ONE-WAY FUNCTIONS AND LOCAL DESCRIPTIONS OF RANDOM DATA

DIMITRIOS MYRISIOTIS
GREAT BAY UNIVERSITY
CHINA

ABSTRACT. Sample compression describes a labeled sample by retained examples and extra information from which a decoder recovers the labels. We show that estimating the length of the shortest such description of random data is hard on average if and only if one-way functions exist. As in Hirahara and Nanashima's (FOCS 2023) consistent pointwise programs, the decoder labels each sample point separately, as a hypothesis does, but in our main result the labels are random rather than given by a concept. To label a point, it sees only that point, the retained examples with one extra bit each, a program, and a public random string revealed before the description is chosen. The length counts the choice of retained examples, their extra bits and the program, not the examples themselves. The task is to output a number between the $1/2$ and $3/4$ quantiles of the shortest length over the public string, up to an additive $a\lceil \log_2 m \rceil$ for m points and a fixed integer $a \geq 0$. The main result takes polynomially many, at least $8n$, uniform n-bit points with independent uniform labels and any large enough polynomial time bound per query. If one-way functions exist, every efficient randomized algorithm fails to output such a number for all large n with probability above an inverse polynomial that is the same for all algorithms and time bounds. Otherwise, for every inverse polynomial, some such algorithm fails with at most that probability for infinitely many n. For hardness, labels derived from one-way functions look random yet have short local descriptions, which random labels rarely have. The converse inverts a function that generates descriptions. One-way functions also exist if and only if estimating between the quantiles is hard, at some suitable time bound, for some PAC-style samplable data of the same sizes, labeled by a concept independent of the points.

CONTENTS

1. Introduction . . . 2
1.1. Our Results . . . 3
1.2. Our Techniques . . . 7
1.3. Other Related Work . . . 8
1.4. Discussion . . . 10
1.5. Open Problems . . . 10
AI Disclosure . . . 10
1.6. Paper Organization . . . 10
2. Preliminaries . . . 11
2.1. Certificates and Sampling . . . 12
2.2. Cryptographic Background . . . 14
3. Counting and Coding . . . 14
4. Hardness on Uniform Datasets . . . 16
4.1. Imported Tools and Random Queries . . . 16
4.2. The Uniform-Data Theorem . . . 19
4.3. An Unconditional Precision–Error Bound . . . 22
4.4. Independent Concepts and Uniform Data . . . 23
5. Easiness without One-Way Functions . . . 24
6. Hardness on PAC-Style Samples from One-Way Functions . . . 29
References . . . 31
A. Precision of Quantile Recovery . . . 33
A.1. Exact Computation with a Lower Margin . . . 33
A.2. Exact Computation at a Random Threshold . . . 35
B. Global and Pointwise Descriptions . . . 37

1. Introduction

Sample compression represents a labeled sample by retained examples and information used to reconstruct its labels [LW86, FW95]. Moran–Yehudayoff construct sample compression schemes for every class of finite VC dimension [MY16]. Since the reconstructed labels must agree with the sample, a compression acts like a hypothesis that agrees with the sample. We ask how hard it is to estimate the length of the shortest such description when the data are random. We answer it in terms of one-way functions: the task is hard on average exactly when one-way functions exist.

We study reconstruction through a pointwise interface, as for a hypothesis: to label a query point, the decoder receives that point and the retained points, and returns one label. It does not receive the other sample points or the entire label vector. A certificate consists of three things: the set of retained points, one arbitrary extra bit for each retained point, called its tag, and a side program that the decoder runs. For instance, retaining every point with its label as its tag gives a lookup certificate, which is valid whenever no point repeats with conflicting labels and the decoder has time for the lookup.

A certificate is charged in bits for naming the retained set, for the tags and for the program; the contents of the retained points are not charged, since they are sample points. Retained points are available to every query, whereas a short program for the whole label vector need not yield a pointwise rule, because answering one query may require the other sample points.

The decoder also reads a public random string, the public tape, and a certificate may be chosen after that tape is revealed. Goldberg–Kabanets–Lu–Oliveira use the same order of choices in their conditional probabilistic complexity [GKLO22, Definition 17], and our coding argument reads hash functions from the public tape. The minimum certificate length is therefore a random variable over the public tape. We ask for a value between two fixed quantiles of this minimum; the characterizations use its $1/2$ and $3/4$ quantiles, with an additive tolerance $a\lceil \log m \rceil$ for any fixed integer $a \geq 0$ in the uniform-data case. The gap between the two quantiles lets an algorithm estimate the public-tape probabilities instead of computing them exactly.

Recall that a one-way function is a polynomial-time computable function that is hard to invert on a uniform input. Every randomized polynomial-time algorithm, given the output and the unary input length, finds a preimage only with negligible probability, that is, with probability eventually below each inverse polynomial. Both description complexity and learning have been tied to one-way functions or other cryptographic primitives. Liu–Pass characterize one-way functions through the average-case hardness of time-bounded Kolmogorov complexity [LP20], where a program prints a whole string. Their sublinear-time characterization, stated for nonuniform security, uses programs that answer requested coordinates of a truth table [LP21a].

On the learning side, Impagliazzo–Levin connected inversion hardness with the computational feasibility of universal extrapolation [IL90]. Blum–Furst–Kearns–Lipton relate prediction hardness to cryptographic primitives when concepts and examples are sampled independently [BFKL94, Definitions 1–2]. Nanashima connected one-way functions to heuristic PAC learnability [Nan21]. In the separated sampling model of Blum–Furst–Kearns–Lipton, where the labels come from a concept drawn separately from the points, Hirahara–Nanashima later studied minimum consistent pointwise programs [HN23]. Ilango–Loff–Oliveira state a connection between compression of random partial truth tables and uniform PAC learning, which they describe as implicit in earlier work [ILO20, Appendix A, Theorem 36].

These results leave a concrete question about random finite data: whether, when the points and their labels are independent and uniform, estimating the length of the shortest certificate is hard on average exactly when one-way functions exist. We show that it is, for one-way functions secure against uniform algorithms rather than the nonuniform security of the sublinear-time characterization. Here hardness means that every efficient algorithm fails, at all large lengths, with probability above a fixed inverse polynomial. Without one-way functions, for each inverse polynomial error, some algorithm errs with at most that probability on infinitely many lengths. In our main theorem the queries are the sample's own uniformly random points, given by their contents rather than by an index, and the labels are independent uniform bits rather than values of a concept.

We also treat PAC-style samples, which descend from Valiant's learning model [Val84]. In such a sample, the points are drawn independently from one distribution, a concept is chosen independently of

the points, and each point is labeled by that concept. For these samples we obtain a second characterization of one-way functions.

1.1. **Our Results.** Table 1 summarizes the results, with the data distribution, the assumption and the approximation allowed in each.

Table 1: Selected results for charged local descriptions. Hardness is eventual for every PPT heuristic; easiness without one-way functions holds on infinitely many point lengths.

Setting	Assumption and approximation	Guarantee
Independent uniform points and labels	One-way functions; any fixed $a\lceil \log m \rceil$ additive tolerance	Common inverse-polynomial failure bound; characterization in Corollary 1.4
PAC-style samples	One-way functions; additive tolerance $m/4$	Constant failure bound on one sampler; Corollary 1.5
Every PAC-style sampler	No one-way functions; separated or inverse-polynomially shrinking quantiles	Any inverse-polynomial error at the given adequate clock; Theorem 1.2 and Corollary 5.2
Exact quantile recovery	No one-way functions; lower margin or independent random threshold	Exactness at the same clock; Corollary 1.6 and Theorem 1.7
Global versus local	No cryptographic assumption	Linear separation on cyclic data; typical PAC reverse bound; Theorem 1.8 and Corollary B.3

All logarithms have base two. Write n for the point length, m for the sample size, and t for the time allowed for each query, the query clock. A dataset has ordered point vector $x = (x_1, \ldots, x_m)$, with each $x_i \in \{0,1\}^n$, and label vector $y = (y_1, \ldots, y_m) \in \{0,1\}^m$. Fix an efficiently universal pointwise decoder U, whose precise conventions are in Section 2; simulating a program on U has polynomial overhead in the simulated time and in the lengths of the program, the list of retained points and the query.

A certificate selects k indices from $\{1, \ldots, m\}$, supplies one arbitrary tag bit per retained point, and supplies a finite binary program β. Its charged length is

$$\lceil \log(m+1) \rceil + \left\lceil \log \binom{m}{k} \right\rceil + k + |\beta|,$$

where $|\beta|$ is the number of program bits. The first two terms charge the retained-set size and its index in the lexicographic ordering of k-element subsets. We also call the charged length the price of the certificate.

On a query, U receives only the increasing-index list of retained points and tags, the program, the query, and a uniform public t-bit tape. If the run halts within t steps and its first output symbol is a bit, that bit is the returned label; otherwise, after a timeout or an invalid output, the label is zero. A certificate is valid when these returned labels agree with all the sample labels. It may be chosen after the public tape is revealed. For $0 < \theta \leq 1$, the quantile $\mathsf{pSC}^t_\theta(x,y)$ is the smallest nonnegative integer s such that, on at least a θ fraction of public tapes, some valid certificate has charged length at most s. Its value is ∞ if no such length exists. Section 2 gives the formal model and sampling conventions.

A PPT algorithm is a probabilistic polynomial-time algorithm; a heuristic is a PPT algorithm that receives 1^n and a dataset and outputs an integer or ∞. A polynomial clock is adequate for a PAC-style sampler D when it supplies the public bits and running time of the fixed coding program for D. That program applies an affine hash read from the public tape to its payload, the short string that follows its fixed header, to obtain a seed for the concept sampler, and evaluates the resulting concept. The formal definition is Definition 3.2. Throughout, a polynomial schedule means a polynomially bounded, polynomial-time computable integer schedule. For a PAC-style samplable distribution, both sampling procedures and concept evaluation run in polynomial time.

Uniform-data hardness. The first theorem gives hardness on a fixed distribution of independent uniform points and independent uniform labels.

Theorem 1.1 (Informal; restated as Theorem 4.5)**.** *Assume one-way functions exist. Fix a polynomial sample schedule $m(n) \geq 8n$, constants $0 < \theta_1 \leq \theta_2 \leq 1$, and an integer $a \geq 0$. There are polynomials $t_0(n)$ and $p(n) \geq 1$ with the following property. Every polynomial query clock $t \geq t_0$ permits lookup through a certificate retaining all sample points. At every such clock and on independent uniform points and labels, every PPT algorithm, which may output ∞, eventually has probability strictly below $1 - 1/p(n)$ of returning a value in*

$$[\mathsf{pSC}^t_{\theta_1}(X,Y) - a\lceil \log m \rceil, \mathsf{pSC}^t_{\theta_2}(X,Y) + a\lceil \log m \rceil].$$

Taking $a = 0$ and $\theta_1 = \theta_2$ gives hardness of exact computation at any fixed positive quantile. The failure bound $1/p(n)$ is the same for all algorithms. The polynomials p and t_0, however, may depend on the schedule m, the quantile levels θ_1, θ_2 and the tolerance coefficient a.

The converse at the given clock. The converse applies to every PAC-style sampler at every adequate query clock, without enlarging that clock. Its algorithm returns a value between two separated quantiles, with any requested inverse-polynomial error on infinitely many point lengths. Here D_n denotes the distribution of datasets that the sampler D produces at point length n.

Theorem 1.2 (Informal; restated as Theorem 5.1)**.** *Assume one-way functions do not exist. Fix a polynomial sample-size schedule $m(n) \geq 1$ and a PAC-style samplable D. Fix also a polynomial clock t adequate for D, constants $0 < \theta_1 < \theta_2 \leq 3/4$, and a positive polynomial $p(n) \geq 1$. There is a PPT algorithm $B = B_{D,t,p,\theta_1,\theta_2}$ such that, for infinitely many n,*

$$\Pr_{(x,y)\sim D_n, B}\left[\mathsf{pSC}^t_{\theta_1}(x,y) \leq B(1^n,x,y) \leq \mathsf{pSC}^t_{\theta_2}(x,y)\right] \geq 1 - \frac{1}{p(n)}.$$

Easiness on infinitely many lengths is exactly the negation of eventual hardness. The theorem does not assert easiness at all large lengths or exact computation of a single quantile.

Hardness on concept-labeled samples. For PAC-style samples, the hardness direction gives hardness even with the additive tolerance $m/4$, much larger than the logarithmic tolerance of Theorem 1.1: every algorithm eventually fails with probability greater than 0.1.

Theorem 1.3 (Informal; restated as Theorem 6.2)**.** *Assume one-way functions exist and fix a polynomial sample-size schedule $m(n) \geq 8n$. There are a PAC-style samplable D and an adequate polynomial time schedule t such that every PPT B, for all sufficiently large n, satisfies*

$$\Pr_{D_n,B}\left[\mathsf{pSC}^t_{1/2}(x,y) - m/4 \leq B(1^n,x,y) \leq \mathsf{pSC}^t_{3/4}(x,y) + m/4\right] < 0.9.$$

These results give two characterizations of one-way functions. For uniform data, the hardness direction is Theorem 1.1. For the matching converse we draw the concept from an independent function family, whose values on any m distinct points are independent uniform bits. Repeated points receive the same label, and repeated points occur with negligible probability for a polynomial sample size. The resulting PAC-style distribution is therefore statistically close to independent uniform points and labels (Lemma 4.8), so Theorem 1.2 transfers infinitely-often easiness to uniform data.

The uniform-data characterization. Let D_{ind} draw uniform points and independently choose a Boolean function whose values on every set of at most m distinct inputs are independent uniform bits. The function family has polynomial-time sampling and evaluation, as constructed in Lemma 4.8. The next statement requires its clock to be adequate for this fixed sampler D_{ind} and to permit lookup through a certificate retaining all sample points. For the hardness direction, Theorem 1.1 holds at every clock above a fixed polynomial, so a single clock can meet both requirements. The quantile is not known to be monotone in the clock, so hardness at a larger clock does not follow from hardness at a smaller one. It follows instead from the proof of Theorem 1.1, which uses only a short certificate for the generator-derived labels and the counting bound of Lemma 3.1. Both of these remain valid at every larger clock (Theorem 4.5 and Corollary 5.4).

Corollary 1.4 (Informal; restated as Corollary 5.4)**.** *Fix a polynomial sample schedule $m(n) \geq 8n$, an integer $a \geq 0$, and the efficiently universal decoder of Section 2.1. One-way functions exist if and only if the following holds for some polynomial clock t, adequate for D_{ind} and permitting lookup through a certificate retaining all sample points, and some positive polynomial $p(n) \geq 1$. Every PPT algorithm eventually has success strictly below $1 - 1/p(n)$ on independent uniform points and labels in returning a value in*

$$[\mathsf{pSC}^t_{1/2} - a\lceil \log m \rceil, \mathsf{pSC}^t_{3/4} + a\lceil \log m \rceil].$$

Without one-way functions, every such clock admits a heuristic with any requested inverse-polynomial error on infinitely many lengths. Repeated uniform points can receive conflicting labels, an event of negligible probability; on such datasets the minimum certificate length is infinite, and algorithms may output ∞. Both directions treat these datasets as part of the distribution.

An unconditional estimator complements this characterization. Given a nonnegative integer tolerance d, it returns ∞ on datasets with conflicting repeated points and otherwise the price of the lookup certificate retaining all sample points, decreased by d but not below zero. For quantile endpoints $0 < \theta_1 \leq \theta_2 \leq 1$ and a clock permitting this lookup, Theorem 4.6 bounds the probability that it fails to return a value in $[\mathsf{pSC}^t_{\theta_1} - d, \mathsf{pSC}^t_{\theta_2} + d]$ on independent uniform points and labels by

$$\min\left\{1, \frac{2^{c_{\mathrm{lookup}}+1}(m+1)}{\theta_2} 2^{-2d}\right\},$$

where c_{lookup} is the fixed lookup-program length. Thus every requested inverse-polynomial error $1/p$ is attainable at tolerance $O(\log m + \log p)$. This does not contradict the hardness theorem: there the tolerance coefficient a is fixed first and the error polynomial p is chosen afterward, whereas the estimator needs a tolerance that grows with $\log p$.

The PAC-style characterization. For the PAC-style characterization, Theorem 1.3 supplies hardness even with tolerance $m/4$. Since the interval without tolerance is contained in the interval with tolerance $m/4$, this hardness also holds without the tolerance. Conversely, Theorem 1.2 gives infinitely-often easiness for the same sampler and adequate clock if one-way functions do not exist. This yields the following equivalence.

Corollary 1.5 (Informal; restated as Corollary 6.3)**.** *Fix any polynomial sample-size schedule $m(n) \geq 8n$ and the efficiently universal decoder of Section 2.1. The following are equivalent:*

(1) *One-way functions exist.*
(2) *There are a PAC-style samplable D and an adequate polynomial t for which every PPT algorithm eventually has success below 0.9 in returning a value between $\mathsf{pSC}^t_{1/2} - m/4$ and $\mathsf{pSC}^t_{3/4} + m/4$.*
(3) *There are a PAC-style samplable D, an adequate polynomial t, and a positive polynomial $p(n) \geq 1$ for which every PPT algorithm eventually has success strictly below $1 - 1/p(n)$ in returning a value in $[\mathsf{pSC}^t_{1/2}, \mathsf{pSC}^t_{3/4}]$.*

The converse works at the given adequate query clock, while the hardness construction chooses a suitable clock.

The PAC-style characterization also holds when the two probability thresholds approach $2/3$ at any inverse-polynomial rate. For a polynomial-time computable integer polynomial $\nu(n) \geq 12$, take the endpoints $2/3 - 1/\nu(n)$ and $2/3 + 1/\nu(n)$. Corollary 6.4 states that one-way functions exist if and only if, on a suitable PAC-style distribution, every PPT heuristic eventually succeeds with probability strictly below 0.9 in returning a value between those quantiles. Without one-way functions, Corollary 5.2 gives any requested inverse-polynomial error on infinitely many lengths, without increasing the decoder clock. The shrinking window narrows the probability thresholds; the interval of description lengths may contain several integers. A converse that computes a single fixed quantile exactly remains open (Open Problem 1).

When exact recovery is possible. A separated pair of quantiles tolerates uncertainty in the public-tape probabilities, but exact recovery at $2/3$ does not follow from that argument. Call the $2/3$ quantile the crossing price: the cheapest price whose cumulative public-tape probability reaches $2/3$. The following consequence assumes only that the probability at the next cheaper price stays a margin below $2/3$.

Corollary 1.6 (Informal; restated as Corollary A.1)**.** *Assume one-way functions do not exist. Fix a PAC-style sampler, its adequate clock, and a requested inverse-polynomial error. Suppose that, at all sufficiently large point lengths and except on a suitably small fraction of datasets, the following margin holds. The public-tape probability of admitting a certificate strictly cheaper than the* $2/3$ *quantile is below* $2/3$ *by at least a fixed inverse polynomial. Then a PPT algorithm computes that quantile with the requested error on infinitely many point lengths.*

Exactness is also possible without a margin assumption if the threshold is chosen at random. A random threshold from a fine grid is, with high probability, far from the polynomially many cumulative probabilities at the relevant integer prices.

Theorem 1.7 (Informal; restated as Theorem A.2)**.** *Assume one-way functions do not exist. For every PAC-style sampler, adequate clock and inverse-polynomial requested error, there is an explicit polynomial-size grid of thresholds in* $(1/2,3/4)$ *and one PPT estimator. The estimator receives a uniformly random grid index as input. On infinitely many point lengths, it outputs the quantile at the corresponding threshold with the requested error, where the probability is over the dataset, the grid index and the estimator's coins. The decoder clock is unchanged.*

The grid depends on the sampler, the sample-size schedule and the error bound. This average over thresholds does not give exact recovery at every threshold or at the fixed value $2/3$. Both results are proved in Section A.

Global and pointwise descriptions. A global decoder V receives the complete ordered point vector and outputs all labels. Write $E(x)$ for a self-delimiting encoding of the complete ordered point vector. A global program receives $E(x)$ and outputs all labels within time T; it may be selected after the public tape is revealed, and only its bits are charged. Write $\mathsf{pK}^T_{\mathrm{all},\theta}(y|E(x))$ for the resulting global public-randomness description quantile. The formal machine and encoding appear in Definition B.1.

Theorem 1.8 (Informal; restated as Theorem B.2)**.** *Fix polynomial schedules* $m(n)\geq 1$ *and* $t(n)\geq 1$*, the pointwise decoder* U*, and the global decoder* V*. There are a polynomial global clock* T *and constants* c, c_{cyc} *such that, for every dataset and every* $0<\theta\leq 1$*,*

$$\mathsf{pK}^T_{\mathrm{all},\theta}(y|E(x)) \leq \mathsf{pSC}^t_\theta(x,y)+c.$$

Draw $X_i=(B_i,\widetilde{X}_i)$ *independently and uniformly from* $\{0,1\}^n$*, where* B_i *is the first bit and* $\widetilde{X}_i$ *is the remaining suffix. Set* $Y_i=B_{i+1}$*, with* $B_{m+1}=B_1$*. Then, for every* $0<\theta\leq 1$*, the global quantile* $\mathsf{pK}^T_{\mathrm{all},\theta}(Y|E(X))$ *is at most* c_{cyc} *on every point vector, whereas*

$$\Pr_X[\mathsf{pSC}^t_\theta(X,Y)\leq s]\leq 2^{2s+2-m}/\theta \qquad (s\in\mathbb{Z}_{\geq 0}).$$

In particular, the local price exceeds $\lfloor m/3\rfloor$ with exponentially high probability for fixed positive θ, while the global price stays constant. The global clock T is at least t and large enough for two programs: the compiled certificates, which evaluate a local certificate on every sample point, and the cyclic-label program. The cyclic labels depend on the full point vector rather than on a concept drawn independently of the points, so this separation does not settle the reverse comparison on PAC-style samples. That comparison bounds the local price by the global price plus an allowance; the next paragraph states this bound. The complete proof of Theorem 1.8, with a version conditioned on distinct points, is in Section B.

