mathbf{x}$ is associated with a probability distribution $\rho_{\mathbf{x}}$, from which Alice's random string is selected when Bob gets $\mathbf{x}$.

In such protocols, the distribution of the t-th query index depends not only on the answers $\mathbf{a} = (a_1, \ldots, a_{t-1})$ to the first $t-1$ queries, but also on the input $\mathbf{x}$. We thus denote this distribution by $\xi_{\mathbf{x},\mathbf{a}}$. The index distribution of the protocol on $\mathbf{x}$ is now defined as $\mu_{\mathbf{x}} = \{\mu_{\mathbf{x},\mathbf{i}}\}_{\mathbf{i}}$, where $\mu_{\mathbf{x},\mathbf{i}} \stackrel{\text{def}}{=} \xi_{\mathbf{x},\mathbf{x}_{\mathbf{i}}}$. Similar adaptations apply to the answer distribution and to the index-answer distribution.

4 Canonical forms for symmetric functions

In this section we show that, without loss of generality, any sampling algorithm computing a "symmetric" function has a certain canonical form. Namely, it is private-coin and its index distribution on any input is uniform with replacement. We extend previously known such canonical form simulations to a larger class of functions.

4.1 Symmetric functions

Given an input $\mathbf{x} \in \mathcal{X}^n$ and a permutation $\pi \in S_n$, we define the permutation of $\mathbf{x}$ according to π to be the input $\pi(\mathbf{x})$ which satisfies for all $i \in [n]$, $\pi(\mathbf{x})_i = \mathbf{x}_{\pi^{-1}(i)}$. A symmetric function is one which is invariant under permutations of its inputs: $f(\pi(\mathbf{x})) = f(\mathbf{x})$ for all $\mathbf{x}$ and π. Many natural functions like average, minimum, and median are symmetric. A generalization of this notion is the following: for some function $g : \mathcal{Y} \times S_n \to \mathcal{Y}$, f is called g-symmetric, if for all $\mathbf{x}$ and π, $g(f(\pi(\mathbf{x})), \pi) = f(\mathbf{x})$. That is, $f(\pi(\mathbf{x}))$ is not necessarily the same as $f(\mathbf{x})$, but with a simple transformation, one can use $f(\pi(\mathbf{x}))$ to get $f(\mathbf{x})$. Symmetric functions are g-symmetric with $g(y, \pi) = y$. Another interesting sub-class of g-symmetric functions are ones in which $f(\pi(\mathbf{x})) = \pi(f(\mathbf{x}))$. In this case $g(y, \pi) = \pi^{-1}(y)$. We call such functions permutation-commutative. The identity function, for example, is permutation-commutative. Next, we define the notion corresponding to g-symmetric functions, when dealing with approximations of functions:

Definition 4.1. Let $f : \mathcal{X}^n \to \mathcal{Y}$ be a function with approximation notion $A_{f,\epsilon}$. For a function $g : \mathcal{Y} \times S_n \to \mathcal{Y}$, we call f (g, ϵ)-symmetric, if for all inputs $\mathbf{x} \in \mathcal{X}^n$, for all permutations $\pi \in S_n$, and for all $y \in A_{f,\epsilon}(\pi(\mathbf{x}))$, $\quad g(y, \pi) \in A_{f,\epsilon}(\mathbf{x})$.

We show that any sampling algorithm approximating a (g, ϵ)-symmetric function can be simulated efficiently by a private-coin algorithm whose index distribution on all input is uniform with replacement. We present two variants of the simulation: an *index-oblivious* one, which works only if the original algorithm uses at most $O(\sqrt{n})$ queries and if g depends only on its first argument, and an index-aware simulation which works for almost all other algorithms. (We do not have a simulation for the uninteresting case in which the original algorithm uses more than $n/2$ queries).

Theorem 4.2. *Let $\epsilon \geq 0$, let $0 < \delta < \frac{1}{2}$, and let Δ be any sampling protocol that (ϵ, δ)-approximates a (g, ϵ)-symmetric function $f : \mathcal{X}^n \to \mathcal{Y}$. Let q be the worst-case query cost of Δ and let q_E be its expected query cost. Then, there is a private-coin sampling protocol $\hat{\Delta}$ that $(\epsilon, 2\delta)$-approximates f and whose index distribution on all inputs is uniform with replacement. Furthermore,*

Low query cost case *($q \leq \sqrt{2\delta n}$): If g depends only on its first argument, then $\hat{\Delta}$ is index-oblivious. The worst-case query cost of $\hat{\Delta}$ is at most q and its expected query cost is at most $(1 - \delta)q_E + \delta q$. If in addition $q \leq \sqrt[3]{2n \cdot q_E}$, then the expected query cost of $\hat{\Delta}$ is at most $2q_E$.*

High query cost case *($q \leq \frac{n}{2} - \sqrt{\frac{n}{2\delta}}$): The worst-case query cost of $\hat{\Delta}$ is at most $2q$ and its expected query cost is at most $2q_E$.*

The theorem follows from two simulation lemmas. Lemma 4.3 shows how to simulate any sampling algorithm approximating a (g, ϵ)-symmetric function by a sampling algorithm whose queries are uniform without replacement. Lemma 4.7 shows how to simulate uniform queries without replacement by uniform queries with replacement. Proofs of similar lemmas can be found in [BKS01, Bar02].

Lemma 4.3. *Let $\epsilon \geq 0$, let $0 < \delta < 1$, and let Δ be any sampling protocol that (ϵ, δ)-approximates a (g, ϵ)-symmetric function $f : \mathcal{X}^n \to \mathcal{Y}$. Let q be the worst-case query cost of Δ and let q_E be its expected query cost. Then, there is a private-coin sampling protocol $\hat{\Delta}$ that (ϵ, δ)-approximates f and whose index distribution on all inputs is uniform without replacement. If g depends only its first argument, then $\hat{\Delta}$ is index-oblivious. The worst-case query cost of $\hat{\Delta}$ is at most q and its expected query cost is at most q_E.*

Proof. The basic intuition of the proof is the following: given an input $\mathbf{x}$, $\hat{\Delta}$ selects a random permutation $\sigma \in S_n$ and simulates Δ on $\sigma(x)$. Every time Δ would like to query some index i of $\sigma(\mathbf{x})$, $\hat{\Delta}$ returns $\mathbf{x}_{\sigma^{-1}(i)}$. Since Δ (ϵ, δ)-approximates f, it is likely to output a value $y \in A_{f,\epsilon}(\sigma(\mathbf{x}))$. $\hat{\Delta}$ now computes $g(y, \sigma)$. Since f is (g, ϵ)-symmetric, then $y \in A_{f,\epsilon}(\sigma(\mathbf{x}))$

implies $g(y,\sigma) \in A_{f,\epsilon}(\mathbf{x})$, and therefore $\hat{\Delta}$ indeed (ϵ,δ)-approximates f. The random choice of σ makes the queries of $\hat{\Delta}$ to $\mathbf{x}$ uniform without replacement, regardless of how the queries of Δ are distributed.

The above intuition can be made into a formal argument, but unfortunately it gives only a public-coin protocol, since both Alice and the referee depend on the same randomly chosen permutation σ. In order to construct a private-coin protocol, we resort to a more involved argument, in which Alice makes uniform queries without replacement and the referee has to "figure out" the permutation to which these queries correspond.

Some notational comments are in order. First, all the operators that appear with a "hat" refer to the simulating protocol $\hat{\Delta}$, while the ones that appear without a hat refer to Δ. Second, for a permutation $\sigma \in S_n$ and for any sequence $\mathbf{i}$ of t distinct indices $i_1,\ldots,i_t \in [n]$, $\sigma(\mathbf{i})$ is the sequence $\sigma(i_1),\ldots,\sigma(i_t)$. Finally, for an input $\mathbf{x}$ and a sequence $\mathbf{i}$ of t indices $i_1,\ldots,i_t$, $\mathbf{x}_{\mathbf{i}}$ is the sequence $\mathbf{x}_{i_1},\ldots,\mathbf{x}_{i_t}$.

Without loss of generality, we assume Δ is a public-coin protocol. We can therefore assume that it is index-oblivious. That is, the referee's decision function D takes as input arguments only the public random string and the answer transcript (and not the index transcript).

Furthermore, we assume: (1) that the worst-case query cost of the protocol is at most n; and (2) that Alice never queries the same index twice within these n queries. If Δ does not satisfy these properties, we can simulate it by a new protocol $\tilde{\Delta}$ that also (ϵ,δ)-approximates f, whose query costs are at most those of Δ, and that satisfies the two properties. The simulation is simple: suppose there are some input $\mathbf{x}$, public random string r, and step t, such that Δ does not stop by the t-th step when running on $(\mathbf{x},r)$, and such that the t-th index queried, i_t, has already been queried before. Since Alice queried i_t before, she knows the answer to the query and does not have to wait for Bob to send her this answer. So Alice of the new protocol $\tilde{\Delta}$ simply "shortcuts" the second query of i_t, by answering it on her own, and immediately moving on to the next query. The referee of $\tilde{\Delta}$ simulates the referee of Δ and also makes the same shortcuts. Note that one shortcut can be followed immediately by another shortcut, and so on, resulting in a a chain of shortcuts. The chain always ends, when either a new query index is reached, or when the referee decides to stop and output a value. The referee and Alice can synchronize the chain of shortcuts, since they share the same random string.

We next define the protocol $\hat{\Delta}$. In this protocol Alice and the referee get independent private random strings. Alice gets a uniformly chosen permutation Π in S_n. The referee gets a random pair (R,L), where R has the same distribution as the public random string of the protocol Δ and L is a uniformly chosen integer in $[n!]$ (R and L are independent random variables). Bob, as usual, gets an input $\mathbf{x}$.

For $t = 1,\ldots,n$, the t-th query of Alice is $\Pi(t)$, regardless of the answers to previous queries. For $t > n$, Alice queries an arbitrary value (e.g., always 1). This already implies that the index distribution of $\hat{\Delta}$ on every input is uniform without replacement.

Before we define the decision function of the referee, we prove that the answer transcript of $\hat{\Delta}$ when Alice is given a permutation π and Bob is given an input $\mathbf{x}$ is identical to the answer transcript of Δ when Alice is given a random string r and Bob is given an input $\sigma(\mathbf{x})$. Here σ is some permutation, which depends on $\mathbf{x}, r, \pi$, and t.

Claim 4.4. *Fix any input $\mathbf{x}$ and any random string r. Let $\mathbf{i} = (i_1, \ldots, i_t)$ be any sequence of t distinct indices. Let $\mathbf{j} = (j_1, \ldots, j_t) \stackrel{\text{def}}{=} A_t(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{t-1}}, r)$ be the index transcript of Δ at step t, when the public random string is r and the answers to the first $t-1$ queries are $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{t-1}}$. (By our assumption about Δ, $\mathbf{j}$ consists of t distinct indices.) Then, for any permutation π satisfying $\pi(1, \ldots, t) = \mathbf{i}$ and for any permutation σ satisfying $\sigma(\mathbf{i}) = \mathbf{j}$, $\hat{\Delta}_t(\mathbf{x}, \pi) = (\mathbf{i}, \mathbf{x}_{\mathbf{i}})$, and $\Delta_t(\sigma(\mathbf{x}), r) = (\sigma(\mathbf{i}), \mathbf{x}_{\mathbf{i}})$.*

Proof. The statement $\hat{\Delta}_t(\mathbf{x}, \pi) = (\mathbf{i}, \mathbf{x}_{\mathbf{i}})$ directly follows from the definition of Alice in the protocol $\hat{\Delta}$. We prove the other statement by induction on t. The base case, $t = 0$, is trivial since the transcripts are all empty at this point. Assume then that the condition holds for $t - 1$, and we will show it holds for t.

Fix any $\mathbf{i}$ and any σ satisfying $\sigma(\mathbf{i}) = \mathbf{j}$, where $\mathbf{j} = (j_1, \ldots, j_t) = A_t(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{t-1}}, r)$. Let $\mathbf{i}'$ and $\mathbf{j}'$ denote, respectively, the prefixes of $\mathbf{i}$ and $\mathbf{j}$ of length $t-1$. We have $\sigma(\mathbf{i}') = \mathbf{j}'$, and thus can use the induction hypothesis on σ to derive: $\Delta_{t-1}(\sigma(\mathbf{x}), r) = (\sigma(\mathbf{i}'), \mathbf{x}_{\mathbf{i}'})$. We are thus left to prove that the t-th index queried in Δ on $(\sigma(\mathbf{x}), r)$ is $\sigma(i_t)$ and that its answer is $\mathbf{x}_{i_t}$. Note that $\sigma(i_t) = j_t$. By the definition of the sequence $\mathbf{j}$, j_t is the t-th index queried in Δ when the public random string is r and the answers to the first $t-1$ queries are $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{t-1}}$. Therefore, j_t is exactly the t-th index queried in Δ on $(\sigma(\mathbf{x}), r)$. The answer to this query is the following: $\sigma(\mathbf{x})_{j_t} = \sigma(\mathbf{x})_{\sigma(i_t)} = \mathbf{x}_{i_t}$. $\square$

For a random string r, a sequence of t query answers $\mathbf{a} = (a_1, \ldots, a_t)$ is called a final answer transcript of Δ on r, if $D(\mathbf{a}, r) \neq \text{CONT}$ but $D(\mathbf{a}', r) = \text{CONT}$ for every prefix $\mathbf{a}'$ of $\mathbf{a}$. That is, if the referee of Δ sees the sequence of query answers $a_1, \ldots, a_t$, the first step at which he decides to stop and give an output is t. We denote by $\mathcal{I}_{\mathbf{x},r}$ the set of all index transcripts $\mathbf{i}$, for which $\mathbf{x}_{\mathbf{i}}$ is a final answer transcript of Δ on r.

For an index transcript $\mathbf{i} = (i_1, \ldots, i_t) \in \mathcal{I}_{\mathbf{x},r}$, let $\mathcal{P}_{\mathbf{x},r,i}$ be the collection of all permutations π that satisfy $\pi(1, \ldots, t) = \mathbf{i}$. Let $\mathcal{S}_{\mathbf{x},r,i}$ be the collection of all permutations σ that satisfy $\sigma(\mathbf{i}) = \mathbf{j}$, where $\mathbf{j} = A_t(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{t-1}}, r)$. The following claim shows that both $\mathcal{P}_{\mathbf{x},r} \stackrel{\text{def}}{=} \{\mathcal{P}_{\mathbf{x},r,\mathbf{i}}\}_{\mathbf{i} \in \mathcal{I}_{\mathbf{x},r}}$ and $\mathcal{S}_{\mathbf{x},r} \stackrel{\text{def}}{=} \{\mathcal{S}_{\mathbf{x},r,\mathbf{i}}\}_{\mathbf{i} \in \mathcal{I}_{\mathbf{x},r}}$ are partitions of S_n with certain properties:

Claim 4.5. *Fix any input $\mathbf{x}$ and any random string r. Then, $\mathcal{P}_{\mathbf{x},r}$ and $\mathcal{S}_{\mathbf{x},r}$ are partitions of S_n with the following properties:*

1. $\forall \mathbf{i} \in \mathcal{I}_{\mathbf{x},r}, \quad |\mathcal{P}_{\mathbf{x},r,\mathbf{i}}| = |\mathcal{S}_{\mathbf{x},r,\mathbf{i}}|$.

2. $\forall \mathbf{i} \in \mathcal{I}_{\mathbf{x},r}, \ \forall \pi \in \mathcal{P}_{\mathbf{x},r,\mathbf{i}}, \ \forall \sigma \in \mathcal{S}_{\mathbf{x},r,\mathbf{i}}, \quad \hat{\Delta}_t(\mathbf{x}, \pi) = (\mathbf{i}, \mathbf{x}_{\mathbf{i}})$, *and* $\Delta_t(\sigma(\mathbf{x}), r) = (\sigma(\mathbf{i}), \mathbf{x}_{\mathbf{i}})$, *where* $t = |\mathbf{i}|$.

Proof. The first property is obvious from the definitions of $\mathcal{P}_{\mathbf{x},r,\mathbf{i}}$ and $\mathcal{S}_{\mathbf{x},r,\mathbf{i}}$, since they are both of size $(n-t)!$. The second property follows from Claim 4.4. We are left to prove that $\mathcal{P}_{\mathbf{x},r}$ and $\mathcal{S}_{\mathbf{x},r}$ are partitions of S_n.

We first prove that $\mathcal{P}_{\mathbf{x},r}$ is a partition of S_n. Consider any $\mathbf{i} \neq \mathbf{i}'$. We need to prove that $\mathcal{P}_{\mathbf{x},r,\mathbf{i}}$ and $\mathcal{P}_{\mathbf{x},r,\mathbf{i}'}$ are disjoint. Let $\mathbf{i} = (i_1, \ldots, i_t)$ and $\mathbf{i}' = (i'_1, \ldots, i'_{t'})$. Note that one final answer transcript cannot be a prefix of another final answer transcript, which implies that also $\mathbf{i}$ cannot be a prefix of $\mathbf{i}'$ and vice versa. Therefore, there exists some position $\ell \leq \min(t, t')$, such that $(i_1, \ldots, i_{\ell-1}) = (i'_1, \ldots, i'_{\ell-1})$ and $i_\ell \neq i'_\ell$. Now, for every $\pi \in \mathcal{P}_{\mathbf{x},r,\mathbf{i}}$, $\pi(\ell) = i_\ell$, while for every $\pi' \in \mathcal{P}_{\mathbf{x},r,\mathbf{i}'}$, $\pi'(\ell) = i'_\ell$. Therefore, $\mathcal{P}_{\mathbf{x},r,\mathbf{i}}$ and $\mathcal{P}_{\mathbf{x},r,\mathbf{i}'}$ are disjoint.

We next prove that $\mathcal{P}_{\mathbf{x},r}$ covers S_n. Consider any permutation $\pi \in S_n$. Let $\mathbf{i}$ be the shortest prefix of the sequence $\pi(1), \ldots, \pi(n)$, for which $\mathbf{x}_\mathbf{i}$ is a final answer transcript of Δ on r. $\mathbf{i}$ must exist, because by our assumption about Δ, it always stops within n steps. Now, by definition, $\pi \in \mathcal{P}_{\mathbf{x},r,\mathbf{i}}$. Therefore, $\mathcal{P}_{\mathbf{x},r}$ covers S_n, and thus it is a partition of S_n.

We now prove that $\mathcal{S}_{\mathbf{x},r}$ is a partition of S_n. Consider any $\mathbf{i} \neq \mathbf{i}'$, $\mathbf{i} = (i_1, \ldots, i_t)$,$\mathbf{i}' = (i'_1, \ldots, i'_{t'})$. As before, there must be an ℓ s.t. $(i_1, \ldots, i_{\ell-1}) = (i'_1, \ldots, i'_{\ell-1})$ and $i_\ell \neq i'_\ell$. Let $\mathbf{j} = (j_1, \ldots, j_t) = A_t(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{t-1}}, r)$ and let $\mathbf{j}' = (j'_1, \ldots, j'_{t'}) = A_t(\mathbf{x}_{i'_1}, \ldots, \mathbf{x}_{i'_{t'-1}}, r)$. By definition, for any $\sigma \in \mathcal{S}_{\mathbf{x},r,\mathbf{i}}$, $\sigma(\mathbf{i}) = \mathbf{j}$, and for any $\sigma' \in \mathcal{S}_{\mathbf{x},r,\mathbf{i}'}$, $\sigma'(\mathbf{i}') = \mathbf{j}'$.

Since the first ℓ indices queried by Alice in Δ depend only on the public random string r and on the answers to the first $\ell - 1$ queries, and since $(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{\ell-1}}) = (\mathbf{x}_{i'_1}, \ldots, \mathbf{x}_{i'_{\ell-1}})$, then $(j_1, \ldots, j_\ell) = (j'_1, \ldots, j'_\ell)$. It follows that for all $\sigma \in \mathcal{S}_{\mathbf{x},r,\mathbf{i}}$ and for all $\sigma' \in \mathcal{S}_{\mathbf{x},r,\mathbf{i}'}$, $\sigma(i_\ell) = j_\ell = j'_\ell = \sigma'(i'_\ell)$. But now, since $i_\ell \neq \imath'_\ell$, then $\sigma \neq \sigma'$ for all such σ, σ'. In other words, $\mathcal{S}_{\mathbf{x},r,\mathbf{i}}$ and $\mathcal{S}_{\mathbf{x},r,\mathbf{i}'}$ are disjoint.

We are left to prove that $\mathcal{S}_{\mathbf{x},r}$ covers S_n. Note that $|\mathcal{S}_{\mathbf{x},r}| = |\mathcal{P}_{\mathbf{x},r}|$ and for all $\mathbf{i}$, $|\mathcal{S}_{\mathbf{x},r,\mathbf{i}}| = |\mathcal{P}_{\mathbf{x},r,\mathbf{i}}|$. Therefore, the size of the union of the sets in $\mathcal{S}_{\mathbf{x},r}$ is identical to the size of the union of the sets in $\mathcal{P}_{\mathbf{x},r}$, which we already know to be $n!$. This implies that $\mathcal{S}_{\mathbf{x},r}$ covers S_n. $\square$

We are now ready to define the decision function of the referee. Fix some input $\mathbf{x}$ and some random string r. Transcripts of $\hat{\Delta}$ when $\mathbf{x}$ is the input are of the form $(\mathbf{i}, \mathbf{x}_\mathbf{i})$, where $\mathbf{i}$ is a sequence of t distinct indices. Given such a transcript, if $D(\mathbf{x}_\mathbf{i}, r) = \textsc{cont}$, then we define $\hat{D}(\mathbf{i}, \mathbf{x}_\mathbf{i}, (r, \ell)) = \textsc{cont}$, for all values of ℓ. That is, the referee of $\hat{\Delta}$ does not stop as long as the referee of Δ would have continued on the given transcript. If $D(\mathbf{x}_\mathbf{i}, r) = y$ for some $y \in \mathcal{Y}$, it means that $\mathbf{x}_\mathbf{i}$ is a final answer transcript of Δ on r, and thus $\mathbf{i} \in \mathcal{I}_{\mathbf{x},r}$. We then define $\hat{D}(\mathbf{i}, \mathbf{x}_\mathbf{i}, (r, \ell)) = g(y, \sigma)$, where σ is the ℓ-th permutation (under some arbitrary order on S_n) in $\mathcal{S}_{\mathbf{x},r,\mathbf{i}}$.

