Electronic Colloquium on Computational Complexity, Report No. 94 (2024)

Interactive Proofs for General Distribution Properties

Tal Herman[*]
Weizmann Institute of Science

Guy N. Rothblum[†]
Apple

May 19, 2024

Abstract

Suppose Alice has collected a small number of samples from an unknown distribution, and would like to learn about the distribution. Bob, an untrusted data analyst, claims that he ran a sophisticated data analysis on the distribution, and makes assertions about its properties. Can Alice efficiently verify Bob's claims using fewer resources (say in terms of samples and computation) than would be needed to run the analysis herself?

We construct an interactive proof system for any distribution property that can be decided by uniform polynomial-size circuits of bounded depth: the circuit gets a complete description of the distribution and decides whether it has the property. Taking N to be an upper bound on the size of the distribution's support, the verifier's sample complexity, running time, and the communication complexity are all sublinear in N: they are bounded by $\widetilde{O}(N^{1-\alpha} + D)$ for a constant $\alpha > 0$, where D is a bound on the depth of the circuits that decide the property. The honest prover runs in $\mathsf{poly}(N)$ time and has quasi-linear sample complexity. Moreover, the proof system is *tolerant:* it can be used to approximate the distribution's distance from the property. We show similar results for any distribution property that can be decided by a bounded-space Turing machine (that gets as input a complete description of the distribution). We remark that even for simple properties, deciding the property without a prover requires quasi-linear sample complexity and running time. Prior work [Herman and Rothblum, FOCS 2023] demonstrated sublinear interactive proof systems, but only for the much more restricted class of *label-invariant* distribution properties.

[*] Part of this work was done while the author was at Apple. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819702) and from the Simons Foundation Collaboration on the Theory of Algorithmic Fairness.

[†] Part of this work was done while the author was at the Weizmann Institute of Science. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 819702) and from the Simons Foundation Collaboration on the Theory of Algorithmic Fairness.

ISSN 1433-8092

Contents

1 Introduction

What can we learn about the properties of an unknown distribution by drawing i.i.d. samples? How many samples are required, and what is the computational complexity of learning? These are foundational questions. An emerging line of works asks a new question:

Can an untrusted prover convince a verifier that an unknown distribution has some property? How efficient can verification be? What is the cost of generating the proof?

We are interested in proofs that can be verified using fewer samples and computational resources than it would take to perform (without a prover) a data analysis that determines whether the distribution has the property. We are particularly interested in *doubly-efficient proof systems*, where the proof can be generated in polynomial time and sample complexity.

We focus on verifying distribution properties via an interactive proof system [GMR85], where a probabilistic verifier has sampling access to the distribution and communicates with an untrusted prover. This continues a study of proof systems for distribution properties initiated by Chiesa and Gur [CG18]. Drawing inspiration from the property testing literature [GGR98, RS96], the prover's claim is that the distribution has (or is close to having) a property. If the prover's claim is approximately correct, the verifier accepts with high probability. If the claim is *far* from correct, i.e. the distribution is far from the property, then no matter what strategy a cheating prover might follow, the verifier rejects with high probability.

Recently, Herman and Rothblum [HR22, HR23] showed that the family of *label invariant* distribution properties (see below) has doubly-efficient proof systems with efficient verification: for many well-studied properties in this family, verification can be almost quadratically more efficient than deciding the property in the standalone setting (without an untrusted prover). In a label-invariant distribution property (sometimes referred to as a *symmetric* property), changing the labels of elements in the support of a distribution does not change membership in the property.[1] While several well-studied properties are label-invariant, this is a restrictive class: many interesting data analyses consider the particular features of data elements, and are thus inherently not label-invariant. Examples include verifying a distribution's distance from being monotone [BKR04, RS05, RV20] or a junta [ABR16], verifying the results of machine learning algorithms that try to learn to predict an element's label from its features, or verifying demographic statistics that depend the features of each individual. Thus, while prior work demonstrated that there are interesting properties for which verification can be more efficient than performing the analysis, it was not clear whether this can be the case for a more general class of analyses.

1.1 This Work: Proof Systems for General Distribution Properties

We show that a rich class of distribution properties can be verified using sample complexity, communication and verifier time that are *sublinear* in the support size of the distribution. Our new doubly-efficient proof system applies to any distribution property that can be approximately decided by a circuit of bounded depth (or a Turing machine of bounded space, see Appendix A. For ease of presentation we focus on bounded-depth circuits for most of this introduction).

[1] More formally, for a distribution D over the domain $[N]$, and a permutation $\pi : [N] \to [N]$, we let $\pi(D)$ be the distribution obtained by sampling from D and applying the permutation π to the outcome. A property $\mathcal{P}$ is *label-invariant* if for every distribution $D \in \mathcal{P}$, and every permutation π over D's domain, $\pi(D) \in \mathcal{P}$.

We proceed to detail this result: a *distribution property* is a set of distributions (similarly to the way a *language* is a set of strings), parameterized by the size of the domain N. We measure the distance of a distribution D from a property $\mathcal{P}$ by D's total variation distance to the closest distribution in $\mathcal{P}$. We study families of properties characterized by the computational complexity of deciding, given the full specification of a distribution, whether it is δ_c-close to the property (the YES case), or δ_f-far from the property (the NO case). A circuit (or TM) for this problem gets as its input the parameters δ_c, δ_f and a list $((i, D[i]))_{i \in [N]}$ specifying the probability of each element in the support. Taking $\rho = (\delta_f - \delta_c) \geq \mathsf{poly}(1/N)$, the probabilities are specified up to to precision $\mathsf{poly}(\rho)$, and the representation is thus of length $\widetilde{O}(N)$.[2] Note that this is a purely computational problem: the distribution is fully specified, there is no need to draw samples. We say that a distribution property can be $\rho(N)$-approximately decided (or "approximated") by a circuit family of depth $D(N)$ and size $S(N)$ if there exists a family of logspace uniform Boolean circuits with fan-in 2 with the specified depth and size that decides the aforementioned decision problem for every δ_c, δ_f s.t. $(\delta_f - \delta_c) \geq \rho(N)$.

Theorem 1.1 (Main result: IPs for depth-bounded properties)**.** *There exists a constant $\alpha > 0$ s.t. for every approximation parameter $\rho = \rho(N) \in (0, 1)$, and every property that can be $\rho(N)$-approximately decided by a circuit family of depth $D = D(N)$ and size $S = S(N)$, there is an interactive proof system as follows. The prover and the verifier both get as input an integer N and proximity parameters $\varepsilon_c, \varepsilon_f \in [0, 1]$ s.t. $\varepsilon_f - \varepsilon_c \geq \Theta(\rho)$, as well as sampling access to a distribution D over the domain $[N]$, where*

- *Completeness: if D is ε_c-close to the property and the prover follows the protocol, then the verifier accepts with all but small constant probability.*
- *Soundness: if D is ε_f-far from the property, then, no matter how the prover cheats, the verifier rejects with all but small constant probability.*
- *Efficient verification: the verifier's sample complexity is $N^{1-\alpha} \cdot \mathsf{poly}(1/\rho, \log(N))$. The protocol's communication complexity and verifier runtime are $N^{1-\alpha} \cdot \mathsf{poly}(1/\rho, \log(N)) + D \cdot \mathsf{polylog}(S)$. The protocol has $O(D \cdot \log(S))$ rounds.*
- *Doubly-efficient prover: the honest prover's sample complexity is $N \cdot \mathsf{poly}(1/\rho, \log(N))$ and its runtime is $\mathsf{poly}(S) + (N \cdot \mathsf{poly}(1/\rho), \log(N))$.*

See Appendix A for a statement for bounded-space computations. We emphasize that the protocol achieves *tolerant* verification [PRR06]: the verifier should accept even if the distribution is not in the property, so long as it is *close to the property.* The complexity is polynomial in the gap $\rho = (\varepsilon_f - \varepsilon_c)$ between the distances. Tolerant verification can be used to approximately verify the distribution's distance to the property: if the prover claims the distance is δ, we can verify this (up to distance ρ) by setting $\varepsilon_c = \delta$ and $\varepsilon_f = \delta + \rho$ in our proof system. We remark that in the distribution testing setting (without a prover) tolerant testing for many well-studied problems requires quasi-linear sample complexity (see below).

We make several remarks about the protocol's complexities. The sample and communication complexities are sublinear: their dependence on N is $\widetilde{O}(N^{1-\alpha})$. For the sample complexity, the

[2] One can consider different natural representations. Many of these are equivalent under NC or log-space reductions. They are thus interchangable for our purposes, and we fix the presentation discussed above.

concrete bound achieved by our protocol is $\widetilde{O}(N^{1-\frac{1}{19}})$ and for the communication complexity it is $\widetilde{O}(N^{1-\frac{1}{38}})$, see the discussion concluding Section 2 for further details and open questions that could lead to improvements in the exponent. In contrast, for distribution testing without a prover, and especially for tolerant testing, many properties require linear or quasi-linear $\Omega(N/\log(N))$ sample complexity. For many of these properties, verification requires $\Omega(\sqrt{N})$ samples (regardless of the communication or round complexities) [CG18, HR22]. Finally, we remark that our protocol makes extensive use of private coins. For clarity of exposition, protocols are presented as if the honest prover has *perfect* knowledge of the distribution, but this idealized honest prover can implemented by an honest prover that learns a sufficiently-accurate approximation to the distribution.

Huge domain, bounded support. We extend Theorem 1.1 to distributions over a huge domain $\mathcal{U}$, so long as the support size of the distribution is bounded (in both the YES case and the NO case). The complexities are only poly-logarithmic in $|\mathcal{U}|$ (verification is sublinear in the support size). This is quite a natural setting, e.g. if the distribution is over at most M individuals, but each individual's representation can come from a rich domain. We show the extension using a general domain-reduction technique, see Section 7 for further details and formal statements.

Corollary 1.2. *The result of Theorem 1.1 can be extended to properties of distributions over a domain $\mathcal{U} = \mathcal{U}(N)$ where the support is of size at most $M = M(N)$. The bound on the support size needs to hold for both completeness and soundness. In terms of complexity, the verifier's sample complexity is $\widetilde{O}((M \cdot \log(|\mathcal{U}|))^{1-\alpha} \cdot \mathsf{poly}(1/\rho))$. The communication complexity and verifier runtime are $(\widetilde{O}(M^{1-\alpha} \cdot \mathsf{poly}(\log(|\mathcal{U}|), (1/\rho)) + D \cdot \log(S))$. The honest prover's sample complexity is $\widetilde{O}(M) \cdot \log|\mathcal{U}| \cdot \mathsf{poly}(1/\rho)$ and its runtime is $(\mathsf{poly}(S) + \widetilde{O}(M) \cdot \mathsf{poly}(\log(|\mathcal{U}|), (1/\rho))$.*

Comparison to known results. Compared to the [HR23] protocol, our protocol applies to a much more general class of properties (namely, properties that are not label-invariant). The verifier's complexity in our protocol is $\widetilde{O}(N^{1-\alpha})$, whereas they have a $\widetilde{O}(\sqrt{N})$ verifier. Our round complexity is also considerably larger, whereas they had a 2-message protocol (the protocol in Appendix A, which applies to bounded-depth computations, has constant round complexity). Chiesa and Gur [CG18] showed a result for general distribution properties: verification requires quasi-linear communication and verification time, but only $O(\sqrt{N})$ samples. The main distinction with our work is that we focus on verification that is simultaneously efficient in terms of the verifier's running time, of the communication complexity, and of the sample complexity.