For PAC-style samples, the separation is limited by a reverse bound. Draw a dataset from a PAC-style sampler. Fix any global clock T, any $0<\theta\leq 1$ and any $0<\varepsilon<1$. Let r be the global θ-quantile price of the dataset at clock T. Then, at a local clock adequate for the sampler, the following holds with probability at least $1-\varepsilon$ over the dataset. If r is finite, then the local $3/4$-quantile price is at most

$$r+2\log(r+1)+\log\frac{1}{\theta\varepsilon}+\lceil\log(m+1)\rceil+c_Q+5,$$

where c_Q is a constant depending on the sampler. Corollary B.3 proves this using the PAC coding bound and counting global programs.

1.2. **Our Techniques.** The hardness direction plants a short pointwise description in labels computed from the output of a conditional pseudorandom generator, and shows by counting that an accurate estimator would distinguish the generator from uniform. The converse inverts a function that generates certificates and uses the inverter to recover tape minima, the minimum certificate lengths on individual public tapes.

For the uniform-data theorem, we use a generator whose output no polynomial-time test distinguishes from uniform with more than a small constant advantage, after its seed is conditioned on a suitable nonempty event. The first seed-length prefix of this conditional output has Shannon entropy within a logarithmic term of the seed length. The conditional generator is imported from Liu–Pass [LP25, Definition A.1 and Lemma A.3].

The labels are computed from this output as follows. A query point is mapped through a small-bias space [NN93, AGHP92] to a vector whose inner product with every fixed nonzero vector is nearly balanced between zero and one on a uniform point. The inner product of this vector with the generator output is the label of the query point. We call a sample planted when its labels are produced in this way from a conditional generator output.

The vectors of the independent query points form the rows of a measurement matrix. A short program stores a generator seed and evaluates the corresponding measurement on each new query, without receiving the sample points. Simply treating the entire label vector as a compressed string would not provide this pointwise evaluator. The sample's labels are the measurements of the vector built from the generator output at the m query points. A collision bound (Lemma 4.4) shows that these labels keep nearly all of that vector's entropy given the points, while a rank bound makes a uniform generator output close to independent uniform labels. On most planted samples, the entropy bound controls the ratio of their planted probability to their uniform probability. This ratio bound transfers a heuristic's success on uniform data to success on planted data.

Local meta-complexity uses entropy-preserving generators and conversion of entropy bounds into density bounds [LP21a]. Carmosino and Juvekar use entropy-preserving generators in the proof of their hardness theorem for parallel Kolmogorov complexity [CJ26, Theorem 1.12, stated formally as Theorem 7.18]. In our setting, the entropy and rank estimates for the measurement matrix concern independent random query rows, and Lemma 4.4 proves them. Certificates that retain points are handled by counting (Lemma 3.1), which charges every retained-set choice and tag.

The reduction from an estimator to a distinguisher uses the point length itself as the generator's security parameter. It keeps the first $n + O(\log n)$ output bits of the generator on an n-bit seed and appends independent uniform bits, obtaining a vector slightly longer than the sample's label vector. The pointwise certificate stores the seed and the appended independent bits, whose total length remains below the sample size by a logarithmic amount. A uniform string in place of the generator output gives nearly independent uniform labels, while a conditional generator output gives the planted distribution. The same reduction applies at each point length, including when the sample-size schedule is nonmonotone.

A counting argument bounds the number of label vectors with short certificates. Public-randomness coding is established in time-bounded description complexity [LOZ22]. The coding argument reads a hash function from the public tape, an affine extension of Carter–Wegman's binary linear family [CW79], in the form stated by Liu–Pass [LP23, Proposition A.4]. The hashing argument follows Hirahara–Nanashima's coding lemma for consistent concept seeds [HN23, Lemma C.6] and Liu–Pass's implementation [LP23, Appendix A]; here the hash maps short payloads directly to concept-sampler seeds that produce the required labels. This gives a coding bound in terms of conditional label probability. The coding bound ensures that a target dataset is not too unlikely under a generator of certificates at the relevant lengths. The certificate generator of the converse reproduces the full charged representation and preserves the public tape, while the pointwise decoder still receives only the query, the retained points with their tags, the program and the public tape.

The inversion-and-domination method follows Liu–Pass [LP23]. Our generator explicitly charges subset ranks, tags and programs and preserves the original decoder clock. We call the input lengths at which the inverter succeeds successful. Strictly increasing padding aligns them with infinitely many point lengths n, even for nonmonotone sample schedules. Verified inversions recover enough tape minima for empirical quantile estimation; the algorithm need not identify which lengths are successful.

For the PAC-style hardness, one-way functions supply short pseudorandom-function keys, and these keys together with the independent-function sampler give two distributions with sharply separated certificate lengths. An algorithm that approximates the quantile price accurately would therefore distinguish the two distributions. This comparison uses the established mixture of pseudorandom and random distributions [IRS22, Section 4.1 of ECCC TR21-082]. The independent-function branch supplies a concept independent of the points and exactly reproduces random-function labels on every sample. For the matching uniform-data converse, a standard independent function family supplies nearly uniform labels on random points [Vad12].

The local/global separation uses cyclic labels. Its counting step adapts the directed-cycle guessing-game bound of Riis [Rii07, Section 3], by fixing retained first bits before counting cyclic assignments.

1.3. **Other Related Work.** Liu–Pass's characterization [LP20] has been extended in several directions. Ren–Santhanam characterize parallel cryptography through the hardness of KT [RS21]. Liu–Pass study the hardness of Kt and the possibility of basing cryptography on $\mathsf{EXP} \neq \mathsf{BPP}$ [LP21b], and relate one-way functions to the hardness of conditional time-bounded Kolmogorov complexity [LP22]. Hirahara captures one-way functions through the NP-hardness of meta-complexity [Hir23], and Hirahara–Lu–Oliveira relate one-way functions to the gap between two-sided and one-sided error in the average-case complexity of approximating pKt [HLO25]. Hirahara also proves NP-hardness of learning programs and of the minimum circuit size problem for partial functions [Hir22]. A consistent labeled sample specifies such a partial function, but our task is the average-case estimation of a charged local description length.

Allender–Cheraghchi–Myrisiotis–Tirumala–Volkovich study average-case hardness of conditional KT minimization [ACM$^+$21]. Their indexed decoder receives the full conditioning string, and the objective charges program length plus runtime. Here the pointwise clock is fixed and the objective is a public-tape quantile of the charged certificate length.

Liu–Pass later treat probabilistic descriptions under arbitrary samplable distributions and use coding, distribution domination, and inversion to obtain infinitely-often easiness without one-way functions [LP23]. Liu–Pass also study local decompression, in which a program answers a requested coordinate of a truth table [LP21a]. Our uniform-data theorem, in contrast, uses independently sampled query points and charges retained-point choices and tags as well as program bits.

Liu–Pass's recent boundary characterization uses randomized programs, so that one program must succeed on many tapes [LP26]. Its boundary problem compares two time bounds. Our certificate quantifiers allow a different certificate on each tape, and our converse is proved at the given query clock. Hirahara–Lu–Nanashima obtain infinitely-often efficient average-case conditional coding for randomized Kolmogorov complexity without one-way functions [HLN24, Theorem 4.11 of the full version]. Their reconstruction polynomial is chosen by the coding theorem, and one randomized program must produce the whole output with constant probability. Here the decoder clock is given, its inputs are local, and the certificate may vary with the public tape.

Hirahara–Ilango–Lu–Nanashima–Oliveira show that average-case conditional coding for probabilistic time-bounded Kolmogorov complexity over every samplable distribution of pairs holds if and only if infinitely-often one-way functions do not exist [HIL$^+$23, Theorem 1 of the full version]. Their decoder reads the whole conditioning string, and their time bound is chosen by the theorem. Our PAC-style coding bound in Lemma 3.3 needs no assumption, because the concept is drawn independently of the points.

The PAC-style coding bound is a variant of Lu–Oliveira–Zimand's general public-randomness coding lemma [LOZ22, Lemma 31 of the full version], which gives success probability $2/3$ with a function family produced by a pseudorandom generator for constant-depth circuits. We prove the version used here directly; the selected concept seed can then be evaluated on any query point. Our affine implementation follows the established hashing method in Liu–Pass's Appendix A and in the proof of Hirahara–Nanashima's Lemma C.6, and records an explicit public-tape budget and additive constant.

Ilango–Ren–Santhanam use a mixture of pseudorandom-function truth tables and uniform strings to obtain average-case meta-complexity hardness [IRS22]. Our hardness proof adapts this mixture to independently sampled query points, using the independent-function sampler for the second branch. Building on their method, Zimand characterizes one-way functions by the average-case hardness of almost-optimal compression [Zim24, Theorem 1]. If one-way functions exist, then there are a polynomial-time samplable distribution and a constant c with the following property. Every efficient algorithm fails with probability greater than $1/100$ to output a program for the sampled string whose length is within $c \log n$ of its Kolmogorov complexity. There the program describes the whole string and the program's running time is unbounded; our certificates answer one query at a time within the query clock.

Among the predecessors cited above, Hirahara–Nanashima's study of minimum consistent pointwise programs under separated distributions is closest to our converse [HN23]. Their Appendix C derives exact search at every query clock for which a coding hypothesis holds, assuming that infinitely-often one-way functions do not exist [HN23, Lemma C.3]. The same appendix obtains deterministic coding for separated distributions under an additional derandomization assumption, at clocks at least a polynomial chosen by their coding lemma [HN23, Lemma C.6 and Theorem C.7]. Their coding proof already uses the set of concept seeds consistent with the sample. Our public tape permits this coding method without their derandomization assumption.

Lu–Santhanam study two-threshold conditional probabilistic complexity under arbitrary samplable joint distributions [LS24]. Their decoder can inspect the entire conditioning string, whereas ours sees only the retained points and the current query. Their eventual easiness assumes that infinitely-often one-way functions do not exist; our converse assumes only the absence of one-way functions and concludes infinitely-often easiness.

Goldberg–Kabanets–Lu–Oliveira define conditional probabilistic complexity with a program chosen after the public tape [GKLO22, Definition 17]. This order of choices is already part of the established whole-output measure. A whole-conditioning program cannot be used unchanged as our hypothesis: its input includes points unavailable to the local decoder. In the other direction, a global decoder can simulate a certificate by evaluating it on every sample point, but this requires all of those evaluations and an encoding of the retained-point choices. Theorem B.2 makes this compiler comparison explicit and gives a linear separation for general labeled data, using the same public-tape order in both models. The global quantity defined there is a comparison model; it is not identified with a randomized description whose program must be chosen before the tape.

Hirahara–Nanashima's later characterization concerns a multiplicative-gap agnostic learning problem with a sampling-advice parameter [HN26]. Its title refers to Pessiland, one of Impagliazzo's five worlds of average-case complexity [Imp95]. Its objective is prediction over an example distribution. Here the objective is exact agreement on a given finite dataset, measured by two public-tape quantiles of certificate length. The retained-point indices, their arbitrary tags, and the side program are charged, and the certificate is selected after the public tape is revealed.

The connection between compression of random partial truth tables and uniform PAC learning is stated by Ilango–Loff–Oliveira [ILO20, Appendix A, Theorem 36], who describe it as implicit in earlier work. Their comparison tests small circuit representations against independent labels and uses holdout examples and learning reductions. Our task estimates the shortest charged local certificate on a given sample, including its public-tape quantiles. The connection between random partial-function compression and learning is prior work; the contribution here is the quantitative uniform-data description-estimation theorem.

Liu–Pass's sublinear-time characterization uses indexed local decompression and an average-case notion that conditions separately on yes and no instances [LP21a, Definition 2.4 of the full version]. Its equivalence is stated for nonuniform security. Carmosino–Juvekar study parallel local descriptions and establish related hardness results under stronger pseudorandom-function assumptions [CJ26]. Their model has indexed access and charges program length plus processor count, with a time cap. These local-description results are direct precedents; our uniform-data theorem differs in using ordinary uniform one-way functions and independently sampled queries, with all retained-set choices charged. The survey of Goldberg–Nanashima–Oliveira places sample compression, partial-function minimization and learning in their broader meta-complexity context [GNO26].

1.4. **Discussion.** The characterization concerns a fixed universal decoder and description-length pricing. A short certificate may consist of retained-point choices, arbitrary tags, or a program; the theorem bounds their total charged length. It does not provide a scheme with a bounded number of retained examples. The public tape is revealed before the certificate is selected, so different tapes may have different shortest certificates. This order of choices is part of the computational problem. The planted short descriptions in both hardness proofs retain no sample points. The lower bound nevertheless counts the larger certificate class allowing retained points. These two facts do not establish that retaining points is essential to either characterization.

One-way functions give eventual hardness for each fixed heuristic, with one inverse-polynomial failure bound shared by all heuristics. For each requested inverse-polynomial error bound, their absence gives an estimator that succeeds on infinitely many point lengths. The converse does not efficiently identify those point lengths. Suppose the inverter used in the converse succeeded at every sufficiently large input length, rather than only at infinitely many. Since padding assigns input lengths to each point length, every sufficiently large point length would then have a successful input length. The converse's proof would then give easiness at every sufficiently large point length.

The hardness direction also makes additive $m/4$ approximation to the single $2/3$ quantile hard, because such an approximation belongs to the tolerant interval in Corollary 1.5.

The shrinking-window converse brings the two probability thresholds arbitrarily close at an inverse-polynomial rate while preserving the given adequate clock. The returned length is still between two quantiles. Certificate prices are integers, and their cumulative public-tape probabilities are multiples of 2^{-t}; nothing in our analysis keeps these probabilities an inverse polynomial away from $2/3$. An inverse-polynomial window around $2/3$ need not determine on which side of that threshold a cumulative probability lies, and therefore need not identify the exact quantile. Section A proves exact recovery under a lower margin and at an independent random threshold. Neither result resolves the cumulative probability at a prescribed threshold for all datasets in the support of the sampler.

1.5. **Open Problems.** We state one open problem, about exact recovery of a single quantile when one-way functions do not exist.
A single quantile. The current converse estimates a value between two quantiles.

Open Problem 1. Assume one-way functions do not exist. Fix a polynomial sample-size schedule $m(n) \geq 1$, a PAC-style samplable D and an adequate polynomial time schedule t. For every positive polynomial $p(n) \geq 1$, is there a PPT algorithm B such that $\mathbf{Pr}_{(x,y)\sim D_n,B}[B(1^n,x,y) = \mathsf{pSC}^t_{2/3}(x,y)] \geq 1 - 1/p(n)$ for infinitely many n?

The shrinking-window result narrows the two thresholds without identifying an exact crossing. The lower-margin and random-threshold theorems give complementary partial answers, but neither supplies the fixed-threshold guarantee in this problem. Write $g^t_w(x,y)$ for the minimum charged length on tape w. At a candidate price s, exact computation must decide whether $\mathbf{Pr}_w[g^t_w(x,y) \leq s]$ is below $2/3$ or at least $2/3$ even when that probability is arbitrarily close to $2/3$. The open problem retains the fixed universal decoder and its original adequate clock.

AI Disclosure. *Substantial AI contribution.* This work was developed with substantial assistance from Codex and Claude Code. These tools were used to discover almost all of the proofs in this work, find relevant literature, and write parts of the prose; the author wrote the rest. The author posed the problem, guided the development, and is responsible for any errors.

1.6. **Paper Organization.** Section 2 defines the measure, sampling model and one-way-function convention. Section 3 develops the counting and coding bounds. The main uniform-data result follows in Section 4. Section 5 gives the inversion-based converse and the matching uniform-data characterization, and Section 6 gives PAC-style hardness. The main text concludes with the PAC-style characterization and its shrinking-window consequence (Corollaries 6.3 and 6.4). Section A proves the lower-margin and random-threshold exact-recovery results in the first appendix. Section B proves the comparison between global and pointwise descriptions in the second appendix. The AI disclosure before this subsection identifies the tools used in developing this work.

2. PRELIMINARIES

This section fixes notation, the formal certificate model and the sampling conventions, and then recalls the cryptographic definitions used later. All logarithms have base two. The point length and security parameter n is a positive integer. A probabilistic polynomial-time algorithm, abbreviated PPT, uses independent uniform random bits and has a running time bounded by a fixed polynomial in its input length. For a positive integer m, write $[m] = \{1, \ldots, m\}$, and write U_ℓ for the uniform distribution on ℓ-bit strings. For a finite set A, the number of its elements is denoted by $|A|$. For a binary string v, the notation $|v|$ denotes its length. The symbols $\lceil z \rceil$ and $\lfloor z \rfloor$ denote the least integer at least z and the greatest integer at most z, respectively. The string 1^n consists of n ones and supplies the security parameter in unary.

The binary field $\mathbb{F}_2 = \{0,1\}$ uses addition and multiplication modulo two; $\oplus$ denotes coordinatewise addition in this field. For $u, v \in \mathbb{F}_2^L$, their inner product is $\langle u, v \rangle = \sum_{i=1}^{L} u_i v_i$, computed modulo two.

For nonnegative functions f, g of n, the notation $f = O(g)$ means that there is a constant C with $f(n) \le C g(n)$ for all sufficiently large n. When g is positive, the notation $f = o(g)$ means that the ratio $f(n)/g(n)$ tends to zero. Constants in these bounds may depend on previously fixed samplers, algorithms and parameters, but not on n. The notation poly denotes an unspecified polynomial bound in its listed arguments. We write $\exp(v) = e^v$ for the natural exponential and ln for its inverse, the natural logarithm; every unmarked log still has base two.

A finite random variable assigns a value to each outcome of a finite probability experiment. Its distribution assigns each value its probability, and its support consists of the values with positive probability. For an event $\mathscr{A}$, its indicator $\mathbf{1}_{\mathscr{A}}$ is one on that event and zero elsewhere. For a real-valued random variable Z, its expectation is $\mathbb{E}[Z] = \sum_z z \mathbf{Pr}[Z = z]$, and its variance is $\mathrm{Var}(Z) = \mathbb{E}[(Z - \mathbb{E}[Z])^2]$. If $\mathbf{Pr}[\mathscr{B}] > 0$, conditional probability means $\mathbf{Pr}[\mathscr{A} \mid \mathscr{B}] = \mathbf{Pr}[\mathscr{A} \cap \mathscr{B}] / \mathbf{Pr}[\mathscr{B}]$; conditional distributions use this formula for each value. Random variables are independent when their joint probabilities are the products of their marginal probabilities; pairwise independence requires this only for each pair. These conventions also apply after fixing a conditioning value of positive probability.

The following probability bounds will control the errors of our estimates.

Theorem 2.1 (Markov and Chebyshev inequalities [Vad12, Lemmas 2.20 and 3.27])**.** *For every nonnegative finite random variable Z and real $u > 0$,*

$$\mathbf{Pr}[Z \ge u] \le \frac{\mathbb{E}[Z]}{u}.$$

For every real-valued finite random variable Z and real $u > 0$,

$$\mathbf{Pr}[|Z - \mathbb{E}[Z]| \ge u] \le \frac{\mathrm{Var}(Z)}{u^2}.$$

For events $\mathscr{A}_1, \ldots, \mathscr{A}_k$, their union has probability at most $\sum_{i=1}^{k} \mathbf{Pr}[\mathscr{A}_i]$. This union bound follows by taking expectations in $\mathbf{1}_{\bigcup_i \mathscr{A}_i} \le \sum_i \mathbf{1}_{\mathscr{A}_i}$. A random variable Z with values in $\{0,1\}$ and mean v has variance $v(1-v) = 1/4 - (v - 1/2)^2 \le 1/4$. For pairwise independent real variables, expanding the square in the variance of their sum leaves only the individual variances: every mixed term is zero because $\mathbb{E}[(Z_i - \mathbb{E}[Z_i])(Z_j - \mathbb{E}[Z_j])] = 0$ when $i \ne j$. Hence, for every positive integer T, the sum of T independent $\{0,1\}$-valued variables has variance at most $T/4$. Their average has the variance of the sum divided by T^2, which is at most $T/(4T^2) = 1/(4T)$.

An error is negligible if, for each inverse polynomial, it is at most that inverse polynomial at all sufficiently large parameters.

Definition 2.2 (Negligible error)**.** A nonnegative function $\varepsilon(n)$ is negligible if, for every positive polynomial p, it satisfies $\varepsilon(n) \le 1/p(n)$ at all sufficiently large n.

We compare probability distributions by the following distance.

Definition 2.3 (Total variation distance)**.** For distributions P, Q on the same finite set, their total variation distance (also called statistical distance) is $\mathrm{TV}(P, Q) = \frac{1}{2} \sum_z |P(z) - Q(z)|$.

The total variation distance between P and Q bounds the difference between the probabilities that they assign to any event, including the acceptance event of a randomized test.

Computational indistinguishability restricts the comparison to efficient tests. The notion goes back to Goldwasser–Micali and Yao [GM84, Yao82].

Definition 2.4 (Computational indistinguishability)**.** Two sequences of distributions, indexed by the security parameter n, are computationally indistinguishable if every PPT test given 1^n has a negligible difference in acceptance probability on them.

A distribution over functions is accessed through an oracle, which answers each queried string with the function's value there. A test with such oracle access may make polynomially many adaptive queries, and the same acceptance-probability condition defines indistinguishability of two distributions over functions.

A pseudorandom generator is a deterministic polynomial-time map G that sends every r-bit seed to a string of a fixed length $\ell(r) > r$, where ℓ is polynomially bounded and computable. Its security requirement is that $G(U_r)$ and $U_{\ell(r)}$ are computationally indistinguishable for tests given 1^r. The seed is the generator's random input; its stretch is $\ell(r) - r$.