One subtle point that needs to be addressed regards the ability of the referee to know the set of permutations $\mathcal{S}_{\mathbf{x},r,\mathbf{i}}$, even though he does not know $\mathbf{x}$. $\mathcal{S}_{\mathbf{x},r,\mathbf{i}}$ consists of all the permutations σ that satisfy the condition $\sigma(\mathbf{i}) = \mathbf{j}$, where $\mathbf{j} = (j_1, \ldots, j_t) = A_t(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{t-1}}, r)$. The referee knows both the string r and the answers $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{t-1}}$, and can therefore reconstruct the sequence $\mathbf{j}$. Since he knows also the sequence $\mathbf{i}$, he can reconstruct $\mathcal{S}_{\mathbf{x},r,\mathbf{i}}$.

Note that the index transcript $\mathbf{i}$ is needed in the decision function only in order to select the permutation σ used in the output step. If the function g does not depend on its second argument (that is, $g(y, \sigma) = g(y)$), then the referee of $\hat{\Delta}$ is in fact index-oblivious.

We next prove that the stopping time of $\hat{\Delta}$ on $(\mathbf{x}, \pi, (r, \ell))$ is identical to the stopping time of Δ on $(\sigma(\mathbf{x}), r)$, where σ is some permutation depending on $\mathbf{x}, r, \pi$, and ℓ. Moreover, if the output of Δ on $(\sigma(\mathbf{x}), r)$ is y, then the output of $\hat{\Delta}$ on $(\mathbf{x}, \pi, (r, \ell))$ is $g(y, \sigma)$.

Claim 4.6. *Fix any input* $\mathbf{x}$ *and any random string* r*. Then,*

1. $\forall \mathbf{i} \in \mathcal{I}_{\mathbf{x},r},\ \forall \pi \in \mathcal{P}_{\mathbf{x},r,\mathbf{i}},\ \forall \sigma \in \mathcal{S}_{\mathbf{x},r,\mathbf{i}},\ \forall \ell,\quad \hat{T}(\mathbf{x}, \pi, (r, \ell)) = T(\sigma(\mathbf{x}), r)$.

2. $\forall \mathbf{i} \in \mathcal{I}_{\mathbf{x},r},\ \forall \pi \in \mathcal{P}_{\mathbf{x},r,\mathbf{i}},\ \forall \ell,\quad \hat{\Delta}_{out}(\mathbf{x}, \pi, (r, \ell)) = g(\Delta_{out}(\sigma(\mathbf{x}), r), \sigma)$*, where* σ *is the* ℓ*-th permutation in* $\mathcal{S}_{\mathbf{x},r,\mathbf{i}}$*.*

Proof. Fix any $\mathbf{i} \in \mathcal{I}_{\mathbf{x},r}$, any $\pi \in \mathcal{P}_{\mathbf{x},r,\mathbf{i}}$, and any $\sigma \in \mathcal{S}_{\mathbf{x},r,\mathbf{i}}$. Let $t = |\mathbf{i}|$. Using Claim 4.5, the first t answers given both in Δ, when running on $(\sigma(\mathbf{x}), r)$, and in $\hat{\Delta}$, when running on $(\mathbf{x}, \pi)$, are $\mathbf{x_i} = \mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_t}$. Since $\mathbf{x_i}$ is a final answer transcript of Δ on r, Δ stops exactly at step t. Now, since $\hat{\Delta}$ sees the same answer transcript as Δ, and since it simulates Δ exactly, it also stops at step t. We conclude that $\hat{T}(\mathbf{x}, \pi, (r, \ell)) = T(\sigma(\mathbf{x}), r) = t$.

As for the second part, given any $\pi \in \mathcal{P}_{\mathbf{x},r,\mathbf{i}}$, $\hat{\Delta}$ stops on $(\mathbf{x}, \pi, (r, \ell))$ at step t and outputs $g(y, \sigma)$, where y is the output of Δ when given a public random string r and after seeing the answer transcript $\mathbf{x_i}$. $y = \Delta_{\text{out}}(\sigma(\mathbf{x}), r)$ for any $\sigma \in \mathcal{S}_{\mathbf{x},r,\mathbf{i}}$, by Claim 4.5. ☐

We now use this claim to prove that $\hat{\Delta}$ (ϵ, δ)-approximates f using q queries in worst-case and q_E queries on average. Since the random variables Π, R, and L are mutually independent, we can write the success probability of $\hat{\Delta}$ on an input $\mathbf{x}$ as follows:

$$\begin{aligned}
&\Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \Pi, (R, L)) \in A_{f,\epsilon}(\mathbf{x})) = \\
&= \sum_{r \in \mathcal{R}, \pi \in S_n} \Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \pi, (r, L)) \in A_{f,\epsilon}(\mathbf{x})) \cdot \Pr(\Pi = \pi) \cdot \Pr(R = r).
\end{aligned} \tag{1}$$

$\Pr(\Pi = \pi) = 1/n!$, because Π is uniform on S_n. By Claim 4.5, $\mathcal{P}_{\mathbf{x},r} = \{\mathcal{P}_{\mathbf{x},r,\mathbf{i}}\}_{\mathbf{i} \in \mathcal{I}_{\mathbf{x},r}}$ is a partition of S_n. We can therefore rewrite the RHS of Equation (1) as:

$$\frac{1}{n!} \cdot \sum_{r \in \mathcal{R}} \Pr(R = r) \sum_{\mathbf{i} \in \mathcal{I}_{\mathbf{x},r}} \sum_{\pi \in \mathcal{P}_{\mathbf{x},r,\mathbf{i}}} \Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \pi, (r, L)) \in A_{f,\epsilon}(\mathbf{x})). \tag{2}$$

Using the second part of Claim 4.6, we can rewrite (2) as:

$$\frac{1}{n!} \cdot \sum_{r \in \mathcal{R}} \Pr(R = r) \sum_{\mathbf{i} \in \mathcal{I}_{\mathbf{x},r}} \sum_{\pi \in \mathcal{P}_{\mathbf{x},r,\mathbf{i}}} \frac{1}{|\mathcal{S}_{\mathbf{x},r,\mathbf{i}}|} \cdot \sum_{\sigma \in \mathcal{S}_{\mathbf{x},r,\mathbf{i}}} \Pr(g(\Delta_{\text{out}}(\sigma(\mathbf{x}), r), \sigma) \in A_{f,\epsilon}(\mathbf{x})). \tag{3}$$

Using the fact f is (g, ϵ)-symmetric, we can bound (3) from below by:

$$\frac{1}{n!} \cdot \sum_{r \in \mathcal{R}} \Pr(R = r) \sum_{\mathbf{i} \in \mathcal{I}_{\mathbf{x},r}} \sum_{\pi \in \mathcal{P}_{\mathbf{x},r,\mathbf{i}}} \frac{1}{|\mathcal{S}_{\mathbf{x},r,\mathbf{i}}|} \cdot \sum_{\sigma \in \mathcal{S}_{\mathbf{x},r,\mathbf{i}}} \Pr(\Delta_{\text{out}}(\sigma(\mathbf{x}), r) \in A_{f,\epsilon}(\sigma(\mathbf{x}))). \tag{4}$$

The terms in the sum over π are independent of π. This sum has $|\mathcal{P}_{\mathbf{x},r,\mathbf{i}}|$ terms, which by Claim 4.5 is identical to $|\mathcal{S}_{\mathbf{x},r,\mathbf{i}}|$. We can therefore rewrite (4) as:

$$\frac{1}{n!} \cdot \sum_{r \in \mathcal{R}} \Pr(R = r) \sum_{\mathbf{i} \in \mathcal{I}_{\mathbf{x},r}} \sum_{\sigma \in \mathcal{S}_{\mathbf{x},r,\mathbf{i}}} \Pr(\Delta_{\text{out}}(\sigma(\mathbf{x}), r) \in A_{f,\epsilon}(\sigma(\mathbf{x}))). \tag{5}$$

Using the fact that $\mathcal{S}_{\mathbf{x},r} = \{\mathcal{S}_{\mathbf{x},r,\mathbf{i}}\}_{\mathbf{i}\in\mathcal{I}_{\mathbf{x},r}}$ is a partition of S_n, we rewrite (5) as:

$$\begin{aligned} \frac{1}{n!} \cdot \sum_{r\in\mathcal{R}} \Pr(R=r) \sum_{\sigma\in S_n} \Pr(\Delta_{\text{out}}(\sigma(\mathbf{x}), r) \in A_{f,\epsilon}(\sigma(\mathbf{x}))) &= \\ = \frac{1}{n!} \sum_{\sigma\in S_n} \Pr(\Delta_{\text{out}}(\sigma(\mathbf{x}), R) \in A_{f,\epsilon}(\sigma(\mathbf{x}))). \end{aligned}$$

Each of the probabilities in the sum are at least $1-\delta$, due to the correctness of Δ. We conclude that the success probability of $\hat{\Delta}$ is at least $1-\delta$.

As for the query cost analysis, the worst-case query cost of $\hat{\Delta}$ is at most the worst-case query cost of Δ, since by the first part of Claim 4.6 for all choices of $\mathbf{x}, \pi, r, \ell$, there is some permutation σ, such that the stopping time of $\hat{\Delta}$ on $(\mathbf{x}, \pi, (r,\ell))$ is identical to the stopping time of Δ on $(\sigma(\mathbf{x}), r)$.

The analysis of the expected query cost is slightly trickier. Using a similar analysis to the one of the success probability, we can rewrite the expected query cost of $\hat{\Delta}$ on $\mathbf{x}$ as:

$$\mathrm{E}\left[\hat{T}(\mathbf{x}, \Pi, (R, L))\right] = \frac{1}{n!} \cdot \sum_{r\in\mathcal{R}} \Pr(R=r) \cdot \sum_{\mathbf{i}\in\mathcal{I}_{\mathbf{x},r}} \sum_{\pi\in\mathcal{P}_{\mathbf{x},r,\mathbf{i}}} \mathrm{E}\left[\hat{T}(\mathbf{x}, \pi, (r, L))\right]. \tag{6}$$

By the first part of Claim 4.6, we can rewrite the RHS of Equation (6) as:

$$\frac{1}{n!} \cdot \sum_{r\in\mathcal{R}} \Pr(R=r) \cdot \sum_{\mathbf{i}\in\mathcal{I}_{\mathbf{x},r}} \sum_{\pi\in\mathcal{P}_{\mathbf{x},r,\mathbf{i}}} \frac{1}{|\mathcal{S}_{\mathbf{x},r,\mathbf{i}}|} \sum_{\sigma\in\mathcal{S}_{\mathbf{x},r,\mathbf{i}}} \mathrm{E}\left[T(\sigma(\mathbf{x}), r)\right]. \tag{7}$$

Using now similar derivations to what was done in the analysis of the success probability, we further rewrite (7) as follows:

$$\begin{aligned} \frac{1}{n!} \cdot \sum_{r\in\mathcal{R}} \Pr(R=r) \cdot \sum_{\mathbf{i}\in\mathcal{I}_{\mathbf{x},r}} \sum_{\sigma\in\mathcal{S}_{\mathbf{x},r,\mathbf{i}}} \mathrm{E}\left[T(\sigma(\mathbf{x}), r)\right] &= \frac{1}{n!} \cdot \sum_{r\in\mathcal{R}} \Pr(R=r) \cdot \sum_{\sigma\in S_n} \mathrm{E}\left[T(\sigma(\mathbf{x}), r)\right] \\ &= \frac{1}{n!} \cdot \sum_{\sigma\in S_n} \mathrm{E}\left[T(\sigma(\mathbf{x}), R)\right]. \end{aligned}$$

Each of the terms in the last sum is at most q_E, since q_E is the expected query cost of Δ. It follows that the whole sum is also at most q_E, implying that the expected query cost of $\hat{\Delta}$ is at most q_E. $\square$

Lemma 4.7. *Let $\epsilon \geq 0$, let $0 < \delta < \frac{1}{2}$, and let Δ be a private-coin sampling protocol that (ϵ, δ)-approximates a function $f : \mathcal{X}^n \to \mathcal{Y}$ and whose index distribution on all inputs is uniform without replacement. Let q be the worst-case query cost of Δ and let q_E be its expected query cost. Then, there is a private-coin sampling protocol $\hat{\Delta}$ that $(\epsilon, 2\delta)$-approximates f and whose index distribution on all inputs is uniform with replacement. Furthermore,*

Low query cost case *($q \leq \sqrt{2\delta n}$): If Δ is index-oblivious, then also $\hat{\Delta}$ is index-oblivious. The worst-case query cost of $\hat{\Delta}$ is at most q and its expected query cost is at most $(1-\delta)q_E + \delta q$. If in addition $q \leq \sqrt[3]{2n \cdot q_E}$, then the expected query cost of $\hat{\Delta}$ is at most $2q_E$.*

High query cost case *($q \leq \frac{n}{2} - \sqrt{\frac{n}{2\delta}}$): The worst-case query cost of $\hat{\Delta}$ is at most $2q$ and its expected query cost is at most $2q_E$.*

Proof. The main idea of the construction is the following: $\hat{\Delta}$ uses queries chosen uniformly at random with replacement to simulate the uniform queries without replacement of Δ. There are two basic cases: either the total number of queries q Δ performs is small (at most $O(\sqrt{n})$) or not. In the former case, by a birthday argument, q uniform queries with replacement are very likely to contain no collisions, and are therefore identical to uniform queries without replacement. In the latter case, $\hat{\Delta}$ chooses $2q$ queries with replacement; with high probability, at least q of them are distinct, and can therefore be used as uniform queries without replacement for the simulation of Δ. Note that in the former case $\hat{\Delta}$ need not know the indices of the queries selected; this implies that if Δ was index-oblivious, then so is $\hat{\Delta}$. In the latter case, however, $\hat{\Delta}$ has to know the indices of the queries chosen, because it needs to pick the q distinct ones. Therefore, $\hat{\Delta}$ cannot be index-oblivious even if Δ is.

The formal proof of the above is given next. We are using similar notations to those introduced in the proof of Lemma 4.3.

Since Δ is a private-coin protocol and since its index distribution on any input is uniform without replacement, we can assume that Alice and the referee get two independent random strings: Alice gets a uniformly chosen permutation Π and the referee gets some random string R. Alice's query at the t-th step, for $t \le n$, is $\Pi(t)$.

Low query cost case. We start with the simulation for the case $q \le \sqrt{2\delta n}$. The protocol $\hat{\Delta}$ is defined as follows. Alice and the referee are given two independent private random strings. The random string of Alice is an infinite sequence $\mathbf{J}$ of indices $J_1, J_2, \ldots \in [n]$, which are chosen uniformly and independently. At the t-th step, Alice queries J_t, regardless of the previous queries and their answers. This already implies that the index distribution of $\hat{\Delta}$ on all inputs is uniform with replacement.

The random string R of the referee is the same random string used by the referee of Δ. Suppose, first, that Δ is not index-oblivious. Given a transcript $(\mathbf{i}, \mathbf{a})$, the referee of $\hat{\Delta}$ simulates the referee of Δ, as long as the index transcript $\mathbf{i}$ consists only of distinct indices. Otherwise, the referee immediately stops and outputs an arbitrary value. Formally, if $\mathbf{i}$ consists of distinct indices, then for all r and $\mathbf{a}$, $\hat{D}(\mathbf{i}, \mathbf{a}, r) = D(\mathbf{i}, \mathbf{a}, r)$; otherwise, $\hat{D}(\mathbf{i}, \mathbf{a}, r) = y$, where $y \in \mathcal{Y}$ is some arbitrary value. If Δ is index-oblivious, then the referee of $\hat{\Delta}$ is identical to the referee of Δ: for all r and $\mathbf{a}$, $\hat{D}(\mathbf{a}, r) = D(\mathbf{a}, r)$.

It is clear from the above description that if Δ is index-oblivious, then so is $\hat{\Delta}$. We are left to analyze the correctness of $\hat{\Delta}$ and its query cost. For that we use the following connection between Δ and $\hat{\Delta}$:

Claim 4.8. *Let $\mathbf{x}$ be any input and let r be any random string for the referee. Let $\mathbf{j} = (j_1, j_2, \ldots)$ be any random string for Alice, for which the first q indices $j_1, \ldots, j_q$ are distinct. Let π be any permutation satisfying $\pi(1, \ldots, q) = (j_1, \ldots, j_q)$. Then, $\hat{T}(\mathbf{x}, \mathbf{j}, r) = T(\mathbf{x}, \pi, r)$ and $\hat{\Delta}_{out}(\mathbf{x}, \mathbf{j}, r) = \Delta_{out}(\mathbf{x}, \pi, r)$.*

Proof. Suppose, initially, that Δ is not index-oblivious. For all $t \le q$, the t-th index Alice queries in Δ when given π is $\pi(t) = j_t$, which is the same index queried by Alice at the t-th step in $\hat{\Delta}$ when given $\mathbf{j}$. Therefore, the transcript of $\hat{\Delta}$ on $(\mathbf{x}, \mathbf{j})$ at step q is identical to the transcript of Δ on $(\mathbf{x}, \pi)$ at step q. That is, $\hat{\Delta}_q(\mathbf{x}, \mathbf{j}) = \Delta_q(\mathbf{x}, \pi)$.

Since the query indices in $\hat{\Delta}_q(\mathbf{x}, \mathbf{j})$ are distinct, then, by definition, for all r and for all $t \le q$, $\hat{D}(\hat{\Delta}_t(\mathbf{x}, \mathbf{j}), r) = D(\Delta_t(\mathbf{x}, \pi), r)$. Since the worst-case query cost of Δ is at most q, it stops on $(\mathbf{x}, \pi, r)$ at some step $t \le q$ and outputs $D(\Delta_t(\mathbf{x}, \pi), r)$. It follows that also $\hat{\Delta}$ stops at the same step on $(\mathbf{x}, \mathbf{j}, r)$, and outputs the same value. In other words, $\hat{T}(\mathbf{x}, \mathbf{j}, r) = T(\mathbf{x}, \pi, r)$ and $\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{j}, r) = \Delta_{\text{out}}(\mathbf{x}, \pi, r)$.

The case of an index-oblivious Δ is proven using the same argument, except for replacing transcripts by answer transcripts. □

Let E denote the event that $J_1, \ldots, J_q$, the first q elements of Alice's random string, are distinct. We prove:

Claim 4.9. *The distribution of the random variable $\hat{\Delta}_{out}(\mathbf{x}, \mathbf{J}, R)$ given the event E is identical to the distribution of the random variable $\Delta_{out}(\mathbf{x}, \Pi, R)$. Similarly, the distribution of the random variable $\hat{T}(\mathbf{x}, \mathbf{J}, R)$ given the event E is identical to the distribution of the random variable $T(\mathbf{x}, \Pi, R)$.*

Proof. We prove the claim for the random variables $\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) \mid E$ and $\Delta_{\text{out}}(\mathbf{x}, \Pi, R)$. An identical argument works for the random variables $\hat{T}(\mathbf{x}, \mathbf{J}, R) \mid E$ and $T(\mathbf{x}, \Pi, R)$.