An application to verifying machine learning from samples. Theorem 1.1 is very general, and can be applied in different settings. We highlight an application to a particular machine learning task. Suppose that Alice and Bob can both sample from a dataset $X = ((x_1, y_1), \ldots, (x_N, y_N))$ of labeled examples. Alice asks Bob to run an empirical risk minimization algorithm M on the entire dataset, to produce a classifier whose empirical loss is approximately minimal in a benchmark class $\mathcal{H}$ w.r.t a loss function L. Bob responds with a classifer h with loss β and claims that h is an approximate empirical loss minimizer, that is

$$\beta = \sum_{i \in [N]} L(h(x_i), y_i) \leq \min_{h' \in \mathcal{H}} \left(\sum_{i \in [N]} L(h'(x_i), y_i) \right) + \varepsilon. \tag{1}$$

Alice can draw a few samples from the dataset to verify that h's empirical loss is approximately β, but how can she verify that the empirical loss is minimal within $\mathcal{H}$ without running M herself on the entire dataset X? This is similar in spirit to verifying PAC learning [GRSY21, GJK+24], but here we are focused on the empirical loss (rather than the loss w.r.t. the underlying distribution), and we assume that M is guaranteed to output an (approximate) empirical risk minimizer.

If the algorithm M can be run in bounded-depth or bounded-space, then Alice can use the protocol of Theorem 1.1 to verify the claim in Equation (1) using only $\widetilde{O}(N^{1-\alpha} \cdot \mathsf{poly}(1/\varepsilon))$ samples. If, for example, the size of the dataset is roughly the VC dimension of $\mathcal{H}$ (the optimal sample complexity for agnostic learning), then the verification is sublinear in the VC dimension: in this setting, for any task that has a bounded-space or bounded-depth learning algorithm with optimal sample complexity, verification is provably more efficient than learning would be!

To verify the empirical loss, Alice and Bob run the protocol on the circuit that, on input a distribution describing X (a list of the elements, each with probability $1/N$), runs M, computes the loss of the resulting empirical risk minimizer h^*, and outputs 1 if this best loss is at least $(\beta - \varepsilon)$. If Bob was honest, then this circuit should accept the dataset X and Alice will accept. If, however, there is a risk minimizer h^* whose empirical loss is smaller than h's by $\Omega(\varepsilon)$, then X is at Hamming distance $\widetilde{\Omega}(\varepsilon)$ from a dataset X' for which the best loss is $(\beta - \varepsilon)$. Thus, the distribution induced by sampling from X is at a similar statistical distance from satisfying the circuit, and Alice will reject w.h.p. We remark that the distribution can be over a huge domain, but its support is over the N elements $\{(x_i, y_i)\}$ (see Section 7). Note that the labels (i.e. the features) of data items are quite important for the property being verified, so verification for label invariant properties (as in [HR23]) does not seem helpful. We also remark that in recent work, Gur *et al.* [GJK+24] give a general result for verifying machine learning using few labeled examples, assuming that the prover is unbounded and the verifier is allowed to draw many unlabeled samples from the underlying distribution. In the application we consider here, the prover knows the entire distribution (random samples from the dataset), but our verifier does not use access to additional unlabeled examples: it is truly sublinear in the size of the dataset. We conclude by remarking that the novelty here is that verification can be performed *using only sample access* to the dataset X (otherwise, if the verifier has query access, it can use an interactive proof of proximity (IPP), see Section 1.2).

1.2 Further Related Work

We study the verification of distribution properties via interactive proofs. Interactive proof systems were introduced by Goldwasser, Micali and Rackoff [GMR85] in the context of proving computational statements about an input that is fully known to the prover and the verifier. In our work, the distribution can be thought of as the input, but it is not fully known to the verifier. We aim for verification without examining the distribution in its entirety, using minimal resources (samples, communication, runtime, etc.). Our work builds on a line of work that studies the power of sublinear time verifiers, who cannot read the entire input [EKR04, RVW13, GR18], on verifying properties of distributions using a small number of samples [CG18, HR22, HR23], and on verifying the result of machine learning algorithms using a small number of labeled examples [GRSY21, GJK+24]. Goldberg and Rothblum [GR22] study *sample-based* IPPs, where the input is a string $x \in \{0,1\}^n$ and the verifier has sampling access to uniformly random input locations, i.e. samples (i, x_i) where i is uniform in $[n]$. There is some similarity to our model in the fact that the verifier only gets samples from a certain distribution, but the setting is quite different: the input is still a string (rather than a distribution), and the samples are guaranteed to be uniformly random over the support $\{(i, x_i)\}_{i \in [n]}$. Aaronson *et al.* [AGRR23] define and construct *distribution-free* IPPs, where the verifier should reject inputs that are far from the language according to a distance measured by an unknown underlying distribution D over the input locations $[n]$. The verifier has query access to x (as usual in an IPP), and also gets sampling access to (i, x_i) where i is chosen by D. Here too, the verifier has access to samples from an unknown distribution, but the setting is again quite different in talking about properties of strings rather than distributions. On a technical level, the *protocol* behind Theorem 1.1 does build on an IPP where some of the verifier's queries are sampled by a distribution Y, but the fact that a query was sampled from Y should not be revealed to the prover (this is different from the setting and the construction in [AGRR23]), see Section 4.3.

2 Technical Overview of Theorem 1.1

At a very high level idea, the protocol is structured as follows:

- The prover and the verifier define a representation of the distribution D *as a bit string* X_D of quasi-linear length in N, and a bounded-depth circuit C. If D is close to the property then C accepts X_D, and if D is far from the property, then X_D is far (in Hamming distance) from satisfying C. The representation is via a hash table, where each element z in the support of D is hashed to several locations (using several hash functions). The number of locations z is hashed into is proportional to the probability that D assigns to z. The honest prover knows X_D explicitly. The verifier, on the other hand, only has restricted access to X_D: it can draw samples from D, and each sample tells it about the value of a number of locations in X_D. A central challenge is that the verifier doesn't know what z's probability by D is, so it doesn't know which hash entries z affects. We elaborate on this below, but for now assume the verifier knows the probability of each sample z, so it can also determine the values of the appropriate locations in the hash table.

- The prover and the verifier run an Interactive Proof of Proximity (IPP, see below) for verifying that X_D satisfies the circuit. The IPP verifier needs to query a sublinear number of locations in the input X_D. Our verifier cannot do so, as its access to X_D is very restricted. Thus, the verifier asks the (untrusted) prover to provide the values of X_D at the locations queried by the IPP. Of course, a cheating prover might lie about some values. To detect this kind of cheating, the verifier chooses its query set in the IPP to include hidden queries about locations chosen by drawing samples from D (as above). This means that the verifier knows the value of X_D at some of the locations queried by the IPP, and the prover doesn't know exactly which locations those are (the prover does know, for example, that the hidden queries are to non-empty hash table entries, but it doesn't know much more). We use a *robust* IPP, where the prover needs to lie on many input locations to pass the verifier's tests, including also on hidden queries. We carefully design tests that ensure the prover has to cheat on some of the hidden queries, which will lead the verifier to reject.

- A sample z drawn from D conveys information about one or more locations in the hash table X_D. The set of locations depends on z's probability mass (by D), but the verifier doesn't know z's probability. It can ask the prover for the probability, but the prover might lie. Thus, we use a protocol of Herman and Rothblum [HR23], which allows the verifier to check alleged probabilities provided by an untrusted prover for a collection of samples from D. The protocol guarantees that the probabilities will be good approximations w.h.p. Our verifier needs to know the probabilities *before* running the IPP (so it knows which locations to query), but it needs to keep the hidden samples z secret from the prover. In our protocol, the verifier guesses these probabilities and runs the IPP using hidden queries derived from its guesses. After the IPP ends and the prover provides the alleged values of X_D in the queried locations, the verifier reveals the hidden samples and uses the [HR23] protocol to obtain approximations for their probabilities. We design tests that ensure the prover has to lie on the values of (many) hidden queries where the verifier's guesses were good enough. If the verifier learns the locations of queries where its guesses were good enough, it will see that the prover lied on such locations, and reject w.h.p. The prover can try to hide the fact that the verifier's guesses were good by misreporting those elements' probabilities, but this will be detected by the [HR23] protocol.

We proceed with a more comprehensive overview of the protocol. We begin in Section 2.1 with a warm-up for the case where there is a promise that the distribution D is exactly uniform over an (unknown) set of size S. In Section 2.2 we extend the warm-up to the case of a general distribution *where the verifier knows the probability of each sample.* That warm-up already describes the way we represent a distribution as a bit string in the full protocol. Finally, we highlight challenges and ideas from the full protocol in Section 2.3.

2.1 Warm-up I: Uniform Over a Set of Size S

We begin with a warm-up: we assume that the unknown distribution is uniform over an (unknown) set of a known size $S = \Theta(N)$. Membership of the distribution in the property depends on the specific set over which it is uniform (the property is not label-invariant). We assume that this promise holds in both the YES case and the NO case. While this is mainly intended as a warm-up, the restriction can be interesting in its own right: for example, it would apply to a distribution obtained by random sampling from a population of individuals with distinct features (further, the case where $S \ll N$ can be handled using a general domain-reduction technique, see Section 7).

Representation. A key step in our construction is representing the distribution D (over domain $[N]$) as an M-bit string X_D. We highlight two important properties of the representation: the first is *correctness:* D can be *reconstructed* from X_D (up to small statistical distance) by a logspace uniform NC^1 circuit. The circuit takes as input the string X_D and outputs the list containing each element's probability by D. We refer to (the functionality computed by) this circuit as the *reconstruction function.* The second property is *distance preservation:* the reconstruction of any string that is close to X_D in Hamming distance gives a distribution that is close to D in statistical distance. By composing the reconstruction circuit with the approximate decision circuit guaranteed by Theorem 1.1, we get a bounded-depth circuit that accepts X_D if D is close to the property, and rejects X_D if D is far. The prover and the verifier run an IPP for the language specified by this circuit on the input X_D. If the verifier had query access to X_D it could now verify that X_D is in the language $\mathcal{L}$, and hence D is not far from the property. The challenge is that the verifier only has limited access to X_D, so we will need to carefully design tests for verifying that $X_D \in \mathcal{L}$ using only limited access to X_D (more on this below).

For the warm-up we use a simple representation: X_D is an N-bit string, where $X_D[z] = 1$ iff z is one of the S elements in D's support. The reconstruction of a given string $X' \in \{0,1\}^N$ verifies that it has Hamming weight S (otherwise it rejects), and outputs the distribution that is uniform over the 1's of X_D (entries where X_D equals 1). The correctness property follows: under the promise that D is uniform over S elements, the reconstruction of X_D always outputs (a representaiton of) D. For distance preservation, if X' is δ-close to X_D in Hamming distance, then either the reconstruction rejects it or it outputs a distribution D' that is uniform over S elements, where the supports of D and of D' differ in at most $(\delta \cdot N)$ elements, so D' is $\Theta(\delta)$-close to D.