2.1. **Certificates and Sampling.** Fix an efficiently universal Turing machine U once and for all. Every program for U consists of a self-delimiting header followed by an arbitrary finite payload. The header names a fixed procedure, its length depends only on that procedure, and it shows where the payload starts.

Simulating a program on U has polynomial overhead in the simulated running time and in the lengths of the program, the anchor list and the query defined below. This overhead does not depend on the total public-tape length, because public bits are read sequentially as the simulated procedure requests them. The program and every random tape the machine reads, such as the public tape defined below, are sequential inputs, read one symbol at a time; the machine is not told their lengths. Each step reads at most one new symbol of a sequential input, so a run of a given number of steps reads at most that many initial symbols of each.

A polynomial schedule is a polynomially bounded, polynomial-time computable integer function of n; it need not be a polynomial expression. Throughout, $m = m(n) \geq 1$ and $t = t(n) \geq 1$ denote polynomial schedules, quantified in each statement. The schedule t is the query clock: the number of steps allowed for answering one query. A dataset consists of an ordered point vector $x = (x_1, \ldots, x_m)$ in $(\{0,1\}^n)^m$ and a label vector $y \in \{0,1\}^m$. A heuristic is a PPT algorithm that receives 1^n and a dataset; the input 1^n makes the security parameter explicit.

Definition 2.5 (Public-randomness sample certificate)**.** A certificate is a triple (A, a, β), where $A \subseteq [m]$ is the retained-point index set, $a \in \{0,1\}^{|A|}$ is a string of tags, and β is a finite binary string, the program. The points whose indices belong to A are the retained points, also called anchors. Write L_A for the anchor list: the increasing-index list of the anchors, with their tags from a. As a sequential input, L_A is the string $x_{i_1} a_1 \cdots x_{i_k} a_k$, where $i_1 < \cdots < i_k$ enumerate A and $a_1, \ldots, a_k$ are the tags. Each of the inputs β and L_A, and the query z introduced below, is followed by a blank end marker, the only symbol on these inputs that is not a bit.

Fix a public tape $w \in \{0,1\}^t$. The hypothesis of the certificate answers a query $z \in \{0,1\}^n$ by running $U(\beta, L_A, w, z)$ for at most t steps. If the run halts within t steps and its first output symbol is a bit, the hypothesis returns that bit; otherwise it returns zero. The hypothesis therefore returns zero after a timeout, meaning that the run has not halted within t steps. It also returns zero after an invalid output, meaning that the run halts without writing an output symbol or its first output symbol is not a bit. The hypothesis receives only L_A, w, z, β. It receives neither the other sample points nor their indices nor the sample size as free input.

The tags are arbitrary bits, charged in the price defined below, and they need not be the true labels. The certificate is valid on the tape w if, on that tape, the hypothesis returns y_i on the query x_i for every $i \in [m]$.

For $0 \leq k \leq m$, the binomial coefficient $\binom{m}{k}$ counts the k-element subsets of $[m]$. The rank of A is its zero-based index in the lexicographic ordering of the $|A|$-element subsets of $[m]$, each written as an

increasing sequence of indices. The certificate's charged length, also called its price, is

$$\lambda_m(k)+k+|\beta|, \qquad \lambda_m(k)=\lceil\log(m+1)\rceil+\left\lceil\log\binom{m}{k}\right\rceil, \qquad k=|A|.$$

The two terms in $\lambda_m(k)$ charge for the retained-set size and its rank, respectively. The rank is charged even though the pointwise hypothesis does not receive it. The additional k bits charge for the tags, and $|\beta|$ charges for the entire program. For each tape w, let $g^t_w(x,y)$ be the minimum charged length of a certificate valid on w, or ∞ if no such certificate exists. For $0<\theta\le 1$, the public-randomness sample-compression quantile is

$$\mathsf{pSC}^t_\theta(x,y)=\min\left\{s\in\mathbb{Z}_{\ge 0}:\mathop{\mathbf{Pr}}_{w}[g^t_w(x,y)\le s]\ge\theta\right\},$$

where w is uniform on $\{0,1\}^t$ and the minimum of an empty set is ∞.

Thus $\mathsf{pSC}^t_\theta(x,y)$ is the smallest nonnegative integer s such that, on at least a θ fraction of public tapes, some valid certificate has charged length at most s. Every finite minimum is attained, since the lengths are nonnegative integers. The order of quantifiers is public randomness first and the existence of a certificate second, so a certificate may depend on w. The quantiles are nondecreasing in θ, since the set in the definition shrinks as θ increases.

A heuristic is later asked to output a value between two quantiles, up to an additive tolerance, so we fix conventions for the value ∞. Quantiles take values in $\mathbb{Z}_{\ge 0}\cup\{\infty\}$. For every real $d\ge 0$ we set $\infty-d=\infty+d=\infty$, and every integer is smaller than ∞. Intervals are closed intervals of the ordered set $\mathbb{Z}\cup\{\infty\}$. Hence an interval whose endpoints are $\infty-d$ and $\infty+d$ contains exactly ∞. A heuristic estimating a quantile outputs either a binary integer or a fixed symbol for ∞; such an output is well formed, and any other output counts as a failure.

Full-anchor lookup retains $A=[m]$ and uses the observed labels as tags. Its fixed program scans the anchor list L_A sequentially, compares each stored point with the query, and returns the first matching tag, or zero if no point matches. The scan reaches the list's end marker only when no stored point matches; it uses no sample-size or index input. Its running time on U is polynomial in the lengths of the anchor list and the query; this bound covers the comparisons, the read of the end marker and the simulation overhead.

A clock permits full-anchor lookup if, at every point length, this program halts within the clock on every list of $m(n)$ tagged points and every query of that length. Call a dataset consistent when every repeated point has the same label at each occurrence. On a consistent dataset, each sample point x_i, used as a query, has a match and all matching tags agree, so this program is valid on every public tape at any clock that permits full-anchor lookup.

For example, suppose that every label is the first bit of its point. A fixed program of some constant length c_1 that returns the first bit of the query, with no anchors, is valid on every public tape at any clock that lets it halt on every n-bit query. Every quantile is then at most $\lambda_m(0)+c_1=\lceil\log(m+1)\rceil+c_1$. By contrast, Lemma 3.1 below shows that, for every point vector and every integer $0\le d\le m$, at most a $2^{1-d}/\theta$ fraction of all label vectors satisfy $\mathsf{pSC}^t_\theta(x,y)\le m-d$.

PAC stands for probably approximately correct; the distributions below draw a concept independently of the sampled points, as in the PAC model of learning.

Definition 2.6 (PAC-style samplable distribution)**.** A PAC-style samplable distribution $D=(\mu,Q)$ consists of a point sampler μ and a concept sampler Q. Both samplers run in polynomial time, and their fixed tape-length schedules are polynomial-time computable. The point sampler $\mu(1^n;\rho)$ uses a tape of length $r_\mu(n)$, a fixed polynomial schedule, and outputs an n-bit point. The concept sampler $Q(1^n;r)$ uses a tape of length $q(n)$, a fixed polynomial schedule, and outputs a deterministic concept program c_r. The concept evaluation $c_r(z)$ takes at most $\tau_Q(n)$ steps for every r,z of the indicated lengths, where τ_Q is a fixed polynomial schedule. Unused sampler random bits are padded so these tape lengths are fixed. Draw r uniformly, draw the x_i independently from μ and independently of r, and set $y_i=c_r(x_i)$. Write $D(y|x)$ for the conditional label probability and $\mu^m(x)$ for the point-vector probability:

$$D(y|x)=2^{-q(n)}\,|\{r:c_r(x_i)=y_i\text{ for all }i\}|, \qquad \mu^m(x)=\prod_{i=1}^{m}\mathbf{Pr}[\mu(1^n;\rho)=x_i].$$

The joint mass of (x,y) is the product $\mu^m(x)D(y|x)$; this product form expresses the independence of the points from the concept. That independence is the hypothesis used below. Write D_n for the resulting distribution of datasets at point length n and $\mathrm{supp}(D_n)$ for its support. A dataset is supported when it has positive probability under D_n; then $D(y|x) > 0$. We also call D a PAC-style sampler.

2.2. **Cryptographic Background.** A statement depending on the parameter n holds eventually if it holds for all sufficiently large n; when the statement concerns an algorithm, the threshold may depend on that algorithm. A one-way function is efficient to evaluate but hard to invert on a uniform input.

Definition 2.7 (One-way functions)**.** For a deterministic polynomial-time function f, the inversion experiment draws u uniformly from $\{0,1\}^n$ and gives the inverter $(1^n, f(u))$. The inverter succeeds if it outputs a string v with $f(v) = f(u)$. A one-way function is such a function whose inversion success is negligible for every PPT inverter. A weak one-way function is such a function with a positive polynomial R for which every PPT inverter fails eventually with probability at least $1/R(n)$. That is, each inverter has a length n_0 beyond which its failure probability at every $n \geq n_0$ is at least $1/R(n)$.

The converse in Section 5 needs inverters against the weak notion. Amplification shows that the weak notion is equivalent to the strong one.

Theorem 2.8 (Amplification [Yao82, Gol01, Gol95])**.** *Weak one-way functions exist if and only if one-way functions exist.*

The theorem is implicit in Yao's work [Yao82]. A full proof is in Goldreich's *Foundations of Cryptography*, Section 2.3 [Gol01]. Goldreich's earlier fragments state it as Theorem 2.8 with a full proof, for the same strong notion against PPT inverters given 1^n [Gol95, Definitions 2.1–2.2 and Theorem 2.8]. The weak notion there requires failure probability greater than $1/p(n)$. A function that is weak one-way in our sense with R is therefore weak one-way in that sense with $p = 2R$. Conversely, a function that is weak one-way in that sense with p is weak one-way in our sense with $R = p$. Liu–Pass state the same convention of PPT adversaries as ours, with the weak-hardness bound required for all sufficiently large n [LP20, Definitions 2.1–2.2 and Theorem 2.3]. We use Theorem 2.8 in the following contrapositive form. If no one-way function exists, then no weak one-way function exists, so for every polynomial-time function f and every positive polynomial R, some PPT inverter $\mathscr{I}$ satisfies

$$\Pr_{u\in\{0,1\}^N,\mathscr{I}}[\mathscr{I}(1^N, f(u)) \in f^{-1}(f(u))] \geq 1 - 1/R(N)$$

for infinitely many N. This inverter comes from negating eventual weak hardness, so the bound is asserted only for infinitely many N and not for all sufficiently large N.

3. Counting and Coding

The charged representation lets us count short certificates without specifying their decoding behavior.

Lemma 3.1 (Counting)**.** *For every query clock $t \geq 1$, every point vector x, every integer $s \geq 0$, and every $0 < \theta \leq 1$,*

$$\left|\left\{y : \mathsf{pSC}^t_\theta(x,y) \leq s\right\}\right| \leq \frac{2^{s+1}}{\theta}.$$

Moreover, for every $m \geq 1$ and every integer $s \geq 0$, at most 2^{s+1} certificates have charged length at most s.

Proof. We count certificates for each fixed public tape and then average over the tapes. Fix w. A certificate fixes its hypothesis on the tape w, so it is valid for at most one labeling, namely the vector of its returned labels on $x_1, \ldots, x_m$. Since $g^t_w(x,y)$ is a minimum over valid certificates, each labeling y with $g^t_w(x,y) \leq s$ has a valid certificate of length at most s. As a certificate is valid for at most one labeling, the number of such labelings is at most the number of certificates of length at most s.

For each anchor size k, there are $\binom{m}{k}2^k$ choices of the retained-point set and tags. If $J = s - \lambda_m(k) - k \geq 0$, the number of program strings of length at most J is $2^{J+1} - 1 < 2^{J+1}$. If $J < 0$, that number is

zero. Summing over k counts all certificates of length at most s; this count depends neither on w nor on the points. Since $2^{\lambda_m(k)} \geq (m+1)\binom{m}{k}$, this count is at most

$$\sum_{k=0}^{m} \binom{m}{k} 2^k 2^{s-\lambda_m(k)-k+1} \leq \sum_{k=0}^{m} \frac{2^{s+1}}{m+1} = 2^{s+1}.$$

Terms with $J < 0$ may be included in this upper bound because their actual count is zero. This proves the second assertion. If $\mathsf{pSC}^t_\theta(x,y) \leq s$, then the quantile definition and monotonicity of $\mathbf{Pr}_w[g^t_w(x,y) \leq s']$ in s' give $\mathbf{Pr}_w[g^t_w(x,y) \leq s] \geq \theta$. Averaging the labeling count over w now gives

$$\begin{aligned} \theta \left|\{y : \mathsf{pSC}^t_\theta(x,y) \leq s\}\right| &\leq \sum_y \mathbf{Pr}_w[g^t_w(x,y) \leq s] \\ &= \mathbb{E}_w \left|\{y : g^t_w(x,y) \leq s\}\right| \\ &\leq 2^{s+1}. \end{aligned}$$

Dividing by θ proves the claim. □

The upper bound below connects conditional probability to the length of a local hypothesis. The public tape lets the certificate select a concept-sampler seed that produces the required labeling.

The coding bound of Lemma 3.3 below is a variant of the coding lemma of Lu–Oliveira–Zimand for public randomness [LOZ22, Lemma 31 of the full version]. Their function family is produced by a pseudorandom generator for constant-depth circuits, and their lemma gives success probability $2/3$ with a payload whose length exceeds the negative logarithm of the target probability by a constant. Hashing the set of concept seeds consistent with a sample is the argument in the proof of Hirahara–Nanashima's coding lemma for separated distributions [HN23, Lemma C.6]. They derive that lemma from earlier work and combine it with a derandomization assumption. We use the affine binary hash family $v \mapsto Mv \oplus b$, which Liu–Pass attribute to Carter–Wegman [CW79] in [LP23, Proposition A.4]. Hirahara–Nanashima hash concept seeds and then index a preimage; here the public tape maps each short payload directly to a concept seed. We prove this version directly and specify the tape budget and the additive constant in our certificate model.

The coding program and the clocks it needs are defined first.

Definition 3.2 (Coding program and adequate clocks)**.** Fix a PAC-style sampler $D = (\mu, Q)$. Fix once and for all a self-delimiting header σ_Q, depending only on D, of length c_Q, for the following procedure. On a query z and payload v, it computes $n = |z|$ and $q = q(n)$. It reads the payload up to its end marker and sets $j = |v|$. It reads a $q \times j$ binary matrix M and a q-bit vector b from the first $q(j+1)$ public bits. It then computes $r = Mv \oplus b$, runs $Q(1^n; r)$, and returns the concept value $c_r(z)$. A polynomial schedule t is adequate for D if two conditions hold for every n. First, $t(n) \geq q(n)(q(n)+3)$. Second, let v be any payload of length at most $q(n)+2$. On every query of length n and every public tape, the program $\sigma_Q v$ with the empty anchor list must halt on U within $t(n)$ steps.

The first condition makes the public tape long enough to hold the hash matrix and vector that the program reads, and the second lets the program halt on every query; the proof of Lemma 3.3 uses both. The next lemma concerns an adequate clock and a supported dataset. It shows that, on at least three quarters of the public tapes, some payload for the coding program gives a valid certificate; the payload may depend on the tape. The price of this certificate exceeds the negative logarithm of the conditional label probability by at most $\lceil \log(m+1) \rceil + c_Q + 3$.

Lemma 3.3 (PAC-style coding)**.** *Fix a PAC-style sampler $D = (\mu, Q)$, with σ_Q and c_Q as in Definition 3.2. For every adequate t and every supported dataset,*

$$\mathsf{pSC}^t_{3/4}(x,y) \leq \log \frac{1}{D(y|x)} + \lceil \log(m+1) \rceil + c_Q + 3.$$

Adequate schedules exist: some polynomial t_D makes every polynomial schedule $t \geq t_D$ adequate.

Proof. The proof reads an affine hash from the public tape and shows, by a second-moment argument, that on at least three quarters of the tapes some short payload hashes to a concept seed producing the labels y on x. It then checks the running time and the price of the resulting certificate.

Write $q = q(n)$ for the concept sampler's random-tape length. Let $\delta = D(y|x) > 0$ be the conditional label probability and let R be the set of concept seeds producing y on x. Then $|R| = \delta 2^q$ and $\delta \geq 2^{-q}$. Put $j = \lceil \log(1/\delta) \rceil + 2$, so $j \leq q+2$. Read a uniformly random $q \times j$ binary matrix M and a uniformly random q-bit vector b from the public tape. For $v \in \{0,1\}^j$, set $H(v) = Mv \oplus b$. If $v \neq v'$, their difference $v \oplus v'$ is nonzero. Each independent row of M has a uniform inner product with this difference, so $M(v \oplus v')$ is uniformly distributed. Conditionally on every value of $M(v \oplus v')$, the independent uniform vector b makes $H(v)$ uniform. Consequently $H(v)$ is uniform and independent of $M(v \oplus v')$. Since $H(v') = H(v) \oplus M(v \oplus v')$, the q-bit strings $H(v)$ and $H(v')$ are independent and uniform.

If $q = 0$, the sole concept seed is the empty string, $\delta = 1$, and every $H(v)$ lies in R with probability one. The same formulas below hold with variance zero in this case. Let X_v indicate $H(v) \in R$. The count $N_R = \sum_v X_v$ is the number of hash inputs whose images lie in the desired seed set. Pairwise independence gives

$$\mathbb{E}N_R = 2^j \delta \geq 4, \qquad \operatorname{Var} N_R = \sum_v \delta(1-\delta) \leq \mathbb{E}N_R.$$

The lower bound holds because $2^j \geq 4/\delta$ by the choice of j. The event $N_R = 0$ is a deviation of N_R from its mean by $\mathbb{E}N_R$, so Chebyshev's inequality (Theorem 2.1) gives

$$\mathbf{Pr}[N_R = 0] \leq \frac{\operatorname{Var} N_R}{(\mathbb{E}N_R)^2} \leq \frac{1}{\mathbb{E}N_R} \leq \frac{1}{4}.$$

For a payload v, consider the certificate with an empty anchor list and the program $\sigma_Q v$ of Definition 3.2. It finds the end of the payload by scanning its sequential input: it reads at most $q+2$ payload bits and then the blank end marker, so no length oracle is needed. It reads the same matrix and vector on every query and computes $r = H(v)$, so it returns $c_r(z)$ on every query z. Thus if $N_R > 0$, some payload v gives a valid certificate, provided that the program halts within the clock, which is checked below. The header is independent of the dataset and public tape; only v is chosen after seeing them.

The matrix and vector use $q(j+1) \leq q(q+3)$ public bits; an adequate clock supplies them and lets the program halt on every query, so the certificate is evaluated as described. All computations of the program have a uniform polynomial bound in n, q, τ_Q and the sampler runtime. Hence some polynomial $t_D \geq q(q+3)$ bounds their running time on U, and every polynomial schedule $t \geq t_D$ is adequate. The certificate has no anchors, so its price is $\lambda_m(0) + |\sigma_Q| + j$, where $\lambda_m(0) = \lceil \log(m+1) \rceil$, $|\sigma_Q| = c_Q$ and $j \leq \log(1/\delta) + 3$ by the choice of j; hence

$$\lambda_m(0) + |\sigma_Q| + j \leq \lceil \log(m+1) \rceil + c_Q + \log(1/\delta) + 3.$$

Such a certificate is available on at least $3/4$ of the tapes, which proves the quantile bound. □

4. Hardness on Uniform Datasets

The PAC-style hardness distribution of Theorem 6.2, proved later in Section 6, chooses its concept independently of the points. We now obtain hardness when both the points and their labels are independently uniform. The proof uses a conditional entropy-preserving generator and a small-bias encoding, both stated precisely in Theorems 4.2 and 4.3 below. The encoding makes each query point select a linear measurement of the generator output. We first show that the measurements lose little entropy and that a uniform vector produces nearly uniform labels.

4.1. Imported Tools and Random Queries.

For a finite random variable V, let $p_V(v) = \mathbf{Pr}[V = v]$ be its probability mass function. Its support is the set of values v with $p_V(v) > 0$. Its Shannon entropy in bits is $H(V) = \mathbb{E}[-\log p_V(V)]$. Conditional entropy averages the entropy of the conditional distribution:

$$H(V|X) = \sum_{x : \mathbf{Pr}[X=x] > 0} \mathbf{Pr}[X = x] H(V|X = x).$$

Conditioning values of probability zero contribute nothing and need not be assigned a conditional distribution. For finite random variables V, X, Y we use the chain rule in both orders,

$$H(V,Y|X) = H(Y|X) + H(V|Y,X) = H(V|X) + H(Y|V,X).$$

Both expansions follow from the factorizations $\mathbf{Pr}[v,y|x] = \mathbf{Pr}[y|x]\,\mathbf{Pr}[v|y,x] = \mathbf{Pr}[v|x]\,\mathbf{Pr}[y|v,x]$ of the joint conditional probability. Suppose that Y is a function of (V,X), so that $H(Y|V,X) = 0$. Suppose moreover that V is independent of X, so that $H(V|X) = H(V)$. Equating the two expansions gives

$$H(Y|X) = H(V) - H(V|Y,X).$$

We use the concavity of the logarithm in the following form, from which the entropy bound for a finite support follows.

Theorem 4.1 (Logarithm concavity [BV04, Sections 3.1.5 and 3.1.8])**.** *For every positive integer k, positive numbers $z_1,\dots,z_k$, and nonnegative weights $p_1,\dots,p_k$ summing to one,*

$$\sum_{i=1}^{k} p_i \log z_i \le \log\left(\sum_{i=1}^{k} p_i z_i\right).$$

A distribution supported on at most N values has entropy at most $\log N$. Indeed, let its positive masses be $p_1,\dots,p_{N'}$ with $N' \le N$; concavity of the logarithm, applied with $z_i = 1/p_i$, gives $\sum_{i=1}^{N'} p_i \log(1/p_i) \le \log(\sum_{i=1}^{N'} p_i/p_i) = \log N' \le \log N$.