Fix any $y \in \mathcal{Y}$. Denote by $\mathbf{J}^q$ the prefix $(J_1, \ldots, J_q)$ of $\mathbf{J}$. Let $\mathcal{T}$ denote the set of all sequences $\mathbf{j} = (j_1, \ldots, j_q)$ of q distinct indices. Since the random variables $\mathbf{J}$ and R are independent (even given the event E), we can rewrite the probability that $\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) = y$ given E as follows:

$$\Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) = y \mid E) \;=\; \sum_{\mathbf{j} \in \mathcal{T}} \Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) = y \mid \mathbf{J}^q = \mathbf{j}) \cdot \Pr(\mathbf{J}^q = \mathbf{j} \mid E). \tag{8}$$

For a sequence $\mathbf{j} \in \mathcal{T}$, let $\mathcal{P}_{\mathbf{j}}$ be the set of all permutations π satisfying $\pi(1, \ldots, q) = \mathbf{j}$. Using Claim 4.8, we can rewrite the RHS of Equation (8) as follows:

$$\begin{aligned} &\sum_{\mathbf{j} \in \mathcal{T}} \Pr(\mathbf{J}^q = \mathbf{j} \mid E) \cdot \frac{1}{|\mathcal{P}_{\mathbf{j}}|} \sum_{\pi \in \mathcal{P}_{\mathbf{j}}} \Pr(\Delta_{\text{out}}(\mathbf{x}, \pi, R) = y) \;= \\ &\quad= \sum_{\mathbf{j} \in \mathcal{T}} \Pr(\mathbf{J}^q = \mathbf{j} \mid E) \cdot \Pr(\Delta_{\text{out}}(\mathbf{x}, \Pi, R) = y \mid \Pi(1, \ldots, q) = \mathbf{j}). \end{aligned} \tag{9}$$

By a symmetry argument, the distribution of the random variable $\mathbf{J}^q$ given the event E is uniform on $\mathcal{T}$. Similarly, the random variable $\Pi(1, \ldots, q)$ is uniformly distributed in $\mathcal{T}$. We can thus rewrite (9) as:

$$\sum_{\mathbf{j} \in \mathcal{T}} \Pr(\Pi(1, \ldots, q) = \mathbf{j}) \cdot \Pr(\Delta_{\text{out}}(\mathbf{x}, \Pi, R) = y \mid \Pi(1, \ldots, q) = \mathbf{j}) \;=\; \Pr(\Delta_{\text{out}}(\mathbf{x}, \Pi, R) = y).$$

We conclude that $\Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) = y \mid E) = \Pr(\Delta_{\text{out}}(\mathbf{x}, \Pi, R) = y)$ for all $y \in \mathcal{Y}$, implying that the random variables $\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) | E$ and $\Delta_{\text{out}}(\mathbf{x}, \Pi, R)$ have the same distribution. □

We next analyze the correctness of $\hat{\Delta}$. Let $\mathbf{x}$ be any input. Let $\overline{E}$ denote the complement of E. Then,

$$
\begin{aligned}
&\Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) \notin A_{f,\epsilon}(\mathbf{x})) = \\
&\quad = \Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) \notin A_{f,\epsilon}(\mathbf{x}) \mid E) \cdot \Pr(E) + \Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) \notin A_{f,\epsilon}(\mathbf{x}) \mid \overline{E}) \cdot \Pr(\overline{E}) \\
&\quad \leq \Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) \notin A_{f,\epsilon}(\mathbf{x}) \mid E) + \Pr(\overline{E}) \\
&\quad = \Pr(\Delta_{\text{out}}(\mathbf{x}, \Pi, R) \notin A_{f,\epsilon}(\mathbf{x})) + \Pr(\overline{E}) \quad \text{(By Claim 4.9)} \\
&\quad \leq \delta + \Pr(\overline{E}). \quad \text{(By the correctness of } \Delta\text{)}
\end{aligned}
$$

We next show that $\Pr(\overline{E}) \leq \delta$, implying that $\hat{\Delta}$ $(\epsilon, 2\delta)$-approximates f. Define $\binom{q}{2}$ indicator random variables $X_{k\ell}$, such that $X_{k\ell} = 1$ iff $J_k = J_\ell$. $J_1, \ldots, J_q$ are not distinct if and only if $\sum_{k,\ell} X_{k\ell} \geq 1$. The expectation of $\sum_{k,\ell} X_{k\ell}$ is $\binom{q}{2}\frac{1}{n} \leq \frac{q^2}{2n}$. Therefore, using Markov's inequality, $\Pr(\overline{E}) \leq \frac{q^2}{2n} \leq \delta$.

We next turn to the analysis of the query cost of $\hat{\Delta}$. We begin with the worst-case query cost. Fix any input $\mathbf{x}$ and any random strings $\mathbf{j}$ and r. If the first q indices in $\mathbf{j}$ are distinct, then by Claim 4.8, $\hat{T}(\mathbf{x}, \mathbf{j}, r) = T(\mathbf{x}, \pi, r)$. The latter is at most q, since the worst-case query cost of Δ is q. Suppose, then, that there are collisions in the first q indices of $\mathbf{j}$. If Δ is not index-oblivious, then $\hat{\Delta}$ can detect these collisions and stop immediately. Thus in this case $\hat{\Delta}$ indeed has worst-case query cost of at most q.

The case that Δ is index-oblivious is slightly more intricate. Let $j_1, \ldots, j_q$ be the first q indices in $\mathbf{j}$ (which may consist of repetitions). Define an input $\mathbf{x}'$ as follows: for $t = 1, \ldots, q$, $\mathbf{x}'_t = \mathbf{x}_{j_t}$; for $q < t \leq n$, $\mathbf{x}'_t = x$, where $x \in \mathcal{X}$ is some arbitrary value. Let π be the identity permutation. Note that $(\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_q}) = (\mathbf{x}'_1, \ldots, \mathbf{x}'_q)$ is both the answer transcript of $\hat{\Delta}$ on $(\mathbf{x}, \mathbf{j})$ at step q and the answer transcript of Δ on $(\mathbf{x}', \pi)$ at step q. Now, since the decision functions of the referees of $\hat{\Delta}$ and of Δ are identical, then when they both get to see the above answer transcript and both get the same random string r, then they both stop at the same step t. $t \leq q$, because the worst-case query cost of Δ is at most q.

As for the expected query cost,

$$
\begin{aligned}
&\max_{\mathbf{x}} \mathrm{E}\left[\hat{T}(\mathbf{x}, \mathbf{J}, R)\right] \\
&\quad = \max_{\mathbf{x}} \left(\mathrm{E}\left[\hat{T}(\mathbf{x}, \mathbf{J}, R) \mid E\right] \Pr(E) + \mathrm{E}\left[\hat{T}(\mathbf{x}, \mathbf{J}, R) \mid \overline{E}\right] \Pr(\overline{E})\right) \\
&\quad \leq \max_{\mathbf{x}} \left(\mathrm{E}\left[\hat{T}(\mathbf{x}, \mathbf{J}, R) \mid E\right] \cdot \left(1 - \Pr(\overline{E})\right) + q \cdot \Pr(\overline{E})\right) \\
&\quad = \max_{\mathbf{x}} \left(\mathrm{E}\left[T(\mathbf{x}, \Pi, R)\right] \cdot \left(1 - \Pr(\overline{E})\right) + q \cdot \Pr(\overline{E})\right) \quad \text{(By Claim 4.9)} \\
&\quad \leq q_E(1 - \delta) + q\delta.
\end{aligned}
$$

Note that if $q \leq \sqrt[3]{2n \cdot q_E}$, then $q \cdot \Pr(\overline{E}) \leq q^3/(2n) \leq q_E$. Therefore, in this case the expected query cost of $\hat{\Delta}$ is at most $q_E(1 - \delta) + q_E < 2q_E$.

High query cost case. We now proceed to the case $q \leq \frac{n}{2} - \sqrt{\frac{n}{2\delta}}$. For the description below we will need to define the following operators on transcripts. Let $\mathbf{i} = (i_1, \ldots, i_t)$ be an index transcript and let $\mathbf{a} = (a_1, \ldots, a_t)$ be a corresponding answer transcript. Let d be the number of distinct indices among $i_1, \ldots, i_t$, and let $k_1, \ldots, k_d$ be the positions of the first occurrences of these indices. We call $k_1, \ldots, k_d$ the distinct index positions of $\mathbf{i}$. For each $1 \leq \ell \leq d$, we denote by $\text{DIP}_\ell(\mathbf{i})$ the prefix of $k_1, \ldots, k_d$ of length ℓ. The d distinct indices, $i_{k_1}, \ldots, i_{k_d}$, are called the distinct index subsequence of $\mathbf{i}$. For each $1 \leq \ell \leq d$, we denote by $\text{DIS}_\ell(\mathbf{i})$ the prefix of $i_{k_1}, \ldots, i_{k_d}$ of length ℓ. $\text{DIS}_\ell(\mathbf{i}, \mathbf{a})$ is the corresponding subsequence of the transcript: $(i_{k_1}, \ldots, i_{k_\ell}, a_{k_1}, \ldots, a_{k_\ell})$.

The simulating algorithm $\hat{\Delta}$ in this case is identical to the simulating algorithm in the case $q \leq \sqrt{2\delta n}$, except for the decision function of the referee. We thus already know that it is private-coin and that its index distribution on all inputs is uniform with replacement.

Given a transcript $(\mathbf{i}, \mathbf{a})$, the referee extracts the distinct index subsequence: $(\mathbf{i}', \mathbf{a}') = \text{DIS}_d(\mathbf{i}, \mathbf{a})$ (d is the number of distinct indices in $\mathbf{i}$), and runs the referee of Δ on the transcript $(\mathbf{i}', \mathbf{a}')$. That is, $\hat{D}(\mathbf{i}, \mathbf{a}, r) = D(\mathbf{i}', \mathbf{a}', r)$ for all r. If the number of queries made so far is $2q$, the referee stops, even if the simulation of Δ has not finished. If the simulation has not produced an output value thus far, the referee outputs an arbitrary value. Formally, if $|\mathbf{i}| = 2q$ and if $D(\mathbf{i}', \mathbf{a}', r) = \text{CONT}$, then $\hat{D}(\mathbf{i}, \mathbf{a}, r) = y$, where $y \in \mathcal{Y}$ is some arbitrary value.

The above definition immediately implies that the worst-case query cost of $\hat{\Delta}$ is at most $2q$. It also implies that $\hat{\Delta}$ is not index-oblivious, even if Δ is, since the referee needs to know the index transcript $\mathbf{i}$ in order to find the positions of the distinct indices.

We are left to analyze the correctness of $\hat{\Delta}$ and its expected query cost. Similar to Claim 4.8, we now have:

Claim 4.10. *Let $\mathbf{x}$ be any input, let r be any random string for the referee, and let $\mathbf{j} = (j_1, j_2, \ldots)$ be any random string for Alice, for which q out of the first $2q$ elements are distinct. Let $(k_1, \ldots, k_q) = \text{DIP}_q(\mathbf{j})$ be the positions of the first occurrences of these q elements. Let π be any permutation satisfying $\pi(1, \ldots, q) = \text{DIS}_q(\mathbf{j}) = (j_{k_1}, \ldots, j_{k_q})$. Then, $\hat{\Delta}_{out}(\mathbf{x}, \mathbf{j}, r) = \Delta_{out}(\mathbf{x}, \pi, r)$ and $\hat{T}(\mathbf{x}, \mathbf{j}, r) = k_T$, where $T = T(\mathbf{x}, \pi, r)$.*

Proof. For all $t \leq q$, the t-th index Alice queries in Δ when given π is $\pi(t) = j_{k_t}$, which is the same index queried by Alice at the k_t-th step in $\hat{\Delta}$ when given $\mathbf{j}$. Therefore, if $(\mathbf{i}, \mathbf{a})$ is the transcript of $\hat{\Delta}$ on $(\mathbf{x}, \mathbf{j})$ at step k_t, then $(\mathbf{i}', \mathbf{a}')$ is the transcript of Δ on $(\mathbf{x}, \pi)$, where $(\mathbf{i}', \mathbf{a}') = \text{DIS}_t(\mathbf{i}, \mathbf{a})$.

By the definition of the referee's decision function, for all r and $t \leq q$, $\hat{D}(\hat{\Delta}_{k_t}(\mathbf{x}, \mathbf{j}), r) = D(\Delta_t(\mathbf{x}, \pi), r)$. Since the worst-case query cost of Δ is at most q, it stops on $(\mathbf{x}, \pi, r)$ at some step $t \leq q$ and outputs $D(\Delta_t(\mathbf{x}, \pi), r)$. It follows that $\hat{\Delta}$ stops at step k_t on $(\mathbf{x}, \mathbf{j}, r)$, and outputs the same value. In other words, $\hat{T}(\mathbf{x}, \mathbf{j}, r) = k_{T(\mathbf{x}, \pi, r)}$ and $\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{j}, r) = \Delta_{\text{out}}(\mathbf{x}, \pi, r)$. $\square$

We use $\Pi, \mathbf{J}$, and R to denote the same random variables as above. We define now E to be the event that at least q of $J_1, \ldots, J_{2q}$ are distinct. We prove:

Claim 4.11. *The distribution of the random variable $\hat{\Delta}_{out}(\mathbf{x}, \mathbf{J}, R)$ given the event E is identical to the distribution of the random variable $\Delta_{out}(\mathbf{x}, \Pi, R)$.*

Proof. Denote by $\mathbf{J}^{2q}$ the prefix $(J_1, \ldots, J_{2q})$ of $\mathbf{J}$. Let $\mathcal{S}$ be the set of all sequences in $[n]^{2q}$, which contain at least q distinct indices. Recall that for a sequence $\mathbf{j} \in \mathcal{S}$, $\textsc{dis}_q(\mathbf{j})$ denotes the subsequence consisting of the first occurrences of the q distinct indices. Let $\mathcal{T}$ denote the set of all sequences in $[n]^q$ that consist of q distinct indices.

Fix any $y \in \mathcal{Y}$. Since $\mathbf{J}$ and R are independent given E, we can rewrite the probability $\Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) = y \mid E)$ as follows:

$$\Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) = y \mid E) \;=\; \sum_{\mathbf{j} \in \mathcal{S}} \Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) = y \mid \mathbf{J}^{2q} = \mathbf{j}) \cdot \Pr(\mathbf{J}^{2q} = \mathbf{j} \mid E). \tag{10}$$

For a sequence $\mathbf{j} \in \mathcal{S}$, let $\mathcal{P}_{\mathbf{j}}$ be the set of permutations π that satisfy $\pi(1, \ldots, q) = \textsc{dis}_q(\mathbf{j})$. Using Claim 4.10, we can rewrite the RHS of Equation (10) as:

$$\begin{aligned}
&\sum_{\mathbf{j} \in \mathcal{S}} \Pr(\mathbf{J}^{2q} = \mathbf{j} \mid E) \cdot \frac{1}{|\mathcal{P}_{\mathbf{j}}|} \sum_{\pi \in \mathcal{P}_{\mathbf{j}}} \Pr(\Delta_{\text{out}}(\mathbf{x}, \pi, R) = y) \;= \\
&\quad = \sum_{\mathbf{j} \in \mathcal{S}} \Pr(\mathbf{J}^{2q} = \mathbf{j} \mid E) \cdot \Pr(\Delta_{\text{out}}(\mathbf{x}, \Pi, R) = y \mid \Pi(1, \ldots, q) = \textsc{dis}_q(\mathbf{j})).
\end{aligned} \tag{11}$$

We next rearrange the terms of the sum and group them according to the value of $\textsc{dis}_q(\mathbf{j})$. Thus, the RHS of (11) is written as:

$$\begin{aligned}
&\sum_{\mathbf{j}' \in \mathcal{T}} \left(\Pr(\Delta_{\text{out}}(\mathbf{x}, \Pi, R) = y \mid \Pi(1, \ldots, q) = \mathbf{j}') \cdot \sum_{\mathbf{j} \in \mathcal{S}, \text{DIS}_q(\mathbf{j}) = \mathbf{j}'} \cdot \Pr(\mathbf{J}^{2q} = \mathbf{j} \mid E) \right) \;= \\
&\quad = \sum_{\mathbf{j}' \in \mathcal{T}} \Pr(\Delta_{\text{out}}(\mathbf{x}, \Pi, R) = y \mid \Pi(1, \ldots, q) = \mathbf{j}') \cdot \Pr(\textsc{dis}_q(\mathbf{J}) = \mathbf{j}' \mid E).
\end{aligned} \tag{12}$$

By a symmetry argument, the distribution of the random variable $\textsc{dis}_q(\mathbf{J})$ given the event E is uniform on $\mathcal{T}$. Similarly, $\Pi(1, \ldots, q)$ is uniformly distributed in $\mathcal{T}$. We can thus rewrite the RHS of (12) as:

$$\sum_{\mathbf{j}' \in \mathcal{T}} \Pr(\Delta_{\text{out}}(\mathbf{x}, \Pi, R) = y \mid \Pi(1, \ldots, q) = \mathbf{j}') \cdot \Pr(\Pi(1, \ldots, q) = \mathbf{j}') \;=\; \Pr(\Delta_{\text{out}}(\mathbf{x}, \Pi, R) = y).$$

We conclude that $\Pr(\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R) = y \mid E) \;=\; \Pr(\Delta_{\text{out}}(\mathbf{x}, \Pi, R) = y)$, for all $y \in \mathcal{Y}$, and therefore the distributions of the random variables $\hat{\Delta}_{\text{out}}(\mathbf{x}, \mathbf{J}, R)|E$ and $\Delta_{\text{out}}(\mathbf{x}, \Pi, R)$ are identical. $\square$

We next analyze the correctness of $\hat{\Delta}$. Let $\mathbf{x}$ be any input. Let $\overline{E}$ denote the complement of E. Then,

$$\begin{aligned}
&\Pr(\hat{\Delta}_{\text{out}}(\mathbf{x},\mathbf{J},R) \notin A_{f,\epsilon}(\mathbf{x})) = \\
&\quad = \Pr(\hat{\Delta}_{\text{out}}(\mathbf{x},\mathbf{J},R) \notin A_{f,\epsilon}(\mathbf{x}) \mid E) \cdot \Pr(E) + \Pr(\hat{\Delta}_{\text{out}}(\mathbf{x},\mathbf{J},R) \notin A_{f,\epsilon}(\mathbf{x}) \mid \overline{E}) \cdot \Pr(\overline{E}) \\
&\quad \leq \Pr(\hat{\Delta}_{\text{out}}(\mathbf{x},\mathbf{J},R) \notin A_{f,\epsilon}(\mathbf{x}) \mid E) + \Pr(\overline{E}) \\
&\quad = \Pr(\Delta_{\text{out}}(\mathbf{x},\Pi,R) \notin A_{f,\epsilon}(\mathbf{x})) + \Pr(\overline{E}) \quad \text{(By Claim 4.11)} \\
&\quad \leq \delta + \Pr(\overline{E}). \quad \text{(By the correctness of } \Delta)
\end{aligned}$$

We next show that $\Pr(\overline{E}) \leq \delta$, implying that $\hat{\Delta}$ $(\epsilon, 2\delta)$-approximates f. Define $\binom{2q}{2}$ indicator random variables $X_{k\ell}$, such that $X_{k\ell} = 1$ iff $J_k = J_\ell$. If the sequence $J_1, \ldots, J_{2q}$ consists of at most q distinct values, then $\sum_{k,\ell} X_{k\ell} \geq q$ (because the way to create the least number of collisions is to have two occurrences for each of the q values). On the other hand, the expectation of $\sum_{k,\ell} X_{k\ell}$ is $\binom{2q}{2}\frac{1}{n} \leq \frac{2q^2}{n}$. Using the pairwise independence of $X_{k\ell}$ and Chebyshev's inequality we obtain:

$$\begin{aligned}
\Pr(\overline{E}) \;&\leq\; \Pr(\sum_{k,\ell} X_{k\ell} \geq q) \;\leq\; \Pr(\left|\sum_{k,\ell} X_{k\ell} - \mathrm{E}\left[\sum_{k,\ell} X_{k\ell}\right]\right| \geq q - \frac{2q^2}{n}) \\
&\leq\; \frac{\textsc{Var}\left[\sum_{k,\ell} X_{k\ell}\right]}{q^2\left(1-\frac{2q}{n}\right)^2} \;=\; \frac{\binom{2q}{2}\frac{1}{n}\frac{n-1}{n}}{q^2\left(1-\frac{2q}{n}\right)^2} \;\leq\; \frac{\frac{2q^2}{n}}{q^2(1-\frac{2q}{n})^2} \;=\; \frac{2}{n\left(1-\frac{2q}{n}\right)^2}.
\end{aligned}$$

The last expression is at most δ for $q \leq n/2 - \sqrt{n/(2\delta)}$.

We next turn to the analysis of the expected query cost of $\hat{\Delta}$. The event that the random infinite sequence $\mathbf{J}$ consists of less than n distinct elements has probability measure 0. We can thus expand the expected query cost of $\hat{\Delta}$ on any input $\mathbf{x}$ as follows:

$$\mathrm{E}\left[\hat{T}(\mathbf{x},\mathbf{J},R)\right] \;=\; \sum_{\mathbf{j}'\in\mathcal{T}} \mathrm{E}\left[\hat{T}(\mathbf{x},\mathbf{J},R) \mid \textsc{dis}_n(\mathbf{J}) = \mathbf{j}'\right] \cdot \Pr(\textsc{dis}_n(\mathbf{J}) = \mathbf{j}'), \tag{13}$$

where $\mathcal{T} \subseteq [n]^n$ is the set of all $n!$ sequences of n distinct elements.

For any infinite sequence of indices $\mathbf{j}$, which consists of n distinct indices, let $(k_1, \ldots, k_n) = \textsc{dip}_n(\mathbf{j})$ be the distinct index positions of $\mathbf{j}$. For every $1 \leq \ell \leq n$, we call the difference $k_\ell - k_{\ell-1}$ the ℓ-th delay of $\mathbf{j}$, and denote it by $\textsc{delay}_\ell(\mathbf{j})$. (We define $k_0 = 0$). Note that for every $1 \leq t \leq n$, $k_t = \sum_{\ell=1}^{t} \textsc{delay}_\ell(\mathbf{j})$.