Interactive Proofs of Proximity (IPP). We briefly recap IPPs and a specific protocol used in our construction. IPPs, defined and studied by [EKR04, RVW13], are interactive proofs in which the verifier has query access to the input and runs in sub-linear time. The soundness requirement is relaxed to rejecting w.h.p. inputs that are *far* from the language. A sequence of works [RVW13, RRR16, RR20] culminated in IPPs for any language decidable by a bounded-depth circuit or a bounded-space Turing machine:

Theorem 2.1 (IPP of [RR20], informal.). *For any language that can be decided by log-space uniform circuits of depth D and polynomial size, and any desired proximity parameter δ there is a doubly-efficient public-coins* IPP *for δ-proximity with communication complexity $\widetilde{O}(\delta \cdot n + D)$. At the end of the interaction,* before it accesses the input*, the verifier either rejects, or it outputs a partition $\{P_i \subset [n]\}_{i \in [(n/u)]}$ of the input into subsets or blocks of size $u = \widetilde{O}(1/\delta)$, and a decision predicate $\phi_i : \{0,1\}^u \to \{0,1\}$ for each block, where:*

- **Completeness.** *If the input x is in the language, then for every i: $\phi_i(x|P_i) = 1$.*
- **Soundness.** *There is a universal constant $\beta > 0$ s.t. if the input x is δ-far from the language*

$$\Pr_{i \in [(n/u)]} [\phi_i(x|P_i) = o] \geq \beta.$$

The parition and the decision predicates are succinctly described by small NC^1 circuits.

See Theorem 4.7 for a formal statement. We remark that this statement makes explicit some helpful aspects of the [RR20] protocol: the interaction in the protocol is run without the verifier ever needing to examine the input. At the end of this interaction the input is partitioned into subsets of size u, and the prover has made (via the decision predicates) claims about each such subset. In the soundness case, the claims will be false for a constant fraction of the subsets. Thus, the verifier can choose a random subset, query its bits, and check if the decision predicate accepts. This gives constant soundness and query complexity $\widetilde{O}(1/\delta)$. In our work, we repeat the sampling process r times: we derive r claims about disjoint subsets of the input bits. In the soundness case, a constant fraction of the claims are false. In particular, the set of queries bits will be at absolute distance $\Theta(r)$ from satisfying all of those subsets' decision predicates. We leverage this *robustness* in our construction (see Section 4.2 for further discussion about *robust* IPPs).

Running the IPP. The simple representation described above already highlights a key difficulty: the verifier doesn't have query access to X_D. The only access that the verifier *does* have is by sampling random elements from D: for any sampled element $z \sim D$, the verifier knows that $X_D[z] = 1$. Thus, the verifier has sample access to the 1's of X_D. However, for an arbitrary query index, or even for a uniformly random index in $[N]$, the verifier has no idea whether X_D is 0 or 1.

Nonetheless, in our construction, the prover and the verifier run the IPP of Theorem 2.1 to verify that $X_D \in \mathcal{L}$. In the IPP, the verifier either rejects, or it outputs a (succinct description of

a) partitioning of the input into (n/u) sets $\{P_i\}_{i\in[(n/u)]}$, each of size u, and a decision predicate ϕ_i for each set in the partition. If X_D is far from $\mathcal{L}$, then $\phi_i(X_D|P_i) = 0$ for a constant fraction of these sets. We emphasize that the verifier can run the (public coins) interactive phase of the IPP without *any* access to the input X_D. Of course, the verifier can't query the input's values at any of the sets P_i, so it cannot directly check whether the decision predicates accept. Instead, the verifier picks r of the sets $(P_{i_1}, \ldots, P_{i_r})$, and asks the prover to send the values of the input at each of those sets. It checks that the decision predicates accept the values sent by the prover and rejects immediately if this is not the case. To avoid these checks failing, the prover needs to lie on at least one of the bits in $\Omega(r)$ of the sets. In our protocol, the verifier embeds *hidden D-queries*, locations that were sampled from D, among its queries. Our goal is showing that the prover cannot cheat on many of the query locations without also claiming that $X_D[z] = 0$ for some hidden D-query z. Since the verifier knows that z was sampled from D, it knows that $X_D[z] = 1$, and it can catch the cheating prover and reject. In the protocol, for each $\ell \in [r]$, the verifier chooses the partition P_{i_ℓ} by either (with probability 1/2) choosing a random partition (we refer to these as *U-queries*), or (with probability (1/2)), choosing $z_\ell \sim D$ and taking P_{i_ℓ} to be the set in the partition that includes z_ℓ (a hidden D-query). We emphasize that the verifier only sends to the prover the vector $(i_1, \ldots, i_r)$ of sets in the partition that were chosen. It does not reveal how each subset was chosen (uniformly at random or via a D-sample). The prover responds with the claimed values $\bar{v} \in \{0,1\}^{r\cdot u}$ of the input in all locations specified by the verifier's queries. We show that: (i) there are many "1→0 errors" in the values sent by the prover (queried locations where the prover claimed the value is 0, but the true value is 1), and (ii) at least one of the 1→0 errors is on a hidden D-query.

Detecting errors in v. To show the properties claimed above, we need the input to be permuted using an almost $k = \Theta(log(N))$-wise independent permutation Π from $[N]$ to $[N]$, so each set in the partition is approximately k-wise independent. We remark that if we don't permute the input, the support of D might be isolated to certain subsets of the partition, and the prover can avoid any cheating on those. The verifier chooses the permutation and sends it to the prover before running the IPP. Note that the circuit verifying membership in $\mathcal{L}$ needs to invert the permutation, but this can be done in NC^1 (see Section 3.1). The following claim shows that w.h.p. there will be many 1→0 errors in the values v sent by the prover (the probability is over the verifier's public and secret coin tosses, including the choice of the permutation Π).

Claim 2.2 (Many 1→0 errors)**.** *There exists $r = \widetilde{O}(t)$ s.t. for any cheating prover strategy, w.h.p. there are $\Omega(r)$ query subsets P_{i_ℓ} where there is a 1→0 error.*

Proof sketch. While the IPP guarantees that there will be $\Omega(r)$ errors on the U-queries, they might all be 0→1 errors (the cheating prover can choose where to insert adversarial errors). However, this would result in the verifier seeing significantly more 1-answers than it would expect. To circumvent this type of cheating, we add an additional *1-counting test*: the verifier checks that the fraction of 1-values in $\bar{v}$ in the prover's answers to U-queries is close to its expectation. Taking $r = \widetilde{O}(t)$, we show that w.h.p. the prover must insert $\Omega(r)$ errors of the 1→0 type (or be detected). □

We need to show that there will also be at least one 1→0 error on a hidden D-query. The following claim shows that for any query to a 1-location in X_D, the conditional probability that the query was a hidden D-query is not much smaller than $(1/t)$ (this is the best we could hope for, since at most a $(1/t)$-fraction of the queries are hidden D-queries). In particular, this means that w.h.p. at least one of the (many) 1→0 errors will be on a hidden D-query.

Claim 2.3 (Density of D-queries)**.** *W.h.p. over the verifier's choice of* Π *and its coins, for every query* q *in the verifier's query set s.t.* $X_D[q] = 1$ *simultaneously, the probability that* q *is a hidden* D*-query* conditioned on the entire tuple of queries made by the verifier *is* $\Omega(1/(t \cdot \log N))$*. Moreover, for queries in different sets* P_{i_ℓ} *these probabilities are independent.*

Complexity. The protocol's sample complexity is $r = \widetilde{O}(t)$. The communication complexity is dominated by the IPP, where it is $\widetilde{O}(N/t)$, and by the cost of sending $\bar{v}$, which is $(r \cdot t)$. Taking $t \approx N^{1/3}$ gives $\widetilde{O}(N^{2/3})$ communication and $\widetilde{O}(N^{1/3})$ samples.

Discussion. Looking ahead, one thing that makes the warm-up case much easier than the general case, is that the verifier knows *with certainty* that the answers on hidden D-queries should be 1, so it can immediately reject as soon as there's even a single $1 \to 0$ error on such a query. In the general case, on the other hand, the verifier's access to X_D is much more restricted: if there are many $1 \to 0$ errors in the prover's answers this can be detected, but the cost of detecting is inversely polynomial in the fraction of errors. This is the main reason that we incur much larger sample and communication complexities in our general protocol.

2.2 Warm-up II: Known Sample Probabilities

A representation for general distributions. We begin with the case that the distribution is (γ/N)-*grained* for $\gamma \in (0,1)$: the probability of each element is an integer multiple of (γ/N) [GR21]. We represent the distribution using a hash table $X_D \in \{0,1\}^M$, hashing units of (γ/N) mass in D to M "hash buckets". Hashing is performed using a $k = \Theta(\log(N))$-wise independent hash function $h : [N] \times [N/\gamma] \to [M]$ that is chosen by the verifier. The representation is computed as follows: for each element z in the distribution's support, let $(T \cdot \gamma/N)$ be z's probability by D (the distribution is grained, so T is an integer). For each such z, we set the entries $\{h(z,t)\}_{t \in [T]}$ in X_D to be 1. The remaining entries are set to 0. The reconstruction procedure gets a string, and for each element $z \in [N]$ it looks for the longest "prefix" of 1's along the hash locations corresponding to z, i.e. the largest T_z s.t. $\forall t \in [T_z], X_D[h(z,t)] = 1$. The probability of z is set to $(T_z \cdot \gamma/N)$ (these probabilities can be normalized so they sum up to 1). Note that this computation can be performed in NC^1: the probabilities of different elements can be computed in parallel. The non-grained case is handled by randomized rounding (see below).

We remark that the reconstruction is not perfect (unlike Section 2.1): hash collisions might cause some probabilities to be over-estimated. Taking a hash table of size $M = \widetilde{O}(N/\gamma)$, we can bound the effect of such collisions. To argue that strings that are close to X_D in hamming distance will be reconstructed into distributions that are close to D, we need to add some additional checks to the reconstruction procedure. For example, we check that for any entry in the hash table, the number of pairs (z,t) (recovered in the reconstruction procedure) hashed to that entry is at most $O(\log(N))$, otherwise the reconstruction procedure rejects. This bounds the effect that flipping an entry in X_D from 0 to 1 can have on the resulting distribution. See Section 5 for the full details.

Key points in the protocol. We extend the analysis to the case where the verifier knows the probability $D[z]$ of each element sampled from D. This means that the verifier can still sample from the 1-entries in X_D by drawing $z \sim D$, choosing a random hash index $t \in [T_z = D[z] \cdot (N/\gamma)]$ and considering the entry $h(z,t)$. We show that Y_D is close to uniform over 1-entries in X_D.

The ability to sample from the 1-entries of X_D puts us in a situation that is quite similar to the protocol for uniform distributions. In particular, we run the IPP on X_D exactly as described in Section 2.1. The distribution Y_D plays the same role as D in the verifier's choice of query subsets. The prover responds to the verifier's queries with alleged values $\bar{v}$ for those queries. For Claim 2.2 to hold, the verifier needs to know a high-probability upper bound for the number of 1-entries within its U-queries. We show that the *expected* number of 1's is the very close for any underlying distribution D (this is why we need randomized rounding for non-grained distributions), and the observed number of 1's will be close to the expectation w.h.p. The proof of Claim 2.3 also follows by carefully accounting for the effect of hash collisions.