The notation $U_r|E_r$ means a uniform seed conditioned to belong to E_r. Distinguishing advantage is the absolute difference between a test's two acceptance probabilities.

Theorem 4.2 (Conditional entropy-preserving generation)**.** *Assume one-way functions exist. There are constants $\alpha > 0$ and $0 < \xi < 1$ with the following properties* [LP25, Definition A.1 and Lemma A.3]*. There is a polynomial-time generator $\overline{G}_r$ with seed length r and output length $r + r^{\xi}$ rounded to an integer; we use only that this length is at least $r + r^{\xi} - 1$. There are nonempty events $E_r \subseteq \{0,1\}^r$ for all sufficiently large r. The first r bits of $\overline{G}_r(U_r|E_r)$ have entropy at least $r - \alpha \log r$. For every PPT distinguisher given 1^r and the challenge, its distinguishing advantage between $\overline{G}_r(U_r|E_r)$ and a uniform output of the same length is eventually at most $1/40$.*

This is Liu–Pass's conditional entropy-preserving PRG theorem, using $\varepsilon = 1/10$ in their Lemma A.3 and Definition A.1 [LP25]. The conditional construction with sublinear stretch first appears in Liu–Pass [LP21a, Definition 5.2 and Lemma 5.3 of the full version], and that construction starts from a generator with superpolynomial security. The cited later formulation states the version obtained from ordinary one-way functions, secure against every PPT distinguisher with advantage below $1/40$, which is the form used here. That formulation leaves nonemptiness of the events implicit in the conditioning on them. Neither efficient recognition nor efficient sampling of E_r is asserted or required.

We normalize the stretch as follows. Choose a fixed integer $\gamma \ge 2$ with $1/\gamma < \xi$ and truncate the generator to its first $r + \lfloor r^{1/\gamma} \rfloor$ output bits, calling the resulting generator G_r. For all sufficiently large r the truncation is available. At the finitely many other lengths, G_r pads the generator's output with zeros to this length, so G_r is one uniform polynomial-time procedure; only its behavior at large r is used. The integer root can be computed by binary search and integer exponentiation, so this normalization is uniform and polynomial time. The entropy of the whole truncated output is at least that of its first r bits. A distinguisher for the truncation also distinguishes the original generator by truncating its challenge; the truncated uniform challenge is exactly uniform. Thus both guarantees remain valid for G_r.

The inner product and matrix products in this section are over $\mathbb{F}_2$. For a rational accuracy parameter, its binary encoding length is the combined bit length of its numerator and positive denominator.

Theorem 4.3 (Small-bias spaces)**.** *For every positive integer L and rational $0 < \varepsilon < 1$, the following construction exists* [AGHP92, Construction 3 and Proposition 3]*. There is an explicitly computable map $S : \{0,1\}^h \to \mathbb{F}_2^L$, with $h = O(1 + \log L + \log(1/\varepsilon))$, such that*

$$\left|\mathbb{E}_{z\sim U_h} (-1)^{\langle v, S(z)\rangle}\right| \le \varepsilon \qquad \text{for every nonzero } v \in \mathbb{F}_2^L.$$

The map and its seed-length schedule are uniform. Given L, the rational ε and the seed, computing the full vector takes time polynomial in L, $1/\varepsilon$ and the binary encoding length of ε.

Small-bias spaces were introduced by Naor–Naor [NN93]; we use the powering construction of Alon–Goldreich–Håstad–Peralta [AGHP92, Construction 3 and Proposition 3]. The construction works in a

field with 2^ω elements, where $\omega = \lceil \log(L/\varepsilon) \rceil$. Since $L/\varepsilon > 1$, we have $\omega \geq 1$. The ceiling convention gives $\log(L/\varepsilon) \leq \omega < \log(L/\varepsilon) + 1$, hence

$$L/\varepsilon \leq 2^\omega < 2L/\varepsilon.$$

We specify the field representation used by the construction. A field is a set with addition and multiplication satisfying the usual arithmetic laws, in which every nonzero element has a multiplicative inverse. A polynomial over $\mathbb{F}_2$ is a finite sum of powers of a formal variable with binary coefficients; its degree is the largest exponent with a nonzero coefficient. A polynomial is monic when its leading coefficient is one, and is irreducible when it has positive degree and cannot be written as a product of two positive-degree polynomials.

For a monic irreducible polynomial f of degree ω, the quotient $\mathbb{F}_2[Z]/(f)$ is the field whose elements are polynomials of degree below ω, with addition coefficientwise and multiplication followed by reduction to the remainder modulo f [Vad12, Remark 3.25]. The coefficients, in increasing degree order, identify these elements with ω-bit vectors; addition then agrees with binary vector addition, so this identification is linear. The leading coefficient of f is fixed at one, and its remaining ω coefficients encode f in exactly ω bits. The construction uses the lexicographically least such polynomial, found by the search described below, so S is a fixed map.

Read the seed as two elements χ, ψ of this field, each written as an ω-bit string by the coefficient identification just specified. For $i \in [L]$, the ith coordinate of $S(\chi, \psi)$ is the inner product of the strings of χ^{i-1} and ψ. Proposition 3 of Alon–Goldreich–Håstad–Peralta bounds the bias against every nonzero v by $(L-1)2^{-\omega} < \varepsilon$, and the seed length is $h = 2\omega \leq 2\log(L/\varepsilon) + 2 = O(1 + \log L + \log(1/\varepsilon))$.

The construction is explicit. To compute ω, test successive integers starting at zero until $2^\omega \geq L/\varepsilon$. The first successful integer is $\lceil \log(L/\varepsilon) \rceil$; there are $\omega + 1$ tests, each implementable by binary integer arithmetic on the numerator, denominator and a power of two. This preprocessing has time polynomial in the parameter encoding lengths and $\log(L/\varepsilon)$.

A representation of the field, that is, an irreducible polynomial of degree ω over $\mathbb{F}_2$, is an ω-bit string. About a $1/\omega$ fraction of these strings are valid representations, and validity can be tested deterministically in polynomial time [AGHP93]. Exhaustive search therefore finds the lexicographically least valid representation in time $2^\omega \operatorname{poly}(\omega)$, which is polynomial in L/ε. Given the representation, the L powers of χ and the L inner products take $L\operatorname{poly}(\omega)$ further time. Together with the computation of ω above, this gives the stated uniform running-time bound.

The next lemma gives two guarantees for the query encoding: a uniform vector produces nearly uniform labels, while a random vector whose distribution has bounded support loses little entropy. The rank of a binary matrix is the maximum number of linearly independent rows. Here independence means that no nonempty subset of those rows sums to zero over $\mathbb{F}_2$.

Lemma 4.4 (Random-query encoding)**.** *Fix positive integers n, L, m, a real $\varepsilon \geq 0$ and a real $r \geq 0$. Let $S : \{0,1\}^n \to \mathbb{F}_2^L$ have bias at most ε on uniform inputs, that is,*

$$\left| \mathbb{E}_{Z \sim U_n} (-1)^{\langle v, S(Z) \rangle} \right| \leq \varepsilon \qquad \text{for every nonzero } v \in \mathbb{F}_2^L.$$

Let $X_1, \ldots, X_m$ be independent uniform points. Write $X = (X_1, \ldots, X_m)$ for their vector. Let M_X be the matrix whose ith row is $S(X_i)$. Then

$$\mathbf{Pr}[\operatorname{rank} M_X < m] \leq 2^{m-L} + (1+\varepsilon)^m - 1. \tag{1}$$

The statistical distance between $(X, M_X U_L)$ and (X, U_m) is at most the right side of Equation (1)*, where the uniform variables are independent of X. If an $\mathbb{F}_2^L$-valued random variable V is independent of X and has support of size at most 2^r, then*

$$H(M_X V | X) \geq H(V) - \log\left(1 + 2^{r-m}(1+\varepsilon)^m\right). \tag{2}$$

Proof. We first bound row dependence to obtain the statistical-distance estimate. We then bound fiber sizes to obtain the entropy estimate. For a uniform n-bit point Z, let $\phi(v) = \mathbb{E}_Z (-1)^{\langle v, S(Z) \rangle}$. The functions $v \mapsto (-1)^{\langle v, u \rangle}$ are called binary characters. For every $u \in \mathbb{F}_2^L$, their orthogonality gives

$$\mathbf{1}_{u=0} = 2^{-L} \sum_{v \in \mathbb{F}_2^L} (-1)^{\langle v, u \rangle}.$$

Indeed, the sum is 2^L when $u = 0$. Otherwise choose a coordinate c for which $u_c = 1$. Pair each v with $v \oplus e_c$, where e_c has a one in coordinate c and zeros elsewhere. The two characters have opposite signs and cancel. Fix a nonempty set J of row indices, and write $j = |J|$. Applying the orthogonality identity to $u = \bigoplus_{i \in J} S(X_i)$, taking expectations, and using the independence of the points X_i gives

$$\mathbf{Pr}\left[\bigoplus_{i \in J} S(X_i) = 0\right] = 2^{-L} \sum_{v} \phi(v)^j \leq 2^{-L} + \varepsilon^j,$$

since $\phi(0) = 1$ and $|\phi(v)| \leq \varepsilon$ for nonzero v. A matrix over $\mathbb{F}_2$ has dependent rows exactly when some nonempty subset of them sums to zero. Summing the displayed bound over all such subsets yields

$$\mathbf{Pr}[\operatorname{rank} M_X < m] \leq (2^m - 1)2^{-L} + \sum_{j=1}^{m} \binom{m}{j} \varepsilon^j \leq 2^{m-L} + (1+\varepsilon)^m - 1.$$

For each fixed X with full row rank, row elimination gives m distinct pivot columns. To see this, successively swap a nonzero row into the next position and add it to the other rows to clear its pivot column. These operations preserve the solution set when also applied to the right-hand side; row independence ensures that no zero row remains before all m pivots are found. For any prescribed right-hand side, choose the $L - m$ nonpivot coordinates of u freely, and then solve uniquely for the pivot coordinates in reverse order. Thus $u \mapsto M_X u$ is onto $\mathbb{F}_2^m$, with exactly 2^{L-m} preimages of each vector. A uniform U_L consequently gives exactly uniform $M_X U_L$. For the remaining X, the conditional statistical distance is at most one. Averaging these conditional distances proves the statistical-distance assertion.

For the entropy assertion let T be the support of V. The fiber size $N(v,X)$ counts support vectors that produce the same query labels as v:

$$N(v,X) = |\{u \in T : M_X u = M_X v\}|.$$

If $u \neq v$, then

$$\mathbf{Pr}[\langle S(X_i), u \oplus v \rangle = 0] = \frac{1 + \phi(u \oplus v)}{2} \leq \frac{1+\varepsilon}{2}.$$

Independence of the rows shows, for every $v \in T$, that

$$\mathbb{E}_X N(v,X) \leq 1 + (|T| - 1)2^{-m}(1+\varepsilon)^m \leq 1 + 2^{r-m}(1+\varepsilon)^m.$$

Given X and $Y = M_X V$, the conditional variable V takes values in the fiber of M_X over Y within T, which has cardinality $N(V,X)$. Since V is independent of X, averaging the last bound over V gives the same bound for $\mathbb{E}N(V,X)$. The entropy bound for a finite support and concavity of the logarithm imply

$$H(V|Y,X) \leq \mathbb{E} \log N(V,X) \leq \log \mathbb{E} N(V,X) \leq \log\left(1 + 2^{r-m}(1+\varepsilon)^m\right).$$

Applying the conditional chain rule stated above proves Equation (2). □

4.2. **The Uniform-Data Theorem.** Let $D_{0,n}$ draw $m(n)$ independent uniform n-bit points and $m(n)$ independent uniform label bits. A repeated point can receive conflicting labels; then no certificate is valid, every quantile is infinite, and a heuristic is permitted to output ∞.

Theorem 4.5 (Uniform-data hardness; restatement of Theorem 1.1)**.** *Assume one-way functions exist. Fix a polynomially bounded, polynomial-time computable integer schedule $m(n) \geq 8n$. Fix constants $0 < \theta_1 \leq \theta_2 \leq 1$ and an integer $a \geq 0$. There are polynomials $t_0(n)$ and $p(n) \geq 1$ such that, for every polynomial query clock $t \geq t_0$, every PPT algorithm B eventually satisfies*

$$\Pr_{(X,Y) \sim D_{0,n}, B}\left[\mathsf{pSC}^t_{\theta_1}(X,Y) - a\lceil \log m(n) \rceil \leq B(1^n, X, Y) \leq \mathsf{pSC}^t_{\theta_2}(X,Y) + a\lceil \log m(n) \rceil\right] < 1 - \frac{1}{p(n)}.$$

Every such clock t permits the full-anchor lookup procedure on U.

Proof. We prove the contrapositive. A heuristic that meets the claimed success threshold on uniform data infinitely often will give a distinguisher for the conditional generator at those same point lengths. We first encode generator outputs as labels, which have small quantiles, and transfer the heuristic's success to those labels by bounding their density relative to uniform labels. On uniform labels, Lemma 3.1 shows that small quantiles are rare, so a test that compares the heuristic's output with a threshold separates the two label distributions. The proof has five steps. First, it fixes parameters and a concept family with

short certificates. Second, it bounds the density of the planted labels, those computed from generator outputs, relative to uniform labels. Third, it transfers the heuristic's success to planted labels. Fourth, it applies the counting bound to uniform labels. Fifth, it builds the distinguisher.

Take G_n, α, γ, E_n as in Theorem 4.2 and its normalization, at seed length n. Set $\bar{a} = 2a + 4$ and $\Delta_n = a\lceil \log m(n) \rceil$. The tolerance Δ_n is added twice in the counting bound below, once at each end of the estimation interval, and accounts for $2a$ of the coefficient $\bar{a}$, which multiplies $\ell = \lceil \log m \rceil$ in the parameters below. The surplus four in $\bar{a}$ then leaves a factor m^{-3} in that bound.

For sufficiently large n, write $m = m(n)$ and $\ell = \lceil \log m \rceil$. The label vector has m bits, while the concept below is described by r payload bits, slightly fewer than m. Define the encoded vector length L, the retained generator stretch j, the padding length b, and the payload length r by

$$L = m + 2\ell, \qquad j = (\bar{a} + 2)\ell, \qquad b = m - n - \bar{a}\ell, \qquad r = n + b = m - \bar{a}\ell.$$

Since $m \geq 8n$ and m is polynomially bounded, $b \geq 0$ and $n \leq r \leq m$ eventually. Indeed, $m - n \geq 7m/8$, whereas $\bar{a}\lceil \log m \rceil = o(m)$. Also $j = O(\log n) \leq \lfloor n^{1/\gamma} \rfloor$ eventually. We can therefore keep only the first $n + j$ bits of G_n and append b bits to obtain a vector of length $L = n + j + b$. Use the small-bias bound $\varepsilon = m^{-3}$. Theorem 4.3 supplies a uniform map S_n with seed length $h = O(1 + \log L + \log(1/\varepsilon)) = O(\log n)$. Consequently $h \leq n$ eventually; use the first h query bits as its seed and ignore the remaining query bits. This gives a map on n-bit points with bias at most ε, since the first h bits of a uniform point are a uniform seed.

Fix n_0 large enough for these conditions, the entropy guarantee and the availability of the generator truncation. The length n_0 does not depend on any heuristic. The generator's security bound is applied later to one fixed distinguisher, and the length from which that bound holds may depend on that distinguisher. For $n \geq n_0$, a payload consists of $u \in \{0,1\}^n$ and $\zeta \in \{0,1\}^b$. For a string, prefix_{n+j} retains its first $n + j$ bits, and $\|$ denotes concatenation. Define the pointwise concept by

$$F_{u,\zeta}(z) = \left\langle S_n(z), \mathrm{prefix}_{n+j}(G_n(u)) \,\|\, \zeta \right\rangle.$$

A fixed self-delimiting header computes n from the query length, computes the schedules m, L, j, b, r, h, reads exactly r successive payload bits, and evaluates this expression. It uses no anchors and ignores public randomness. It receives no sample-size oracle or other points: m is computed from the fixed built-in schedule $m(n)$. The code for that schedule contributes only a fixed constant to the header. Below n_0 let this procedure return zero; these finite defaults do not affect an eventual assertion. Let c_F be the fixed header length.

Let t_0 be a polynomial that bounds the simulated running time on U of this fixed procedure and of the full-anchor lookup procedure, including the finite-small-n cases. All procedures are polynomial time, and efficient universality gives such a polynomial. Fix any polynomial clock $t \geq t_0$. For each payload (u, ζ), consider the certificate with no anchors whose program is the header followed by the payload $u\zeta$. A run that halts within t_0 steps halts within t steps with the same output, and the procedure ignores the public tape, so this certificate is valid at the clock t on every dataset labeled by $F_{u,\zeta}$. Likewise t permits full-anchor lookup, and nothing else in this proof depends on the clock, since Lemma 3.1 holds at every clock. Consequently every dataset labeled by any $F_{u,\zeta}$ has

$$\mathsf{pSC}^t_{\theta_2}(X, Y) \leq s_n, \qquad s_n = r + \lceil \log(m+1) \rceil + c_F. \tag{3}$$

The three terms of s_n are the payload length r, the charge $\lambda_m(0) = \lceil \log(m+1) \rceil$ for an empty retained set, and the header length c_F; the same certificate succeeds on every public tape. For analysis only, draw

$$V = \mathrm{prefix}_{n+j}(G_n(U_n | E_n)) \,\|\, U_b$$

independently of uniform points X, with the padding U_b also independent of the conditional seed. Set $Y = M_X V$, where M_X is the matrix whose ith row is $S_n(X_i)$, as in Lemma 4.4, and call this joint distribution P_n. Every planted dataset is labeled by some $F_{u,\zeta}$: if $V = \mathrm{prefix}_{n+j}(G_n(u)) \,\|\, \zeta$, then $Y_i = \langle S_n(X_i), V \rangle = F_{u,\zeta}(X_i)$ for every i. The planted distribution P_n need not be efficiently samplable; the distinguisher will receive the generator challenge directly. The variable V has support of size at most $2^{n+b} = 2^r$. Its first n bits have entropy at least $n - \alpha \log n$; the additional retained generator bits

cannot decrease this entropy, and the independent padding adds b bits. Thus $H(V) \geq r - \alpha \log n$. By Lemma 4.4,

$$H(Y|X) \geq r - \alpha \log n - \delta_n, \qquad \delta_n = \log(1 + 2^{r-m}(1+\varepsilon)^m) = O(m^{-4}).$$

For the last bound use $2^{r-m} = 2^{-\bar{a}\ell} \leq m^{-4}$ and $(1+m^{-3})^m \leq \exp(m^{-2})$. For sufficiently large m the exponential is at most two, and $\log(1+v) \leq v/\ln 2$ bounds δ_n by $2m^{-4}/\ln 2$.

Write $P_n(y|x)$ for the planted conditional mass. Each supported label vector has at least one seed in E_n and one padding string producing it, so $P_n(y|x) \geq 1/(|E_n|2^b) \geq 2^{-r}$. For a planted sample, let $Z = -\log P_n(Y|X)$. Then $0 \leq Z \leq r$ and

$$\mathbb{E}(r - Z) = r - H(Y|X) \leq \alpha \log n + \delta_n.$$

Choose the fixed integer $C = \lceil 100(\alpha+1) \rceil$ and define the good event $\mathscr{G} = \{Z \geq r - C \log m\}$, on which the planted conditional mass is at most $2^{-r} m^C$. Markov's inequality gives

$$P_n(\mathscr{G}^c) \leq \frac{\alpha \log n + \delta_n}{C \log m} < \frac{1}{50}$$

for all sufficiently large n, since $n \leq m$ and $\delta_n = o(\log m)$. On $\mathscr{G}$ the conditional density relative to uniform labels satisfies

$$\frac{P_n(y|x)}{2^{-m}} \leq 2^{m-r} m^C \leq 2^{\bar{a}} m^{C+\bar{a}}. \tag{4}$$

The last inequality uses $\ell < \log m + 1$. Both distributions have the same uniform point marginal, so this is also their joint density ratio. The same density bound applies to the intersection of $\mathscr{G}$ with any event that also involves independent algorithm coins: apply it to the event for each fixed choice of the coins, and then average over the coins.

Choose a positive integer polynomial $p(n)$ satisfying $p(n) \geq m(n)^{C+\bar{a}+2}$ for every n. A polynomial majorant exists because the fixed schedule m is polynomially bounded. This choice does not require m itself to be a polynomial expression. Suppose a PPT B has success at least $1 - 1/p(n)$ on $D_{0,n}$ for infinitely many n. Let $\mathscr{E}$ be its failure event, including outputs outside the specified extended-integer interval or malformed outputs. At each such sufficiently large n, Equation (4) implies

$$P_n(\mathscr{G} \cap \mathscr{E}) \leq 2^{\bar{a}} m^{C+\bar{a}} / p(n) \leq 2^{\bar{a}}/m^2.$$

Put $T_n = s_n + \Delta_n$. On a planted sample where B succeeds, its output is at most $\mathsf{pSC}^t_{\theta_2} + \Delta_n \leq s_n + \Delta_n = T_n$ by Equation (3). Thus the efficiently recognizable test that accepts a well-formed finite output $B \leq T_n$ has planted acceptance probability at least

$$P_n[B \leq T_n] \geq \frac{49}{50} - \frac{2^{\bar{a}}}{m^2}. \tag{5}$$

Under $D_{0,n}$, success together with acceptance implies $\mathsf{pSC}^t_{\theta_1} \leq T_n + \Delta_n = s_n + 2\Delta_n$. For each fixed point vector, Lemma 3.1 bounds the fraction of label vectors with $\mathsf{pSC}^t_{\theta_1} \leq s$ by $2^{s+1-m}/\theta_1$. Applying it for each fixed X and averaging over the uniform marginal of X gives

$$\pi_n := D_{0,n}[B \leq T_n] \leq \frac{1}{p(n)} + \frac{2^{s_n + 2\Delta_n + 1 - m}}{\theta_1}.$$

On an inconsistent dataset, one with a repeated point carrying two different labels, no certificate is valid. An output counted as correct must then be ∞, which this test rejects, so such datasets enter π_n only through the failure term $1/p(n)$. For the second term,

$$\frac{2^{s_n + 2\Delta_n + 1 - m}}{\theta_1} = \frac{2^{c_F+1}}{\theta_1} 2^{-(\bar{a}-2a)\lceil \log m \rceil + \lceil \log(m+1) \rceil} \leq \frac{2^{c_F+3}}{\theta_1} m^{-3}.$$

This bound uses $\bar{a} - 2a = 4$, so that $2^{-4\lceil \log m \rceil} \leq m^{-4}$, together with $2^{\lceil \log(m+1) \rceil} \leq 2(m+1) \leq 4m$.