For a sequence $\mathbf{j}' \in \mathcal{T}$ of n distinct indices, let $\pi_{\mathbf{j}'}$ be the permutation satisfying $\pi_{\mathbf{j}'}(1,\ldots,n) = \mathbf{j}'$. We know from Claim 4.10 that for all $\mathbf{x}$ and r, and for all $\mathbf{j}$ s.t. $\textsc{dis}_n(\mathbf{j}) = \mathbf{j}'$, $\hat{T}(\mathbf{x},\mathbf{j},r) = \min(k_T, 2q)$, where $T = T(\mathbf{x},\pi_{\mathbf{j}'},r)$. Therefore, $\hat{T}(\mathbf{x},\mathbf{j},r) \leq k_T = \sum_{\ell=1}^{T} \textsc{delay}_\ell(\mathbf{j})$. We can thus bound the expected value of $\hat{T}(\mathbf{x},\mathbf{J},R)$ given the event "$\textsc{dis}_n(\mathbf{J}) = \mathbf{j}'$" as follows:

$$\mathrm{E}\left[\hat{T}(\mathbf{x},\mathbf{J},R) \mid \textsc{dis}_n(\mathbf{J}) = \mathbf{j}'\right] \;\leq\; \mathrm{E}\left[\sum_{\ell=1}^{T(\mathbf{x},\pi_{\mathbf{j}'},R)} \textsc{delay}_\ell(\mathbf{J}) \mid \textsc{dis}_n(\mathbf{J}) = \mathbf{j}'\right]. \tag{14}$$

We cannot immediately use linearity of expectation for rewriting the expectation on the RHS of (14), since the number of summands in the sum is a random variable itself. So we first expand on the values of R (while exploiting the independence of R and $\mathbf{J}$), and then apply the linearity of expectation:

$$\begin{aligned}
&\mathrm{E}\left[\sum_{\ell=1}^{T(\mathbf{x},\pi_{\mathbf{j}'},R)} \textsc{delay}_\ell(\mathbf{J}) \mid \textsc{dis}_n(\mathbf{J}) = \mathbf{j}'\right] \; = \\
&= \sum_{r\in\mathcal{R}} \mathrm{E}\left[\sum_{\ell=1}^{T(\mathbf{x},\pi_{\mathbf{j}'},r)} \textsc{delay}_\ell(\mathbf{J}) \mid \textsc{dis}_n(\mathbf{J}) = \mathbf{j}', R = r\right] \cdot \Pr(R = r) \\
&= \sum_{r\in\mathcal{R}} \sum_{\ell=1}^{T(\mathbf{x},\pi_{\mathbf{j}'},r)} \mathrm{E}\left[\textsc{delay}_\ell(\mathbf{J}) \mid \textsc{dis}_n(\mathbf{J}) = \mathbf{j}', R = r\right] \cdot \Pr(R = r).
\end{aligned} \tag{15}$$

The random variable $\textsc{delay}_\ell(\mathbf{J})$ depends only on $\mathbf{J}$ and is independent of R. Moreover, for every choice of $\mathbf{j}'$, by a symmetry argument, $E(\textsc{delay}_\ell(\mathbf{J}) \mid \textsc{dis}_n(\mathbf{J}) = \mathbf{j}') = E(\textsc{delay}_\ell(\mathbf{J})) = n/(n-\ell+1)$. $T(\mathbf{x},\pi_{\mathbf{j}'},r) \le q \le n/2$ for all $\mathbf{x},\pi_{\mathbf{j}'},r$, because the worst-case query cost of Δ is at most $n/2$. Thus, for all $\ell \le T(\mathbf{x},\pi_{\mathbf{j}'},r)$, $E(\textsc{delay}_\ell(\mathbf{J})) \le 2$. We conclude that the last expression in Equation 15 can be bounded as follows:

$$\begin{aligned}
&\sum_{r\in\mathcal{R}} \sum_{\ell=1}^{T(\mathbf{x},\pi_{\mathbf{j}'},r)} \mathrm{E}\left[\textsc{delay}_\ell(\mathbf{J}) \mid \textsc{dis}_n(\mathbf{J}) = \mathbf{j}', R = r\right] \cdot \Pr(R = r) \;\le \\
&\quad\le\; 2 \cdot \sum_{r\in\mathcal{R}} T(\mathbf{x},\pi_{\mathbf{j}'},r) \cdot \Pr(R = r) \;=\; 2 \cdot \mathrm{E}\left[T(\mathbf{x},\pi_{\mathbf{j}'},R)\right].
\end{aligned}$$

Substituting this bound back in the expression for the expected query cost of $\hat{\Delta}$ on $\mathbf{x}$ (Equation 13), we have:

$$\begin{aligned}
\mathrm{E}\left[\hat{T}(\mathbf{x},\mathbf{J},R)\right] &= \sum_{\mathbf{j}'\in\mathcal{T}} \mathrm{E}\left[\hat{T}(\mathbf{x},\mathbf{J},R) \mid \textsc{dis}_n(\mathbf{J}) = \mathbf{j}'\right] \cdot \Pr(\textsc{dis}_n(\mathbf{J}) = \mathbf{j}') \\
&\le 2 \cdot \sum_{\mathbf{j}'\in\mathcal{T}} \mathrm{E}\left[T(\mathbf{x},\pi_{\mathbf{j}'},R)\right] \cdot \Pr(\textsc{dis}_n(\mathbf{J}) = \mathbf{j}').
\end{aligned}$$

By a symmetry argument, the random variable $\textsc{dis}_n(\mathbf{J})$ is distributed uniformly on $\mathcal{T}$. Thus, for all $\mathbf{j}' \in \mathcal{T}$, $\Pr(\textsc{dis}_n(\mathbf{J}) = \mathbf{j}') = 1/n!$, which is the same as $\Pr(\Pi = \pi)$, for any $\pi \in S_n$. Also note that the mapping $\mathbf{j}' \mapsto \pi_{\mathbf{j}'}$ is a 1-1 mapping from $\mathcal{T}$ to S_n. We thus have:

$$\mathrm{E}\left[\hat{T}(\mathbf{x},\mathbf{J},R)\right] \;\le\; 2 \cdot \sum_{\pi\in S_n} \mathrm{E}\left[T(\mathbf{x},\pi,R)\right] \cdot \Pr(\Pi = \pi) \;=\; 2 \cdot \mathrm{E}\left[T(\mathbf{x},\Pi,R)\right] \;=\; 2q_E. \qquad \square$$

4.2 Row-symmetric and column-symmetric functions

For a set $\mathcal{X}$, $M_{m\times n}(\mathcal{X})$ denotes the set of all $m \times n$ matrices whose entries are elements of $\mathcal{X}$. For a matrix $\mathbf{A} \in M_{m\times n}(\mathcal{X})$ and for a permutation $\sigma \in S_m$, we denote by $\sigma_R(\mathbf{A})$ the matrix obtained from $\mathbf{A}$ by permuting its rows according to σ. That is, the i-th row of $\sigma_R(\mathbf{A})$ is the $\sigma^{-1}(i)$-th row of $\mathbf{A}$. Similarly, for a permutation $\pi \in S_n$, $\pi_C(\mathbf{A})$ is the matrix obtained from $\mathbf{A}$ by permuting its columns according to π.

A function f on $M_{m\times n}(\mathcal{X})$ is called row-symmetric, if it is invariant under permutations of its rows. That is, $f(\mathbf{A}) = f(\sigma_R(\mathbf{A}))$ for all $\mathbf{A}$ and σ. The rank of a matrix is an example of a row-symmetric function. Similar to the definitions of g-symmetric functions and permutation-commutative functions, we define g-row-symmetric functions and row-permutation-commutative functions. The low-rank matrix approximation problem and the matrix reconstruction problem are permutation-commutative (see Section 7). Again, we define the corresponding notion for approximation of functions:

Definition 4.12 ((g, ϵ)-row-symmetric functions). Let $f : M_{m\times n}(\mathcal{X}) \to \mathcal{Y}$ be a function with approximation notion $A_{f,\epsilon}$. For a function $g : \mathcal{Y} \times S_m \to \mathcal{Y}$, we call f (g, ϵ)-row-symmetric, if for all input matrices $\mathbf{A} \in M_{m\times n}(\mathcal{X})$, for all permutations $\sigma \in S_m$, and for all $y \in A_{f,\epsilon}(\sigma_R(\mathbf{A}))$, $\quad g(y, \sigma) \in A_{f,\epsilon}(\mathbf{A})$.

We define(g, ϵ)-column-symmetric functions analogously.

A standard sampling algorithm computing a function over matrices queries one entry of the input matrix at a time. We call a sampling algorithm that queries a full row of the input matrix at a time a row-querying sampling algorithm. The query cost of such an algorithm is the number of full rows it queries.

We prove that any sampling algorithm Δ approximating a (g, ϵ)-row symmetric function can be simulated by a sampling algorithm Δ' with the following properties: (1) it is row-querying; (2) its queries are uniform with replacement; (3) it uses private coins; (4) the number of rows it queries is at most twice the number of queries of Δ. If the number of queries used by Δ is small enough (at most $O(\sqrt{m})$), then Δ' is also index-oblivious.

Theorem 4.13. *Let $\epsilon \geq 0$, let $0 < \delta < \frac{1}{2}$, and let Δ be any sampling protocol that (ϵ, δ)-approximates a (g, ϵ)-row-symmetric function $f : M_{m\times n}(\mathcal{X}) \to \mathcal{Y}$. Let q be the worst-case query cost of Δ and let q_E be its expected query cost. Then, there is a private-coin, row-querying, sampling protocol $\hat{\Delta}$ that $(\epsilon, 2\delta)$-approximates f and whose index distribution on all input matrices is uniform with replacement. Furthermore,*

Low query cost case *($q \leq \sqrt{2\delta n}$): If g depends only on its first argument, then $\hat{\Delta}$ is index-oblivious. The worst-case query cost of $\hat{\Delta}$ is at most q and its expected query cost is at most $(1 - \delta)q_E + \delta q$. If in addition $q \leq \sqrt[3]{2n \cdot q_E}$, then the expected query cost of $\hat{\Delta}$ is at most $2q_E$.*

High query cost case *($q \leq \frac{n}{2} - \sqrt{\frac{n}{2\delta}}$): The worst-case query cost of $\hat{\Delta}$ is at most $2q$ and its expected query cost is at most $2q_E$.*

An analogous theorem holds for (g, ϵ)-column symmetric functions.

Proof. The theorem follows from the following simple observation: we view each input matrix $\mathbf{A} \in M_{m\times n}(\mathcal{X})$ as an n-dimensional vector $\mathbf{v_A}$ whose entries are elements of $\mathcal{X}^m$. That is, the i-th row of $\mathbf{A}$ becomes the i-th entry of $\mathbf{v_A}$. We denote by $f' : (\mathcal{X}^m)^n \to \mathcal{Y}$ the function induced by f and this input transformation. Now it is trivial to verify that since f is (g, ϵ)-row-symmetric, then f' is (g, ϵ)-symmetric.

The protocol Δ that (ϵ, δ)-approximates f induces a protocol Δ' that (ϵ, δ)-approximates f'. Δ' simulates Δ as follows. When Δ queries some entry (i, j) of $\mathbf{A}$, Δ' queries the i-th entry of $\mathbf{v_A}$; the answer to this query consists of all the entries in the i-th row of $\mathbf{A}$ and in particular (i, j). We conclude that Δ' indeed (ϵ, δ)-approximates f' and that the number queries it uses is always the same as the number of queries used by Δ.

Now, since f' is (g, ϵ)-symmetric, we can apply Theorem 4.2 and get a new sampling protocol $\hat{\Delta}$, which satisfies all the conditions stated above. $\hat{\Delta}$ can be viewed also as a protocol for f that always queries full rows of the input matrix. $\square$

5 Reduction from classification to sampling

In this section we show that any sampling algorithm that approximates a function f derives a statistical classifier for the query distributions of an appropriately chosen family of inputs to f. We start by defining statistical classification and then describe the reduction.

5.1 Statistical classification

Loosely speaking, statistical classification is defined as follows. A "black box" contains a distribution μ, which is guaranteed to be one of ℓ known distributions $\mu_1, \ldots, \mu_\ell$ on the same domain $\mathcal{B}$. (We do not consider here the "Bayesian" scenario, in which there is some prior distribution on $[\ell]$). A classifier is a randomized oracle algorithm that has to determine the identity of μ. Each oracle call produces a sample from μ. At the end of its execution, the classifier announces a guess for the identity of μ. This guess is required to be correct with probability at least $1-\delta$, for all possible choices of μ (the probability is over both the samples from μ and the coin tosses of the classifier).

Formally, a sequential classifier C for ℓ sequential distributions $\mu_1, \ldots, \mu_\ell$ on a domain $\mathcal{B}$ (recall the definition from Section 2.5) is a communication protocol between two players: Alice and a "referee". Alice gets as input an index $j \in [\ell]$ and the referee gets a random string $r \in \mathcal{R}$. The referee's random string is independent of the distributions $\mu_1, \ldots, \mu_\ell$. The protocol proceeds in rounds as follows. The t-th round starts with Alice sending a sample $b \in \mathcal{B}$ to the referee. b is chosen according to the distribution $\mu_{j,\mathbf{b}}$, where $\mathbf{b} = (b_1, \ldots, b_{t-1})$ is the sequence of previous samples Alice generated from μ_j. The referee, after seeing b, applies a "decision function" $D : \mathcal{B}^* \times \mathcal{R} \to [\ell] \cup \{\textsc{cont}\}$, which determines his decision: whether to continue getting samples from Alice, or to stop and declare a decision. The decision is based on the referee's random string and on the t samples generated so far.

The stopping time of C on j and r, denoted $T(j, r)$, is the random variable corresponding to the step t at which C stops when Alice gets the input j and the referee gets the random string r. The worst-case sample cost of C is $\max_j \max_r \max_{\mu_j}(T(j,r))$. Here the maximum is over the choice of the input distribution j, over the random string of the referee, and over the samples drawn from μ_j. The expected sample cost of C is $\max_j \mathrm{E}[T(j,R)]$. The expectation is over the random string of the referee and the samples drawn from μ_j.

The decision of C on j and r, denoted $C_{\mathrm{out}}(j,r)$, is the random variable $D(C_{T(j,r)}(j),r)$. C is said to be a δ-error classifier for $\mu_1,\ldots,\mu_\ell$, if for all $j \in [\ell]$, $\Pr(C_{\mathrm{out}}(j,R)=j) \geq 1-\delta$.

The transcript of a protocol at step t is the sequence $\mathbf{b}=(b_1,\ldots,b_t)$ of samples sent by Alice during the first t rounds of the protocol. Unlike the sampling scenario, the transcript here is not fully determined by Alice's input j; it is a random variable, depending on the distribution μ_j. We denote this random variable by $C_t(j)$. Like in sampling protocols, it will be convenient to think of Alice as sending an infinite sequence of samples and consider steps t exceeding the stopping time of the referee.

5.2 The reduction

In the reduction described below, we show that given any **private-coin** sampling protocol that approximates a function f and given any collection of inputs, on which f takes very "different" values, there is a classifier for the query distributions corresponding to these inputs. We start by defining what it means for two inputs to have "different" f-values:

Definition 5.1 (Disjoint inputs). Let $f:\mathcal{X}^n \to \mathcal{Y}$ be a function with approximation notion $A_{f,\epsilon}$. Two inputs $\mathbf{x},\mathbf{x}' \in \mathcal{X}^n$ are called ϵ-disjoint, if $A_{f,\epsilon}(\mathbf{x}) \cap A_{f,\epsilon}(\mathbf{x}') = \emptyset$.

Theorem 5.2. *Let Δ be a private-coin sampling protocol that (ϵ,δ)-approximates a function $f:\mathcal{X}^n \to \mathcal{Y}$. Let q be the worst-case query cost of Δ and let q_E be its expected query cost. Let $\mathcal{S}=\{\mathbf{x}_1,\ldots,\mathbf{x}_\ell\}$ be any set of inputs to f, each two of which are ϵ-disjoint. Then,*

1. *There is a δ-error sequential classifier C for the index-answer distributions $\psi_{\mathbf{x}_1},\ldots,\psi_{\mathbf{x}_\ell}$, whose worst-case sample cost is at most q and expected sample cost is at most q_E.*
2. *If Δ is index-oblivious, there is a δ-error sequential classifier C for the answer distributions $\alpha_{\mathbf{x}_1},\ldots,\alpha_{\mathbf{x}_\ell}$, whose worst-case sample cost is at most q and expected sample cost is at most q_E.*

Proof. We start by describing the classifier for the index-answer distributions.

Informally, the classifier C, given an index-answer distribution $\psi_{\mathbf{x}_j}$, simulates the sampling protocol Δ on $\mathbf{x}_j$. The role of Alice and Bob in the protocol Δ is simulated by Alice alone in C: Alice produces samples from the index-answer distribution $\psi_{\mathbf{x}_j}$; this distribution is identical by definition to the distribution of index-answer pairs Alice and Bob produce in Δ when given the input $\mathbf{x}_j$. Since the random string used by the referee of Δ is independent of the random string used by Alice (recall that Δ is a private-coin protocol), the referee of C can simulate the referee of Δ.

In the end of the simulation, Δ gives some output. C decides that the input distribution is $\psi_{\mathbf{x}_j}$ if the output of Δ belongs to $A_{f,\epsilon}(\mathbf{x}_j)$. If the output of Δ does not belong to any of the sets $A_{f,\epsilon}(\mathbf{x}_1),\ldots,A_{f,\epsilon}(\mathbf{x}_\ell)$, C outputs an arbitrary decision (e.g., "1"). Note that C's decision is well defined, since the sets $A_{f,\epsilon}(\mathbf{x}_1), \ldots, A_{f,\epsilon}(\mathbf{x}_\ell)$ are pairwise disjoint. Since Δ produces a value in $A_{f,\epsilon}(\mathbf{x}_j)$ with probability at least $1-\delta$, the classifier makes the right decision with the same probability.

The formal proof of the above is given next. We denote the operators corresponding to the classifier C with "hats" and the ones corresponding to Δ without a hat.

The referee of C is given a random string R, which has the same distribution as the random string of the referee of Δ. The decision function of the referee is defined as follows. For all r, if $D(\mathbf{i}, \mathbf{a}, r) = \textsc{cont}$, then $\hat{D}(\mathbf{i}, \mathbf{a}, r) = \textsc{cont}$. If $D(\mathbf{i}, \mathbf{a}, r) = y$ for some $y \in \mathcal{Y}$, then $\hat{D}(\mathbf{i}, \mathbf{a}, r) = j$, where j is the (only) index satisfying $y \in A_{f,\epsilon}(\mathbf{x}_j)$. If no such j exists, then $\hat{D}(\mathbf{i}, \mathbf{a}, r) = 1$.

We observe that the transcript of C on j and the transcript of Δ on $\mathbf{x}_j$ are identically distributed. Let R' denote the random string given to Alice in the protocol Δ. By definition, for any step t, the distribution of the random variable $\Delta_t(\mathbf{x}_j, R')$ corresponding to the transcript of Δ on $(\mathbf{x}_j, R')$ at step t is distributed according to $\psi_{\mathbf{x}_j,t}$—the t-wise distribution induced by $\psi_{\mathbf{x}_j}$ (recall the definition from Section 2.5). However, $\psi_{\mathbf{x}_j,t}$ is also the distribution of the random variable $C_t(j)$, corresponding to the transcript of C on input j at step t. Thus, $\Delta_t(\mathbf{x}_j, R')$ and $C_t(j)$ have the same distribution.

We use this observation to prove that C errs with probability at most δ on any given input j. The error probability of C on j is:

$$\Pr(C_{\text{out}}(j) \neq j) \;=\; \Pr(\hat{D}(C_{\hat{T}(j,R)}(j), R) \neq j). \tag{16}$$

Since the referee of C does not output j only if the referee of Δ does not output a value in $A_{f,\epsilon}(\mathbf{x}_j)$ on the given transcript, we can bound the RHS of (16) as follows:

$$\Pr(\hat{D}(C_{\hat{T}(j,R)}(j), R) \neq j) \;\leq\; \Pr(D(C_{\hat{T}(j,R)}(j), R) \notin A_{f,\epsilon}(\mathbf{x}_j)). \tag{17}$$

We expand the RHS of (17) by considering all the possible values of the stopping time:

$$\Pr(D(C_{\hat{T}(j,R)}(j), R) \notin A_{f,\epsilon}(\mathbf{x}_j)) \;=\; \sum_{t=0}^{\infty} \Pr(D(C_t(j), R) \notin A_{f,\epsilon}(\mathbf{x}_j) \;\wedge\; \hat{T}(j,R) = t). \tag{18}$$

The event "$\hat{T}(j,R) = t$" can be rewritten as the disjunction of the $t+1$ following events: "$\hat{D}(C_\ell(j), R) = \textsc{cont}$", for $\ell = 0, \ldots, t-1$, and "$\hat{D}(C_t(j), R) \neq \textsc{cont}$". By the definition of $\hat{D}$, these are identical to the events "$D(C_\ell(j), R) = \textsc{cont}$", for $\ell = 0, \ldots, t-1$, and "$D(C_t(j), R) \neq \textsc{cont}$". We can thus rewrite the RHS of (18) as follows:

$$\sum_{t=0}^{\infty} \Pr(\bigwedge_{\ell=0}^{t-1} D(C_\ell(j), R) = \textsc{cont} \;\wedge\; D(C_t(j), R) \notin A_{f,\epsilon}(\mathbf{x}_j) \cup \{\textsc{cont}\}). \tag{19}$$

Let us denote by P_t the predicate that corresponds to the event at the t-th summand of the above sum. Note that P_t is a function only of $C_t(j)$ and R (because $C_\ell(j)$ is a prefix of $C_t(j)$ for $\ell < t$). By our observation above the random variables $C_t(j)$ and $\Delta_t(\mathbf{x}_j, R')$

are identically distributed. Since R is independent of both $C_t(j)$ and $\Delta_t(\mathbf{x}_j, R')$ (recall that Δ is a private-coin protocol), then also the random pairs $(C_t(j), R)$ and $(\Delta_t(\mathbf{x}_j, R'), R)$ are identically distributed. It follows that the event "$P_t(C_t(j), R)$" has the same probability as the event "$P_t(\Delta_t(\mathbf{x}_j, R'), R)$. We can thus rewrite (19) as:

$$\sum_{t=0}^{\infty} \Pr(\bigwedge_{\ell=0}^{t-1} D(\Delta_\ell(\mathbf{x}_j, R'), R) = \text{CONT} \ \wedge \ D(\Delta_t(\mathbf{x}_j, R'), R) \notin A_{f,\epsilon}(\mathbf{x}_j) \cup \{\text{CONT}\}). \tag{20}$$

But now, by similar derivations to the ones we did above for C, the expression (20) is exactly the error probability of Δ on $\mathbf{x}_j$. This probability at most δ, due to the correctness of Δ. We conclude that the classifier C also errs with at most that much probability.