2.3 Towards General Distributions

In the full protocol, the verifier doesn't know the probabilities of elements sampled from D. Instead, it uses a query distribution Y_D over $[M]$ defined as follows: the verifier draws $z \sim D$ and guesses its probability using a (roughly) log-uniform distribution: it guesses $\tau \in [1, \log(N/\gamma)]$ uniformyl at random, takes $T_z = 2^\tau$ and chooses $t \sim [T_z]$. The query location is $h(z,t)$. We say that the guess was "good" if $t \leq (D[z] \cdot N/\gamma)$. If the guess is good, then the value of X_D in the queried location should be 1. Indeed, conditioning Y_D on the guess being good gives a sample distribution that is "well-spread" over the 1's in X_D. We show that the guess is good with (inverse) logarithmic probability, and the set of hidden D-queries with good guesses plays an important role in the analysis. The verifier proceeds to run the IPP as in the Section 2.2, picking subsets of the partition either uniformly at random or by drawing $q \sim Y_D$ and choosing the subset that includes q. This process is repeated r times, resulting in a query set that is sent to the prover. The prover responds with the alleged values $\bar{v}$ of X_D at all queried locations. As before, there must be many $1 \to 0$ errors on the U-queries. We further show that many of these errors will fall on hidden D-queries, similarly to Claim 2.3 (this entails showing upper and lower bounds for the probability that the query distribution Y_D assigns to any 1-entry of X_D). In fact, we show that the fraction of $1 \to 0$ errors is also roughly maintained among the hidden D-queries *with good guesses* (note that, at this point, neither the prover nor the verifier know which queries are in this set).

At this point, we diverge from the warm-ups. There, the verifier knew that the value of X_D on any hidden D-query should be 1. Here, however, it only knows that the value should be 1 if the guess was good. To separate the good guesses from the other D-queries, the verifier needs to learn (good approximations to) the probabilities of its samples from D. To do so, we employ the protocol of [HR23], where the verifier sends its samples to the prover, who responds with their alleged probabilities. If the alleged probabilities are far from the truth then the verifier rejects w.h.p. (see Theorem 3.9 for a formal statement). The verifier checks that, for each D-query where the alleged probability indicates that its guess was good, the value specified by $\bar{v}$ is 1 (otherwise it rejects immediately). The cheating prover wants to convince the verifier that the good-guess query locations where it inserted $1 \to 0$ errors were not, in fact, good guesses. To do so, it needs to report alleged probabilities that are lower than the true probabilities for the elements that lead to those queries. We argue that such mis-reporting will lead to rejection in the [HR23] protocol.

Complexity. As in the prior sections, the number of samples drawn from D in the IPP is r, and the communication complexity is $\widetilde{O}((N/u) + r \cdot u)$. The fraction of $1 \to 0$ errors on the hidden D-queries with good guesses will be $\widetilde{\Omega}(1/u)$. To run the [HR23] protocol on the samples used in the IPP we need r to be $\widetilde{O}(\sqrt{N}/\sigma^c)$, where c is a constant and σ is a measure of the distance between the alleged and true probabilities (this omits several additional steps and technical conditions, see Theorem 3.9). To avoid detection, the cheating prover's lies must create a $\sigma = \widetilde{\Omega}(1/u^2)$ distance. We get that the sample complexity is $\widetilde{O}(\sqrt{N} \cdot u^{2c})$, and the communication complexity is $\widetilde{O}((N/u) + (\sqrt{N} \cdot u^{2c+1}))$. Balancing these terms, we take $u = \widetilde{\Theta}(N^{1/(4(c+1))})$ and get

$$\textit{samples} = \widetilde{O}\left(\sqrt{N} \cdot N^{\frac{2c}{4(c+1)}}\right) = \widetilde{O}\left(N^{1-\frac{1}{2(c+1)}}\right).$$
$$\textit{communication} = \widetilde{O}\left(\sqrt{N} \cdot N^{\frac{2c+1}{4(c+1)}}\right) = \widetilde{O}\left(N^{1-\frac{1}{4(c+1)}}\right).$$

Taking $c = 8.5$, as in [HR23], gives sample complexity $\widetilde{O}\left(N^{1-\frac{1}{19}}\right)$ and communication $\widetilde{O}\left(N^{1-\frac{1}{38}}\right)$.

Digest. Our protocol can be viewed as a reduction from verifying membership in a general distribution property, to verifying the probabilities of a collection of samples from the distribution. This latter task has been explored in recent work [HR23], but is not yet well understood. In particular, to the best of our knowledge the only know lower bound on the sample complexity is $\Omega(\sqrt{N}/\sigma^2)$ [CG18, HR22]. A protocol matching this lower bound, i.e. with $c = 2$, would improve the sample complexity for general properties to $\widetilde{O}(N^{1-\frac{1}{6}})$ and the communication to $\widetilde{O}(N^{1-\frac{1}{12}})$. In terms of the reduction itself, one primary source of overhead is that the prover can "cover up" an η fraction of $1 \to 0$ errors (among the hidden D queries with good guesses) by "moving" a quadratically smaller mass in the reported probabilities ($\sigma \approx \eta^2$). Improving the quadratic dependence in the reduction is a fascinating open question for future work.

Organization of Paper. Preliminaries and definition are in Section 3. In Section 4 we cover the IPPs used in our result. In Section 5 we construct the representation of the distribution D as a string X_D over $\{0,1\}^M$, and prove that this construction features the desired properties outlined above. The full protocol is in Section 6. Lastly, in Section 7 we extend our main result to distribution properties *over large domains* (see Definition 3.4).

3 Preliminaries

For an integer $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, \ldots, n\}$. For integers $r, t \in \mathbb{N}$, we identify the set $[r \cdot t]$ with the set $[r] \times [t]$ in the natural way. For a vector $x \in \Sigma^n$ and a set $S \subseteq [n]$, we use $(x|S)$ or $x[S]$ to denote the restriction of the vector to the subset of coordinates in S. For a discrete set S, a partition $\mathcal{P}$ of S is a collection of subsets, where each $P \in \mathcal{P}$ is a subset of S, the subsets are all disjoint, and their union is the set S.

A machine $\mathcal{A}$ with *oracle access* (or query access) to a string $x \in \{0,1\}^n$ is an oracle Turing Machine with access to the function $f : [n] \to \{0,1\}$ where $\forall i \in [n], f(i) = x_i$. We refer to x as $\mathcal{A}$'s *implicit* input. The machine $\mathcal{A}$ can also have *explicit* input. We denote the output of $\mathcal{A}$ with oracle access to $x \in \{0,1\}^n$ and with explicit input z by $\mathcal{A}^x(z) \in \{0,1\}$.

Let $x, y \in \Sigma^n$ be two strings of length $n \in \mathbb{N}$ over a (finite) alphabet Σ. We define the (relative Hamming) distance of x and y as $\mathrm{Ham}(x,y) = \Delta(x,y) \stackrel{\text{def}}{=} |\{x_i \neq y_i : i \in [n]\}| / n$. If $\Delta(x,y) \leq \varepsilon$, then we say that x is ε-close to y, and otherwise we say that x is ε-far from y. We define the distance of x from a (non-empty) set $S \subseteq \Sigma^n$ as $\Delta(x,S) \stackrel{\text{def}}{=} \min_{y \in S} \Delta(x,y)$. If $\Delta(x,S) \leq \varepsilon$, then we say that x is ε-close to S and otherwise we say that x is ε-far from S. We extend these definitions from strings to functions by identifying a function with its truth table.

Definition 3.1 (NC^1 circuit families)**.** *Throughout this work we use NC^1 to refer to the class of logspace uniform Boolean circuits of logarithmic depth and constant fan-in. Namely, $\mathcal{L} \in \mathsf{NC}^1$ if there exists a logspace Turing machine M that on input 1^n outputs a full description of a logarithmic depth circuit $C : \{0,1\}^n \to \{0,1\}$ such that for every $x \in \{0,1\}^n$ it holds that $C(x) = 1$ if and only if $x \in \mathcal{L}$.*

Definition 3.2. *The total variation distance (alt. statistical distance) between distributions P and Q over a finite domain X is defined as:*

$$\delta_{TV}(P, Q) = \frac{1}{2} \sum_{x \in X} |P(x) - Q(x)|$$

Theorem 3.3 (Folklore distribution learner [Gol17])**.** *There exists an algorithm that given sample access to a distribution P over the domain $[N]$, and an accuracy parameter $\alpha \in (0,1)$, it runs in time $\widetilde{O}(N/\alpha^2)$, takes $O(N/\alpha^2)$ samples, and with probability at least 0.99 outputs a full description of a distribution P_{approx} such that $\delta_{TV}(P, P_{approx}) \leq \alpha$.*

Definition 3.4 (Distribution property)**.** *We say the $\mathcal{P} = (\mathcal{P}_N)_{N \in \mathbb{N}}$ is a distribution property if $\mathcal{P}_N \subseteq \Delta_N$, where Δ_N is the set of all distributions over domain $[N]$.*
We say that $\mathcal{P} = (\mathcal{P}_N)_{N \in \mathbb{N}}$ is a distribution property over a large domain $\mathcal{U} = (\mathcal{U}_N)$, if $\mathcal{P}_N$ is a collection of distributions over domain $\mathcal{U}_N$ with support of size at most N, and we assume $|\mathcal{U}_N| = \omega(N)$.

Definition 3.5 (Distribution tester for property $\mathcal{P}$)**.** *Let δ be some distance measure between distributions, $\mathcal{P}$ a distribution property. A tester T of property Π is a probabilistic oracle machine, that on input parameters N and ε, and oracle access to a sampling device for a distribution D over a domain of size $[N]$, outputs a binary verdict that satisfies the following two conditions:*

1. *If $D \in \mathcal{P}_N$, then $\Pr(T^D(N, \varepsilon) = 1) \geq 2/3$.*
2. *If $\delta_{TV}(D, \mathcal{P}_N) > \varepsilon$, then $\Pr(T^D(N, \varepsilon) = 0) \geq 2/3$.*

In the context of this work, the relevant distance measure is *statistical distance* as defined above. An extension of this definition, introduced by Parnas, Ron, and Rubinfeld [PRR06] is the following:

Definition 3.6 ($(\varepsilon_c, \varepsilon_f)$-tolerant distribution property tester)**.** *For parameters $\varepsilon_c, \varepsilon_f \in [0,1]$ such that $\varepsilon_c < \varepsilon_f$, a $(\varepsilon_c, \varepsilon_f)$-tolerant tester T of property Π is a probabilistic oracle machine, that on inputs $N, \varepsilon_c, \varepsilon_f$ and given oracle access to a sampling device for distribution D over a domain of size N, outputs a binary verdict that satisfies the following two conditions:*

1. *If $\delta(D, \Pi_N) \leq \varepsilon_c$, then $\Pr(T^D(N, \varepsilon_c, \varepsilon_f) = 1) \geq 2/3$.*
2. *If $\delta(D, \Pi_N) \geq \varepsilon_f$, then $\Pr(T^D(N, \varepsilon_c, \varepsilon_f) = 0) \geq 2/3$.*

Note that a tolerant distribution test is for some property Π is at least as hard as a standard non-tolerant tester for the same property.

Definition 3.7 (Proof system for tolerant distribution testing problems)**.** *A proof system for a tolerant distribution testing problem $\mathcal{P}$ with parameters ε_c and ε_f is a two-party game, between a verifier executing a probabilistic polynomial time strategy V, and a prover that executes a strategy P. Given that both V and P have black-box sample access to distribution D over the domain $[N]$, and are given N, the interaction should satisfy the following conditions:*

- ***Completeness:*** *For every D over domain of size at most N, such that $\delta_{TV}(D, \mathcal{P}_N) \leq \varepsilon_c$, the verifier V, after interacting with the prover P, accepts with probability at least $2/3$.*

- ***Soundness:*** *For every D over domain of size at most N such that $\delta_{TV}(D, \mathcal{P}_N) \geq \varepsilon_f$, and every cheating strategy P^*, the verifier V, after interacting with the prover P^*, rejects with probability at least $2/3$.*

The complexity measures associated with the protocol are: the sample complexity of the verifier as as the honest prover (strategy P), the communication complexity, the runtime of both agents, and the round complexity (how many messages were exchanged).