We now construct one PPT generator distinguisher. On input 1^n and an $(n + \lfloor n^{1/\gamma} \rfloor)$-bit challenge, reject if $n < n_0$. Otherwise retain its first $n + j$ bits, append b fresh uniform bits, draw m independent uniform n-bit points, and label them by the small-bias measurements of this vector. Run B with fresh coins

and accept exactly when its output is well formed, finite and at most T_n. All schedules and transformations are polynomial-time computable in n, so this is a single PPT test. No inversion of the sample-size schedule is required.

A conditional-generator challenge produces exactly the planted distribution P_n. A uniform challenge followed by independent padding produces exactly U_L. For the latter challenge, let τ_n be the statistical distance between the resulting sample and independent uniform points and labels; Lemma 4.4 bounds it by

$$\tau_n \le 2^{m-L} + (1+m^{-3})^m - 1 \le m^{-2} + 2m^{-2} = 3m^{-2}$$

for all sufficiently large n. Here $L-m = 2\ell$ gives $2^{m-L} \le m^{-2}$, and $e^v - 1 \le 2v$ for $0 \le v \le 1$ gives the second term's bound. Since statistical distance bounds the change in the probability of every event, the acceptance probability on a uniform challenge is at most $\pi_n + 3m^{-2}$. At each sufficiently large parameter where B has the assumed success, Equation (5) and the bounds on π_n and τ_n give distinguishing advantage at least

$$\frac{49}{50} - \frac{2^{\bar{a}}+3}{m^2} - \frac{1}{p(n)} - \frac{2^{c_F+3}}{\theta_1} m^{-3} = \frac{49}{50} - o(1).$$

These are infinitely many generator security parameters n, contradicting its eventual $1/40$ guarantee for this fixed test. Thus every PPT B eventually has success strictly below the stated threshold. □

Taking $a = 0$ and $\theta_1 = \theta_2$ proves hardness of exact computation at every fixed positive quantile. Every fixed real multiple of $\log m$ is bounded by the tolerance of this theorem for some fixed integer a. Exact computation has no matching converse here: the no-one-way-function converse, Theorem 5.1, requires two distinct quantile thresholds.

4.3. **An Unconditional Precision–Error Bound.** In Theorem 4.5 the coefficient a of the logarithmic tolerance is fixed before the error polynomial p is chosen. If instead the error rate is a fixed inverse polynomial, a sufficiently large logarithmic tolerance permits an unconditional estimate.

Theorem 4.6 (Uniform-label estimation)**.** *Fix a polynomial sample schedule m, a polynomial clock t permitting full-anchor lookup, and constants $0 < \theta_1 \le \theta_2 \le 1$. There is a constant c_{lookup} and a deterministic polynomial-time estimator taking $(1^n, x, y, d(n))$ as input, with the tolerance supplied in binary, with the following property. For every polynomial-time computable nonnegative integer tolerance $d(n)$ of polynomial bit length, its probability of failing to return a value in*

$$[\mathsf{pSC}^t_{\theta_1}(X,Y) - d(n), \mathsf{pSC}^t_{\theta_2}(X,Y) + d(n)]$$

on independent uniform points and labels is at most

$$\min\left\{1, \frac{2^{c_{\mathrm{lookup}}+1}(m(n)+1)}{\theta_2} 2^{-2d(n)}\right\}.$$

Proof. On a consistent dataset the estimator returns the price of the full-anchor lookup certificate lowered by the tolerance, but not below zero; this price bounds every quantile from above, and certificate counting shows that quantiles far below it are rare. A fixed program implementing full-anchor lookup has length c_{lookup}. The full-anchor lookup certificate retains all m points with their observed labels as tags, and its price is

$$B_m = m + \lceil \log(m+1) \rceil + c_{\mathrm{lookup}},$$

because $\binom{m}{m} = 1$. On a consistent dataset, this certificate answers every query correctly, independently of the public tape. The stipulated clock permits this procedure, so every positive quantile is at most B_m. On an inconsistent dataset no pointwise decoder can answer both occurrences correctly, and every positive quantile is infinite. The estimator first detects such a conflict by comparing all pairs of points and their labels and returns ∞ in that case. Otherwise it returns $v = \max\{0, B_m - d(n)\}$. This uses polynomial-time arithmetic even if d is large, since its bit length is polynomial.

Write $h_i = \mathsf{pSC}^t_{\theta_i}(X,Y)$ on a consistent dataset, and abbreviate $d = d(n)$. If $d \ge B_m$, then $v = 0$, $h_1 - d \le 0$ and $h_2 + d \ge 0$, so the estimate succeeds. If $d < B_m$, then $v = B_m - d \ge h_1 - d$. The only failure is $v > h_2 + d$, equivalently $h_2 < B_m - 2d$. Since finite quantiles are nonnegative integers, this implies $h_2 \le B_m - 2d - 1$. When this last threshold is negative there is no failure. Otherwise, for every

fixed point vector, Lemma 3.1, which holds at every integer threshold $s \geq 0$, bounds the number of such label vectors by

$$\frac{2^{B_m-2d}}{\theta_2}.$$

Each label vector has conditional uniform probability 2^{-m}. Thus the failure probability, also after averaging the point vector, is at most

$$\frac{2^{B_m-2d-m}}{\theta_2} = \frac{2^{\lceil \log(m+1) \rceil + c_{\text{lookup}}-2d}}{\theta_2} \leq \frac{2^{c_{\text{lookup}}+1}(m+1)}{\theta_2} 2^{-2d}.$$

The probability is also at most one, which gives the stated bound. □

Consequently, logarithmic additive tolerance suffices for every requested inverse-polynomial error. For each positive polynomial p and fixed quantile thresholds $0 < \theta_1 \leq \theta_2 \leq 1$, the required tolerance is $O(\log m + \log p)$. For example, choose an integer $K \geq 1/\theta_2$, a positive integer polynomial $P(n) \geq p(n)$, and take

$$d(n) = \left\lceil \frac{1}{2} \log \left(2^{c_{\text{lookup}}+1} K(m(n)+1)P(n) \right) \right\rceil.$$

For this d, the failure-probability bound of Theorem 4.6 is at most $1/(\theta_2 KP(n)) \leq 1/p(n)$. The tolerance is the least integer d with $2^{2d} \geq 2^{c_{\text{lookup}}+1} K(m(n)+1)P(n)$, so integer arithmetic computes it exactly, without approximating a threshold probability. This does not contradict Theorem 4.5, which fixes the coefficient a of the tolerance first and chooses the error polynomial afterward, whereas here the tolerance grows with the error polynomial. The bounds locate the precision–error tradeoff at a logarithmic scale; they do not identify optimal constants.

4.4. **Independent Concepts and Uniform Data.** A sufficiently independent concept family approximates uniform datasets, because collisions between uniform points are rare. Section 5 uses this approximation to transfer its converse to uniform data.

Theorem 4.7 (Independent function families)**.** *For positive integers n, b, k with $k \leq 2^n$, there is a uniform family of k-wise independent functions from $\{0,1\}^n$ to $\{0,1\}^b$* [Vad12, Corollary 3.34 and Remark 3.25]. *A uniformly random member of the family can be chosen with exactly $k \max\{n,b\}$ random bits, and evaluated deterministically in time polynomial in n, b, k. For every list of at most k distinct inputs, its outputs are independent and uniform.*

This is the polynomial-based construction recorded in Vadhan's survey, whose historical notes credit Lancaster and Joffe for limited-independence constructions [Vad12, Section 3.7]. The deterministic construction of binary fields discussed in Remark 3.25 makes the evaluation uniform; the evaluating program receives no advice about the field.

Lemma 4.8 (PAC approximation to uniform data)**.** *Fix a polynomial sample-size schedule $m(n)$. There is a fixed PAC-style sampler D_{ind} with uniform points such that*

$$\text{TV}(D_{\text{ind},n}, D_{0,n}) = \mathbb{E}_X[1 - 2^{N_{\text{dist}}(X)-m(n)}] \leq \binom{m(n)}{2} 2^{-n},$$

where $N_{\text{dist}}(X)$ is the number of distinct points in X.

Proof. We construct the independent-function sampler and compute its conditional distance from independent uniform labels. Put $k(n) = \min\{m(n), 2^n\}$, so that $k(n) \leq 2^n$, as Theorem 4.7 requires, and no point vector has more than $k(n)$ distinct points. This integer is computable by polynomial-time arithmetic, and it is at most $m(n)$, so it is polynomially bounded even when its second argument is exponential. Use Theorem 4.7 with output length one. The concept sampler draws a uniformly random member of the family with $q(n) = k(n)n$ random bits, independently of the points. Its fixed evaluation procedure obtains n from the query length, computes m, k, q, reads exactly q payload bits, and evaluates that family member. The construction is deterministic polynomial time once the seed is supplied, so it meets Definition 2.6. In particular it supplies one fixed sampler D_{ind} for the chosen schedule m, independently of every error polynomial or heuristic.

Fix X; it has $N_{\mathrm{dist}}(X) \le k(n)$ distinct points. By Theorem 4.7, the concept labels of these distinct points are independent uniform bits, and each label is repeated at every occurrence of its point. There are exactly $2^{N_{\mathrm{dist}}(X)}$ consistent label vectors, each of conditional probability $2^{-N_{\mathrm{dist}}(X)}$. Uniform independent labels give all 2^m vectors probability 2^{-m}. The conditional statistical distance is therefore

$$\begin{aligned}\frac{1}{2}\sum_{y}|D_{\mathrm{ind}}(y|X)-2^{-m}| &= \frac{1}{2}\left(2^{N_{\mathrm{dist}}(X)}(2^{-N_{\mathrm{dist}}(X)}-2^{-m})+(2^m-2^{N_{\mathrm{dist}}(X)})2^{-m}\right)\\ &= 1-2^{N_{\mathrm{dist}}(X)-m}.\end{aligned}$$

The two distributions have the same point marginal; averaging proves the identity. The conditional distance is zero on distinct points and at most one otherwise. Every pair of independent uniform points is equal with probability 2^{-n}. A union bound over the $\binom{m}{2}$ pairs gives the asserted upper bound. □

5. Easiness without One-Way Functions

The coding bound of Lemma 3.3 lets us compare the dataset distribution with the output distribution of a function that generates certificates. Inverting that function recovers the certificates used to estimate the quantiles. We adapt Liu–Pass's verified-inversion and distribution-domination argument [LP23, Lemma 4.3]; Hirahara–Nanashima's search for a minimum consistent local program is a related predecessor [HN23, Appendix C].

Theorem 5.1 (Robust easiness without one-way functions; restatement of Theorem 1.2)**.** *Assume one-way functions do not exist. Fix a polynomial sample-size schedule $m(n) \ge 1$ and a PAC-style samplable D. Fix also a polynomial clock t adequate for D, constants $0 < \theta_1 < \theta_2 \le 3/4$, and a positive polynomial $p(n) \ge 1$. There is a PPT algorithm $B = B_{D,t,p,\theta_1,\theta_2}$ such that, for infinitely many n,*

$$\Pr_{(x,y)\sim D_n,B}\left[\mathsf{pSC}^t_{\theta_1}(x,y) \le B(1^n,x,y) \le \mathsf{pSC}^t_{\theta_2}(x,y)\right] \ge 1-\frac{1}{p(n)}.$$

The upper endpoint $3/4$ is the quantile that Lemma 3.3 bounds. The proof uses it only to place every relevant tape below the certificate-length cutoff.

Proof. The proof has four stages. First, we build a polynomial-time function f, the certificate generator, whose valid outputs list a charged length, a public tape and a dataset labeled by a sampled certificate, and we compare its output distribution with the dataset distribution (Equation (6)). Second, the absence of one-way functions gives an inverter for f that succeeds at infinitely many input lengths, which we call successful. Third, the algorithm B inverts f on independent public tapes, verifies every returned inverse, and outputs an empirical quantile of the recovered certificate lengths.

Fourth, we identify the parameters at which B succeeds. Each parameter n is assigned a finite interval of input lengths for f, defined below; these intervals are disjoint and cover all sufficiently large lengths, so infinitely many of them contain a successful length. We show that at every n whose interval contains a successful length, the output of B lies between the θ_1- and θ_2-quantiles except with probability at most $1/p(n)$.

Before the first stage we fix conventions. Replace the requested error polynomial by an integer-coefficient polynomial majorant, still denoted p, whose values are polynomial-time computable. Proving an error bound of $1/p(n)$ for this majorant proves the requested, possibly weaker bound. Suppress the superscript t in g^t_w and in pSC^t_θ. Let $c'(n) = \lceil \log(m+1) \rceil + c_Q + 3$ be the overhead in the coding bound. Let $L(n) = q(n) + c'(n)$ be the certificate-length cutoff, where $q(n)$ is the length of the concept sampler's random tape. Encoding a proposed charged length between zero and $L(n)$ uses $b(n) = \lceil \log(L(n)+1) \rceil$ bits. Encoding an anchor size between zero and $m(n)$ uses $b_k(n) = \lceil \log(m(n)+1) \rceil$ bits. The coding lemma (Lemma 3.3) and $D(y|x) \ge 2^{-q}$ show that all supported datasets satisfy

$$s_1 := \mathsf{pSC}_{\theta_1}(x,y) \le s_2 := \mathsf{pSC}_{\theta_2}(x,y) \le \mathsf{pSC}_{3/4}(x,y) \le L(n).$$

We construct a polynomial-time function that samples certificates and then outputs their labelings. Its core is the concatenation of the following seven fields; a core is valid when it passes all the rejection tests below. The length field ℓ uses b bits. The public tape w uses t bits. The point-sampler tapes $\rho_1, \dots, \rho_m$

use mr_μ bits in total. The anchor-size field k uses b_k bits. The subset-rank field κ uses m bits. The tag field α also uses m bits. The program field π uses L bits.

The rejection tests and decoding below turn a valid core into a certificate (A,a,β) of charged length ℓ. Reject a core if $\ell > L$ or $k > m$. Use the first $\lceil \log \binom{m}{k} \rceil$ bits of κ as a rank, rejecting ranks at least $\binom{m}{k}$. Unrank the subset A in increasing lexicographic order and take the first k bits of α as a. If $e = \ell - \lambda_m(k) - k$ is negative, reject the core; otherwise take the first e bits of π as β. Since $\ell \le L$, the program field π, which has L bits, is long enough. Compute $x_i = \mu(1^n;\rho_i)$ and evaluate on each x_i the t-step hypothesis that this certificate defines on the tape w, to obtain y.

Binomial coefficients and subset ranks are computable in polynomial time using integer arithmetic on at most $m+1$ bits. For unranking, scan the candidate indices, count completions by the corresponding binomial coefficient, and subtract that count whenever the desired rank is beyond the current block. This uses at most m scans and polynomially many arithmetic operations.

Write $H(n) = b + t + mr_\mu + b_k + 2m + L$ for the core width and put

$$N(n) = n + \max_{1 \le i \le n} H(i).$$

The schedule $N(n)$ is strictly increasing, polynomially bounded, and computable in polynomial time. On an arbitrary input u of length $N \ge N(1)$, choose the largest n with $N(n) \le N$. Parse the first $H(n)$ bits as a core and treat the rest as a pad η. A valid core gives the unambiguous output

$$f(u) = (1, n, N, \ell, w, x, y, \eta).$$

An invalid core, or an input shorter than $N(1)$, gives an output tagged zero together with its length. This defines one deterministic polynomial-time function on all binary strings. Computing n by testing indices up to N is polynomial in N. Outputting n, N, η ensures that an inverse of a valid output has exactly the intended length, parameter, and pad.

For a tuple (ℓ,w,x,y), let $F_n(\ell,w,x,y)$ be the probability that a uniform core produces that tuple with the valid-output tag. Its total mass may be less than one. Conditioned on selecting ℓ, any fixed certificate of charged length exactly ℓ is drawn with probability

$$2^{-b_k} 2^{-\lceil \log \binom{m}{k} \rceil} 2^{-k} 2^{-e} = 2^{-\lambda_m(k)-k-e} = 2^{-\ell}.$$

Unused field tails do not change this probability. The length field equals a given ℓ with probability $2^{-b} \ge 1/(2(L+1))$. Independently of these fields, the t-bit tape field equals w with probability 2^{-t} and the point-sampler fields produce x with probability $\mu^m(x)$. Taking a minimizing certificate, whose labels are y, shows that whenever $g_w(x,y) \le L$,

$$F_n(g_w,w,x,y) \ge \frac{2^{-t}\mu^m(x)2^{-g_w(x,y)}}{2(L+1)}. \tag{6}$$

No invariance under program padding is used: the sampled program has exactly the minimizing length.

Choose a rational θ_* strictly between θ_1 and θ_2. Choose a positive rational $\varepsilon < \min\{\theta_* - \theta_1, \theta_2 - \theta_*\}$. These fixed rational constants can be built into the algorithm even when the endpoints are not computable reals. Choose an increasing positive polynomial R satisfying, for every n,

$$R(n) \ge \frac{8}{\varepsilon} 2^{c'(n)} (L(n)+1) p(n).$$

The factor $2^{c'(n)}$ is polynomially bounded, since

$$2^{c'(n)} \le 2^{cQ+4}(m(n)+1).$$

The other schedules are polynomially bounded as well, so an increasing polynomial majorant R exists. Since one-way functions do not exist, the contrapositive form of Theorem 2.8 gives a PPT inverter $\mathscr{I}$ that inverts f with average success at least $1 - 1/R(N)$ for infinitely many lengths N. Call these lengths successful. Each interval $[N(n), N(n+1))$ is finite, so infinitely many successful lengths yield infinitely many parameters n whose interval contains a successful length.

On input $(1^n, x, y)$, the algorithm B draws

$$T = \left\lceil \frac{5p(n)}{2\varepsilon^2} \right\rceil$$

independent tapes w_j. For each tape, it tries every integer $\ell \in [0, L]$ and every integer $N \in [N(n), N(n+1))$. At each attempt it draws a fresh uniform pad of length $N - H(n)$ and calls $\mathscr{I}$ on $(1^N, (1, n, N, \ell, w_j, x, y, \eta))$. A returned inverse u' is accepted only after checking $|u'| = N$ and $f(u') = (1, n, N, \ell, w_j, x, y, \eta)$. An accepted inverse contains a valid certificate of charged length ℓ. Let $\widehat{g}_j$ be the smallest accepted ℓ, or ∞ if no attempt succeeds. Every accepted length is the charged length of a valid certificate on w_j, so $\widehat{g}_j \geq g_{w_j}$ on every outcome; we call this soundness.

For $s \in [0, L]$, the observed fraction of recovered lengths at most s is

$$\widehat{P}(s) = T^{-1} \left| \{ j : \widehat{g}_j \leq s \} \right|.$$

Output the smallest s with $\widehat{P}(s) \geq \theta_*$, and output $L+1$ if none exists. The number of attempts is at most $T(L+1)N(n+1)$, which is polynomially bounded. Each attempt makes one call to $\mathscr{I}$ and one evaluation of f, both in polynomial time.

We now carry out the fourth stage. Fix a parameter n whose interval contains a successful length N_*. For a tuple $v = (\ell, w, x, y)$ produced by a valid core, let $\mathrm{fail}(v)$ be the inverter failure probability averaged over its coins and an independent uniform pad at that length. Since the core and pad are independent under a uniform N_*-bit input,

$$\sum_v F_n(v) \, \mathrm{fail}(v) \leq 1/R(N_*).$$

The omitted invalid outputs can only increase the full average failure. For a supported dataset, the coding bound $C(x, y)$ and its good-tape set are

$$C(x, y) = \log(1/D(y|x)) + c', \quad \mathrm{Good}(x, y) = \{ w : g_w(x, y) \leq C(x, y) \}.$$

For every supported dataset, $D(y|x) \geq 2^{-q}$ implies $C(x, y) \leq q + c' = L$. Thus every tape in $\mathrm{Good}(x, y)$ satisfies the cutoff hypothesis of Equation (6).