We now turn to the analysis of the sample cost. For any value of his random string r, the referee of C stops on any given transcript exactly at the same step as the referee of Δ would have stopped given r and the same transcript. Since the latter always stops within q steps, then also the referee of C always stops within q steps. We conclude that the worst-case sample cost of C is at most q.

To show that C has expected sample cost of at most q_E, we prove that the random variables corresponding to the stopping times of C on input j and Δ on input $\mathbf{x}_j$ are identically distributed. It would then follow that in particular their expectations are the same.

Fix any step t. The event that C stops on j at step t ("$\hat{T}(j, R) = t$") can be written as some predicate Q_t which depends only on $C_t(j)$ and on R. Q_t is the disjunction of the $t+1$ events $D(C_\ell(j), R) = \text{CONT}$, for $\ell = 0, \ldots, t-1$, and $D(C_t(j), R) \neq \text{CONT}$. Since $(C_t(j), R)$ has the same distribution as $(\Delta_t(\mathbf{x}_j, R'), R)$, then the event "$Q_t(C_t(j), R)$" has the same probability as the event "$Q_t(\Delta_t(\mathbf{x}_j, R'), R)$". But now it is easy to verify that this event is exactly the event that Δ stops on $\mathbf{x}_j$ at step t ("$T(\mathbf{x}_j, R', R) = t$"). Since this holds for all t, we conclude that the random variables $\hat{T}(j, R)$ and $T(\mathbf{x}_j, R', R)$ are indeed identically distributed. This completes the proof of the first part of the theorem.

The proof of the second part is almost identical to the proof of the first part. Since in this case the protocol Δ is index-oblivious, the decision function of the referee takes only the answer transcript and the random string as inputs (and not the index transcript). This implies that the decision function of the simulating classifier C can also take only an answer transcript and a random string as inputs. Thus, the resulting classifier works for the answer distributions and not just for the index-answer distributions. $\square$

6 Classification lower bound

In this section we present a new lower bound on the sample complexity of statistical classification in terms of the Jensen-Shannon divergence. We mention the implication to query complexity lower bounds of functions in the end of the section.

The lower bound is proved in two steps: first we show a lower bound on the error probability of the classifier (a.k.a. the "misclassification error") in terms of the Jensen-Shannon divergence among the distributions that are being classified. We then prove a decomposition property of the Jensen-Shannon divergence among i.i.d. distributions. The two steps together yield a lower bound on the sample cost of classification of i.i.d. distributions.

The following theorem gives a lower bound on the misclassification error of a classifier that uses q samples to classify sequential distributions $\mu_1, \ldots, \mu_\ell$ in terms of the Jensen-Shannon divergence among $\mu_{1,q}, \ldots, \mu_{\ell,q}$:

Theorem 6.1 (Misclassification lower bound). *Let λ be any distribution on $[\ell]$. The classification error δ of any classifier C of sample cost q for sequential distributions $\mu_1, \ldots, \mu_\ell$ on $\mathcal{B}$ satisfies:*

$$H_2(\delta) + \delta \log(\ell - 1) \geq H(\lambda) - JS_\lambda(\mu_{1,q}, \ldots, \mu_{\ell,q}).$$

Bounding $H_2(\delta)$ by $2\delta \log(1/\delta)$ gives an explicit bound:

$$\delta \;\geq\; \Omega\left(\frac{1}{\log \ell \cdot \log \log \ell} \cdot (H(\lambda) - JS_\lambda(\mu_{1,q}, \ldots, \mu_{\ell,q}))\right).$$

This is an exponential improvement (in terms of ℓ) over the bound of Lin [Lin91], who showed $\delta \geq \frac{1}{4(\ell-1)}(H(\lambda) - JS_\lambda(\mu_{1,q}, \ldots, \mu_{\ell,q}))^2$.

Proof. The proof is based on Fano's inequality (cf. [CT91]) from information theory, which gives a lower bound on the error probability of predicting the value of a random variable X from the observation of another random variable Y:

Theorem 6.2 (Fano's inequality [Fan52]). *Let X and Y be two random variables on domains $\mathcal{X}$ and $\mathcal{Y}$ respectively. Let $g : \mathcal{Y} \to \mathcal{X}$ be a prediction function, and let $\delta = \Pr(g(Y) \neq X)$ be the prediction error probability. Then, $H_2(\delta) + \delta \log(|\mathcal{X}| - 1) \;\geq\; H(X \mid Y)$.*

Consider now the classifier C that uses at most q samples to classify $\mu_1, \ldots, \mu_\ell$. For each $d = 1, \ldots, \ell$, let $B_d \sim \mu_{d,q}$ denote the random variable corresponding to the first q samples drawn from μ_d.

If we select an index $d \in [\ell]$ according to λ and run the classifier with μ_d as input, we can recover d with probability at least $1 - \delta$. Therefore, if $D \sim \lambda$ is the random variable corresponding to the choice of d, the decision function of the referee in the classifier C induces a prediction function $g_C : \mathcal{B}^q \times \mathcal{R} \to [\ell]$, which predicts the value of D based on q samples drawn from μ_D and based on the random string of the referee. Thus, the prediction error, $\Pr(g_C(B_D, R) \neq D)$, is at most δ. Here, B_D denotes the random variable $\pi(D, B_1, \ldots, B_\ell)$, where $\pi(d, \mathbf{b}_1, \ldots, \mathbf{b}_\ell) = \mathbf{b}_d$. R is the random variable corresponding to the random string used by the classifier's referee. Using Fano's inequality, we have the following lower bound on δ:

$$H_2(\delta) + \delta \log(\ell - 1) \geq H(D \mid g_C(B_D, R)).$$

We now expand the conditional entropy, using the information theory properties mentioned in Proposition 2.3. By the definition of mutual information,

$$H(D \mid g_C(B_D, R)) = H(D) - I(D;\; g_C(B_D, R)).$$

By the data processing inequality,

$$I(D;\ g_C(B_D, R)) \le I(D;\ B_D, R).$$

By the chain rule for mutual information,

$$I(D;\ B_D, R) = I(D;\ R) + I(D;\ B_D \mid R).$$

Since D and B_D are jointly independent of R, then $I(D;\ R) = 0$ and $I(D;\ B_D \mid R) = I(D;\ B_D)$. We conclude:

$$H_2(\delta) + \delta \log(\ell - 1) \ \ge\ H(D) - I(D;\ B_D).$$

We complete the proof by noting that $I(D;\ B_D) = JS_\lambda(\mu_{1,q}, \ldots, \mu_{\ell,q})$ (Proposition 2.4). $\square$

Proposition 6.3 (Decomposition of Jensen-Shannon). *Let $\mu_{1,q} = \nu_1^q, \ldots, \mu_{\ell,q} = \nu_\ell^q$ be i.i.d. distributions on $\mathcal{B}^q$. Then,*

$$JS_\lambda(\mu_{1,q}, \ldots, \mu_{\ell,q}) \ \le\ q \cdot JS_\lambda(\nu_1, \ldots, \nu_\ell).$$

Proof. Let $B_j \sim \mu_{j,q}$ and $X_j \sim \nu_j$. Since $\mu_{j,q} = \nu_j^q$, $B_j = X_j^q$; that is, B_j consists of q independent copies of X_j. Also, let $D \sim \lambda$, $B_D = \pi(D, B_1, \ldots, B_\ell)$, and $X_D = \pi(D, X_1, \ldots, X_\ell)$, where $\pi(d, x_1, \ldots, x_\ell) = x_d$. From Proposition 2.4 we have that $JS_\lambda(\mu_{1,q}, \ldots, \mu_{\ell,q}) = I(D;\ B_D)$ and $JS_\lambda(\nu_1, \ldots, \nu_\ell) = I(D;\ X_D)$,

By the definition of mutual information, $I(D;\ B_D) = H(B_D) - H(B_D \mid D)$. By the subadditivity of entropy, $H(B_D) \le q \cdot H(X_D)$. Note that conditioned on $D = d$, $B_D = X_d^q$. Therefore,

$$H(B_D \mid D) \ =\ \sum_d \lambda(d) H(X_d^q) \ =\ \sum_d \lambda(d) \cdot q \cdot H(X_d) \ =\ q \cdot H(X_D \mid D).$$

The next to the last equality follows from the fact the joint entropy of independent random variables is the sum of their entropies. We thus have:

$$I(D;\ B_D) \ \le\ q \cdot H(X_D) - q \cdot H(X_D \mid D) \ =\ q \cdot I(D;\ X_D).$$

$\square$

We can now immediately derive the sample complexity lower bound for classification:

Theorem 6.4. *Let $\mu_1, \ldots, \mu_\ell$ be sequential i.i.d. distributions on $\mathcal{B}$ with base distributions $\nu_1, \ldots, \nu_\ell$, respectively. Let λ be any distribution on $[\ell]$. Then, the worst-case sample cost q of any δ-error classifier C for $\mu_1, \ldots, \mu_\ell$ satisfies:*

$$q \ \ge\ \frac{1}{JS_\lambda(\nu_1, \ldots, \nu_\ell)} \left(H(\lambda) - \delta \log(\ell - 1) - H_2(\delta) \right).$$

Choosing λ to be uniform on $[\ell]$ gives: $q \geq \Omega(\frac{\log \ell}{JS_\lambda(\nu_1,\ldots,\nu_\ell)} \cdot (1-\delta))$.

Using the reduction from classification to sampling, we obtain the following query complexity lower bound:

Theorem 6.5 (Main theorem). *Let Δ be any private-coin sampling algorithm that (ϵ,δ)-approximates a function $f : \mathcal{X}^n \to \mathcal{Y}$. Let q be the worst-case query cost of Δ. Let $\mathcal{S} = \{\mathbf{x}_1,\ldots,\mathbf{x}_\ell\}$ be any set of pairwise ϵ-disjoint inputs, such that the index-answer distributions of Δ on $\mathbf{x}_1,\ldots,\mathbf{x}_\ell$ are all i.i.d. Let $\nu_{\mathbf{x}_1},\ldots,\nu_{\mathbf{x}_\ell}$ be the base distributions of these index-answer distributions. Let λ be any distribution on $\mathcal{S}$. Then,*

$$q \;\geq\; \frac{1}{JS_\lambda(\nu_{\mathbf{x}_1},\ldots,\nu_{\mathbf{x}_\ell})}\left(H(\lambda) - \delta\log(\ell-1) - H_2(\delta)\right).$$

If Δ is index-oblivious, an analogous lower bound holds with the answer distributions replacing the index-answer distributions.

7 Applications

All the proofs appearing in this section share the same basic framework. Therefore, we start the section with a "recipe" (or a "meta-proof"), which shows how to use our main theorem to derive specific lower bounds for specific functions. In the proofs of the actual applications we thus provide only the components that are unique to each applications.

7.1 The lower bound recipe

We next describe the recipe for obtaining query complexity lower bounds for specific functions. The recipe described below is applicable to (g,ϵ)-symmetric functions $f : \mathcal{X}^n \to \mathcal{Y}$, for which g depends on both of its arguments. Slight modifications (mentioned below) are necessary when either g depends only on its first argument, or when f is a (g,ϵ) row/column symmetric function over matrices.

Our goal is to show a lower bound on the worst-case query cost q of any sampling algorithm Δ that (ϵ,δ)-approximates f.

We first argue that, WLOG, $q \leq \frac{n}{2} - \sqrt{\frac{n}{2\delta}}$. Because, if not, then $q \geq \Omega(n)$, which is the best possible up to constant factors (there is always an algorithm that computes the function exactly with n queries).

1. Prove that f is (g,ϵ)-symmetric. It then follows from the second part of Theorem 4.2 that there exists a private-coin sampling algorithm $\hat{\Delta}$ of worst-case query cost $2q$ that $(\epsilon,2\delta)$-approximates f and whose index distribution on all inputs is uniform with replacement. Since this index distribution is i.i.d., then so is the index-answer distribution of $\hat{\Delta}$. We denote by $\nu_{\mathbf{x}}$ the base of the index-answer distribution of $\hat{\Delta}$ on input $\mathbf{x}$.

2. Construct a family of pairwise disjoint inputs. We have to come up with some collection of inputs $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$, such that for all $j \neq j'$, $\mathbf{x}_j$ and $\mathbf{x}_{j'}$ are ϵ-disjoint. The choice of $\mathcal{S}$ depends on the particular f at hand.

3. Bound the dissimilarity among inputs in $\mathcal{S}$. We need to pick some distribution λ on $[\ell]$ and prove an upper bound of γ on $JS_\lambda(\nu_{\mathbf{x}_1}, \ldots, \nu_{\mathbf{x}_\ell})$.

It now immediately follows from Theorem 6.5 that $2q \geq \frac{1}{\gamma} \cdot (H(\lambda) - 2\delta \log(\ell - 1) - H_2(2\delta))$, which gives us the lower bound on q.

When g depends only on its first argument When g depends only on its first argument (e.g., when f is ϵ-symmetric) and when the lower bound we are shooting for is at most $O(\sqrt{n})$, we can get an improved lower bound by using the answer distributions instead of the index-answer distributions.

Formally, we first argue that, WLOG, $q \leq \sqrt{2\delta n}$. Because if not, then $q \geq \Omega(\sqrt{n})$, which is better than the lower bound we are aiming at. We can then apply the first part of Theorem 4.2 and conclude that there is a private-coin, *index-oblivious*, sampling algorithm $\hat{\Delta}$ of worst-case query cost q that $(\epsilon, 2\delta)$-approximates f and whose index distribution on all inputs is uniform with replacement. Since the index distribution of $\hat{\Delta}$ is i.i.d, then so is its answer distribution. We denote now by $\nu_{\mathbf{x}}$ the base of the answer distribution of $\hat{\Delta}$ on $\mathbf{x}$.

We next construct a set $\mathcal{S}$ of pairwise ϵ-disjoint inputs as before and show $JS_\lambda(\nu_{\mathbf{x}_1}, \ldots, \nu_{\mathbf{x}_\ell})$ is at most γ, for some choice of λ (note that now we bound the dissimilarity among the answer distributions and not among the index-answer distributions). Finally, we apply Theorem 6.5. Note that since $\hat{\Delta}$ is index-oblivious, we can apply the theorem with the answer distributions rather than the index-answer distributions.

When f is row/column symmetric . When $f : M_{m \times n}(\mathcal{X}) \to \mathcal{Y}$ is a (g, ϵ) row-symmetric function over matrices, we think of f as a function of the form $f : (\mathcal{X}^n)^m \to \mathcal{Y}$ and apply the recipe described above. What we get is a lower bound on the query cost of *row-querying* sampling algorithms that (ϵ, δ)-approximate f. But since any entry-querying algorithm for f of worst-case query cost q can be simulated by a row-querying algorithm of worst-case query cost at most q, the same lower bound holds for entry-querying algorithms. An equivalent arguments holds for column-symmetric functions.

Specialized recipe for our applications The above recipe is general and works for any (g, ϵ) (row / column) symmetric function. It turns out, however, that in all of the applications we consider in this paper the particular ways in which we construct the family of disjoint inputs $\mathcal{S}$ and bound the dissimilarity among its inputs are almost the same. We thus next elaborate on Steps 2 and 3 of the recipe above. We caution that this part of the recipe is not necessarily applicable to any (g, ϵ) (row/column) symmetric function.

2.1. Choose an "abundance parameter" and a "similarity parameter". We pick an integer t and a real number $\gamma > 0$. The aimed lower bound will be $\Omega(\frac{t}{\gamma} \cdot (1 - 2\delta))$.

2.2. Find an appropriate mapping from subsets of $[t]$ to inputs of f. We find a mapping (which depends on the particular f) from subsets of $[t]$ of size $t/2$ to inputs of f that satisfies two properties:

1. **Disjointness:** For any two subsets $\mathcal{F}, \mathcal{F}'$ whose intersection size is at most $\frac{11}{24}t$, the corresponding inputs $\mathbf{x}, \mathbf{x}'$ are ϵ-disjoint.

2. **Entropy invariance:** For any two subsets $\mathcal{F}, \mathcal{F}'$, $H(\nu_{\mathbf{x}}) = H(\nu_{\mathbf{x}'})$, where $\mathbf{x}, \mathbf{x}'$ are the inputs corresponding to $\mathcal{F}, \mathcal{F}'$.

Using such a mapping we can construct a large set of pairwise ϵ-disjoint inputs as follows. Let $\mathcal{F}_1, \ldots, \mathcal{F}_\ell$ be a $(\frac{t}{2}, \frac{11}{24}t)$-design of size $\ell = 2^{\Omega(t)}$ (Proposition 2.5). The set of pairwise ϵ-disjoint inputs is the family $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$ of inputs corresponding to $\mathcal{F}_1, \ldots, \mathcal{F}_\ell$.

We next show how to prove $JS_\lambda(\nu_{\mathbf{x}_1}, \ldots, \nu_{\mathbf{x}_\ell}) \le \gamma$, for λ which is the uniform distribution on $[\ell]$. We prove that $JS_\lambda(\nu_{\mathbf{x}_1}, \ldots, \nu_{\mathbf{x}_\ell}) \le JS_\lambda(\nu_{\mathbf{y}_1}, \ldots, \nu_{\mathbf{y}_{\ell^*}})$, and show the latter is at most γ. $\mathbf{y}_1, \ldots, \mathbf{y}_{\ell^*}$ are the inputs corresponding to the family of all the $\ell^* = \binom{t}{t/2}$ subsets of $[t]$ of size $t/2$. λ^* is the uniform distribution on $[\ell^*]$. We thus need to prove that:

Claim 7.1. *$JS_\lambda(\nu_{\mathbf{x}_1}, \ldots, \nu_{\mathbf{x}_\ell}) \le JS_{\lambda^*}(\nu_{\mathbf{y}_1}, \ldots, \nu_{\mathbf{y}_{\ell^*}})$.*

In order to prove the claim, we need to verify the following fact. The proof of this fact depends on the particular f we are dealing with. Let $\nu_{\mathbf{x}_\lambda}$ and $\nu_{\mathbf{y}_{\lambda^*}}$ denote, respectively, the λ-weighted and λ^*-weighted average distributions of $\nu_{\mathbf{x}_1}, \ldots, \nu_{\mathbf{x}_\ell}$ and $\nu_{\mathbf{y}_1}, \ldots, \nu_{\mathbf{y}_{\ell^*}}$. Then,

3.1. Show that $H(\nu_{\mathbf{y}_{\lambda^*}}) \ge H(\nu_{\mathbf{x}_\lambda})$.

Proof (of Claim 7.1). Let $D \sim \lambda$, $D^* \sim \lambda^*$, $X_D \sim \nu_{\mathbf{x}_\lambda}$, and $Y_{D^*} \sim \nu_{\mathbf{y}_{\lambda^*}}$.

By Proposition 2.4, $JS_\lambda(\nu_{\mathbf{x}_1}, \ldots, \nu_{\mathbf{x}_\ell}) = I(D;\, X_D)$ and $JS_{\lambda^*}(\nu_{\mathbf{y}_1}, \ldots, \nu_{\mathbf{y}_{\ell^*}}) = I(D^*;\, Y_{D^*})$. Using the fact given in Step 3.1, $H(X_D) = H(\nu_{\mathbf{x}_\lambda}) \le H(\nu_{\mathbf{y}_{\lambda^*}}) = H(Y_{D^*})$. Therefore, $I(D;\, X_D) = H(X_D) - H(X_D \mid D) \le H(Y_{D^*}) - H(X_D \mid D)$. By the definition of conditional entropy and the fact λ is independent of $\nu_{\mathbf{x}_j}$ for all j, $H(X_D \mid D) = \frac{1}{\ell}\sum_{j \in [\ell]} H(\nu_{\mathbf{x}_j})$. Similarly, $H(Y_{D^*} \mid D^*) = \frac{1}{\ell^*}\sum_{j \in [\ell^*]} H(\nu_{\mathbf{y}_j})$. Using the entropy invariance property, all the entropies appearing in both sums are identical. Therefore, $H(X_D \mid D) = H(Y_{D^*} \mid D^*)$. Thus, $I(D;\, X_D) \le H(Y_{D^*}) - H(Y_{D^*} \mid D^*) = I(D^*;\, Y_{D^*})$. □

To obtain the dissimilarity upper bound, we need to bound $JS_{\lambda^*}(\nu_{\mathbf{y}_1}, \ldots, \nu_{\mathbf{y}_{\ell^*}})$:

3.2. Show that $JS_{\lambda^*}(\nu_{\mathbf{y}_1}, \ldots, \nu_{\mathbf{y}_{\ell^*}}) \le \gamma$.

It follows that also $JS_\lambda(\nu_{\mathbf{x}_1}, \ldots, \nu_{\mathbf{x}_\ell}) \le \gamma$, and therefore, by Theorem 6.5, $2q \ge \frac{1}{\gamma} \cdot (\log \ell \cdot (1 - 2\delta) - H_2(2\delta)) = \Omega(\frac{t}{\gamma} \cdot (1 - 2\delta))$.

Remark: When we apply the above recipe to functions over matrices, we denote the input matrices corresponding to the design by $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ and the inputs corresponding to the family of all subsets of $[t]$ of size $t/2$ by $\mathbf{B}_1, \ldots, \mathbf{B}_{\ell^*}$.

7.2 The election problem

The input for the election problem, EP, is a sequence $\mathbf{x} \in [m]^n$ of n votes to m parties. The vote frequency of party $i \in [m]$, $v_i(\mathbf{x})$, is the number of votes party i gets in $\mathbf{x}$. EP($\mathbf{x}$) is the vote distribution $V_{\mathbf{x}} \stackrel{\text{def}}{=} (v_1(\mathbf{x})/n, \ldots, v_m(\mathbf{x})/n)$. We consider additive approximation w.r.t. statistical distance (i.e., half of the L_1 distance). That is, $\mu \in A_{\mathrm{EP},\epsilon}(\mathbf{x})$, iff $||\mu - V_{\mathbf{x}}|| < \epsilon$.