Definition 3.8 (Label invariant distribution property)**.** *A distribution property $\mathcal{P}$ is called label invariant if for all $N \in \mathbb{N}$, it holds that any permutation σ over N elements satisfies that $D \in \mathcal{P}_N$ if and only if $\sigma(D) \in \mathcal{P}_N$.*

Obtaining a verified tagged sample. Herman and Rothblum [HR23] constructed protocols for verifying label-invariant properties of an unknown distribution D over $[N]$. Their protocol allows a verifier, interacting with the untrusted prover, to obtain a set $x_1, \ldots, x_S \sim D$ of i.i.d. samples from D, together with alleged probabilities for those samples. The probability that the verifier accepts and the alleged probabilities are σ-far from the truth (by the distance measure below) is small.

Theorem 3.9 (Verifiable tagged sample protocol [HR23])**.** *Set $c = 8.5$. For any $\sigma = \sigma(n) \in (0, 1)$, and distribution D over domain $[N]$ such that $D(x) \leq \frac{\sigma^c}{\sqrt{N}}$, there is a doubly efficient two-message protocol between a verifier $\mathcal{V}$, who has sample access to an unknown distribution D, and an untrusted prover, as follows. In its first message, the verifier sends a sample $S = (z_1, \ldots, z_s)$ of size $s = \widetilde{O}\left(\sqrt{N} \cdot \sigma^{-c}\right)$, where each z_i is independently drawn either from D (w.p. $1/2$) or from U_N (w.p. $1/2$), where the information which is which is a secret held by $\mathcal{V}$. The prover responds with the alleged probability $\pi(z_i)$ of each z_i by D. Taking $S_D \subseteq [S]$ to be the subset of samples that were drawn by D, let the distance of the tags be*

$$\Delta_{Z,S_D}(\pi, D) = \frac{1}{|S_D|} \cdot \sum_{i \in S_D} \left(1 - \min\left\{\frac{\pi(z_i)}{D(z_i)}, \frac{D(z_i)}{\pi(z_i)}\right\}\right)$$

Then the following holds:

- *If the prover is honest, then w.h.p. $\Delta_{Z,S_D}(\pi, D) \leq \sigma^2$.*

- *For any cheating prover, the probability that $\Delta_{Z,S_D}(\pi, D) > \sigma$ and $\mathcal{V}$ accepts is small.*

The size of S, the communication complexity, and $\mathcal{V}$'s overall sample complexity are all $\widetilde{O}(\sqrt{N}\sigma^{-c})$. The honest prover sample complexity and runtime are both $\widetilde{O}\left(N \mathsf{poly}(\sigma^{-1})\right)$.

In our construction we run this protocol with tiny (inverse polynomial) error parameter σ, so the polynomial dependence on $(1/\sigma)$ shows up as an N^c factor in the sample and communication complexity bounds of Theorem 1.1, see the discussion in Section 2.3.

3.1 Pairwise and k-Wise Independence

We review definitions and constructions of k-wise independent functions and permutations, a notion of k-wise paritions uses in our work, and a concentration inequality for k-wise random variables.

Definition 3.10 (k-wise independent functions). *A family $\mathcal{F}$ of functions mapping $[N]$ to $[M]$ is k-wise independent if for every $x_1, \ldots, x_k \in [N]$, taking a random function f from the family, the distribution of $f(x_1), \ldots, f(x_k)$ is uniformly random in $[M]^k$.*

Definition 3.11 (k-wise independent permutations). *A family $\mathcal{F}$ of permutations mapping $[N]$ to $[N]$ is k-wise independent if for every $x_1, \ldots, x_k \in [N]$, taking a random function f from the family, the distribution of $f(x_1), \ldots, f(x_k)$ is γ-close to a sequence of k random distinct elements from $[N]$.*

Definition 3.12 (k-wise partition). *A partition P over domain $[N]$ divides its domain into t disjoint ordered sets of equal sizes $\{P_i \in [N]^{(N/t)}\}$. A family $\mathcal{P}$ of partitions over $[N]$ is a k-wise γ-dependent if for every fixed $(i_1, \ldots, i_k)$ and $(j_1, \ldots, j_k)$, for a random partition P from the family, the distribution of $P_{i_1}[j_1], \ldots, P_{i_k}[j_k]$ is γ-close to a sequence of k random distinct elements from $[N]$.*

In this work whenever we refer to a k-wise independent partition, the partition is generated by taking a k-wise permutation of the elements of $[N]$ and then applying a fixed partition.

Claim 3.13 (construction of k-wise independent functions). *For $k = k(N) = O(\log(N))$ there is an ensemble of k-wise independent families of functions from $[N]$ to $[N]$. A random function f in the family can be sampled in $\mathsf{polylog}(N)$ time (and has a $\mathsf{polylog}(N)$-size representation). The value of f on an input $x \in [N]$ can be computed by a $\log(N)$-space uniform circuit of depth $O(\log(N))$ and size $\mathsf{polylog}(N)$.*

Proof sketch. We assume w.l.o.g. that N is a power of 2. The construction is achieved by taking the function to be a random polynomial over a field of size N. Healy and Viola [HV06] showed that exponentiation and iterated addition can be performed by very uniform TC^0 circuits of the appropriate sizes and the claim follows. □

Claim 3.14 (construction of k-wise independent permutations). *There is an ensemble of pairwise families of permutations from $[N]$ to $[N]$. A random permutation π in the family can be sampled in $O(log(N)$ time (and has a $O(log(N))$-size representation). The permutation π can be computed and inverted by a $\log(N)$-space uniform circuit of depth $O(\log(N))$ and siace $\mathsf{polylog}(N)$.*

The above construction is obtained by sampling uniformly random a and b from a finite field, where a is non-zero, and mapping x to $(a \cdot x + b)$.

Claim 3.15 (construction of k-wise independent permutations). *For $k = k(N) = O(\log(N))$ and any desired $\gamma = \gamma(N)$ there is an ensemble of k-wise γ-dependent families of permutations from $[N]$ to $[N]$. A random permutation π in the family can be sampled in $\mathsf{polylog}(N, 1/\gamma)$ time (and has a $\mathsf{polylog}(N, 1/\gamma)$-size representation). The value of π on an input $x \in [N]$ can be computed in $\mathsf{polylog}(N, 1/\gamma)$ time. There is a log-space uniform NC^1 circuit (of $\mathsf{poly}(N, \log(1/\gamma))$ size) that inverts the permutation.*

Proof sketch. We use Naor and Reingold's construction [NR99], which is in the Luby-Rackoff framework [LR88]. The construction uses two (perfectly) pairwise independent permutations from $[N]$ to $[N]$ (see Claim 3.14) and two k-wise independent functions from $[N]$ to $[N]$ (see Claim 3.13). Each of these objects can be evaluated by a $O(\log(N)$-space uniform circuit of depth $\log(N)$ and size $\mathsf{polylog}(N)$, and so can the entire construction. This gives $\gamma' = \tilde{\Omega}(1/\sqrt{N})$ dependence. Composing the construction $O(\log_{\sqrt{N}}(1/\gamma))$ times reduces the dependence to γ-dependence (see [KNR09]). The resulting circuit has depth $O(\log(N/\gamma))$ and size $\mathsf{polylog}(N, 1/\gamma)$. The circuit for inverting the function produces (in parallel) the entire truth table (of size $(N \log(N))$) and performs one lookup to this table. □

Remark 3.16. *Composing a k-wise γ-dependent permutation with a (perfectly) pairwise independent permutation gives a permutation that is simultaneously k-wise γ-dependent and perfectly pairwise independent.*

We conclude with a concentration bound for k-wise independent RVs.

Claim 3.17 (2k-wise independent sampling [Gol17])**.** *For $k \leq n/2$, let $X_1, \ldots X_n \in [0,1]$ be $2k$-wise independent random variables and $\mu = \frac{1}{n}\sum_{i\in[n]} \mathbb{E}[X_i]$. Suppose that $Var[X_i] \leq \beta$ for every $i \in [n]$. Then, for every $\varepsilon > 0$, it holds that:*

$$\Pr\left(\left|\frac{1}{n}\sum_{i\in[n]} X_i - \mu\right| \geq \varepsilon\right) \leq \left(\frac{3k\beta}{n\varepsilon^2}\right)^k$$

4 Interactive Proofs of Proximity (IPPs)

4.1 Succinct Descriptions

Following [RR20], we define a notion of succinct representation of circuits. This notion is helpful for specifying the complexity of the verifier in their IPP protocol (see Section 4.3). Loosely speaking, a function $f : \{0,1\}^n \to \{0,1\}$ has a succinct representation if there is a short string $\langle f \rangle$, of poly-logarithmic length, that describes f. That is, $\langle f \rangle$ can be expanded to a full description of f. The actual technical definition is slightly more involved and in particular requires that the full description of f be an NC^1 (i.e., logarithmic depth) circuit:

Definition 4.1 (Succinct Description of Functions)**.** *We say that a function $f : \{0,1\}^n \to \{0,1\}$ has a* succinct description *if there exists a string $\langle f \rangle$ of length $\mathsf{polylog}(n)$ and a logspace Turing machine M (of constant size, independent of n) such that on input 1^n, the machine M outputs a full description of an NC^1 circuit C s.t. for every $x \in \{0,1\}^n$, $C(\langle f \rangle, x) = f(x)$. In particular, C (and thus f) can be evaluated in $\mathsf{poly}(n)$ time. We refer to $\langle f \rangle$ as the succinct description of f.*

We also define succinct representation for sets $S \subseteq [k]$. Roughly speaking, this means that the set can be described by a string of length $\mathsf{polylog}(k)$:

Definition 4.2 (Succinct Description of Sets)**.** *We say that a set $S \subseteq [k]$ of size s has a succinct description if there exists a string $\langle S \rangle$ of length $\mathsf{polylog}(k)$ and a logspace Turing machine M such that on input 1^k, the machine M outputs a full description of a depth $\mathsf{polylog}(k)$ and size $\mathsf{poly}(s, \log k)$ circuit (of constant fan-in) that on input $\langle S \rangle$ outputs all the elements of S as a list (of length $s \cdot \log(k)$).*

We emphasize that the size of the circuit that M outputs is proportional to the actual size of the set S, rather than the universe size k.

4.2 Robust IPPs: Definition

IPPs, defined and studied by [EKR04, RVW13], are interactive proofs in which the verifier runs in sub-linear time in the input length, where the soundness requirement is relaxed to rejecting inputs that are *far* from the language w.h.p. (for inputs that are not in the language, but are close to it, no requirement is made). We further consider IPPs for pair languages: the input of the verifier is composed of two parts: an *explicit* input $y \in \{0,1\}^{n_{exp}}$, to which the verifier has direct access, and an *implicit* (longer) input $x \in \{0,1\}^n$, to which the verifier has oracle access. The goal is for the verifier to run in time that is sub-linear in n and to verify that x is far from any x' such that the pair (y, x') are in the pair language.