Let the missed-minimum probability $\phi(x, y)$ be the probability, over a uniform tape w, the pad and the inverter's coins, that $w \in \mathrm{Good}(x, y)$ and the attempt at $\ell = g_w$ and $N = N_*$ fails. On $\mathrm{Good}(x, y)$,

$$D(y|x) \leq 2^{c' - g_w(x,y)}.$$

The tuple (g_w, w, x, y) identifies (w, x, y), so no multiplicity is lost when summing over these tuples. Summing over supported datasets and their good tapes gives the following chain. Its first inequality uses this bound on $D(y|x)$, its second uses Equation (6), and its third uses the bound on the inverter's average failure above:

$$\begin{aligned}
\mathbb{E}_{D_n} \phi &= \sum_{(x,y) \in \mathrm{supp}(D_n)} \sum_{w \in \mathrm{Good}(x,y)} \mu^m(x) D(y|x) 2^{-t} \, \mathrm{fail}(g_w, w, x, y) \\
&\leq 2^{c'} \sum_{(x,y) \in \mathrm{supp}(D_n)} \sum_{w \in \mathrm{Good}(x,y)} \mu^m(x) 2^{-g_w} 2^{-t} \, \mathrm{fail}(g_w, w, x, y) \\
&\leq 2^{c'+1} (L+1) \sum_v F_n(v) \, \mathrm{fail}(v) \\
&\leq \frac{2^{c'+1}(L+1)}{R(N_*)} \\
&\leq \frac{\varepsilon}{4p(n)}.
\end{aligned}$$

The last inequality uses $N_* \geq n$ and the monotonicity of R. Markov's inequality implies

$$\Pr_{D_n}[\phi > \varepsilon/2] \leq 1/(2p(n)).$$

Now fix a supported dataset (x, y), and let $P(s) = \Pr_w[g_w \leq s]$ be the fraction of public tapes on which the minimum charged length g_w is at most s. Let $\widetilde{P}(s) = \Pr[\widehat{g}_1 \leq s | x, y]$ be the probability, given the dataset, that a recovered minimum is at most s. We compare P and $\widetilde{P}$ at the two checkpoints $s = s_1 - 1$ and $s = s_2$. The coding lemma gives $s_2 \leq C(x, y)$. Consequently every tape w with $g_w \leq s$ at either checkpoint

lies in Good(x,y). Successful inversion at the minimizing length gives $\widehat{g}_1 \le g_w$, so by soundness $\widehat{g}_1 = g_w$ on that event. Together with soundness, this yields

$$P(s) - \phi \le \widetilde{P}(s) \le P(s) \qquad (s = s_1 - 1, s_2).$$

The quantile definition gives $P(s_2) \ge \theta_2$. Since $s_1 \ge \lambda_m(0) \ge 1$, the lower checkpoint $s_1 - 1$ is a nonnegative integer below the minimum s_1, so the quantile definition also gives $P(s_1 - 1) < \theta_1$. If moreover $\phi(x,y) \le \varepsilon/2$, which holds outside the exceptional datasets bounded by Markov's inequality above, it follows that

$$\widetilde{P}(s_2) \ge \theta_* + \varepsilon/2, \qquad \widetilde{P}(s_1 - 1) < \theta_* - \varepsilon.$$

Conditioned on the dataset, the T trials are independent and their empirical average at each checkpoint has variance at most $1/(4T)$. Chebyshev's inequality gives

$$\mathbf{Pr}[\widehat{P}(s_2) < \theta_*] \le 1/(T\varepsilon^2), \qquad \mathbf{Pr}[\widehat{P}(s_1 - 1) \ge \theta_*] \le 1/(4T\varepsilon^2).$$

If neither event occurs, then $\widehat{P}(s_1 - 1) < \theta_* \le \widehat{P}(s_2)$. Since $\widehat{P}$ is nondecreasing and $s_2 \le L$, the smallest s with $\widehat{P}(s) \ge \theta_*$, which is the algorithm's output, lies in $[s_1, s_2]$. The union of these two events has probability at most $5/(4T\varepsilon^2) \le 1/(2p(n))$. Together with the exceptional-dataset probability, the total failure is at most $1/p(n)$. This holds for each parameter whose interval contains a successful length, and hence for infinitely many n. The algorithm never computes $D(y|x)$, g_w, or the good-tape set; these are only analysis quantities. □

The same construction works when the two probability thresholds are only an inverse polynomial apart. This requires a more accurate quantile estimate but leaves the decoding clock unchanged.

Corollary 5.2 (Shrinking quantile windows)**.** *Assume one-way functions do not exist. Fix a polynomial sample-size schedule $m(n) \ge 1$, a PAC-style samplable D, and a polynomial clock t adequate for D. Fix also a polynomial-time computable integer polynomial $\nu(n) \ge 12$ and a positive polynomial $p(n) \ge 1$. Put $\theta_-(n) = 2/3 - 1/\nu(n)$ and $\theta_+(n) = 2/3 + 1/\nu(n)$. Some PPT algorithm B satisfies, for infinitely many n,*

$$\Pr_{D_n,B}[\mathsf{pSC}^t_{\theta_-(n)}(x,y) \le B(1^n,x,y) \le \mathsf{pSC}^t_{\theta_+(n)}(x,y)] \ge 1 - 1/p(n).$$

Proof. The proof repeats that of Theorem 5.1 with new thresholds, so we list what is kept and what changes. Use an integer-coefficient polynomial majorant of p, so all error parameters are computable. Retain the certificate generator f, the strictly increasing padding schedule $N(n)$, the cutoff $L(n)$, and $c'(n)$ from the proof of Theorem 5.1. These objects do not depend on the probability thresholds. Since $\nu \ge 12$, the new endpoints satisfy $0 < \theta_- < \theta_+ \le 3/4$. The coding bound therefore still gives $s_1 \le s_2 \le L$, where $s_1 = \mathsf{pSC}^t_{\theta_-}$ and $s_2 = \mathsf{pSC}^t_{\theta_+}$.

Set $\theta_* = 2/3$ and $\varepsilon(n) = 1/(2\nu(n))$. Both endpoint distances from θ_* are $1/\nu(n) > \varepsilon(n)$. Choose an increasing positive integer polynomial R such that

$$R(n) \ge \frac{8}{\varepsilon(n)} 2^{c'(n)} (L(n)+1) p(n)$$

for every n. Such a majorant exists because $1/\varepsilon = 2\nu$, $2^{c'}$, L and p are polynomially bounded and positive. The contrapositive of amplification supplies one PPT inverter for f with average failure at most $1/R(N)$ at infinitely many lengths N. Call these lengths successful. As before, the finite intervals $[N(n), N(n+1))$ transfer these lengths to infinitely many parameters n.

At each parameter draw $T = \lceil 5p(n)/(2\varepsilon(n)^2) \rceil$ independent public tapes. Try the same lengths, padding lengths and verified inversions as in the proof of Theorem 5.1. Output the smallest integer $s \in [0, L]$ whose recovered empirical cumulative distribution function $\widehat{P}(s)$ is at least $2/3$, or $L+1$ if none exists. The rational threshold and the integer T are computable by integer arithmetic. Their bit lengths and values are polynomially bounded, as are the numbers of attempted inversions. Thus this defines one PPT algorithm for all n; neither its inverter nor its program is chosen anew at each length.

Fix a parameter whose padding interval contains a successful inversion length N_*. The domination calculation in the proof of Theorem 5.1, which uses Equation (6), depends only on L, c' and the inverter

failure. It gives the same missed-minimum probability $\phi(x,y)$ with

$$\mathbb{E}_{D_n}\phi \le \frac{2^{c'+1}(L+1)}{R(N_*)} \le \frac{\varepsilon(n)}{4p(n)}.$$

Here $N_* \ge n$ and the increasing majorant R gives $R(N_*) \ge R(n)$; no monotonicity of ν is needed. Markov's inequality gives $\mathbf{Pr}_{D_n}[\phi > \varepsilon/2] \le 1/(2p(n))$. For each remaining dataset, soundness and recovery of minimizing certificates give $P(s) - \phi \le \widetilde{P}(s) \le P(s)$ at $s = s_1 - 1, s_2$, exactly as in the proof of Theorem 5.1. The relevant tapes remain below the coding cutoff because $s_2 \le \mathsf{pSC}_{3/4} \le C(x,y)$. Consequently

$$\widetilde{P}(s_2) \ge 2/3 + 1/\nu - \varepsilon/2 \ge \theta_* + \varepsilon/2, \qquad \widetilde{P}(s_1 - 1) < 2/3 - 1/\nu < \theta_* - \varepsilon.$$

The independent tape trials have variance at most $1/(4T)$ at either checkpoint. Chebyshev's inequality bounds the upper-checkpoint failure by $1/(T\varepsilon^2)$ and the lower-checkpoint failure by $1/(4T\varepsilon^2)$. Their sum is at most $5/(4T\varepsilon^2) \le 1/(2p(n))$. Outside those failures the empirical cumulative distribution function first reaches $2/3$ between s_1 and s_2. Including the exceptional datasets gives total error at most $1/p(n)$ at infinitely many parameters n. Every certificate is still verified using the given clock t. □

The converse transfers to uniform data, the distribution $D_{0,n}$ of $m(n)$ independent uniform points with independent uniform labels, through the independent-function sampler of Lemma 4.8, whose statistical distance from uniform data is negligible.

Corollary 5.3 (Uniform-data easiness)**.** *Assume one-way functions do not exist. Fix a polynomial schedule m, the sampler D_{ind} from Lemma 4.8, a polynomial clock t adequate for that sampler, constants $0 < \theta_1 < \theta_2 \le 3/4$, and a positive polynomial $p(n) \ge 1$. Some PPT algorithm B has, for infinitely many n,*

$$\mathbf{Pr}_{D_{0,n},B}[\mathsf{pSC}^t_{\theta_1}(X,Y) \le B(1^n,X,Y) \le \mathsf{pSC}^t_{\theta_2}(X,Y)] \ge 1 - 1/p(n).$$

Proof. Apply Theorem 5.1 to D_{ind}, the given clock t, and error polynomial $2p$. It supplies one PPT B with error at most $1/(2p(n))$ at infinitely many parameters. By Lemma 4.8 the statistical distance is at most $\binom{m(n)}{2}2^{-n}$. Since m and p are polynomially bounded, this is at most $1/(2p(n))$ for all sufficiently large n.

By the remark after Definition 2.3, the statistical distance bounds the difference between the probabilities of the interval-success event under the two distributions. The bound applies although this event involves the algorithm's independent coins and need not be efficiently recognizable. The uniform-data error at each sufficiently large parameter in the infinite set is at most $1/(2p(n)) + 1/(2p(n)) = 1/p(n)$. Datasets with conflicting repeated points, which occur under $D_{0,n}$ but not under D_{ind}, are covered by the same statistical-distance bound. □

Together, the uniform-data hardness of Theorem 4.5 and the uniform-data easiness of Corollary 5.3 characterize one-way functions on the fixed uniform dataset distribution $D_{0,n}$.

Corollary 5.4 (Uniform-data equivalence; restatement of Corollary 1.4)**.** *Fix a polynomially bounded, polynomial-time computable integer schedule $m(n) \ge 8n$ and an integer $a \ge 0$. Let D_{ind} be the fixed sampler constructed in Lemma 4.8 for this schedule. One-way functions exist if and only if the following property holds. There are a polynomial clock t, adequate for D_{ind} and permitting full-anchor lookup, and a positive polynomial $p(n) \ge 1$ such that every PPT algorithm eventually has success strictly below $1 - 1/p(n)$ on $D_{0,n}$ in returning a value in*

$$[\mathsf{pSC}^t_{1/2} - a\lceil \log m \rceil, \mathsf{pSC}^t_{3/4} + a\lceil \log m \rceil].$$

Proof. Suppose one-way functions exist, and take t_0 and p from Theorem 4.5 with $\theta_1 = 1/2$ and $\theta_2 = 3/4$. Let $t_{D_{\mathrm{ind}}}$ be the polynomial of Lemma 3.3 for D_{ind}, and put $t = t_0 + t_{D_{\mathrm{ind}}}$. Since $t \ge t_0$, Theorem 4.5 applies at the clock t, which therefore permits full-anchor lookup. Since $t \ge t_{D_{\mathrm{ind}}}$, the clock t is adequate for D_{ind} by the existence clause of Lemma 3.3. This establishes the forward direction with the additional adequacy condition. Conversely, suppose the property holds with a clock t and a polynomial p, but one-way functions do not exist. Apply Corollary 5.3 at this clock t and with this p, taking $\theta_1 = 1/2$ and

$\theta_2 = 3/4$. It gives a PPT algorithm with success at least $1 - 1/p(n)$ on $D_{0,n}$ infinitely often. The interval $[\mathsf{pSC}^t_{1/2}, \mathsf{pSC}^t_{3/4}]$ guaranteed by that corollary lies inside the interval of the hardness property because $a \geq 0$. Hence every output in the former interval also lies in the latter. This contradicts eventual strict hardness for that same algorithm. □

Consider the hardness property of Corollary 5.4, with the same a and the quantiles $1/2$ and $3/4$. The same argument shows that one-way functions exist if and only if, for some polynomial t_1, this property holds at every polynomial clock $t \geq t_1$. Theorem 4.5 gives the forward direction with $t_1 = t_0$ and one polynomial p for all such clocks. Conversely, the clock $t_1 + t_{D_{\mathrm{ind}}}$ is at least t_1 and, by the existence clause of Lemma 3.3, adequate for D_{ind}, so Corollary 5.3 applies to it and contradicts hardness there.

6. Hardness on PAC-Style Samples from One-Way Functions

A pseudorandom-function key gives a short certificate, while random labels rarely have short certificates. We adapt the pseudorandom-versus-random mixture method of Ilango–Ren–Santhanam [IRS22, Section 4.1, Theorems 36–37 of ECCC TR21-082] to independently sampled query points.

Theorem 6.1 (Pseudorandom functions)**.** *If one-way functions exist, there is a polynomial-time family*

$$F_k : \{0,1\}^n \to \{0,1\}, \qquad k \in \{0,1\}^n.$$

For uniform k, its oracle distribution is computationally indistinguishable from a uniformly random Boolean function for every PPT oracle distinguisher [HILL99, GGM86, Gol95].

We import this statement. The theorem of Håstad–Impagliazzo–Levin–Luby constructs pseudorandom generators from one-way functions [HILL99, Theorem 6.2.2 of the authors' version]. From a length-doubling pseudorandom generator, the construction of Goldreich, Goldwasser and Micali builds pseudorandom functions [GGM86, Theorem 3].

These two works are the original sources. We follow the derivation in Goldreich's fragments, where the seed length is the security parameter [Gol95, Definitions 3.9 and 3.30–3.31]. There, one-way functions give a pseudorandom generator that maps every n-bit seed to a longer string [Gol95, Theorem 3.29]. Keeping the first $n+1$ output bits gives a generator with output length $n+1$. Indeed, a test for the shortened output is a test for the full output that ignores the remaining bits, and the first $n+1$ bits of a uniform string are uniform.

Iterating this generator gives one that maps n-bit seeds to $2n$-bit strings [Gol95, Construction 3.10 and Theorem 3.11]. Applied to it, the construction of Goldreich–Goldwasser–Micali gives, for a uniform seed $s \in \{0,1\}^n$, a function $f_s : \{0,1\}^n \to \{0,1\}^n$ that is efficiently computable from s and indistinguishable from a uniformly random function by every PPT oracle machine [Gol95, Construction 3.33 and Theorem 3.34]. The key is the seed, so keys, inputs and outputs all have n bits. In that source the oracle machine on input 1^n queries only n-bit strings, and our distinguishers query only n-bit sample points.

The Boolean family in the statement takes the first bit of each function value. A test for the Boolean family is a test for the original family that projects every oracle answer to its first bit. The same projection of a uniformly random function is exactly a uniformly random Boolean function, including on repeated queries. The program that evaluates F_k is fixed, so a certificate for labels computed by F_k consists of this program's header and the n-bit key.

Theorem 6.2 (Large-gap robust hardness from one-way functions; restatement of Theorem 1.3)**.** *Assume one-way functions exist and fix a polynomial sample-size schedule $m(n) \geq 8n$. There are a PAC-style samplable D and a polynomial clock t adequate for D such that every PPT B satisfies, for all sufficiently large n,*

$$\Pr_{D_n,B}\left[\mathsf{pSC}^t_{1/2}(x,y) - m/4 \leq B(1^n,x,y) \leq \mathsf{pSC}^t_{3/4}(x,y) + m/4\right] < 0.9.$$

Proof. We mix a pseudorandom-function concept with the independent-function concept constructed in Lemma 4.8. The proof first builds the mixture sampler D and its clock t. It then shows that the pseudorandom branch has a small $3/4$-quantile, while the random-function branch has a large $1/2$-quantile except with small probability. Finally, a heuristic that returns a value in the theorem's tolerant interval,

widened by $m/4$ at each end, with probability at least 0.9 at infinitely many parameters would separate the two branches, contradicting Theorem 6.1.

Use the Boolean pseudorandom functions from Theorem 6.1 and draw the points independently and uniformly from $\{0,1\}^n$. With probability $1/2$, choose F_k for a uniform n-bit key. In the other branch, choose the concept of D_{ind} from Lemma 4.8, using $k_{\mathrm{ind}}(n) = \min\{m(n), 2^n\}$-wise independence. Every point vector has $m(n)$ entries from a set of 2^n strings, so it has at most $k_{\mathrm{ind}}(n)$ distinct points. The labels of this concept on those distinct points are therefore independent uniform bits, and repeated points receive the same label. Consequently this branch has exactly the dataset distribution obtained by querying a uniformly random Boolean function.

Both branches choose their concept independently of the points and have polynomial-time sampling and evaluation. To give the mixture one fixed tape length, draw its branch bit, an n-bit pseudorandom-function key, and all $nk_{\mathrm{ind}}(n)$ independent-family seed bits. Each branch ignores the bits that belong to the other branch's concept. Thus the resulting distribution D is PAC-style samplable. Let t be the sum of the polynomial t_D of Lemma 3.3 for this mixture and a polynomial bound on the time to simulate on U the pseudorandom-function certificate defined below. By the existence clause of Lemma 3.3, t is adequate for D.

We now bound the quantiles in each branch, starting with the pseudorandom one. Let σ_{PRF} be the fixed pseudorandom-function evaluation header, of length c_{PRF}. In the pseudorandom branch the empty-anchor certificate $\sigma_{\mathrm{PRF}}k$ works for every public tape, so

$$\mathsf{pSC}^t_{3/4}(x,y) \le n + \lceil \log(m+1) \rceil + c_{\mathrm{PRF}}.$$

In the random-function branch, sample points collide with probability at most $\binom{m}{2}2^{-n}$. For all sufficiently large n, polynomial boundedness of m and $m \ge 8n$ give $16 \le m \le 2^n$, so distinct points have positive probability. Conditioned on each point vector with distinct entries, all labels are independent and uniform, so each of the 2^m label vectors has conditional probability 2^{-m}. The counting bound of Lemma 3.1 applies with the nonnegative threshold $s = m - 10$ and $\theta = 1/2$:

$$\mathbf{Pr}[\mathsf{pSC}^t_{1/2}(x,y) \le m - 10 | \text{the points are distinct}] \le 4 \cdot 2^{-10}.$$

Without conditioning on distinct points, the exceptional event in the random-function branch is that the points collide or $\mathsf{pSC}^t_{1/2}(x,y) \le m - 10$. Its probability is therefore at most

$$\zeta_n = 4 \cdot 2^{-10} + \binom{m}{2} 2^{-n}.$$

Suppose, for contradiction, that some PPT B returns a value in the tolerant interval of the theorem with probability at least 0.9 at infinitely many parameters n. Each branch has probability $1/2$, so at each such parameter the success of B is the average of its successes in the two branches; each is therefore at least 0.8, since the other is at most one. Because m is polynomially bounded, the inequality $\lceil \log(m+1) \rceil + c_{\mathrm{PRF}} < n$ holds eventually. Since $m \ge 8n$, this implies

$$n + \lceil \log(m+1) \rceil + c_{\mathrm{PRF}} + m/4 < m/2.$$

Thus, by the quantile bound of the pseudorandom branch, success in that branch implies $B(1^n, x, y) < m/2$. In the random-function branch, outside the exceptional event, success implies

$$B(1^n, x, y) \ge \mathsf{pSC}^t_{1/2}(x,y) - m/4 > 3m/4 - 10 > m/2.$$

The last inequality holds once $m > 40$, hence eventually. An oracle distinguisher draws the sample points, queries their labels, runs B, and accepts exactly when B returns a well-formed finite value below $m/2$. At each sufficiently large parameter in that infinite set, its acceptance probability is at least 0.8 for a pseudorandom oracle and at most $0.2 + \zeta_n$ for a uniformly random oracle. Outside the exceptional event, acceptance means that B returned a value below $m/2$, so B did not succeed. In the random-function branch, B fails with probability at most 0.2. This acceptance bound for a uniformly random oracle also uses the exact equality of the random-function branch's dataset distribution with the distribution of

datasets labeled by a uniformly random oracle. The distinguisher's advantage is therefore at least

$$0.6 - 4 \cdot 2^{-10} - \binom{m}{2} 2^{-n} > 0.59$$

eventually, because polynomial m makes the collision term negligible. The distinguisher is PPT: it draws polynomially many points and makes m oracle queries before running B. This contradicts pseudorandom-function security. Every PPT B therefore has the stated success strictly below 0.9 for all sufficiently large n. □

Together, the hardness of Theorem 6.2 and the easiness of Theorem 5.1 give the equivalence stated in the introduction.

Corollary 6.3 (restatement of Corollary 1.5)**.** *Fix any polynomial sample-size schedule $m(n) \geq 8n$ and the efficiently universal decoder of Section 2.1. The following are equivalent:*

(1) *One-way functions exist.*
(2) *There are a PAC-style samplable D and an adequate polynomial t for which every PPT algorithm eventually has success below 0.9 in returning a value between $\mathsf{pSC}^t_{1/2} - m/4$ and $\mathsf{pSC}^t_{3/4} + m/4$.*
(3) *There are a PAC-style samplable D, an adequate polynomial t, and a positive polynomial $p(n) \geq 1$ for which every PPT algorithm eventually has success strictly below $1 - 1/p(n)$ in returning a value in $[\mathsf{pSC}^t_{1/2}, \mathsf{pSC}^t_{3/4}]$.*

Proof. We prove the three implications in order. The implication from item 1 to item 2 is Theorem 6.2. The exact interval in item 3 is contained in the tolerant interval of item 2, so item 2 implies item 3 with $p(n) = 10$. Suppose item 3 holds with some D, t and p, but one-way functions do not exist. Apply Theorem 5.1 to this D, t and p, taking $\theta_1 = 1/2$ and $\theta_2 = 3/4$. It provides one algorithm with success at least $1 - 1/p(n)$ infinitely often. Item 3 says that this same algorithm has success strictly below that bound for every sufficiently large n. Those two conclusions are incompatible. Thus item 3 implies item 1. □

The large-gap hardness also applies to the shrinking windows of Corollary 5.2.