Theorem 7.2. *Let $0 < \epsilon < \frac{1}{96}$, $0 < \delta < \frac{1}{2}$, and $n \geq \Omega(m^2/\epsilon^4 \cdot 1/\delta)$. The worst-case query cost of any sampling algorithm Δ that (ϵ, δ)-approximates* EP *is at least $\Omega(m/\epsilon^2 \cdot (1-2\delta))$.*

Proof. We use the lower bound recipe described in Section 7.1.

1. Symmetry. It is easy to verify that EP is an ϵ-symmetric function. In other words it is (g, ϵ)-symmetric for the function $g(y, \pi) = y$, which depends only on its first argument. Moreover, by the restriction on n the lower bound we are shooting for is $O(\sqrt{n})$. We can thus use the distribution $\nu_{\mathbf{x}}$ on $[m]$, which is the base of the answer distribution induced by uniform queries with replacement to $\mathbf{x}$. Observe that $\nu_{\mathbf{x}}$ happens to be exactly the same as $V_{\mathbf{x}}$ – the vote distribution corresponding to $\mathbf{x}$.

2.1. Abundance and similarity parameters. We choose $t = m$ and $\gamma = O(1/\epsilon^2)$.

2.2. Mapping subsets of $[m]$ to inputs of EP. Given a subset $\mathcal{F} \subseteq [m]$ of size $m/2$, we associate with it any input $\mathbf{x}$ (say, the first in some order on $\mathcal{X}^n$), whose vote distribution is the following: $V_{\mathbf{x}}(i) = \frac{1+24\epsilon}{m}$ for $i \in \mathcal{F}$ and $V_{\mathbf{x}}(i) = \frac{1-24\epsilon}{m}$ for $i \notin \mathcal{F}$. If we think of $\mathcal{F}$ as a collection of half of the parties, $\mathbf{x}$ is some population in which the parties in $\mathcal{F}$ collectively get $\frac{1}{2} + 12\epsilon$ of the votes and the parties not in $\mathcal{F}$ get $\frac{1}{2} - 12\epsilon$ of the votes. The votes to the parties in $\mathcal{F}$ are distributed evenly among them and the votes to the parties not in $\mathcal{F}$ are distributed evenly among them. We need to show that this mapping satisfies the disjointness and entropy invariance properties.

Disjointness. Let $\mathcal{F}, \mathcal{F}' \subseteq [m]$ be two subsets of size $m/2$, such that $|\mathcal{F} \cap \mathcal{F}'| \leq \frac{11}{24}m$. Let $\mathbf{x}, \mathbf{x}'$ be the corresponding inputs. Then, $||V_{\mathbf{x}} - V_{\mathbf{x}'}|| \geq V_{\mathbf{x}}(\mathcal{F} \setminus \mathcal{F}') - V_{\mathbf{x}'}(\mathcal{F} \setminus \mathcal{F}') \geq |\mathcal{F} \setminus \mathcal{F}'| \cdot (\frac{1+24\epsilon}{m} - \frac{1-24\epsilon}{m}) = \frac{m}{24} \cdot \frac{48\epsilon}{m} = 2\epsilon$. Since the statistical distance respects the triangle inequality, it follows that $\mathbf{x}$ and $\mathbf{x}'$ are ϵ-disjoint.

Entropy invariance. Recall that the answer distribution $\nu_{\mathbf{x}}$ is exactly $V_{\mathbf{x}}$. By definition, for any input $\mathbf{x}$ induced by a subset $\mathcal{F} \subseteq [m]$ of size $m/2$, the entropy of $V_{\mathbf{x}}$ depends only on the size of $\mathcal{F}$ and not on its identity. Therefore, for all $\mathcal{F}, \mathcal{F}'$, $H(V_{\mathbf{x}}) = H(V_{\mathbf{x}'})$.

3.1. Prove that $H(\nu_{\mathbf{y}_{\lambda^*}}) \geq H(\nu_{\mathbf{x}_\lambda})$. $\nu_{\mathbf{x}_\lambda} = \frac{1}{\ell}\sum_{j=1}^{\ell} V_{\mathbf{x}_j}$ can be written as the following convex combination: $\nu_{\mathbf{x}_\lambda} = (1-24\epsilon) \cdot U_m + 24\epsilon \cdot F$, where U_m is the uniform distribution on $[m]$, F is the distribution of a uniformly chosen element from $\mathcal{F}_J$, and J is a uniformly chosen index of a set in the design. Formally, $F(i) = \frac{2f_i}{\ell m}$ for $i \in [m]$, where f_i is the number of sets in the design that contain i. Similarly, $\nu_{\mathbf{y}_{\lambda^*}} = \frac{1}{\ell^*}\sum_{j=1}^{\ell^*} V_{\mathbf{y}_j}$ can be written as $(1-24\epsilon) \cdot U_m + 24\epsilon \cdot F^*$, where F^* is the distribution of a uniformly chosen element from $\mathcal{F}^*_{J^*}$ and J^* is a uniformly chosen index of a set in the family of all subsets of $[m]$ of size $m/2$.

Clearly, $F^* = U_m$, and therefore $\nu_{\mathbf{y}_{\lambda^*}} = U_m$. It follows immediately that $H(\nu_{\mathbf{y}_{\lambda^*}}) \geq H(\nu_{\mathbf{x}_{\lambda^*}})$, because both distributions share the same domain ($[m]$), and $\nu_{\mathbf{y}_{\lambda^*}}$ is uniform on that domain.

3.2. Bounding the dissimilarity. We are left to prove that $JS_{\lambda^*}(\nu_{\mathbf{y}_1}, \ldots, \nu_{\mathbf{y}_{\ell^*}}) \leq O(\epsilon^2)$. By definition, $JS_{\lambda^*}(\nu_{\mathbf{y}_1}, \ldots, \nu_{\mathbf{y}_{\ell^*}}) = \frac{1}{\ell^*}\sum_{j=1}^{\ell^*} D_{KL}(\nu_{\mathbf{y}_j} \parallel \nu_{\mathbf{y}_{\lambda^*}})$. Recall that $\nu_{\mathbf{y}_{\lambda^*}} = U_m$ and that $\nu_{\mathbf{y}_j} = V_{\mathbf{y}_j}$. The bound follows from a simple calculation showing that $\forall j \in [\ell^*]$, $D_{KL}(V_{\mathbf{y}_j} \parallel U_m) \leq O(\epsilon^2)$. $\square$

7.3 Low rank matrix approximation

Theorem 7.3. *Let $k \leq \min\{\frac{m}{2}, \frac{n}{2}\}$, $0 \leq \epsilon \leq \frac{1}{\sqrt{24}}$, and $0 < \delta < \frac{1}{2}$. The worst-case query cost of any sampling algorithm Δ that (ϵ, δ)-approximates LRM_k is at least $\Omega((m+n)\cdot(1-2\delta))$.*

Proof. We prove that the worst-case query cost of Δ is at least $\Omega(m\cdot(1-2\delta))$. An analogous argument shows that it is also at least $\Omega(n\cdot(1-2\delta))$. We resort again to the recipe described in Section 7.1.

1. Symmetry. We start by proving that LRM_k is ϵ-row and ϵ-column permutation commutative. Since we are dealing with real-valued matrices, any row permutation of a matrix $\mathbf{A}$ can be written as the product $\mathbf{\Sigma A}$ of a row permutation matrix $\mathbf{\Sigma}$ and of $\mathbf{A}$. Similarly, any column permutation of $\mathbf{A}$ can be written as the product $\mathbf{A\Pi}$, where $\mathbf{\Pi}$ is a column permutation matrix.

Let $\mathbf{A}$ be any $m \times n$ matrix, let $\mathbf{\Sigma}$ be any row permutation matrix, and let $\mathbf{\Pi}$ be any column permutation matrix. Pick any $\mathbf{B}$ in the ϵ-approximation set of $\mathbf{\Sigma A \Pi}$. That is, $||\mathbf{\Sigma A\Pi} - \mathbf{B}||_F \leq ||\mathbf{\Sigma A\Pi} - (\mathbf{\Sigma A\Pi})_k||_F + \epsilon\, ||\mathbf{\Sigma A\Pi}||_F$. We need to show that $\mathbf{\Sigma}^{-1}\mathbf{B}\mathbf{\Pi}^{-1}$ is in the ϵ-approximation set of $\mathbf{A}$.

Note that for any matrix $\mathbf{M}$, any row permutation matrix $\mathbf{U}$, and any column permutation matrix $\mathbf{V}$, (1) $\mathbf{UMV}$ has the same rank as $\mathbf{M}$; (2) $||\mathbf{UMV}||_F = ||\mathbf{M}||_F$; (3) $\mathbf{U}^{-1}$ and $\mathbf{V}^{-1}$ are also permutation matrices; and (4) $(\mathbf{UMV})_k = \mathbf{U}\mathbf{M}_k\mathbf{V}$. It follows that $\mathbf{\Sigma}^{-1}\mathbf{B}\mathbf{\Pi}^{-1}$ is a rank k matrix and

$$\begin{aligned}
\left|\left|\mathbf{A} - \mathbf{\Sigma}^{-1}\mathbf{B}\mathbf{\Pi}^{-1}\right|\right|_F &= ||\mathbf{\Sigma A\Pi} - \mathbf{B}||_F \\
&\leq ||\mathbf{\Sigma A\Pi} - (\mathbf{\Sigma A\Pi})_k||_F + \epsilon\, ||\mathbf{\Sigma A\Pi}||_F \\
&= ||\mathbf{\Sigma A\Pi} - \mathbf{\Sigma A}_k\mathbf{\Pi}||_F + \epsilon\, ||\mathbf{A}||_F \\
&= ||\mathbf{A} - \mathbf{A}_k||_F + \epsilon\, ||\mathbf{A}||_F\,.
\end{aligned}$$

We can thus use the distribution $\nu_{\mathbf{A}}$ on $[m] \times \mathbb{R}^n$, which is the base of the index-answer distribution induced by uniform queries with replacement to rows of $\mathbf{A}$. Thus, $\nu_{\mathbf{A}}$ is uniform on the rows of $\mathbf{A}$.

2.1. Abundance and similarly parameters. We choose $t = 2k$ and $\gamma = 2k/m$.

2.2. Mapping subsets of $[2k]$ to inputs of LRM_k. Given a subset $\mathcal{F} \subseteq [2k]$ of size k, we map it to an input matrix $\mathbf{A}$, all of whose entries are 0, except for the diagonal of its top left $2k \times 2k$ submatrix, which consists of the characteristic vector of $\mathcal{F}$. We next show the mapping satisfies the disjointness and entropy invariance properties:

Disjointness. Let $\mathcal{F}, \mathcal{F}' \subseteq [2k]$ be two subsets of size k, such that $|\mathcal{F} \cap \mathcal{F}'| \leq \frac{11}{12}k$. Let $\mathbf{A}, \mathbf{A}'$ be the two corresponding input matrices. Note that both $\mathbf{A}$ and $\mathbf{A}'$ are of rank k, and therefore in order to be ϵ-disjoint it suffices that $||\mathbf{A} - \mathbf{A}'||_F \geq \epsilon(||\mathbf{A}||_F + ||\mathbf{A}'||_F)$. Indeed, $||\mathbf{A}||_F = ||\mathbf{A}'||_F = \sqrt{k}$, while $||\mathbf{A} - \mathbf{A}'||_F^2 = |\mathcal{F} \setminus \mathcal{F}'| + |\mathcal{F}' \setminus \mathcal{F}| \geq k/6$. The assertion follows now from the fact $\epsilon \leq \frac{1}{\sqrt{24}}$.

Entropy invariance. For any $\mathbf{A}$, $\nu_{\mathbf{A}}$ is uniform on rows of $\mathbf{A}$, and thus $H(\nu_{\mathbf{A}}) = \log m$.

3.1. Prove that $H(\nu_{\mathbf{B}_{\lambda^*}}) \geq H(\nu_{\mathbf{A}_\lambda})$. Let $\mathbf{0}, \mathbf{e}_1, \ldots, \mathbf{e}_n$ denote, respectively, the all-zero and the standard unit vectors in $\mathbb{R}^n$. Recall that the base distribution $\nu_{\mathbf{A}}$ is uniform on rows of $\mathbf{A}$. Thus, if $\mathbf{A}$ was obtained from a subset $\mathcal{F}$ by the mapping, then $\nu_{\mathbf{A}}$ is uniform on the following set of pairs: all the pairs $(i, \mathbf{e}_i)$ with $1 \leq i \leq 2k$ and $i \in \mathcal{F}$, all the pairs $(i, \mathbf{0})$ with $1 \leq i \leq 2k$ and $i \notin \mathcal{F}$, and all the pairs $(i, \mathbf{0})$ with $i > 2k$.

$\nu_{\mathbf{A}_\lambda}$ can be written as the following convex combination: $\nu_{\mathbf{A}_\lambda} = (1 - \frac{2k}{m}) \cdot U + \frac{2k}{m} \cdot F$. U is the uniform distribution on pairs of the form $(i, \mathbf{0})$, where $i > 2k$. F is a distribution on pairs of the form $(i, \mathbf{e}_i)$, $i \in [2k]$, and $(i, \mathbf{0})$, $i \in [2k]$, defined as follows: $F((i, \mathbf{e}_i)) = \frac{f_i}{2k\ell}$ and $F((i, \mathbf{0})) = \frac{\ell - f_i}{2k\ell}$, where f_i is the number of sets in the design that contain i. Similarly, $\nu_{\mathbf{B}_{\lambda^*}} = (1 - \frac{2k}{m}) \cdot U + \frac{2k}{m} \cdot F^*$, where F^* is uniform on the $4k$ pairs $(i, \mathbf{e}_i), i \in [2k], (i, \mathbf{0}), i \in [2k]$.

The following fact is standard in information theory (cf. [CT91], Chapter 2, Exercise #19): if μ is a convex combination $\alpha\mu_1 + (1 - \alpha)\mu_2$ of two distributions with disjoint supports, then $H(\mu) = \alpha H(\mu_1) + (1 - \alpha) H(\mu_2) + H_2(\alpha)$. Note that U and F have disjoint supports, and thus $H(\nu_{\mathbf{A}_\lambda}) = (1 - \frac{2k}{m}) \cdot H(U) + \frac{2k}{m} \cdot H(F) + H_2(\frac{2k}{m})$. Similarly, $H(\nu_{\mathbf{B}_{\lambda^*}}) = (1 - \frac{2k}{m}) \cdot H(U) + \frac{2k}{m} \cdot H(F^*) + H_2(\frac{2k}{m})$. $H(F^*) \geq H(F)$, because F and F^* share the same support, and F^* is uniform on this support. We conclude that $H(\nu_{\mathbf{B}_{\lambda^*}}) \geq H(\nu_{\mathbf{A}_\lambda})$.

3.2. Bounding the dissimilarity. We are left to prove that $JS_{\lambda^*}(\nu_{\mathbf{B}_1}, \ldots, \nu_{\mathbf{B}_{\ell^*}}) \leq O(k/m)$. Note that $\nu_{\mathbf{B}_{\lambda^*}}$ assigns probability $1/2m$ to all the pairs of the form $(i, \mathbf{e}_i)$, $i \in [2k]$, probability $1/2m$ to all the pairs $(i, \mathbf{0})$, $i \in [2k]$, and probability $1/m$ to all the pairs $(i, \mathbf{0})$, $i > 2k$. Therefore, $\forall j \in [\ell^*]$, $D_{KL}(\nu_{\mathbf{B}_j} \parallel \nu_{\mathbf{B}_{\lambda^*}}) = 2k\frac{1}{m} \log \frac{(1/m)}{(1/2m)} = 2k/m$. $\square$

For an input matrix $\mathbf{A}$, the weight of $\mathbf{A}^{(i)}$, the i-th row of $\mathbf{A}$, is $\left|\left|\mathbf{A}^{(i)}\right|\right|_2^2$, and its relative weight is the ratio $\left|\left|\mathbf{A}^{(i)}\right|\right|_2^2 / ||\mathbf{A}||_F^2$. We denote by $P_{\mathbf{A}}$ the distribution over rows of $\mathbf{A}$ induced by their relative weights. We similarly define the relative weight of a column and the column weight distribution $Q_{\mathbf{A}}$. We show that even if the algorithm is given the *exact* (row or column) weight distribution as advice, it still requires $\Omega(k/\epsilon^2)$ queries.

Theorem 7.4. *Let $k \leq \frac{m}{2}$, $0 < \epsilon < \frac{1}{\sqrt{48}}$, and $0 < \delta < 1$. Any private-coin sampling algorithm Δ that (ϵ, δ)-approximates LRM_k and whose queries on input $\mathbf{A}$ are in a set of rows that are chosen independently according to $P_{\mathbf{A}}$ has worst-case query cost of at least $\Omega(\min\{k/\epsilon^2, m\} \cdot (1 - \delta))$.*

An analogous statement and proof hold for algorithms whose queries are in a set of columns that are chosen independently according to $Q_{\mathbf{A}}$.

Proof. Since the queries of Δ are in a set of rows that are chosen independently according to $P_{\mathbf{A}}$, we can simulate it by a private-coin, row-querying, sampling algorithm $\hat{\Delta}$ of worst-case query cost at most q whose index distribution on any input matrix $\mathbf{A}$ is i.i.d. with base distribution $P_{\mathbf{A}}$. It thus suffices to prove a lower bound on the worst-case query cost of $\hat{\Delta}$.

We use the recipe of Section 7.1. In this case we can skip the step of proving the function is row-symmetric, because the theorem itself considers only algorithms whose index distribution on a given input matrix $\mathbf{A}$ is fixed and i.i.d. The base distribution of this index distribution is $P_{\mathbf{A}}$. The index-answer distribution of $\hat{\Delta}$ on $\mathbf{A}$ is also i.i.d., with a base distribution on $[m] \times \mathbb{R}^n$, which we denote by $\nu_{\mathbf{A}}$.

2.1. Abundance and similarity parameters. We choose $t = 2(k-1)$ and $\gamma = O(\max\{\epsilon^2, k/m\})$.

2.2. Mapping subsets of $[2(k-1)]$ to inputs of LRM_k. Let $r \stackrel{\text{def}}{=} \max\{48\epsilon^2 m, 2(k-1)\}$. For simplicity, we assume $2(k-1)$ divides r. Given a subset $\mathcal{F} \subseteq [2(k-1)]$ of size $k-1$, we map it to an input matrix $\mathbf{A}$, all of whose entries are 0, except for the following: (1) the $2k$-th column of $\mathbf{A}$ starts with r 0's and ends with $m-r$ 1's; (2) consider the top left $r \times 2(k-1)$ submatrix of $\mathbf{A}$ and divide it vertically into $r/2(k-1)$ $2(k-1) \times 2(k-1)$ submatrices. The diagonals of these submatrices are the characteristic vector of $\mathcal{F}$.

Disjointness. Let $\mathcal{F}, \mathcal{F}' \subseteq [2(k-1)]$ be two subsets of size $k-1$, such that $|\mathcal{F} \cap \mathcal{F}'| \leq \frac{11}{12}(k-1)$. Let $\mathbf{A}, \mathbf{A}'$ be the two corresponding input matrices. Note that $\mathbf{A}$ and $\mathbf{A}'$ are of rank k; thus, it suffices to prove that $||\mathbf{A} - \mathbf{A}'||_F \geq \epsilon\,(||\mathbf{A}||_F + ||\mathbf{A}'||_F)$: $||\mathbf{A} - \mathbf{A}'||_F^2 = \frac{r}{2(k-1)} \cdot (|\mathcal{F} \setminus \mathcal{F}'| + |\mathcal{F}' \setminus \mathcal{F}|) \geq \frac{r}{2(k-1)} \cdot \frac{k-1}{6} = \frac{r}{12} \geq 4\epsilon^2 m$, while $||\mathbf{A}||_F = ||\mathbf{A}'||_F \leq \sqrt{m}$.

Entropy invariance. All the matrices obtained by the mapping have $m - r/2$ rows of weight 1 and the rest are of weight 0. Thus, for all such $\mathbf{A}$, $H(\nu_{\mathbf{A}}) = \log(m - r/2)$.

3.1. Prove that $H(\nu_{\mathbf{B}_{\lambda^*}}) \geq H(\nu_{\mathbf{A}_\lambda})$. Let $\mathbf{e}_1, \ldots, \mathbf{e}_n$ denote the standard unit vectors in $\mathbb{R}^n$. Let $s \stackrel{\text{def}}{=} m - r/2$. $\nu_{\mathbf{A}_\lambda}$ can be written as the following convex combination: $\nu_{\mathbf{A}_\lambda} = (1 - \frac{r}{2s}) \cdot U + \frac{r}{2s} \cdot F$. U is the uniform distribution on pairs of the form $(i, \mathbf{e}_{2k})$, where $i > r$. F is a distribution on pairs of the form $(i + 2d(k-1), \mathbf{e}_i)$, $i \in [2(k-1)]$, $0 \leq d < r/(2(k-1))$, defined as follows: $F((i + 2d(k-1), \mathbf{e}_i)) = \frac{2f_i}{r\ell}$, where f_i is the number of sets in the design that contain i. Similarly, $\nu_{\mathbf{B}_{\lambda^*}} = (1 - \frac{r}{2s}) \cdot U + \frac{r}{2s} \cdot F^*$, where F^* is uniform on the r pairs $(i + 2d(k-1), \mathbf{e}_i), i \in [2(k-1)], 0 \leq d < r/(2(k-1))$.