We define *robust* IPP*s*, which have a stronger soundness guarantee: the verifier's (sublinear) view of the implicit input should be *far* from a view that would make it accept (given of its random coins and the transcript of its communication with the prover). This notion of robustness follows the PCP and PCPP literature [BGH+06]. Robustness was also considered for probabilistically checkable IPs [RRR16, RR22]. We emphasize that in this work we are concerned with robustness with respect to the bits the verifier reads from the input (rather than robustness w.r.t. queries from the communication transcript).

For our definition, we consider an IPP that proceeds in two phases: a *communication phase*, where the prover and the verifier exchange messages, but the verifier does not query the input, and a subsequent *query phase*, where the verifier queries the input. The verifier then applies a *decision predicate* to its views in the communication phase and the query phase (including any random coins it may toss), and rejects or accepts. See [GRS23] for a more thorough structural study of IPP verifiers. Let Q be the (sublinear) set of bits queried by the verifier in the query phase. In a ρ-robust IPP, if the (implicit) input x is far from the language, then w.h.p. the restriction $(x|Q)$ of the input to the query set is *ρ-far from satisfying* the verifier's decision predicate (w.r.t. the transcript from the communication phase and the verifier's random coins).

Definition 4.3 (Robust Interactive Proof of Proximity (IPP))**.** *A robust* interactive proof of proximity *(*IPP*) for the pair language $\mathcal{L}$ is an interactive protocol with two parties: a (computationally unbounded) prover $\mathcal{P}$ and a computationally bounded verifier $\mathcal{V}$. Both parties get an explicit input $y \in \{0,1\}^{n_{exp}}$, a proximity parameter $\varepsilon \in (0,1)$, and a robustness parameter $\rho \in (0,\varepsilon)$. The verifier also gets oracle access to $x \in \{0,1\}^n$, whereas the prover has full access to x.*

The protocol is divided into two phases. In the interaction phase *the two parties interact, but the verifier does not access the implicit input. The interaction produces a communication transcript τ. In the subsequent* query phase*, the verifier makes its queries Q into the implicit input (based on the explicit input, the transcript τ, and its random coins r). The verifier then applies a decision predicate $\phi_{\mathcal{V}}$ to its views from both phases and accepts or rejects, where the following conditions hold:*

1. **Completeness:** *For every pair $(x,y) \in \mathcal{L}$, $\varepsilon > 0$ and $\rho \in (0,\varepsilon)$ it holds that*

$$\Pr_{(r,\tau,Q)\leftarrow\left(\mathcal{P}(x),\mathcal{V}^x\right)(y,|x|,\varepsilon,\rho)}\Big[\phi_{\mathcal{V}}(y,r,\tau,(x|Q)) = 1\Big] = 1.$$

2. **Soundness:** *For every $\varepsilon > 0$, $\rho \in (0, \varepsilon)$, $y \in \{0,1\}^{n_{exp}}$ and x that is ε-far from the set $\{x' : (x', y) \in \mathcal{L}\}$, and for every computationally unbounded (cheating) prover $\mathcal{P}^*$ it holds that*

$$\Pr_{(r,\tau,Q)\leftarrow\left(\mathcal{P}^*(x),\mathcal{V}^x\right)(y,|x|,\varepsilon,\rho)}\left[\Delta\left((x|Q), \{z : \phi_{\mathcal{V}}(y, r, \tau, z) = 1\}\right) \le \rho\right] \le 1/2.$$

An IPP for $\mathcal{L}$ is said to have $\mathsf{query\ complexity}$ $q = q(n, n_{exp}, \varepsilon)$ if, for every $\varepsilon > 0$ and $(x, y) \in \mathcal{L}$, the verifier $\mathcal{V}$ makes at most $q(|x|, |y|, \varepsilon)$ queries to y when interacting with $\mathcal{P}$. The IPP is said to have $\mathsf{communication\ complexity}$ $\mathsf{cc} = \mathsf{cc}(n, n_{exp}, \varepsilon)$ if, for every $\varepsilon > 0$ and pair $(x, y) \in \mathcal{L}$, the communication between $\mathcal{V}$ and $\mathcal{P}$ consists of at most $\mathsf{cc}(|x|, |y|, \varepsilon)$ bits. If the honest prover's running time is polynomial in n and n_{exp}, then we way that the IPP is *doubly-efficient*.

We make several remarks on the robustness of IPPs.

Remark 4.4 (Optimal robustness)**.** *An* IPP *for ε-proximity cannot (in general) have robustness larger than ε: if the implicit input x is ε-close to the language, then a cheating prover can pretend that the input is a fictitious x' that is in the language (following the honest prover strategy for x'). It might be the case that x and x' differ in a* random *ε-fraction of their indices. Since the verifier would accept if x' was its input, the values it reads from x will not be at a distance large than ε from satisfying the decision predicate.*

Remark 4.5 (Inherent robustness of IPPs)**.** *Any* IPP *has robustness* $(1/q)$*: in the soundness condition, w.h.p. the q bits queried by the verifier will not satisfy the decision predicate. For a natural class of protocols, independent repetitions of the query phase can (roughly) maintain this relative robustness, while increasing the* absolute *distance of the values queried from satisfying the decision predicate (this will be helpful for our construction). If we perform r repetitions of the query phase, then the number of queries can grow by an r multiplicative factor, but we can hope that the absolute distance between the input bits queried and values that would satisfy all r runs increases to $\Omega(r)$. This will be the case, so long as, once we condition on the verifier's view in the communication phase, independent runs of the query phase query disjoint sets of indices w.h.p.*

Remark 4.6 (Generality of two-phase structure)**.** *Any public-coins* IPP *protocol can be transformed into an* IPP *that is divided into a communication phase and a query phase as specified in Definition 4.3. To do so, the verifier sends to the prover any queries made during the interaction (indeed, since the protocol is public coins, the prover should know these values itself, since they can only depend on the coins sent by the verifier). For each query to an index i, the prover sends back the alleged value of $y[i]$. For the remainder of the interaction phase, the verifier takes the prover's claims at face value, and proceeds as if these are the bit values read from the implicit input. Then, in the query phase, the verifier can query the input and substantiate all claims made by the prover. This transformation increases the communication complexity by the query complexity of the original protocol, it increases the round complexity by (at most) a multiplicative factor of 2, and preserves soundness so long as the original protocol was public coins.*

4.3 Robust IPP Constructions

Rothblum and Rothblum [RR20] constructed an IPP for bounded-depth computations (or bounded-space computations) whose query-communication tradeoff is optimal up to $\mathsf{polylog}(n)$ factors. We use several structural properties of their construction: first, it has some inherent robustness (along similar lines to Remark 4.5). Second, the verifier's queries can be made k-wise independent by simply applying a k-wise independent permutation chosen by the verifier to the input (the circuit needs to invert the permutation, this can be done in NC^1 using the construction in Claim 3.15). Lastly, we use the fact that, after the communication phase, the verifier's possible query sets form a partition of the implicit input (thus the absolute distance from satisfying the decision predicate canbe amplified, a la Remark 4.5).

Theorem 4.7 (Robust IPP, k-wise queries)**.** *Let $\delta = \delta(n)$ be a proximity parameter and let $\mathcal{L}$ be a pair language computable by logspace-uniform Boolean circuits of Depth $D = D(n) \geq \log(n)$ and size $S = S(n) \geq n$ with fan-in 2 (where n is the size of the implicit input x and $n_{exp} \leq n$ is the size of the explicit input x_{exp}). Then $\mathcal{L}$ has a public-coin $\Theta(\delta)$-robust IPP for δ-proximity.*

Taking $t = \widetilde{O}(1/\delta)$, for any desired integer $r = r(n)$ and independences parameters $k = k(n) = O(\log(n)), \gamma = \gamma(n) = \mathsf{exp}(-\mathsf{polylog}(n))$ the protocol has

- *Query complexity $q = \widetilde{O}(t \cdot r) = \widetilde{O}\left(\frac{r}{\delta}\right)$.*
- *Communication complexity: $\mathsf{cc} = (\delta \cdot n \cdot \mathsf{polylog}(n) + D \cdot \mathsf{polylog}(S))$.*
- *Round complexity: $O(D \cdot \log(S)) + \mathsf{polylog}(n)$.*
- *Verifier running time: $\left(\left(\delta \cdot n + \frac{r}{\delta}\right) \cdot \mathsf{polylog}(n) + (D + n_{exp}) \cdot \mathsf{polylog}(S)\right)$.*
- *Honest prover running time: $\mathsf{poly}(S)$. .*

Furthermore, at the end of the communication phase, w.h.p. either the verifier rejects, or in time $((\delta \cdot n) \cdot \mathsf{polylog}(n) + ((D + n_{exp}) \cdot \mathsf{polylog}(S)))$ it outputs a succinct description $\langle P \rangle$ of a function $P : [\lceil n/t \rceil] \to [n]^t$ and a succinct description $\langle \phi \rangle$ of a predicate $\phi : [\lceil n/t \rceil] \times \{0,1\}^q \to \{0,1\}$ where

1. *The sets $(P_i = P(i))_{i \in [(n/t)]}$ are a k-wise γ-dependent partition of $[n]$ (see Definition 3.12).*[3]
2. *In the soundness case (i.e. when the input is far from the language) there exists a universal constant $\beta > 0$ s.t.:*

$$\Pr_{i \sim [\lceil n/t \rceil]} [\phi(i, X|P(i)) = 0] \geq \beta. \tag{2}$$

In the protocol, the verifier chooses a uniformly random set $R \subseteq [\lceil n/t \rceil]$ of size r and queries the set $Q = \bigcup_{i \in R} P(i)$. The verifier accepts the implicit input x if and only if $\forall i \in R, \phi(i, (x|P(i))) = 1$. Thus, the protocol is $\Theta(\delta)$-robust.

The theorem follows directly from the IPP protocol of [RR20]. The only novelty is in the k-wise independence requirement from the partition. To obtain this property, the prover and the verifier permute the input using a k-wide γ-dependent permutation from $[n]$ to $[n]$ before running the IPP (see Definition 3.11). The verifier chooses the permutation and sends it to the prover. The verifier then needs to be able to compute the permutation so it knows which input index corresponds to a queried location in the IPP. The circuit on which we run the IPP needs to be able to invert the permutation (so it can undo the permutation and compute the original circuit). Thus, we use a permutation from $[n]$ to $[n]$ that can be represented using $\mathsf{polylog}(n)$ bits, is computable in $\mathsf{polylog}(n)$ time and can be inverted using a logspace-uniform circuit of $\mathsf{poly}(n)$ size and $\mathsf{polylog}(n)$ depth, as in Claim 3.15.

[3] We assume t and (n/t) are integers (otherwise we pad the input with 0's to ensure this is the case).

4.4 A Protocol for Checking D-queries

We use IPPs towards our main construction. The setting is one in which the verifier has limited access to an implicit input: the verifier cannot query the input directly, nor can it sample random indices together with their values, as in the sample-based setting [GR22]. Instead, there is an underlying distribution D over $[n]$ (the indices of the implicit input), and the verifier has (weak) signals about the value of the input at locations drawn from D. In this setup we cannot directly run an IPP on the input (nor can we run a sample-free IPP [AGRR23], see below). Instead, we use a robust IPP to obtain claims about the input (and particularly about the values of the input at locations drawn from D). We use the robustness of the IPP to show that a cheating prover's claims *about the locations drawn from* D are far from the truth. Elsewhere, when we use the protocol, we show how this can be detected using only weak signals about the input's values at those locations.