Corollary 6.4 (Shrinking-window characterization)**.** *Fix a polynomial sample-size schedule $m(n) \geq 8n$. Let $\nu(n) \geq 12$ be a polynomial-time computable integer polynomial. Put $\theta_-(n) = 2/3 - 1/\nu(n)$ and $\theta_+(n) = 2/3 + 1/\nu(n)$. One-way functions exist if and only if there are a PAC-style samplable D and an adequate polynomial clock t for which every PPT algorithm eventually has success strictly below 0.9 in returning a value in*

$$[\mathsf{pSC}^t_{\theta_-(n)}(x,y), \mathsf{pSC}^t_{\theta_+(n)}(x,y)].$$

Proof. Suppose one-way functions exist and use D, t from Theorem 6.2. Since $\nu \geq 12$, we have $1/2 \leq \theta_- < \theta_+ \leq 3/4$. Quantiles are nondecreasing in their probability threshold. Thus every value in the displayed interval is also in $[\mathsf{pSC}^t_{1/2} - m/4, \mathsf{pSC}^t_{3/4} + m/4]$. Success in the smaller interval implies success in the larger one, so the eventual success bound of Theorem 6.2 also bounds success in the smaller interval. Conversely, suppose that one-way functions do not exist but the stated hardness property holds. Apply Corollary 5.2 to this same D, t, ν with $p(n) = 10$. Its single PPT algorithm has success at least 0.9 for infinitely many n. The hardness property bounds the success of that same algorithm strictly below 0.9 for every sufficiently large n, a contradiction. □

REFERENCES

[ACM+21] Eric Allender, Mahdi Cheraghchi, Dimitrios Myrisiotis, Harsha Tirumala, and Ilya Volkovich. One-way functions and a conditional variant of MKTP. In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021)*, volume 213 of *Leibniz International Proceedings in Informatics*, pages 7:1–7:19. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2021.

[AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures & Algorithms*, 3(3):289–304, 1992. Construction 3 and Proposition 3 as numbered in the revised version of February 22, 2002, hosted by Alon.

[AGHP93] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Addendum to "simple constructions of almost k-wise independent random variables". *Random Structures & Algorithms*, 4(1):119–120, 1993.

[BFKL94] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology—CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 278–291. Springer, 1994.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[CJ26] Marco Carmosino and Mandar Juvekar. Parallel Kolmogorov complexity with connections to learning theory and cryptography, 2026. ECCC TR26-176.

[CW79] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[FW95] Sally Floyd and Manfred Warmuth. Sample compression, learnability, and the Vapnik–Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

[GKLO22] Halley Goldberg, Valentine Kabanets, Zhenjian Lu, and Igor C. Oliveira. Probabilistic Kolmogorov complexity with applications to average-case complexity. In *37th Computational Complexity Conference (CCC 2022)*, volume 234 of *Leibniz International Proceedings in Informatics*, pages 16:1–16:60. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2022.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[GNO26] Halley Goldberg, Mikito Nanashima, and Igor C. Oliveira. Learning, compression, and meta-complexity, 2026. arXiv:2610.16535v1, October 12, 2026.

[Gol95] Oded Goldreich. Foundations of cryptography (fragments of a book). Electronic Colloquium on Computational Complexity, Books, Lectures and Surveys, 1995.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2001.

[HIL+23] Shuichi Hirahara, Rahul Ilango, Zhenjian Lu, Mikito Nanashima, and Igor C. Oliveira. A duality between one-way functions and average-case symmetry of information. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1039–1050, 2023. Full version: ECCC TR23-035.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. Authors' version, with Theorem 6.2.2 as the main theorem, hosted by Håstad.

[Hir22] Shuichi Hirahara. NP-hardness of learning programs and partial MCSP. In *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 968–979, 2022.

[Hir23] Shuichi Hirahara. Capturing one-way functions via NP-hardness of meta-complexity. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1027–1038, 2023.

[HLN24] Shuichi Hirahara, Zhenjian Lu, and Mikito Nanashima. Optimal coding for randomized Kolmogorov complexity and its applications. In *Proceedings of the 65th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 369–378, 2024. Full version: arXiv:2409.12744, Theorem 4.11.

[HLO25] Shuichi Hirahara, Zhenjian Lu, and Igor C. Oliveira. One-way functions and pKt complexity. In *Theory of Cryptography—TCC 2024*, volume 15364 of *Lecture Notes in Computer Science*, pages 253–286. Springer, 2025.

[HN23] Shuichi Hirahara and Mikito Nanashima. Learning in Pessiland via inductive inference. In *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 447–457, 2023. Full version: ECCC TR23-100, Appendix C.

[HN26] Shuichi Hirahara and Mikito Nanashima. A sharp characterization of Pessiland, 2026. Revised full version: ECCC TR26-052, revision 1, June 23, 2026. Conference version in Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC 2026), pages 354–364, doi:10.1145/3798129.3800755.

[IL90] R. Impagliazzo and L. A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 812–821, 1990.

[ILO20] Rahul Ilango, Bruno Loff, and Igor C. Oliveira. NP-hardness of circuit minimization for multi-output functions. In Shubhangi Saraf, editor, *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *Leibniz International Proceedings in Informatics*, pages 22:1–22:36. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2020.

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory, Tenth Annual IEEE Conference*, pages 134–147, 1995.

[IRS22] Rahul Ilango, Hanlin Ren, and Rahul Santhanam. Robustness of average-case meta-complexity via pseudorandomness. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1575–1583, 2022. Full version: ECCC TR21-082, under the title Hardness on Any Samplable Distribution Suffices: New Characterizations of One-Way Functions by Meta-Complexity.

[LOZ22] Zhenjian Lu, Igor C. Oliveira, and Marius Zimand. Optimal coding theorems in time-bounded Kolmogorov complexity. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *Leibniz International Proceedings in Informatics*, pages 92:1–92:14. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2022. Full version: arXiv:2204.08312.

[LP20] Yanyi Liu and Rafael Pass. On one-way functions and Kolmogorov complexity. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1243–1254, 2020. Preprint: arXiv:2009.11514.

[LP21a] Yanyi Liu and Rafael Pass. Cryptography from sublinear-time average-case hardness of time-bounded Kolmogorov complexity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 722–735, 2021. Full version used for Definition 5.2 and Lemma 5.3: ECCC TR21-055; also Cryptology ePrint Archive 2021/517.

[LP21b] Yanyi Liu and Rafael Pass. On the possibility of basing cryptography on EXP $\neq$ BPP. In *Advances in Cryptology—CRYPTO 2021, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 11–40. Springer, 2021.

[LP22] Yanyi Liu and Rafael Pass. On one-way functions from NP-complete problems. In *37th Computational Complexity Conference (CCC 2022)*, volume 234 of *Leibniz International Proceedings in Informatics*, pages 36:1–36:24. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2022.

[LP23] Yanyi Liu and Rafael Pass. One-way functions and the hardness of (probabilistic) time-bounded Kolmogorov complexity w.r.t. samplable distributions. In *Advances in Cryptology—CRYPTO 2023, Part II*, volume 14082 of *Lecture Notes in Computer Science*, pages 645–673. Springer, 2023. Full version: ECCC TR23-152, 2023; also available as Cryptology ePrint Archive 2023/1591.

[LP25] Yanyi Liu and Rafael Pass. On one-way functions, the worst-case hardness of time-bounded Kolmogorov complexity, and computational depth. In *Theory of Cryptography—TCC 2024, Part I*, volume 15364 of *Lecture Notes in Computer Science*, pages 222–252. Springer, 2025. Full version used for Definition A.1 and Lemma A.3: ECCC TR23-103, revision 1, September 5, 2023, under the title On One-way Functions and the Worst-case Hardness of Time-Bounded Kolmogorov Complexity.

[LP26] Yanyi Liu and Rafael Pass. One-way functions and boundary hardness of randomized time-bounded Kolmogorov complexity. In *17th Innovations in Theoretical Computer Science Conference (ITCS 2026)*, volume 362 of *Leibniz International Proceedings in Informatics*, pages 97:1–97:19. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2026.

[LS24] Zhenjian Lu and Rahul Santhanam. Impagliazzo's worlds through the lens of conditional Kolmogorov complexity. In *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, volume 297 of *Leibniz International Proceedings in Informatics*, pages 110:1–110:17. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2024. Full version: ECCC TR24-085.

[LW86] N. Littlestone and M. Warmuth. Relating data compression and learnability, 1986. Unpublished manuscript.

[MY16] Shay Moran and Amir Yehudayoff. Sample compression schemes for VC classes. *Journal of the ACM*, 63(3):21:1–21:10, 2016.

[Nan21] Mikito Nanashima. A theory of heuristic learnability. In *Proceedings of Thirty Fourth Conference on Learning Theory (COLT)*, volume 134 of *Proceedings of Machine Learning Research*, pages 3483–3525, 2021.

[NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993.

[Rii07] Søren Riis. Information flows, graphs and their guessing numbers. *The Electronic Journal of Combinatorics*, 14(1):R44, 2007.

[RS21] Hanlin Ren and Rahul Santhanam. Hardness of KT characterizes parallel cryptography. In *36th Computational Complexity Conference (CCC 2021)*, volume 200 of *Leibniz International Proceedings in Informatics*, pages 35:1–35:58. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2021.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Yao82] Andrew C. Yao. Theory and application of trapdoor functions. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 80–91, 1982.

[Zim24] Marius Zimand. On one-way functions and the average time complexity of almost-optimal compression. arXiv:2412.03392v2, 2024. Version 2, January 23, 2025.

APPENDIX A. PRECISION OF QUANTILE RECOVERY

The converse, Theorem 5.1, returns a value between two separated public-tape quantiles. This section gives two exact-recovery consequences and explains what they leave unresolved. Exact recovery is possible at the fixed threshold $2/3$ when the cumulative public-tape probability just below the $2/3$ quantile stays a margin below $2/3$. It is also possible at a randomly chosen threshold, because such a threshold is, with high probability, far from the cumulative probabilities at all relevant prices. Both recovery procedures use the given adequate pointwise clock.

A.1. **Exact Computation with a Lower Margin.** A gap below the target probability suffices for exact computation; no gap above it is needed.

Corollary A.1 (Exactness with a lower margin; restatement of Corollary 1.6)**.** *Assume one-way functions do not exist. Fix a polynomial sample-size schedule $m(n) \geq 1$, a PAC-style samplable D, an adequate polynomial clock t, a positive polynomial $p(n) \geq 1$, and a polynomial-time computable integer polynomial $\nu(n) \geq 1$. Write $h(x,y) = \mathsf{pSC}^t_{2/3}(x,y)$ and $F_{x,y}(s) = \mathbf{Pr}_w[g^t_w(x,y) \leq s]$. If, for all sufficiently large n,*

$$\Pr_{D_n}\left[F_{x,y}(h(x,y)-1) > \frac{2}{3} - \frac{1}{\nu(n)}\right] \leq \frac{1}{2p(n)},$$

then some PPT algorithm B satisfies $\mathbf{Pr}_{(x,y)\sim D_n,B}[B(1^n,x,y) = h(x,y)] \geq 1 - 1/p(n)$ for infinitely many n.

Proof. We reconstruct tape minima by verified inversion and estimate their distribution at the two prices around the quantile. The lower margin makes an empirical crossing distinguish those prices. Put $\Delta = 1/\nu(n)$ and choose a polynomial-time computable integer polynomial $P(n) \geq p(n)$.

We reuse the certificate generator f from the proof of Theorem 5.1 together with its notation, with one change. Here P denotes the integer majorant chosen above, and the tape distribution function written P in that proof is written $F_{x,y}$. The coding overhead is $c'(n) = \lceil \log(m+1) \rceil + c_Q + 3$, the price cutoff is $L(n) = q(n) + c'(n)$, and the padding schedule N assigns input lengths to parameters. For a supported dataset, the coding bound is $C(x,y) = \log(1/D(y|x)) + c'$.

Put $K(n) = 2^{c'(n)+1}(L(n)+1)$ for the factor transferring generator inversion error to dataset error and choose an increasing integer polynomial R satisfying $R(n) \geq 16P(n)K(n)\nu(n)$ for every n. All these factors are polynomially bounded, so such a majorant exists. Since one-way functions do not exist, the contrapositive form of Theorem 2.8 gives a PPT inverter $\mathscr{I}$ with average failure at most $1/R(N)$ on f at infinitely many input lengths N. Call these lengths successful. Because each interval $[N(n), N(n+1))$ is finite, infinitely many distinct parameters n have intervals that contain successful lengths.

Draw $T = \lceil 20P(n)\nu(n)^2 \rceil$ independent public tapes w_j, indexed by $j \in [T]$. For each tape try $\mathscr{I}$ at every $\ell \in [0,L]$ and every $N \in [N(n), N(n+1))$, using independent uniform pads and fresh inverter coins. Verify returned inverses by re-evaluating f. Let $\widehat{g}_j$ be the smallest verified certificate price, or infinity when no inverse is verified. Define the empirical distribution function $\widehat{F}(s) = T^{-1}|\{j \in [T] : \widehat{g}_j \leq s\}|$. Output the least $s \in [0,L]$ with $\widehat{F}(s) \geq 2/3 - \Delta/2$, and output $L+1$ if there is no crossing. The rational threshold is exactly computable and all searches and trials are polynomially bounded.

Fix n whose interval contains a successful inverter length N_* and which is large enough for the margin premise to hold; infinitely many such n exist, since the premise excludes only finitely many lengths. Let $\phi(x,y)$ be the probability, over a uniform public tape, the uniform pads and the inverter's coins, that the tape satisfies $g_w \leq C(x,y)$ and the inversion attempt at its minimizing length and at N_* fails. The domination calculation in Theorem 5.1, which depends on R only through the average inversion error, gives

$$\mathbb{E}_{D_n}\phi \leq K(n)/R(N_*) \leq \Delta/(16P(n)).$$

The second inequality uses $N_* \geq n$ and monotonicity of R. Markov's inequality gives $\mathbf{Pr}_{D_n}[\phi > \Delta/4] \leq 1/(4P(n))$.

For every supported dataset, monotonicity of quantiles and coding give $h \leq \mathsf{pSC}^t_{3/4}(x,y) \leq C(x,y) \leq L$, and $h \geq \lambda_m(0) \geq 1$. We compare the true and reconstructed probabilities at the two prices immediately around the quantile: $s = h-1$ and $s = h$. Both prices are at most $h \leq \mathsf{pSC}_{3/4} \leq C(x,y)$, so every tape contributing to $F_{x,y}(s)$ satisfies $g_w \leq C(x,y)$, the condition of the coding bound. Soundness gives $\widehat{g}_j \geq g_{w_j}$ on every outcome, and successful inversion at the minimizing length gives equality. Consequently, writing $\widetilde{F}(s) = \mathbf{Pr}[\widehat{g}_1 \leq s | x,y]$,

$$F_{x,y}(s) - \phi \leq \widetilde{F}(s) \leq F_{x,y}(s) \qquad (s = h-1, h).$$

Now consider a dataset with $F_{x,y}(h-1) \leq 2/3 - \Delta$, which the margin premise guarantees outside an event of probability at most $1/(2p(n))$, and with $\phi \leq \Delta/4$. For such a dataset,

$$\widetilde{F}(h-1) \leq \frac{2}{3} - \Delta, \qquad \widetilde{F}(h) \geq \frac{2}{3} - \Delta/4.$$

The first inequality combines the upper bound $\widetilde{F}(h-1) \le F_{x,y}(h-1)$ of the previous display with the margin $F_{x,y}(h-1) \le 2/3-\Delta$. The second combines the lower bound $\widetilde{F}(h) \ge F_{x,y}(h)-\phi$ of the previous display with $F_{x,y}(h) \ge 2/3$, which holds because h is the $2/3$ quantile, and with $\phi \le \Delta/4$. The algorithm's threshold $2/3-\Delta/2$ is therefore at least $\Delta/2$ above $\widetilde{F}(h-1)$ and at least $\Delta/4$ below $\widetilde{F}(h)$. Conditioned on the dataset, the trials are independent; each empirical average has variance at most $1/(4T)$. Chebyshev's inequality bounds the probability of a spurious crossing at $h-1$, meaning $\widehat{F}(h-1) \ge 2/3-\Delta/2$, by $1/(T\Delta^2)$. It bounds the probability of a missed crossing at h, meaning $\widehat{F}(h) < 2/3-\Delta/2$, by $4/(T\Delta^2)$. Their sum is at most $5/(T\Delta^2) \le 1/(4P(n))$.

If neither event occurs, monotonicity makes the first crossing exactly h. The algorithm can therefore fail only on one of three events. First, the dataset may violate the margin; this has probability at most $1/(2p(n))$ by the premise. Second, the dataset may have $\phi > \Delta/4$; this has probability at most $1/(4P(n))$ by Markov's inequality above. Third, for a dataset outside the first two events, one of the two Chebyshev events may occur; given such a dataset, this has probability at most $1/(4P(n))$. The total failure probability is therefore at most

$$\frac{1}{2p(n)} + \frac{1}{4P(n)} + \frac{1}{4P(n)} \le \frac{1}{p(n)},$$

where the last step uses $P(n) \ge p(n)$. This holds for infinitely many n. The quantities $h, F_{x,y}, C$ and the good-tape set are used only in the analysis. □

A.2. **Exact Computation at a Random Threshold.** The next result removes the margin assumption by randomizing the probability threshold. The threshold is chosen independently of the dataset from an explicitly specified polynomial-size grid. The guarantee averages over this choice; it does not assert exact recovery at every grid point or at the fixed threshold $2/3$.

Theorem A.2 (Exact recovery at a random threshold; restatement of Theorem 1.7)**.** *Assume one-way functions do not exist. Fix a polynomial sample-size schedule $m(n) \ge 1$, a PAC-style samplable $D = (\mu, Q)$, an adequate polynomial clock t, and a positive polynomial $p(n) \ge 1$. Let $c'(n) = \lceil \log(m+1) \rceil + c_Q + 3$, with c_Q the constant of Lemma 3.3 for D, and let $L(n) = q(n) + c'(n)$ be the resulting integer price cutoff, as in the proof of Theorem 5.1. Choose a polynomial-time computable integer polynomial $P(n) \ge p(n)$, and let $G(n)$ be the least power of two satisfying $G(n) \ge 8P(n)(L(n)+1)$. For $0 \le j < G(n)$ put*

$$\theta_j = \frac{1}{2} + \frac{2j+1}{8G(n)}.$$

There is one PPT algorithm $B = B_{D,t,p,P}$ such that, for infinitely many point lengths n,

$$\mathop{\mathbf{Pr}}_{(x,y)\sim D_n, j, B}\left[B(1^n, x, y, j) = \mathsf{pSC}^t_{\theta_j}(x,y)\right] \ge 1 - \frac{1}{p(n)}.$$

The grid index is uniform on $\{0, \ldots, G(n)-1\}$ and independent of the dataset and the algorithm's coins. The decoder inside every certificate is run for the original $t(n)$ steps.

Proof. We use the single certificate generator from Theorem 5.1 for every grid point. The proof bounds three exceptional events in turn: too many missed inversions on the dataset, a large empirical sampling error, and a random threshold close to a cumulative probability. Outside these events, the algorithm outputs the quantile exactly.

Write $L = L(n)$, $P = P(n)$ and $G = G(n)$ when the parameter is fixed, and put

$$\delta(n) = \frac{1}{64P(n)(L(n)+1)}, \qquad K(n) = 2^{c'(n)+1}(L(n)+1).$$

All these quantities are positive and computable in polynomial time. Both $1/\delta$ and K are bounded by polynomials in n. Since G is the least power of two that is at least $8P(L+1)$, it satisfies $G < 16P(L+1)$. Because G is a power of two, a uniform grid index uses exactly $\log G$ independent fair bits.

Use the generator f, strictly increasing padding schedule N, and dataset-dependent coding bound $C(x,y)$ from the converse proof. Choose an increasing integer polynomial R such that

$$R(n) \geq \frac{8P(n)K(n)}{\delta(n)}$$

at every length. Such a majorant exists because all factors on the right are polynomially bounded. By Theorem 2.8, there is one PPT inverter $\mathscr{I}$ whose average failure on f is at most $1/R(N_*)$ at infinitely many input lengths N_*. Call these lengths successful. Each interval $[N(n),N(n+1))$ is finite, so the successful lengths lie in the intervals of infinitely many distinct parameters n. Fix one such n and a successful length N_* in its interval. The algorithm does not need to know which parameters these are or which length is successful.