$\nu_{\mathbf{A}_\lambda}$ and $\nu_{\mathbf{B}_{\lambda^*}}$ are convex combinations of distributions with disjoint supports. Therefore, $H(\nu_{\mathbf{A}_\lambda}) = (1 - \frac{r}{2s}) \cdot H(U) + \frac{r}{2s} \cdot H(F) + H_2(\frac{r}{2s})$ and $H(\nu_{\mathbf{B}_{\lambda^*}}) = (1 - \frac{r}{2s}) \cdot H(U) + \frac{r}{2s} \cdot H(F^*) + H_2(\frac{r}{2s})$. We have $H(F^*) \geq H(F)$, because F and F^* share the same support, and F^* is uniform on this support. Hence, $H(\nu_{\mathbf{B}_{\lambda^*}}) \geq H(\nu_{\mathbf{A}_\lambda})$.

3.2. Bounding the dissimilarity. We are left to prove that $JS_{\lambda^*}(\nu_{\mathbf{B}_1}, \ldots, \nu_{\mathbf{B}_{\ell^*}}) \leq O(\max\{\epsilon^2, k/m\})$. For every matrix $\mathbf{A}$ induced by a subset $\mathcal{F}$, $\nu_{\mathbf{A}}$ is uniform on the set of pairs $(i + 2d(k-1), \mathbf{e}_i)$, for $i \in \mathcal{F}, d = 0, \ldots, r/(2(k-1)) - 1$ and $(i, \mathbf{e}_{2k})$, $i > r$. $\nu_{\mathbf{B}_{\lambda^*}}$ assigns probability $1/2s$ to all the pairs of the form $(i + 2d(k-1), \mathbf{e}_i)$,$i \in [2(k-1)], d = 0, \ldots, r/(2(k-1))$ and probability $1/s$ to all the pairs of the form $(i, \mathbf{e}_{2k})$, $i > r$. Therefore, for each $j \in [\ell^*]$, $D_{KL}(\nu_{\mathbf{B}_j} \parallel \nu_{\mathbf{B}_{\lambda^*}}) = \frac{r}{2} \cdot \frac{1}{s} \log \frac{(1/s)}{(1/2s)} = r/2s$. Note that $2s = 2(m - r/2) \geq m$. Therefore, $JS_\lambda(\nu_{\mathbf{B}_1}, \ldots, \nu_{\mathbf{B}^*_\ell}) \leq r/m \leq O(\max\{\epsilon^2, k/m\})$. $\square$

7.4 Matrix reconstruction

Theorem 7.5. *Let $0 \leq \epsilon \leq \frac{1}{\sqrt{24}}$ and $0 < \delta < \frac{1}{2}$. Any sampling algorithm Δ that (ϵ, δ)-approximates MR_F has worst-case query cost of at least $\Omega(mn \cdot (1 - 2\delta))$.*

Proof. We use the recipe described in Section 7.1.

1. Symmetry. It is easy to verify that MR_F is ϵ-permutation commutative. We can then use the distribution $\nu_{\mathbf{A}}$ on $[mn] \times \mathbb{R}$, which is the base of the index-answer distribution induced by uniform queries with replacement to entries of $\mathbf{A}$. Thus, $\nu_{\mathbf{A}}$ is uniform on entries of $\mathbf{A}$.

2.1. Abundance and similarly parameters. We choose $t = mn$ and $\gamma = 1$.

2.2. Mapping subsets of $[mn]$ to inputs of MR_F. We think of each matrix as a vector (we span the rows of the matrix one after the other to form an mn-dimensional vector). Given a subset $\mathcal{F} \subseteq [mn]$ of size $mn/2$, we map it to an input matrix $\mathbf{A}$, which is the characteristic vector of $\mathcal{F}$.

Disjointness. Let $\mathcal{F}, \mathcal{F}' \subseteq [mn]$ be two subsets of size $mn/2$, such that $|\mathcal{F} \cap \mathcal{F}'| \leq \frac{11}{24}mn$. Let $\mathbf{A}, \mathbf{A}'$ be the two corresponding input matrices. $||\mathbf{A} - \mathbf{A}'||_F^2 = |\mathcal{F} \setminus \mathcal{F}'| + |\mathcal{F}' \setminus \mathcal{F}| \geq \frac{mn}{12}$, while $\epsilon\left(||\mathbf{A}||_F + ||\mathbf{A}'||_F\right) = 2\epsilon\sqrt{\frac{mn}{2}} \leq \sqrt{\frac{mn}{12}}$, for $\epsilon \leq \frac{1}{\sqrt{24}}$. Thus, $\mathbf{A}$ and $\mathbf{A}'$ are ϵ-disjoint.

Entropy invariance. For any $\mathbf{A}$, $\nu_{\mathbf{A}}$ is uniform on entries of $\mathbf{A}$; thus $H(\nu_{\mathbf{A}}) = \log(mn)$.

3.1. Prove that $H(\nu_{\mathbf{B}_{\lambda^*}}) \geq H(\nu_{\mathbf{A}_\lambda})$. Both $\nu_{\mathbf{A}_\lambda}$ and $\nu_{\mathbf{B}_{\lambda^*}}$ are distributions on pairs of the form $((a, b), 0)$ and $((a, b), 1)$, where $a \in [m]$ and $b \in [n]$. $\nu_{\mathbf{B}_{\lambda^*}}$ is uniform on this support and therefore, $H(\nu_{\mathbf{B}_{\lambda^*}}) \geq H(\nu_{\mathbf{A}_\lambda})$.

3.2. Bounding the dissimilarity. For any $j \in [\ell^*]$, $\nu_{\mathbf{B}_j}$ is uniform on the set of pairs $((a, b), 1)$ with $(a, b) \in \mathcal{F}_j$ and $((a, b), 0)$ with $(a, b) \notin \mathcal{F}_j$. $\nu_{\mathbf{B}_{\lambda^*}}$ is uniform on all the pairs $((a, b), 1)$ and $((a, b), 0)$, where $a \in [m]$ and $b \in [n]$. Therefore, $D_{KL}(\nu_{\mathbf{B}_j} \parallel \nu_{\mathbf{B}_{\ell^*}}) = mn \cdot \frac{1}{mn} \cdot \log \frac{1/mn}{1/(2mn)} = 1$. $\square$

In the following $P_{\mathbf{A}}$ and $Q_{\mathbf{A}}$ denote the weight distributions of the rows and columns of a matrix $\mathbf{A}$.

Theorem 7.6. *Let $0 \leq \epsilon \leq \frac{1}{\sqrt{288}}$ and $0 < \delta < \frac{1}{2}$. Then,*

1. *Any sampling algorithm Δ that (ϵ, δ)-approximates MR_2 has worst-case query cost of at least $\Omega((m + n) \cdot (1 - 2\delta))$.*

2. *Any private-coin sampling algorithm Δ that (ϵ, δ)-approximates MR_2 and whose queries on input $\mathbf{A}$ are in a set of rows that are chosen independently according to $P_{\mathbf{A}}$ has worst-case query cost of at least $\Omega(m \cdot (1 - 2\delta))$.*

3. *Any private-coin sampling algorithm* Δ *that* (ϵ, δ)*-approximates* MR_2 *and whose queries on input* $\mathbf{A}$ *are in a set of columns that are chosen independently according to* $Q_{\mathbf{A}}$ *has worst-case query cost of at least* $\Omega(n \cdot (1 - 2\delta))$.

Proof. We begin with the proof of Part (1). Let q be the worst-case query cost Δ. We first show that $q \geq \Omega(m \cdot (1 - 2\delta))$ and then show that $q \geq \Omega(n \cdot (1 - 2\delta))$ (the two proofs are different this time). As usual, we use the recipe of Section 7.1.

1. Symmetry. It is easy to verify that MR_2 is ϵ-row permutation commutative. We can then use the distribution $\nu_{\mathbf{A}}$ on $[m] \times \mathbb{R}^n$, which is the base of the index-answer distribution induced by uniform queries with replacement to rows of $\mathbf{A}$. Thus, $\nu_{\mathbf{A}}$ is uniform on rows of $\mathbf{A}$.

2.1. Abundance and similarly parameters. We choose $t = m$ and $\gamma = 1$.

2.2. Mapping subsets of $[m]$ **to inputs of** MR_2**.** Given a subset $\mathcal{F} \subseteq [m]$ of size $m/2$, we map it to an input matrix $\mathbf{A}$, which is all-zero, except for the first column, which is the characteristic vector of $\mathcal{F}$.

Disjointness. Let $\mathcal{F}, \mathcal{F}' \subseteq [m]$ be two subsets of size $m/2$, such that $|\mathcal{F} \cap \mathcal{F}'| \leq \frac{11}{24}m$. Let $\mathbf{A}, \mathbf{A}'$ be the two corresponding input matrices. $||\mathbf{A} - \mathbf{A}'||_2^2 = |\mathcal{F} \setminus \mathcal{F}'| + |\mathcal{F}' \setminus \mathcal{F}| \geq \frac{m}{12}$. On the other hand, $\epsilon(||\mathbf{A}||_F + ||\mathbf{A}'||_F) = 2\epsilon\sqrt{\frac{m}{2}}$. Thus, $||\mathbf{A} - \mathbf{A}'||_2 \geq \epsilon(||\mathbf{A}||_F + ||\mathbf{A}'||_F)$ whenever $\epsilon \leq \frac{1}{\sqrt{24}}$, implying $\mathbf{A}$ and $\mathbf{A}'$ are ϵ-disjoint.

Entropy invariance. For any $\mathbf{A}$, $\nu_{\mathbf{A}}$ is uniform on rows of $\mathbf{A}$; thus $H(\nu_{\mathbf{A}}) = \log m$.

3.1. Prove that $H(\nu_{\mathbf{B}_{\lambda^*}}) \geq H(\nu_{\mathbf{A}_\lambda})$**.** Both $\nu_{\mathbf{A}_\lambda}$ and $\nu_{\mathbf{B}_{\lambda^*}}$ are distributions on pairs of the form $(i, \mathbf{0})$ and $(i, \mathbf{e}_1)$, where $i \in [m]$. $\nu_{\mathbf{B}_{\lambda^*}}$ is uniform on this support and therefore, $H(\nu_{\mathbf{B}_{\lambda^*}}) \geq H(\nu_{\mathbf{A}_\lambda})$.

3.2. Bounding the dissimilarity. $\nu_{\mathbf{B}_{\lambda^*}}$ is uniform on the following set of pairs: $(i, \mathbf{e}_1)$ for $i \in [m]$ and $(i, \mathbf{0})$ for $i \in [m]$. For each $j \in [\ell^*]$, $\nu_{\mathbf{B}_j}$ is uniform on the set of pairs $(i, \mathbf{e}_1)$ for $i \in \mathcal{F}_j$ and $(i, \mathbf{0})$ for $i \notin \mathcal{F}_j$. Thefefore, $D_{KL}(\nu_{\mathbf{B}_j} \parallel \nu_{\mathbf{B}_{\lambda^*}}) = m \cdot \frac{1}{m} \cdot \log \frac{1/m}{1/(2m)} = 1$. Thus, $JS_{\lambda^*}(\nu_{\mathbf{B}_1}, \ldots, \nu_{\mathbf{B}_{\ell^*}}) = 1$.

We conclude that $2q \geq \Omega(m \cdot (1 - 2\delta))$. We now prove that $q \geq \Omega(n \cdot (1 - 2\delta))$.

1. Symmetry. It is easy to verify that MR_2 is ϵ-column permutation commutative. We can then use the distribution $\nu_{\mathbf{A}}$ on $[n] \times \mathbb{R}^m$, which is the base of the index-answer distribution induced by uniform queries with replacement to columns of $\mathbf{A}$. Thus, $\nu_{\mathbf{A}}$ is uniform on columns of $\mathbf{A}$.

2.1. Abundance and similarly parameters. We choose $t = n$ and $\gamma = 1$.

2.2. Mapping subsets of $[n]$ **to inputs of** MR_2**.** Given a subset $\mathcal{F} \subseteq [n]$ of size $n/2$, we map it to an input matrix $\mathbf{A}$, all of whose rows are identical and equal to the characteristic vector of $\mathcal{F}$ multiplied by the scalar $\frac{1}{\sqrt{mn}}$.

Disjointness. Let $\mathcal{F}, \mathcal{F}' \subseteq [n]$ be two subsets of size $n/2$, such that $|\mathcal{F} \cap \mathcal{F}'| \leq \frac{11}{24}n$. Let $\mathbf{A}, \mathbf{A}'$ be the two corresponding input matrices. Note that $||\mathbf{A}||_F^2 = ||\mathbf{A}'||_F = m \cdot \frac{n}{2} \cdot \frac{1}{mn} = \frac{1}{2}$.

All the rows of $\mathbf{B} \stackrel{\text{def}}{=} \mathbf{A} - \mathbf{A}'$ are identical and equal to the following:

$$\mathbf{B}_{a,b} = \begin{cases} 0 & \text{if } b \in \mathcal{F} \cap \mathcal{F}' \text{ or } b \notin \mathcal{F} \cup \mathcal{F}' \\ \frac{1}{\sqrt{mn}} & \text{if } b \in \mathcal{F} \setminus \mathcal{F}' \\ -\frac{1}{\sqrt{mn}} & \text{if } b \in \mathcal{F}' \setminus \mathcal{F} \end{cases}$$

Take $\mathbf{x} \in \mathbb{R}^n$ to be the following unit vector:

$$\mathbf{x}_b = \begin{cases} -\frac{1}{\sqrt{n}} & \text{if } b \in \mathcal{F}' \setminus \mathcal{F} \\ \frac{1}{\sqrt{n}} & \text{Otherwise} \end{cases}$$

Then, $\forall a \in [m]$, $(\mathbf{Bx})_a^2 = \frac{1}{n} \cdot \frac{1}{mn} \cdot (|\mathcal{F} \setminus \mathcal{F}'| + |\mathcal{F}' \setminus \mathcal{F}|)^2 \geq \frac{1}{mn^2} \cdot (\frac{n}{12})^2 = \frac{1}{144m}$. Therefore, $||\mathbf{B}||_2^2 \geq \frac{1}{144}$. It follows that $||\mathbf{A} - \mathbf{A}'||_2 \geq \epsilon(||\mathbf{A}||_F + ||\mathbf{A}'||_F)$ (implying $\mathbf{A}$ and $\mathbf{A}'$ are ϵ-disjoint) as long as $\epsilon \leq \frac{1}{\sqrt{288}}$.

The rest of the argument is identical to the proof that $q \geq \Omega(m \cdot (1 - 2\delta))$, and is thus omitted. This completes the proof of Part (1).

Identical mappings of subsets to inputs derive similar lower bounds for row-querying algorithms that query according to $P_{\mathbf{A}}$ and to column-querying algorithms that query according to $Q_{\mathbf{A}}$. In the first mapping the weight of all the non-zero rows in each input matrix is 1, and therefore $P_{\mathbf{A}}$ simply selects uniformly at random from the set of non-zero rows. In the second mapping the weights of all non-zero columns are identical and equal to $\sqrt{\frac{m}{n}}$, implying an algorithm that uses $Q_{\mathbf{A}}$ picks a non-zero column uniformly at random. This changes the analysis above only marginally, and we therefore do not repeat it. $\square$

8 Expected query complexity

In this section we prove a lower bound on the expected query complexity via the KL divergence. The lower bound is based on the reduction from statistical classification and on a lower bound on the expected sample cost of two-class classification. The latter is a well-known result from statistics called the *optimality of the sequential probability ratio test* (cf. [Sie85]):

Theorem 8.1 (cf. [Sie85]). *Let $0 < \delta < 1/e$. Let μ_0, μ_1 be i.i.d. sequential distributions on a domain $\mathcal{B}$ with base distributions ν_0, ν_1. Then, the expected sample cost q_E of any deterministic δ-error classifier C for μ_0, μ_1 satisfies:*

$$q_E > \frac{1}{2 \cdot \min\{D_{KL}(\nu_0 \parallel \nu_1), D_{KL}(\nu_1 \parallel \nu_0)\}} \cdot (1 - 2\delta) \cdot \log\frac{1}{\delta}.$$

The proof appears in [Sie85, DKLR95]. For completeness, we give it in Appendix A.3. The above lower bound holds only for deterministic classifiers, but can easily be extended to randomized classifiers:

Corollary 8.2. *Let $0 < \delta < 1/e$. Let μ_0, μ_1 be i.i.d. sequential distributions on a domain $\mathcal{B}$ with base distributions ν_0, ν_1. Then, the expected sample cost q_E of any (randomized) δ-error classifier C for μ_0, μ_1 satisfies:*

$$q_E \;>\; \frac{1}{8 \cdot \min\left\{D_{KL}(\mu_0 \parallel \mu_1), D_{KL}(\mu_1 \parallel \mu_0)\right\}} \cdot (1 - 8\delta) \cdot \log \frac{1}{\delta}.$$

Proof. For $b = 0, 1$, let $q_b \stackrel{\text{def}}{=} \mathrm{E}\left[T(b, R)\right]$ be the expected sample cost of C when running on input b. The expectation is over both the random samples from μ_b and the random string R of the referee. For each possible value r of the referee's random string, let $\delta_{b,r} \stackrel{\text{def}}{=} \Pr(C_{\text{out}}(b, r) \neq b)$ and $q_{b,r} \stackrel{\text{def}}{=} \mathrm{E}\left[T(b, r)\right]$ be the error probability of the classifier and its expected sample cost, respectively, when running on input b and when the referee gets r as the random string. The probability and the expectation here are only over the random samples from μ_b. We have for $b = 0, 1$,

$$\mathrm{E}\left[\delta_{b,R}\right] \;\leq\; \delta\;, \qquad \mathrm{E}\left[q_{b,R}\right] = q_b.$$

By Markov's inequality, for less than $1/4$ of the values of r, $\delta_{b,r} > 4\delta$, and for less than $1/4$ of the values of r, $q_{b,r} > 4q_b$. Therefore, there is at least one choice of r such that both $\delta_{b,r} \leq 4\delta$ and $q_{b,r} \leq 4q_b$ for $b = 0, 1$.

Let C_r be the deterministic sequential classifier induced by C and the fixing of the referee's random string to r. Note that $\delta_{b,r}$ is the error probability of C_r on input b and $q_{b,r}$ is its expected sample cost on input b. Applying now Theorem 8.1 to C_r, we obtain:

$$4q_b \;>\; \frac{1}{2 \cdot D_{KL}(\mu_b \parallel \mu_{1-b})} \cdot (1 - 8\delta) \cdot \log \frac{1}{8\delta}.$$

□

Using now the reduction from classification to sampling, we deduce the following lower bound on the expected query cost of index-oblivious sampling algorithms:

Theorem 8.3. *Let $0 < \delta \leq 1/e$. Let Δ be an index-oblivious sampling algorithm that (ϵ, δ)-approximates a function $f : \mathcal{X}^n \to \mathcal{Y}$. Let $\mathbf{x}$ and $\mathbf{y}$ be any pair of ϵ-disjoint inputs, such that the answer distributions of Δ on $\mathbf{x}$ and $\mathbf{y}$ are i.i.d. with base distributions $\nu_{\mathbf{x}}$ and $\nu_{\mathbf{y}}$. Then, the expected query cost q_E of Δ satisfies:*

$$q_E \;>\; \frac{1}{8 \cdot \min\left\{D_{KL}(\nu_{\mathbf{x}} \parallel \nu_{\mathbf{y}}), D_{KL}(\nu_{\mathbf{y}} \parallel \nu_{\mathbf{x}})\right\}} \cdot (1 - 8\delta) \cdot \log \frac{1}{8\delta}.$$

The theorem is meaningful only for index-oblivious algorithms, since the KL divergence between the index-answer distributions (as opposed to the answer distributions) of any two different inputs is always infinite. The bound is yet useful for symmetric functions, for which index-oblivious algorithms are the best possible.

We next demonstrate the technique by an optimal lower bound on the expected query complexity of the mean. Given $x_1, \ldots, x_n \in [0, 1]$, the average of $O\left(1/\epsilon^2 \cdot \log(1/\delta)\right)$ uniformly chosen samples from $x_1, \ldots, x_n$ is within ϵ of $\frac{1}{n}\sum_i x_i$ with probability at least $1-\delta$ (Chernoff-Hoeffding bound). Canetti *et al.* [CEG95] proved that this bound is tight w.r.t. worst-case query complexity. Radhakrishnan and Ta-Shma [RT00] (implicitly) extended to the expected query complexity. We provide an elementary proof for the latter (albeit, we need to restrict n more than [RT00] do):

Theorem 8.4. *Let $0 < \epsilon \leq \frac{1}{4}$, $0 < \delta \leq \frac{1}{e}$, and $n \geq \Omega(\frac{1}{\epsilon^4} \cdot \frac{1}{\delta} \cdot \log^4(1/\delta))$. Then, the expected query cost of any sampling algorithm Δ that (ϵ, δ)-approximates the mean is at least $\Omega\left(\frac{1}{\epsilon^2} \cdot \log \frac{1}{\delta}\right)$.*

Proof. Let q be the worst-case query cost of Δ and let q_E be its expected query cost. WLOG, $q \leq q_E \cdot O(\log(1/\delta))$ (cf. [BKS01, Bar02]). Using the restriction on n, we can therefore assume that $q \leq \sqrt{2\delta n}$ and $q \leq \sqrt[3]{2n \cdot q_E}$. Because, if $q \geq \sqrt{2\delta n}$, then $q_E \geq \Omega(q/\log(1/\delta)) \geq \Omega(\sqrt{\delta n}/\log(1/\delta)) \geq \Omega(1/\epsilon^2 \cdot \log(1/\delta))$, which is what we wanted to prove. Similarly, if $q \geq \sqrt[3]{2n \cdot q_E}$, then $q_E \geq \Omega(q/\log(1/\delta)) \geq \Omega(\sqrt[3]{n \cdot q_E}/\log(1/\delta))$, implying $q_E \geq \Omega(\sqrt{n/\log^{3/2}(1/\delta)}) \geq \Omega(1/\epsilon^2 \cdot \log(1/\delta))$.