Relationship to Distribution-Free IPPs. In a recent work, Aaronson *et al.* [AGRR23] define distribution-free IPPs, where the verifier gets sampling access to a distribution D over $[n]$ and should reject inputs that are far from the language, where the distance is measured according to the distribution D. They construct protocols that achieve this goal for general distributions. The verifier's access to the distribution D can be similar to the protocol of Figure 4.7.1.

The major difference is that in our setting, the verifier does not have direct access to the values of the input x at locations drawn from D, it can only get weak signals about their true values. Thus, after running the IPP, the verifier asks the prover to supply the values of all queried indices. The prover's goal is to concentrate its lies outside the D-queries. It is *crucial* that the protocol (including the choice of the query set) does not reveal which queries were made by D. This goal is not present in the work of [AGRR23], and indeed their protocol explicitly reveals the D-queries made by the verifier. On the other hand, in their work they can handle general distributions, whereas we focus on "well spread" distributions (in particular, if the distribution D is very concentrated on a few indices then the prover will always know which queries were made by D-samples).

Protocol overview. We construct a protocol for checking the values of D-queries. Towards this, we use the IPP of Theorem 4.3. The interaction phase is unchanged. Recall that after the interaction, in the query phase, the input has been partitioned into disjoint query sets $\{P_i\}$, each of size t. For a constant β fraction of these query sets, the decision predicate rejects the input (or rather its restriction to that set, i.e. $(x|P_i)$). The verifier chooses r sets to query. Each set is either, w.p. $1/2$, chosen by drawing a sample from D (a "hidden D-query") and querying the set that contains that sample, or, w.p. $1/2$ it is a uniformly random set in the partition (we refer the queries in sets drawn this way as "U-queries"). By the IPP guarantee, the decision predicate will reject a constant fraction of the U-query sets. Intuitively, if D is "well spread" and puts weight at least (μ/n) on each index $j \in [n]$, then the prover can't have confidence about which sets have hidden D queries, nor about where the hidden D-query in a particular query set might be.

After running the IPP and choosing its r query sets, the verifier asks the prover to supply the values of the input on all indices in those query sets. It checks that the values satisfy the decision predicate (otherwise it rejects immediately). To satisfy the decision predicate, the prover needs to lie about the values of $\Omega(r)$ locations in the U-query sets. Since the verifier can (for the most part) only verify values drawn by D, we want to show that the prover also has to cheat on many of the "D-queries". In particular, we need to ensure that, even given the query set, the prover cannot separate the queries that were specified by D-samples from the other queries (and thus cheat only

Protocol for Checking D-queries

Verifier Input. Full access to explicit input $y \in \{0,1\}^{n_{exp}}$, sampling access to a distribution D over $[n]$ (no acces to the implicit input x).
Prover Input. Full access to the explicit and implicit inputs $y \in \{0,1\}^{n_{exp}}, x \in \{0,1\}^n$.

Parameters: Proximity parameter $\delta \in (0,1)$, arity parameter $t \in \mathbb{N}$, number of repetitions r.

Outputs: The verifier rejects or produces public and private outputs.
The public output is a query set $Q \in [n]^{r \cdot t}$, and a vector $\bar{v} \in \{0,1\}^{r \cdot t}$.
The private output includes two disjoint sets $I_D, I_Q \subseteq [r \cdot t]$.

The Protocol:

1. The prover and verifier run the interaction phase of the IPP of Theorem 4.7. The verifier rejects or outputs the (succinct description of the) partition and the decision predicate $\{P_i, \phi_i\}_{i \in [(n/t)]}$.
2. The verifier paritions $[r]$ into two sets, L_D and L_U, where each $\ell \in [r]$ is in L_D w.p. $1/2$ (independently), and otherwise it is in L_U. For each $\ell \in L_U$, the verifier picks an independent partition element $i_\ell \sim U_{[n/t]}$. For each $\ell \in L_D$, the verifier picks $j_\ell \sim D$ and let $i_\ell \in [(n/t)]$ be the index of the (unique) set s.t. $j_\ell \in P_{i_\ell}$. Further, let $h_\ell \in [t]$ be the index of j_ℓ within the parition P_{i_ℓ}.
The verifier sends the selected sets $(i_\ell)_{\ell \in [r]}$ to the prover.
3. The prover sends the alleged values $\bar{v} = ((x|P_{i_\ell}))_{\ell \in [r]} \in \{0,1\}^{r \cdot t}$ of the input at the selected sets.
4. The verifier checks that for each selected set, the alleged values satisfy the decision predicate: $\forall \ell \in [r], \phi_{i_\ell}(\bar{v}_\ell) = 1$, and rejects immediately if this is not the case.
Otherwise, the verifier's public output is $Q = (P_{i_\ell})_{\ell \in [r]}$ and $\bar{v} = (\bar{v}_\ell)_{\ell \in [r]}$.
The private output is $I_D = \{(\ell, h_\ell)\}_{\ell \in L_D}$ and $I_U = \{(\ell, h) : \ell \in L_U, h \in [t]\}$.

Figure 4.7.1: Protocol for Checking D-queries

on the latter). This follows from the fact that the distribution is well spread (we also need a bound on the maximal probability of any element). In fact, we show that for any (large enough) set S of query locations, the *density* of hidden D-queries in that set will be $\widetilde{\Omega}(\mu/t)$, which is not too far from their density within the entire query set. In particular, taking S to be the set of locations on which the prover supplies false answers, we get that there are many false claims about D-queries (in our use of this protocol we will take S to be the set of locations where the prover cheats by saying that x has value 0, whereas the true value is 1). The above applies to any set S chosen by the prover adaptively and adversarially as a function of the queries made by the verifier. The protocol is in Figure 4.7.1 and its guaranteed are in Lemma 4.8.

Lemma 4.8 (Protocol for checking D-queries)**.** *Let $\{D_n\}$ be an ensemble of distributions, where for each $n \in \mathbb{N}$ the distribution D_n is over $[n]$. In the protocol of Figure 4.7.1, applied to languages and parameters as in the* IPP *of Theorem 4.7, the verifier rejects or outputs a tuple $Q \in [n]^{r \cdot t}$ of queries and a vector of alleged values $\bar{v} \in \{0,1\}^{r \cdot t}$. The verifier also produces a private output: two disjoint subsets of the queries, $I_D, I_U \subset [r \cdot t]$. We refer to $Q[I_D]$ as the "D-queries" and to $Q[I_U]$ as the "U-queries".*

The protocol has the following guarantees:

1. **Completeness:** *If $(x,y) \in \mathcal{L}$ and the prover follows the protocol, then the verifier doesn't reject and $(x|Q) = \bar{v}$.*

2. **Robust soundness on U-queries.** *If x is δ-far from $\{x' : (x',y) \in \mathcal{L}\}$, then with all but $\mathsf{exp}(-\Theta(r))$ probability:*

$$\Delta\left((X|Q[I_U]), \bar{v}[I_U]\right) = \Theta(\delta). \tag{3}$$

3. **Query distributions.** *With high probability the sizes of I_D and I_Q is $\Theta(r)$ and the size of I_Q is $\Theta(r \cdot t)$. The D-queries $Q[I_D]$ are independent samples from D. The U-queries are divided into independent batches of size t, where each batch is a draw from a k-wise γ-dependent partition of $[n]$ into partitions of size t.*

4. **Density of D-queries in adversarial sets.** *Let $\mu = \mu(n) \in (0,1)$ be a parameter. If:*

$$\forall j \in [n],\, D[j] \in \left[\frac{\mu}{n}, \frac{t}{n}\right], \tag{4}$$

then for any subset $S \subseteq [r \cdot t]$ of absolute size at least $\Theta\left(\sqrt{r} \cdot t \cdot \log(n)/\mu\right)$, where S can be chosen adaptively and adversarially as a function of the protocol's public outputs, with high probability:

$$\left|S \cap I_D\right| \geq \Theta\left(\frac{\mu}{t \cdot \log(n)} \cdot |S|\right). \tag{5}$$

The probability is over the verifier's coin tosses and its choice of Q_D.

The complexity of the protocol is as in the protocol of Theorem 4.7, where the communication complexity incurs an additional $O(r \cdot t)$ additive overhead.

Proof. The protocol complexity and the claims about the distribution of the queries follow by construction, as does the protocol's completeness.

Robustness on U-queries (Equation (3)). By the robustness of the IPP of Theorem 4.3 (Equation (2)), w.h.p. an almost β-fraction of $\ell \in L_U$ have $\phi_{i_\ell}(x|P_{i_\ell}) = 0$ (since the tests in L_U are chosen uniformly at random). Thus, w.h.p. the absolute distance of $(X|(Q[I_U]))$ from $\bar{v}(Q[I_U])$ is $\Theta(r)$, and the relative distance is $\Theta(1/t) = \Theta(\delta)$.

Density of D-queries (Equation (5)). We analyze the conditional distribution of I_D given Q, $\bar{v}$ and S ($\bar{v}$ and S are a function of Q, so we omit them from the conditioning). The conditional distribution has two components:

1. The set $L_D \subset [r]$ of partitions where there a hidden D-query. By Bayes rule, for each $\ell \in [r]$, the conditional probability that $\ell \in L_D$ is:

$$\Pr[\ell \in L_D | Q] = \frac{D[P_{i_\ell}]/2}{D[P_{i_\ell}]/2 + (t/n)/2},$$

   and the events that $\ell \in L_D$ are all independent.

2. For each $\ell \in L_D$, the index $h_\ell \in [t]$ of the hidden D-element, where:

$$\Pr[(\ell, h_\ell) \in I_D | Q, L_D] = \frac{D[P_{i_\ell}[h_\ell]]}{D[P_{i_\ell}]},$$

and these distributions are independent (for different $\ell \in L_D$).

We want to show that w.h.p. the queries in S are "well represented" in the set I_D. Towards this, we first show that, since the maximal probability D assigns to any element is at most (t/n), w.h.p. over the choice of the partition, there is no set P_i whose probability by D is "too high".

Claim 4.9. *W.h.p. over the choice of the partition $\{P_i\}_{i \in [(n/t)]}$, for every $i \in [(n/t)]$ simultaneously:*

$$D[P_i] = O\left(\log(n) \cdot \frac{t}{n}\right).$$

The proof is by a balls and bins argument, we defer the details for now (see the full proof below). Given this claim, the robustness property follows. For each $\ell \in [r], h_\ell \in [t]$:

$$\begin{aligned}
\Pr[(\ell, h_\ell) \in I_D] &= \frac{D[P_{i_\ell}]/2}{D[P_{i_\ell}]/2 + (t/n)/2} \cdot \frac{D[P_{i_\ell}[h_\ell]]}{D[P_{i_\ell}]} \\
&= \frac{1}{2} \cdot \frac{D[P_{i_\ell}[h_\ell]]}{D[P_{i_\ell}]/2 + (t/n)/2} \\
&= \Omega\left(\mu \cdot \frac{(\mu/n)}{\log(n) \cdot (t/n)}\right) \\
&= \Omega\left(\frac{\mu}{t \cdot \log(n)}\right).
\end{aligned}$$

Moreover, these events are independent for different ℓ's.