For a supported dataset, define $\phi(x,y)$ as the probability that a public tape satisfies $g^t_w(x,y) \leq C(x,y)$ and that the inversion attempt at its minimum price and at N_* fails. The probability includes the uniform pad and inverter coins. The domination calculation in the proof of Theorem 5.1 gives

$$\mathbb{E}_{D_n}\phi \leq \frac{K(n)}{R(N_*)} \leq \frac{\delta(n)}{8P(n)}.$$

The second inequality uses $N_* \geq n$ and the increasing choice of R. Markov's inequality therefore yields

$$\mathop{\mathbf{Pr}}_{D_n}[\phi > \delta/2] \leq \frac{1}{4P}.$$

The algorithm draws

$$T = \left\lceil \frac{8P}{\delta^2} \right\rceil$$

independent public tapes w_i, indexed by $i \in [T]$. For each tape, it tries $\mathscr{I}$ at every integer price $\ell \in [0,L]$ and every input length in $[N(n),N(n+1))$, using fresh inverter coins and independent uniform pads. It verifies every returned inverse by evaluating f again. Let $\widehat{g}_i$ be the smallest verified certificate price in trial i, or ∞ if none is verified. As in the converse, soundness gives $\widehat{g}_i \geq g^t_{w_i}$ on every outcome. If the attempt at the minimum price and at N_* succeeds, equality holds. All attempts and verifications take polynomial time. Define

$$\widehat{F}(s) = \frac{1}{T}\left|\{i \in [T] : \widehat{g}_i \leq s\}\right|, \qquad \widetilde{F}(s) = \mathbf{Pr}[\widehat{g}_1 \leq s | x,y], \qquad F(s) = \mathop{\mathbf{Pr}}_{w}[g^t_w(x,y) \leq s].$$

Conditioned on the dataset, the trials are independent and identically distributed, and they are independent of the grid index j. For each fixed integer s, the estimate $\widehat{F}(s)$ averages T independent indicators with mean $\widetilde{F}(s)$, so its variance is at most $1/(4T)$. Chebyshev's inequality gives

$$\mathbf{Pr}[|\widehat{F}(s) - \widetilde{F}(s)| > \delta/2 | x,y,j] \leq \frac{1}{T\delta^2} \leq \frac{1}{8P}.$$

Fix a supported dataset and an index j, and write $h = \mathsf{pSC}^t_{\theta_j}(x,y)$. The two prices $h-1$ and h are determined by (x,y,j), and $1 \leq h \leq L$ as shown below. Hence, except with probability at most $1/(4P)$, the bound $|\widehat{F}(s) - \widetilde{F}(s)| \leq \delta/2$ holds at both $s = h-1$ and $s = h$.

For each fixed dataset and price s, count the grid points at distance at most δ from $F(s)$. Consecutive grid points are separated by $1/(4G)$. An interval of length 2δ contains at most $8\delta G + 1$ of them: if it contains r points, its first and last have distance $(r-1)/(4G) \leq 2\delta$. Hence a uniform index j satisfies $|F(s) - \theta_j| \leq \delta$ with probability at most $8\delta + 1/G$. A union bound over the $L+1$ prices gives

$$\mathop{\mathbf{Pr}}_{j}[\exists s \in \{0,\ldots,L\}\,(|F(s) - \theta_j| \leq \delta) | x,y] \leq (L+1)\left(8\delta + \frac{1}{G}\right) \leq \frac{1}{4P}.$$

The terms $8\delta(L+1)$ and $(L+1)/G$ are each at most $1/(8P)$, by the choices of δ and G, respectively. This bound holds separately for every dataset.

Given the index j, the algorithm outputs the least $s \in [0,L]$ for which $\widehat{F}(s) \geq \theta_j$, or $L+1$ if there is no such price. All thresholds are rational with polynomial bit length, so these comparisons are exact polynomial-time operations. The construction is the same for every grid point and uses the same inverter.

Suppose none of the three exceptional events listed at the start of the proof occurs, and let h be as above. All grid points lie strictly between $1/2$ and $3/4$. The coding bound therefore gives $h \le \mathsf{pSC}^t_{3/4}(x,y) \le C(x,y) \le L$. Every certificate price is at least $\lceil \log(m+1) \rceil \ge 1$, so $h \ge 1$. By the quantile definition, $F(h-1) < \theta_j \le F(h)$. Outside the third exceptional event, $|F(s) - \theta_j| > \delta$ for every price $s \in \{0,\dots,L\}$, so these inequalities strengthen to

$$F(h-1) < \theta_j - \delta, \qquad F(h) > \theta_j + \delta.$$

Both prices $h-1, h$ are at most $h \le \mathsf{pSC}_{3/4} \le C(x,y)$, so the relevant tapes satisfy $g_w \le C(x,y)$, the condition of the coding bound. Verified inversion and the definition of ϕ imply

$$F(s) - \phi \le \widetilde{F}(s) \le F(s) \qquad (s = h-1, h).$$

Together with $\phi \le \delta/2$ and the empirical bound at these two prices, these inequalities give

$$\widehat{F}(h-1) < \theta_j, \qquad \widehat{F}(h) > \theta_j.$$

Monotonicity makes the first empirical crossing exactly h. The three exceptional events have combined probability at most $3/(4P) \le 1/p(n)$ over the dataset, independent grid index and algorithm coins. The estimate holds at every parameter n whose padding interval contains a successful inverter length, hence at infinitely many point lengths. The quantities F, h, C, ϕ and N_* enter only the analysis. Every verified certificate still evaluates the original decoder at its prescribed clock $t(n)$. □

The lower-margin and random-threshold results give two sufficient conditions for exactness. Open Problem 1 asks for an algorithm that computes the quantile at the fixed threshold $2/3$ with no margin premise, and neither result supplies one.

Appendix B. Global and Pointwise Descriptions

A program that receives all sample points has information unavailable to the pointwise hypothesis. We compare the two interfaces while keeping the public tape before the choice of description. This comparison uses a global analogue defined below. We do not identify that analogue with any probabilistic complexity in which the program must be fixed before the tape is drawn.

Fix an efficiently universal machine V that takes a finite program, a sequential conditioning input, and a public tape, and outputs a finite binary string. As for the pointwise decoder U, a fixed procedure is specified by a self-delimiting header, whose length depends only on that procedure, followed by an arbitrary finite payload. The simulation overhead of V is polynomial in the simulated time and the lengths of the program and conditioning input, independently of unread public bits. For a positive integer e, let $\mathrm{enc}(e)$ consist of the unary length of its binary expansion, a zero separator, and that expansion. This is a self-delimiting encoding. Encode an ordered vector $x = (x_1,\dots,x_m)$ of n-bit points by

$$E(x) = \mathrm{enc}(n)\,\mathrm{enc}(m)\,x_1 \cdots x_m.$$

The global decoder V reads these fields sequentially and needs no length oracle.

Definition B.1 (Global public-randomness description)**.** A global clock is a positive integer-valued function $T = T(n)$. Fix such a clock and draw a uniform public tape $W \in \{0,1\}^T$. Write $h^T_W(x,y)$ for the minimum length of a program that makes V halt with output exactly y within that clock. The program receives conditioning input $E(x)$. Set this minimum to ∞ if no such program exists. For $0 < \theta \le 1$, put

$$\mathsf{pK}^T_{\mathrm{all},\theta}(y \mid E(x)) = \min\{s \in \mathbb{Z}_{\ge 0} : \mathop{\mathbf{Pr}}_W[h^T_W(x,y) \le s] \ge \theta\},$$

with value ∞ when the set is empty. Thus the program may depend on W, just as a sample certificate may depend on its public tape. Only the program length is charged; the complete ordered point vector is the conditioning input.

The comparison allows the global computation enough time to evaluate every sample point. Its clock is not the original pointwise clock.

Theorem B.2 (Global and pointwise reconstruction; restatement of Theorem 1.8)**.** *Fix polynomial schedules $m(n) \geq 1$ and $t(n) \geq 1$, the pointwise decoder U, and the global decoder V. There are a polynomial clock $T(n)$ and constants c, c_{cyc} such that, for every dataset and every $0 < \theta \leq 1$,*

$$\mathsf{pK}^{T}_{\mathrm{all},\theta}(y|E(x)) \leq \mathsf{pSC}^{t}_{\theta}(x,y) + c.$$

Moreover, draw the points $X_i = (B_i, \widetilde{X}_i)$ independently and uniformly from $\{0,1\}^n$, where B_i is the first bit and $\widetilde{X}_i$ is the remaining suffix. Set $Y_i = B_{i+1}$, with $B_{m+1} = B_1$, so that each label is the first bit of the next point, cyclically. Write $B = (B_1, \ldots, B_m)$ for the first-bit vector. The point and label vectors are $X = (X_1, \ldots, X_m)$ and $Y = (Y_1, \ldots, Y_m)$, respectively.

Then, for every $0 < \theta \leq 1$, the global quantile satisfies $\mathsf{pK}^{T}_{\mathrm{all},\theta}(Y|E(X)) \leq c_{\mathrm{cyc}}$ for every point vector, whereas, for every integer $s \geq 0$,

$$\Pr_{X}[\mathsf{pSC}^{t}_{\theta}(X,Y) \leq s] \leq \frac{2^{2s+2-m}}{\theta}.$$

If $\varepsilon_n = \binom{m}{2} 2^{-n} < 1$, then the same probability conditioned on the points being distinct is at most

$$\frac{2^{2s+2-m}}{\theta(1-\varepsilon_n)}.$$

Proof. The proof has three parts. First, a fixed compiler program for V takes a certificate encoding as its payload and simulates the certificate on every sample point, which gives the quantile inequality. Second, a constant-length global program computes the cyclic labels. Third, we count the certificates that can realize the cyclic labels and apply Markov's inequality over the points.

For the first part we fix the encoding of certificates and its length, then define the compiler and bound its running time. Put $b_m = \lceil \log(m+1) \rceil$. The binary encoding of a certificate (A, a, β) with k retained points consists of a b_m-bit retained-set size, a $\lceil \log \binom{m}{k} \rceil$-bit subset rank, its k tags, and the finite program β. Different certificates receive different encodings; that is, the encoding is injective. An impossible size or rank is rejected. For $k = 0$ or $k = m$, the rank field has zero bits. The global decoder learns m from $E(x)$, so these field widths are determined as it parses the encoding. The encoding has length exactly $\lambda_m(k) + k + |\beta|$, where $\lambda_m(k)$ is the retained-set charge of Section 2.1.

We need a uniform simulation bound for all minimizing certificates. During a t-step pointwise computation, at most the first t cells of the sequential program input can be read. If $|\beta| > t$, replacing β by its first t bits changes no input symbol read within those steps: the new end marker lies beyond the cells that can be read. The same replacement works for every query and every fixed public tape. The truncation lowers the price, so every certificate attaining a finite minimum g^t_w has $|\beta| \leq t$. Since $\lceil \log \binom{m}{k} \rceil \leq m$ and $k \leq m$, its encoding has length at most $b_m + 2m + t$. This is a polynomial bound in n.

The compiler is a program for V whose fixed header contains the descriptions of U and the schedules m, t, and whose payload is the certificate encoding. The compiler reads $E(x)$, obtains n, m, and checks the declared vector format and sample schedule. It reads the retained-set size and rank, unranks the subset, reads its tags, and finds the end of the remaining finite program by an ordinary sequential scan. The length bound just established makes this scan polynomial, including the end-marker read. The compiler obtains L_A from the full vector and these fields. It reads the first $t(n)$ public bits and simulates the original $t(n)$-step query procedure separately on each x_i using this same finite tape. It applies the original timeout and invalid-output convention and concatenates the resulting m bits.

The compiler recovers the retained set from its rank with a table of binomial coefficients. For unranking, form the Pascal table of binomial coefficients up to row m. Initialize $\binom{0}{0} = 1$ and set entries outside the range $0 \leq j \leq u$ to zero. For each positive row u, set $\binom{u}{j} = \binom{u-1}{j-1} + \binom{u-1}{j}$. Its $O(m^2)$ entries have at most $m+1$ bits.

At position i, let k' be the number of retained positions still to select, and write ρ for the remaining rank. The maintained invariant is

$$0 \leq \rho < \binom{m-i+1}{k'}.$$

Initially $i = 1$, $k' = k$, and the parsed rank satisfies this bound. If $k' = 0$, select no further positions. If $k' = m - i + 1$, select all remaining positions. Otherwise, the inclusion branch has $\binom{m-i}{k'-1}$ subsets. Select

i and decrease k' if the current rank is smaller than this count. In the other case, subtract that count from the rank and exclude i. In the inclusion case the new rank is smaller than the number of completions after selecting i, so it satisfies the next invariant. In the exclusion case, Pascal's identity gives

$$\binom{m-i+1}{k'} - \binom{m-i}{k'-1} = \binom{m-i}{k'}.$$

Thus the reduced rank, after the subtraction, is nonnegative and smaller than the number of completions after excluding i. Advance i after each decision. Each boundary case has a single completion, and the invariant forces its remaining rank to be zero.

There are at most m decisions, each using polynomial-time arithmetic on these bounded integers. The anchor list L_A has at most m points of length n, each with one tag. There are m simulations, each of t steps. Parsing, schedule evaluation, unranking, and these simulations therefore have a fixed polynomial bound in n. Efficient universality of V supplies a polynomial clock $T \geq t$ covering the whole compiler on every encoding of length at most $b_m + 2m + t$, the bound on encoding lengths shown above. Its fixed header contributes a constant c. Neither c nor T depends on the dataset or the public tape.

A uniform T-bit global tape has a uniform t-bit prefix. Whenever a pointwise tape admits a valid certificate of price at most s, its minimizing certificate compiles to a global program of length at most $s + c$ on every global tape with that prefix. Consequently

$$\Pr_W[h^T_W(x,y) \leq s + c] \geq \Pr_w[g^t_w(x,y) \leq s].$$

Taking $s = \mathsf{pSC}^t_\theta(x,y)$ proves the quantile inequality when this value is finite. When $\mathsf{pSC}^t_\theta(x,y)$ is infinite, the right-hand side of the quantile inequality is ∞, so that inequality holds in the extended nonnegative integers.

For the cyclic construction, a fixed global program parses $E(x)$, reads the first bit of each point, and outputs the next point's first bit, wrapping at the final coordinate. It ignores the public tape and has constant length c_{cyc}. Increase the polynomial T, if necessary, to cover this program as well as the compiler. The compiler implication remains valid with the longer global tape because it still uses only its first t bits. This proves the global upper bound on every vector, including vectors with repeated points.

The bound for a fixed certificate adapts the directed-cycle guessing-game bound of Riis [Rii07, Section 3], after conditioning on its retained first bits. Fix the tails $\widetilde{X}_1, \ldots, \widetilde{X}_m$, a pointwise tape w, and one certificate (A, a, β) with k retained points. Also fix the first bits $(B_\iota)_{\iota \in A}$ and B_1. These choices determine the entire anchor list L_A, including its charged tags. For each position i and bit b, let $f_i(b)$ be the output of the original query procedure on $(b, \widetilde{X}_i)$ with this list, program and tape. Certificate validity requires

$$B_{i+1} = f_i(B_i) \quad (1 \leq i \leq m), \qquad B_{m+1} = B_1.$$

The first equation fixes $B_2 = f_1(B_1)$. If the bits through position i have been fixed, the equation at position i uniquely fixes $B_{i+1} = f_i(B_i)$. Induction through $i = m-1$ therefore determines the remaining vector. The final equation and the fixed retained bits can only reject this vector; they introduce no further choices. There are at most 2^{k+1} assignments of the retained first bits and B_1, even when $1 \in A$. Thus this certificate is valid for at most 2^{k+1} first-bit vectors, out of the 2^m equally likely vectors.

By the second assertion of Lemma 3.1, at most 2^{s+1} certificates have price at most s. This count is independent of the values of the sample points. Every such certificate has $k \leq s$, because its tags alone cost k bits. A union bound therefore gives, for the fixed tails and tape,

$$\Pr_B[g^t_w(X,Y) \leq s] \leq 2^{s+1} 2^{s+1-m} = 2^{2s+2-m}.$$

For each vector X, let $F_s(X)$ be the fraction of pointwise tapes on which the cyclic labels have a certificate of price at most s:

$$F_s(X) = \Pr_w[g^t_w(X, Y(X)) \leq s].$$

Averaging the preceding bound over the tails and then over the uniform public tape gives $\mathbb{E}_X[F_s(X)] \leq 2^{2s+2-m}$. The quantile definition gives

$$\mathsf{pSC}^t_\theta(X, Y(X)) \leq s \quad \Rightarrow \quad F_s(X) \geq \theta.$$

Markov's inequality gives the instantiated estimate

$$\begin{aligned}\theta \mathop{\mathbf{Pr}}_X[\mathsf{pSC}^t_\theta(X,Y) \le s] &\le \theta \mathop{\mathbf{Pr}}_X[F_s(X) \ge \theta] \\ &\le \mathbb{E}_X[F_s(X)] \le 2^{2s+2-m}.\end{aligned}$$

Dividing by θ proves the stated bound. It remains to prove the bound conditioned on the points being distinct. Each pair of independent uniform points coincides with probability 2^{-n}. The union bound makes the probability of any repetition at most ε_n. Let $\mathscr{H}$ be any event, and let $\mathscr{D}$ be the distinctness event. Then $\mathbf{Pr}[\mathscr{H}|\mathscr{D}] \le \mathbf{Pr}[\mathscr{H}]/\mathbf{Pr}[\mathscr{D}]$. Since $\mathbf{Pr}[\mathscr{D}] \ge 1-\varepsilon_n > 0$, this proves the conditional bound. □

At $s = \lfloor m/3 \rfloor$, the probability bound of Theorem B.2 for the cyclic labels is at most $4\theta^{-1}2^{-m/3}$. For fixed positive θ, it is exponentially small in m, while the global price is constant. Polynomial m makes the point-collision error negligible. On distinct points every label vector is realizable by a deterministic lookup function, so the separation persists on consistent datasets, those in which every repeated point carries the same label. However, the cyclic labeling rule uses the entire ordered point vector. It does not draw a concept independently of the points, as a PAC-style sample does. The separation therefore gives no counterexample, for PAC-style samples, to a reverse bound of the pointwise quantile by the global quantile plus an additive allowance. Local certificates may retain sample points. The separation above does not show that retaining them is essential to the characterizations; indeed, the short certificates in the hardness proofs of Theorems 4.5 and 6.2 retain none.

For PAC-style samples, the coding bound also gives a reverse comparison with high probability. The next corollary follows from that bound and counting global programs. Its hypothesis excludes the cyclic labeling rule above, whose labels depend on the sample points.

Corollary B.3 (PAC reverse comparison)**.** *Fix a polynomial sample-size schedule $m(n) \ge 1$, a PAC-style sampler $D = (\mu, Q)$, an adequate pointwise clock t, a global clock T, and $0 < \theta \le 1$. Let $b_m = \lceil \log(m+1) \rceil$, and let c_Q be the constant in Lemma 3.3. For every n and every $0 < \varepsilon < 1$, with probability at least $1-\varepsilon$ over $(X,Y) \sim D_n$, writing*

$$r = \mathsf{pK}^T_{\mathrm{all},\theta}(Y|E(X)),$$

we have

$$\mathsf{pSC}^t_{3/4}(X,Y) \le r + 2\log(r+1) + \log\frac{1}{\theta\varepsilon} + b_m + c_Q + 5.$$

When $r = \infty$, the right-hand side is interpreted as ∞.

Proof. The proof counts, for fixed points, the label vectors of each global price, removes by a union bound those of too small conditional probability, and applies Lemma 3.3 to the rest. Fix a point vector x in the support of the point-vector distribution μ^m, and write $P_x(y) = D(y|x)$. For each integer $j \ge 0$, let

$$\mathscr{A}_j(x) = \{y \in \{0,1\}^m : \mathsf{pK}^T_{\mathrm{all},\theta}(y|E(x)) = j\}.$$

On each fixed global public tape, the number of programs of length at most j is $2^{j+1}-1$. Each program produces at most one label vector, so at most 2^{j+1} label vectors have a description of length at most j on that tape. Every labeling whose global quantile is at most j has such a description on a fraction at least θ of the tapes. Summing, over labelings y, the fraction of tapes on which y has a description of length at most j, and exchanging this sum with the average over tapes, gives

$$\theta\left|\left\{y : \mathsf{pK}^T_{\mathrm{all},\theta}(y|E(x)) \le j\right\}\right| \le 2^{j+1}.$$

The descriptions may differ between tapes; the count includes all programs on each tape. In particular, $|\mathscr{A}_j(x)| \le 2^{j+1}/\theta$.

Let $\mathscr{B}_j(x)$ consist of those $y \in \mathscr{A}_j(x)$ for which

$$P_x(y) < \frac{\theta\varepsilon 2^{-j}}{4(j+1)^2}.$$

The conditional probability of this set satisfies

$$\mathbf{Pr}[Y \in \mathscr{B}_j(x) | X = x] \le |\mathscr{A}_j(x)| \frac{\theta \varepsilon 2^{-j}}{4(j+1)^2} \le \frac{\varepsilon}{2(j+1)^2}.$$

For $u \ge 2$, we have $1/u^2 \le 1/(u(u-1)) = 1/(u-1) - 1/u$. Summing this inequality and including the $u = 1$ term gives $\sum_{u \ge 1} u^{-2} \le 2$. The union bound therefore yields

$$\mathbf{Pr}\left[Y \in \bigcup_{j \ge 0} \mathscr{B}_j(x) | X = x \right] \le \sum_{j \ge 0} \frac{\varepsilon}{2(j+1)^2} \le \varepsilon.$$

Outside this union, a supported labeling with finite global price $r = j$ satisfies

$$\log \frac{1}{P_x(y)} \le j + 2\log(j+1) + \log \frac{1}{\theta \varepsilon} + 2.$$

The PAC-style coding lemma applies at the original adequate clock t and gives

$$\mathsf{pSC}^t_{3/4}(x,y) \le \log \frac{1}{P_x(y)} + b_m + c_Q + 3.$$

Combining these inequalities proves the claimed estimate for finite r. For $r = \infty$, the estimate holds by the stated convention. Averaging the conditional failure bound over X gives total failure probability at most ε. □

For fixed θ, inverse-polynomial ε, and polynomially bounded finite global price, the additive allowance is $O(\log n)$. By monotonicity, the same bound holds for every pointwise quantile at a threshold at most $3/4$. This typical-data comparison uses no assumption about one-way functions and leaves both given clocks unchanged. It does not give a pointwise reverse compiler or determine the exact pointwise quantile.

Email address: dimyrisiotis@gbu.edu.cn