Using the above restrictions on q and using the fact the mean is ϵ-symmetric, we get from the first part of Theorem 4.2 that there exists a private-coin, index-oblivious, sampling algorithm $\hat{\Delta}$ of expected query cost at most $2q_E$ that $(\epsilon, 2\delta)$-approximates the mean and whose index distribution on any input is uniform with replacement. Let us denote by $\nu_{\mathbf{x}}$ the base distribution of this algorithm's answer distribution on input $\mathbf{x}$.

Consider the two inputs $\mathbf{x}$ and $\mathbf{y}$. $\mathbf{x}$ has $(\frac{1}{2}+\epsilon)n$ 0's and rest are 1's; $\mathbf{y}$ has $(\frac{1}{2}-\epsilon)n$ 0's and the rest are 1's. Since the mean of $\mathbf{x}$ is $1/2-\epsilon$ and the mean of $\mathbf{y}$ is $1/2+\epsilon$, $\mathbf{x}$ and $\mathbf{y}$ are ϵ-disjoint.

Note that the answer distribution on $\mathbf{x}$, $\nu_{\mathbf{x}}$, is a Bernoulli distribution with probability of success $\frac{1}{2}-\epsilon$. Similarly, the answer distribution on $\mathbf{y}$, $\nu_{\mathbf{y}}$, is a Bernoulli distribution with probability of success $\frac{1}{2}+\epsilon$. A simple calculation then shows that $D_{KL}(\nu_{\mathbf{x}} \parallel \nu_{\mathbf{y}}) \leq O(\epsilon^2)$. The lower bound now follows from Theorem 8.3. ☐

9 Acknowledgments

I thank T.S. Jayram and D. Sivakumar for their help with the canonical form lemmas for symmetric functions. I thank Ran El-Yaniv for pointing out references in the information theory and statistics literature.

References

[AFK+01] Y. Azar, A. Fiat, A. R. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 619–626, 2001.

[AM01] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 611–618, 2001.

[Bar02] Z. Bar-Yossef. *The Complexity of Massive Data Set Computations*. PhD thesis, Computer Science Division, U.C. Berkeley, 2002.

[BFR+00] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing that distributions are close. In *Proceedings of the 41st IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–269, 2000.

[BJKS02] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002. To appear.

[BKS01] Z. Bar-Yossef, R. Kumar, and D. Sivakumar. Sampling algorithms: Lower bounds and applications. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 266–275, 2001.

[CCMN00] M. S. Charikar, S. Chaudhuri, R. Motwani, and V. Narasayya. Towards estimation error guarantees for distinct values. In *Proceedings of the 19th Annual ACM Symposium on Principles of Database Systems (PODS)*, pages 268–279, 2000.

[CEG95] R. Canetti, G. Even, and O. Goldreich. Lower bounds for sampling algorithms for estimating the average. *Information Processing Letters*, 53:17–25, 1995.

[CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[DFK+99] P. Drineas, A. M. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *Proceedings of the 10th IEEE Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 291–299, 1999.

[DK02] P. Drineas and R. Kannan. Pass efficient algorithm for approximating large matrices. Manuscript, 2002.

[DKLR95] P. Dagum, R. Karp, M. Luby, and S. Ross. An optimal algorithm for monte carlo estimation. In *Proceedings of the 36th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 142–149, 1995.

[DKR02] P. Drineas, I. Kerenidis, and P. Raghavan. Competitive recommendation systems. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 82–90, 2002.

[Fan52] R. M. Fano. Class Notes for Transmission of Information, 1952. Course 6.574, MIT, Cambridge, MA.

[FKV98] A. M. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In *Proceedings of the 39th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 370–378, 1998.

[GT01] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. In *Proceedings of the 42nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 460–469, 2001.

[Gut89] M. Gutman. Asymptotically optimal classification for multiple tests with empirically observed statistics. *IEEE Transactions on Information Theory*, 35(2):401–408, 1989.

[GV96] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

[Kai67] T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, COM-15(1):52–60, 1967.

[KL51] S. Kullback and A. Leibler. On information and sufficiency. *IEEE Transactions on Information Theory*, 22:79–86, 1951.

[KV94] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1994.

[Lin91] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[New91] I. Newman. Private vs. common random bits in communication complexity. *Information Processing Letters*, 39:67–71, 1991.

[NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences (JCSS)*, 49(2):149–167, 1994.

[RT00] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.

[Sie85] D. Siegmund. *Sequential Analysis - Tests and Confidence Intervals*. Springer-Verlag, 1985.

[SV99] L. Schulman and V. V. Vazirani. Majorizing estimators and the approximation of #P-complete problems. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC)*, pages 288–294, 1999.

[Tou74] G. T. Toussaint. On some measures of information and their application to pattern recognition. In *Proceedings of the Conference on Measures of Information and Their Applications*, pages 21–28, Indian Institute of Technology, Bombay, 1974.

[Van68] H. L. Van Trees. *Detection, Estimation, and Modulation Theory.* Jon Wiley & Sons, Inc., 1968.

[Vap98] V. N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, Inc., 1998.

[Ziv88] J. Ziv. On classification with empirically observed statistics and universal data compression. *IEEE Transactions on Information Theory*, 34:278–286, 1988.

A Some proofs

A.1 Jensen-Shannon divergence

Proposition 2.4 (restated) *Let $D \sim \lambda$, $(X_1, \ldots, X_n) \sim (\mu_1, \ldots, \mu_n)$, $\pi(d, x_1, \ldots, x_n) = x_d$, and $X_D = \pi(D, X_1, \ldots, X_n)$. That is, X_D is a sample from the distribution μ_D, where D is chosen according to λ. Then, $JS_\lambda(\mu_1, \ldots, \mu_n) \; = \; I(D;\, X_D)$.*

Proof. We start by stating three facts from information theory used in the proof. For two distributions μ and ν, we denote by (μ, ν) their joint distribution, and by $\mu \times \nu$ their product distribution (i.e., $(\mu \times \nu)(x, y) = \mu(x) \cdot \nu(y)$). The mutual information between two random variables X and Y, $(X, Y) \sim (\mu, \nu)$, has the following characterization in terms of the KL divergence (cf. [CT91]):

$$I(X;\, Y) \; = \; D_{KL}((\mu, \nu) \parallel \mu \times \nu).$$

For a distribution μ and an event A, we denote by $\mu|A$ the conditional distribution of μ given the event A. For joint distributions $\mu = (\mu_X, \mu_Y)$ and $\nu = (\nu_X, \nu_Y)$ on $\mathcal{X} \times \mathcal{Y}$, let $(X, Y) \sim \mu$ and $(W, Z) \sim \nu$. The conditional KL divergence between μ_X and ν_X given μ_Y and ν_Y is defined as:

$$D_{KL}(\mu_X|\mu_Y \parallel \nu_X|\nu_Y) \; \stackrel{\text{def}}{=} \; \sum_{y \in \mathcal{Y}} \mu_Y(y) \cdot D_{KL}(\mu_X|\{Y = y\} \parallel \nu_X|\{Z = y\}).$$

The chain rule for KL divergence is:

$$D_{KL}(\mu \parallel \nu) = D_{KL}(\mu_Y \parallel \nu_Y) + D_{KL}(\mu_X|\mu_Y \parallel \nu_X|\nu_Y).$$

We next use the above facts to prove the proposition. Note that the distribution of X_D is exactly the λ-weighted average distribution $\sum_{i=1}^{n} \lambda(i)\mu_i$. Let us denote this distribution by

μ_λ. The distribution of μ_λ given the event $\{D = i\}$ is exactly μ_i. Thus,

$$
\begin{aligned}
I(D;\ X_D) &= D_{KL}((\lambda, \mu_\lambda) \parallel \lambda \times \mu_\lambda) \\
&\qquad\qquad \text{(KL divergence characterization of mutual information)} \\
&= D_{KL}(\lambda \parallel \lambda) + D_{KL}(\mu_\lambda | \lambda \parallel \mu_\lambda) \\
&\qquad\qquad \text{(Chain rule for KL divergence)} \\
&= 0 + \sum_{i=1}^{n} \lambda(i) \cdot D_{KL}(\mu_\lambda | \{D = i\} \parallel \mu_\lambda) \\
&\qquad\qquad \text{(Definition of conditional KL divergence)} \\
&= \sum_{i=1}^{n} \lambda(i) \cdot D_{KL}(\mu_i \parallel \mu_\lambda) \\
&= JS_\lambda(\mu_1, \ldots, \mu_n)
\end{aligned}
$$

□

A.2 Set designs

Proposition 2.5 (restated) *For every $m \geq 18$, there exists a $(\frac{m}{2}, \frac{11}{24}m)$-design $\mathcal{F}_1, \ldots, \mathcal{F}_\ell \subseteq [m]$ of size $\ell = 2^{\Omega(m)}$.*

Proof. Following [NW94], we construct the design sequentially, where at the ℓ-th step we greedily choose a subset $\mathcal{F}_\ell$ that intersects each of the previously chosen subsets $\mathcal{F}_1, \ldots, \mathcal{F}_{\ell-1}$ at less than $\frac{11}{24}m$ positions. We show that such a choice is possible as long as ℓ is small enough. Unlike [NW94], we need to make sure the sets we choose are not multi-sets.

In order to prove the existence of $\mathcal{F}_\ell$ we choose a random subset X of size $m/2$ (according to some distribution to be specified shortly) and prove that with positive probability X intersects each of $\mathcal{F}_1, \ldots, \mathcal{F}_{\ell-1}$ in less than $\frac{11}{24}m$ positions. Let Y be a random subset of $[m]$ chosen by picking $m/2$ elements of $[m]$ uniformly and independently. Note that $|Y|$ may be smaller than $m/2$. X will be the union of Y and the first $\frac{m}{2} - |Y|$ elements of $[m]$ that do not belong to Y.

Note that for any $1 \leq j \leq \ell - 1$, $|\mathcal{F}_j \cap X| \leq |\mathcal{F}_j \cap Y| + (\frac{m}{2} - |Y|)$. Therefore,

$$\Pr(\exists j, |\mathcal{F}_j \cap X| > \frac{11}{24}m) \ \leq \ \Pr(|Y| < \frac{m}{3}) + \sum_j \Pr(|\mathcal{F}_j \cap Y| > \frac{7}{24}m).$$

In order to bound $\Pr(|Y| < \frac{m}{3})$ we define $\binom{m/2}{2}$ indicator random variables $Z_{i,i'}$, where $Z_{i,i'} = 1$ iff the i-th element and the i'-th element chosen to be included in Y are the same. Let $Z = \sum_{i,i'} Z_{i,i'}$ be the total number of collisions. Note that if $|Y| < \frac{m}{3}$ then $Z > \frac{m}{6}$. Since the probability of a collision is $\frac{1}{m}$, $\mathrm{E}[Z] = \binom{m/2}{2} \cdot \frac{1}{m}$. The $Z_{i,i'}$'s are pairwise independent, and therefore $\mathrm{Var}[Z] = \binom{m/2}{2} \frac{1}{m}(1 - \frac{1}{m}) \leq \mathrm{E}[Z]$. We then apply Chebyshev's inequality and use the fact $\frac{m}{9} \leq \mathrm{E}[Z] \leq \frac{m}{8}$:

$$\Pr(Z > \frac{m}{6}) \ \leq \ \Pr(|Z - \mathrm{E}[Z]| > \mathrm{E}[Z]/3) \ \leq \ \frac{9 \cdot \mathrm{Var}[Z]}{\mathrm{E}^2[Z]} \ \leq \ \frac{81}{m}.$$

Fix some $1 \leq j \leq \ell - 1$. Note that $\mathrm{E}[|\mathcal{F}_j \cap Y|] = m/4$. Therefore, by Chernoff bound, $\Pr(|\mathcal{F}_j \cap Y| > \frac{7}{24}m) \leq e^{-m/288}$. Thus, as long as $\ell < (1 - \frac{81}{m}) \cdot e^{m/288} + 1$, X is as wanted with positive probability. □

A.3 Optimality of the sequential probability ratio test

Theorem 8.1 (restated) *Let $0 < \delta < 1/e$. Let μ_0, μ_1 be i.i.d. sequential distributions on a domain $\mathcal{B}$ with base distributions ν_0, ν_1. Then, the expected sample cost q_E of any deterministic δ-error classifier C for μ_0, μ_1 satisfies:*

$$q_E \;>\; \frac{1}{2 \cdot \min\{D_{KL}(\nu_0 \parallel \nu_1), D_{KL}(\nu_1 \parallel \nu_0)\}} \cdot (1 - 2\delta) \cdot \log \frac{1}{\delta}.$$

Proof. Since C is a deterministic classifier, its referee does not have a random string. Thus, his decision function is of the form $D : \mathcal{B}^* \to \{0,1\} \cup \{\textsc{cont}\}$. The sample cost of C on input $b \in \{0,1\}$ is $\mathrm{E}[T(b)]$, where the expectation is over the samples from μ_b. Our goal is to prove a lower bound on $\max\{\mathrm{E}[T(0)], \mathrm{E}[T(1)]\}$. We will prove a lower bound on $\mathrm{E}[T(0)]$. A similar argument works for $\mathrm{E}[T(1)]$.

Let $B_1, B_2, \ldots$ denote the infinite sequence of random variables on the domain $\mathcal{B}$ obtained by drawing samples according to μ_0. $B_1, B_2, \ldots$ are independent and are distributed according to ν_0. Similarly, let $B'_1, B'_2, \ldots$ denote the infinite sequence of samples from μ_1. $B_1, B_2, \ldots$ are independent and are distributed according to ν_1.

We define L_i to be the "log-likelihood ratio" of ν_0 and ν_1 w.r.t. the i-th sample from μ_0:

$$L_i \;\stackrel{\text{def}}{=}\; \ln \frac{\nu_0(B_i)}{\nu_1(B_i)}.$$

Note that for all i, $\mathrm{E}[L_i] = \ln 2 \cdot D_{KL}(\nu_0 \parallel \nu_1)$. Define $G \stackrel{\text{def}}{=} \sum_{i=1}^{T(0)} L_i$. We use Wald's identity (cf. [Sie85]) to analyze the expectation of G:

Theorem A.1 (Wald's identity). *Let $X_1, X_2, \ldots$ be an infinite sequence of independent and identically distributed random variables with mean μ. Let S be a random variable on $\{0,1,2,\ldots\}$, for which the event $\{S = k\}$ is independent of $X_{k+1}, X_{k+2}, \ldots$ for all k (S is called a* stopping time random variable*). We further assume $\mathrm{E}[S] < \infty$. Then,*

$$\mathrm{E}\left[\sum_{i=1}^{S} X_i\right] = \mu \cdot \mathrm{E}[S].$$

Note that the event "$T(0) = t$" depends only on $B_1, \ldots, B_t$, and is therefore independent of $L_{t+1}, L_{t+2}, \ldots$. Therefore,

$$\mathrm{E}[G] \;=\; \mathrm{E}[T(0)] \cdot \ln 2 \cdot D_{KL}(\nu_0 \parallel \nu_1). \tag{21}$$

Let $\mathcal{A}_0$ denote the set of all finite sequences $\mathbf{b} \in \mathcal{B}^*$ for which $D(\mathbf{b}) = 0$ and let $\mathcal{A}_1$ denote the set of all sequences $\mathbf{b} \in \mathcal{B}^*$ for which $D(\mathbf{b}) = 1$. Let $\mathcal{A} = \mathcal{A}_0 \cup \mathcal{A}_1$. We can rewrite $\mathrm{E}[G]$ as follows:

$$\begin{aligned}
\mathrm{E}[G] &= \sum_{\mathbf{b}=(b_1,\ldots,b_t)\in\mathcal{A}} \mathrm{E}[G \mid B_1 = b_1, \ldots, B_t = b_t] \cdot \Pr(B_1 = b_1, \ldots, .B_t = b_t) \\
&= \sum_{\mathbf{b}\in\mathcal{A}} \sum_{i=1}^{|\mathbf{b}|} \ln \frac{\nu_0(b_i)}{\nu_1(b_i)} \cdot \prod_{i=1}^{|\mathbf{b}|} \nu_0(b_i) \\
&= \sum_{\mathbf{b}\in\mathcal{A}_0} \ln \left(\frac{\prod_{i=1}^{|\mathbf{b}|} \nu_0(b_i)}{\prod_{i=1}^{|\mathbf{b}|} \nu_1(b_i)} \right) \cdot \prod_{i=1}^{|\mathbf{b}|} \nu_0(b_i) \; + \; \sum_{\mathbf{b}\in\mathcal{A}_1} \ln \left(\frac{\prod_{i=1}^{|\mathbf{b}|} \nu_0(b_i)}{\prod_{i=1}^{|\mathbf{b}|} \nu_1(b_i)} \right) \cdot \prod_{i=1}^{|\mathbf{b}|} \nu_0(b_i). \quad (22)
\end{aligned}$$

We next use the "log sum inequality" (cf. [CT91], Chapter 2), which states the following:

Theorem A.2 (Log sum inequality). *For non-negative $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$,*

$$\sum_{i=1}^{n} a_i \ln \frac{a_i}{b_i} \;\geq\; \left(\sum_{i=1}^{n} a_i \right) \ln \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}.$$

Using this inequality, we can bound each of the sums in Equation 22 as follows:

$$\begin{aligned}
&\sum_{\mathbf{b}\in\mathcal{A}_0} \ln \left(\frac{\Pi_{i=1}^{|\mathbf{b}|} \nu_0(b_i)}{\Pi_{i=1}^{|\mathbf{b}|} \nu_1(b_i)} \right) \cdot \Pi_{i=1}^{|\mathbf{b}|} \nu_0(b_i) \;\geq \\
&\quad\geq \left(\sum_{\mathbf{b}\in\mathcal{A}_0} \Pi_{i=1}^{|\mathbf{b}|} \nu_0(b_i) \right) \cdot \ln \left(\frac{\sum_{\mathbf{b}\in\mathcal{A}_0} \Pi_{i=1}^{|\mathbf{b}|} \nu_0(b_i)}{\sum_{\mathbf{b}\in\mathcal{A}_0} \Pi_{i=1}^{|\mathbf{b}|} \nu_1(b_i)} \right) \\
&\quad= \Pr((B_1, \ldots, B_{T(0)}) \in \mathcal{A}_0) \cdot \ln \frac{\Pr((B_1, \ldots, B_{T(0)}) \in \mathcal{A}_0)}{\Pr((B'_1, \ldots, B'_{T(1)}) \in \mathcal{A}_0)}. \quad (23)
\end{aligned}$$

Note that the event "$(B_1, \ldots, B_{T(0)}) \in \mathcal{A}_b$" is identical to the event "$C_{\mathrm{out}}(0) = b$", for $b \in \{0, 1\}$. Similarly, the event "$(B'_1, \ldots, B'_{T(1)}) \in \mathcal{A}_b$" is identical to the event "$C_{\mathrm{out}}(1) = b$". We can therefore rewrite the RHS of Equation 23 as follows:

$$\Pr(C_{\mathrm{out}}(0) = 0) \cdot \ln \frac{\Pr(C_{\mathrm{out}}(0) = 0)}{\Pr(C_{\mathrm{out}}(1) = 0)} \;>\; (1 - \delta) \cdot \ln \frac{1 - \delta}{\delta}.$$

Similarly,

$$\sum_{\mathbf{b}\in\mathcal{A}_1} \ln \left(\frac{\Pi_{i=1}^{|\mathbf{b}|} \nu_0(b_i)}{\Pi_{i=1}^{|\mathbf{b}|} \nu_1(b_i)} \right) \cdot \Pi_{i=1}^{|\mathbf{b}|} \nu_0(b_i) \;\geq\; \Pr(C_{\mathrm{out}}(0) = 1) \cdot \ln \frac{\Pr(C_{\mathrm{out}}(0) = 1)}{\Pr(C_{\mathrm{out}}(1) = 1)}.$$

Let $a = \Pr(C_{\mathrm{out}}(0) = 1)$ and $b = \Pr(C_{\mathrm{out}}(1) = 1)$. Note that $0 \leq a \leq \delta$, while $1 - \delta \leq b \leq 1$. Therefore, $a \ln(a/b) \geq a \ln a$. The function $f(a) = a \ln a$ gets a minimum at $a = 1/e$. Since we assume $\delta \le 1/e$, its minimum in the interval $[0, \delta]$ is at $a = \delta$. Therefore, $\Pr(C_{\text{out}}(0) = 1) \cdot \ln \frac{\Pr(C_{\text{out}}(0)=1)}{\Pr(C_{\text{out}}(1)=1)} \ge \delta \ln \delta$. We conclude that

$$\mathrm{E}[G] \;>\; (1-\delta) \cdot \ln \frac{1-\delta}{\delta} - \delta \ln \frac{1}{\delta} \;\ge\; \frac{1}{2} \cdot (1-2\delta) \cdot \ln \frac{1}{\delta}.$$

Combining this lower bound and Equation 21, we obtain:

$$\mathrm{E}[T(0)] \;>\; \frac{1}{2 \cdot D_{KL}(\nu_0 \parallel \nu_1)} \cdot (1-2\delta) \cdot \log \frac{1}{\delta}.$$

An identical argument shows that:

$$\mathrm{E}[T(1)] > \frac{1}{2 \cdot D_{KL}(\nu_1 \parallel \nu_0)} \cdot (1-2\delta) \cdot \log \frac{1}{\delta}.$$

$\square$