Consider now the set $S \subseteq [r \cdot t]$. By the above (and using the linearity of expectation), the expectation of $|S \cap I_D|$ is $\Omega\left(\frac{\mu}{t \cdot \log(n)} \cdot |S|\right)$. We want to show that $|S \cap I_D|$ is concentrated around its expectation. Towards this, for $\ell \in [r]$, let S_ℓ be the set of locations in S with prefix ℓ. Then:

$$|S \cap I_D| = \sum_{\ell \in [r]} \mathbb{1}_{(I_D \cap S_\ell) \neq \emptyset}.$$

Consider the RVs $\mathbb{1}_{(I_D \cap S_\ell) \neq \emptyset}$. These are r independent $\{0,1\}$ random variables. We use Azuma's inequality to bound the probability that their sum deviates from its expectation:

$$\Pr\left[|S \cap I_D| \leq \left(\mathbb{E}\left[|S \cap I_D|\right] - \Theta\left(\frac{\mu}{t \cdot \log(n)} \cdot |S|\right)\right)\right] = \mathsf{exp}\left(-\Theta\left(\frac{\mu^2 \cdot |S|^2}{r \cdot t^2 \cdot \log^2(n)}\right)\right).$$

This probability is small so long as S is larger than $\Theta\left(\frac{\sqrt{r} \cdot t \cdot \log(n)}{\mu}\right)$.

Proof of Claim 4.9. We begin by restricting our attention to the set M of elements in $[n]$ whose probability by D is above $(100/n)$. The set M is of size at most $(n/100)$, and we want to bound the max contribution that the elements in M make to any one "bin" in the partition (in terms of that bin's probability by D). We analyze this as a balls and bins process: tossing the "balls" in M into the (n/t) bins of the partition (note that the "bins" are of bounded size t, but we can ignore this since w.h.p. the maximum number of elements in any bin will be at most say $(t/50)$).

We recall (rather relaxed) max load bounds for such a processes when c balls are tossed into d bins using a $O(\log(c))$-wise independent hash function. In the "many balls" case, where $c \geq 2d\log(d)$, w.h.p. the max load is $O(c/d)$. In the "few balls" case, where $c < 2d\log(d)$, w.h.p. the max load is at most $O(\log(c))$.

Recall that the maximal probability of any element by D is bounded by (t/n). We partition the elements of M into $B = O(\log(n))$ buckets according to their probabilities, where for $b \in [B]$, the b-th bucket is:

$$M_b = \{z \in [n] : D[z] \in (2^{-b} \cdot \frac{t}{n}, 2^{-(b-1)} \cdot \frac{t}{n}]\},$$

where $|M_b| \leq \frac{2^b \cdot n}{t}$. The max contribution of elements in M to the mass of any set P_i in the partition is thus:

$$\begin{aligned}\sum_{b\in[B]} D[P_i \cap M_b] &\leq \sum_{b\in[B]} 2^{-(b-1)} \cdot \frac{t}{n} \cdot |P_i \cap M_b| \\ &= O\left(\frac{t}{n} \cdot \sum_{b\in[B]} 2^{-(b-1)} \max\{\log|M_b|, \frac{|M_b|}{(n/t)}\}\right) \\ &= O\left(\frac{t}{n} \cdot \sum_{b\in[B]} 2^{-(b-1)} \max\{\log(n), 2^b\}\right) \\ &= O\left(\frac{t \cdot \log(n)}{n}\right).\end{aligned}$$

We emphasize that w.h.p. this bound applies to all sets in the partition simultaneously. The elements not in M can contribute at most $(100t/n)$ probability to any set in the partition, and the claim follows. □

□

5 Representation

In this section we formally define the representation of a distribution D over domain $[N]$ as a string $X_D \in \{0,1\}^M$ for $M = \widetilde{O}\left(N\gamma^{-1}\right)$, where $\gamma \in (0,1)$ is some accuracy parameter. The representation of D is one of the main pillars of our construction. For a detailed discussion about the representation and its desired properties, see section 2.1 for a detailed discussion. Throughout this section we also assume that distribution D satisfies $D(x) \leq N^{-f}$ for some $f \in (0.5, 1)$, this assumption is later justified in Remark 6.2.

Construction 5.1 (String Representation of Distribution D)**.** *Assume D is a distribution over $[N]$, satisfying $D(x) \leq \frac{1}{N^f}$ for some $f \in (0,1)$. The string representation of distribution D is parameterized by:*

- *Granularity parameter $\gamma \in (0,1)$.*
- *Embedding function $h : [N] \times \left[\lceil N^{1-f}/\gamma \rceil\right] \to [M]$, such that $M = 100\left\lceil \frac{N}{\gamma}\log\left(\frac{N}{\gamma}\right)\right\rceil$.*
- *Rounding function $h_\gamma : [N] \to \left[0, \frac{\gamma}{N}, \frac{2\gamma}{N}, \ldots, \gamma\right]$. Denote $h_\gamma(x) = \gamma_x$.*

The representation of D given the parameters above is the string $X_D \in \{0,1\}^M$ defined as so: for every $y \in [M]$, $X_D(y) = 1$ if there exist $(x,t) \in Supp(D) \times \left[\lceil N^{1-f}/\gamma \rceil\right]$ such that $h(x,t) = y$, and $t \cdot \frac{\gamma}{N} \le D(x) + \frac{\gamma_x}{N}$; otherwise $X_D(y) = 0$.

Concerning the functions h and h_γ that play a critical role in the construction above, we show that taking the function h to be randomly chosen from a $10\log(N/\gamma)$-wise independent hash family $\mathcal{H}$, and h_γ chosen from a pairwise independent hash family $\mathcal{H}_\gamma$, the construction above has the following desirable properties:

- Proposition 5.2 shows that the number of entries $y \in [M]$ for which $X_D(y) = 1$ is well concentrated, and can be approximated. This means that every uniformly drawn sample from $[M]$ has a known probability to *hit* a location $y \in [M]$ such that $X_D(y) = 1$. This property of the representation is used in the final protocol presented in Section 6. In a nutshell, in the final protocol, the verifier draws many (almost) uniform samples from $[M]$ and requires the prover to provide the value of X_D in these locations, since the verifier cannot know their values. Since the number of 1's in the string X_D is well concentrated, the prover cannot *lie* and claim that many locations y have value 0 when in fact $X_D(y) = 1$ without also reporting that other locations which have value 1 have value 0. Since our protocol relies on catching the prover when it mislabels entries has having value 0 instead of 1, this property of the representation is key.
- In Proposition 5.3 we argue that X_D can be used to reconstruct a distribution D' over $[N]$ that $\widetilde{O}(\gamma)$ close to D in total variation distance; and we argue that every string close to X_D can either be translated to a distribution close to D, or has a structure that's not characteristic of a string which is the product of Construction 5.1. This last feature is discussed in depth in Section 2.1, and is used to show that there exists a circuit C'_N over which the IPP can be carried.
- We also present in this section the Representation Sampler in Construction 5.4. This mechanism allows drawing locations in $y \in [M]$ in such a way that maintains a connection to distribution D (see Section 2.1 for a high level discussion of this mechanism). In the IPP protocol we employ, the verifier plants samples drawn through this sampler in way that's only weakly traceable by the prover. We show that if the prover lies about many entries as explained above, it will do so over entries drawn according to the sampler. We leverage this point in order to run the Verified Tagged Sample protocol as explain in the Technical Overview.

The proofs to all these propositions can be found in the subsequent sections.

Proposition 5.2. *Fix distribution D over domain $[N]$ such that for all $x \in [N]$, $D(x) \le N^{-f}$, and let X_D be its representation obtained through Construction 5.1 with functions h and h_γ drawn* *from a* $10\log(N/\gamma)$*-wise independent and pairwise independent hash families respectively. With probability at least* 0.98 *over the choice of* h *and* h_γ*:*

$$\left| M \cdot wt(X_D) - \left(\frac{N}{\gamma} - \sum_{k=2}^{\log\left(N/\gamma^{1/2}\right)} \binom{N/\gamma}{k} \left(\frac{1}{M}\right)^{k-1} \right) \right| \leq 500 \log^2(N/\gamma)\sqrt{N/\gamma} \tag{6}$$

Proposition 5.3. *Fix distribution* D *over domain* $[N]$ *such that for every* $x \in [N]$*,* $D(x) \leq N^{-f}$*, and parameter* $\gamma \in (0,1)$*. There exists an algorithm implementable by an* NC^1 *circuit that given* $X \in \{0,1\}^M$*, as well as functions* $h : [N] \times \left[\lceil N^{1-f}/\gamma \rceil\right] \to [M]$ *and* $h_\gamma : [N] \to \left\{0, \frac{\gamma}{N}, \frac{2\gamma}{N}, \ldots, \gamma\right\}$ *satisfies the following:*

- *If* $X = X_D$ *was produced from a distribution* D *using* h *and* h_γ *as described in Construction 5.1: then if the functions* h *and* h_γ *were drawn from a* $10\log(N/\gamma)$*-wise independent and pairwise independent hash families respectively, with probability at least* 0.99 *over the choice of* h *and* h_γ*, the algorithm doesn't reject, and outputs* $\left(D_x^X\right)_{x\in[N]}$ *such that:*

$$\sum_{x\in[N]} \left|D(x) - D_x^X\right| \leq \gamma \cdot \log N$$

- *For any* $\delta \in (0,1)$*, if* X *differs from* X_D *in at most* $\frac{N}{\gamma \log N} \cdot \delta$*, then either the algorithm rejects, or outputs* $\left(D_x^X\right)_{x\in[N]}$ *such that:*

$$\sum_{x\in[N]} \left|D(x) - D_x^X\right| \leq \delta$$

Construction 5.4 (Representation Sampler)**.** *We define following process sampling process:*

1. *First, sample* $(x,t) \in [N] \times \left[\lceil N^{1-f}/\gamma \rceil\right]$ *through distribution* Y^{pair} *defined as follows:*
   (a) *Flip a fair coin* b*. If* $b = 0$*, draw* $x \sim D$*; otherwise, draw* $x \sim U_N$*.*
   (b) *Draw* $t_{\max}$ *uniformly at random from the set* $\left\{\log N, 2\log N, \ldots 2^i \log N, \ldots, \frac{N^{1-f}}{\gamma}\right\}$*.*
   (c) *Draw* t *uniformly at random from the set* $\{1, 2, 3, \ldots, \lceil t_{\max} \rceil\}$*.*
2. *Set* $y = h(x,t)$*, and output* $((x,t), b, y)$*.*

Definition 5.5. *We define the following distributions implicit in the construction above:*

- *Let* $((x,t), b, y)$ *be a sample drawn according to Construction 5.1. Define* Y *to be the marginal distribution over* y *defined over domain* $[M]$*.*
- *Denote by* Y_D^{pair} *the distribution over* $[N] \times \left[\lceil N^{1-f}/\gamma \rceil\right]$ *defined to be* Y^{pair} *restricted to the event that the coin flipped* b *satisfies* $b = 0$*, i.e. that* x *was drawn from* D*.*
- *Denote by* Y_U^{pair} *the distribution over* $[N] \times \left[\lceil N^{1-f}/\gamma \rceil\right]$ *defined to be* Y^{pair} *restricted to the event that the coin flipped* b *satisfies* $b = 1$*, i.e. that* x *was drawn from* $U_{[N]}$*.*

In order to state the important properties of the sampler, we need to first introduce the following notion, which will also be used thoroughly in the process of proving all the propositions in this section:

Definition 5.6 (Trigger Pairs)**.** *Fix some function of* $(\gamma_x)_{x\in[N]}$. *We call a pair* $(x,t)\in Supp(D)\times\left[\left\lceil\frac{N^{1-f}}{\gamma}\right\rceil\right]$ a trigger pair *if* $\frac{\gamma}{N}\cdot t\le D(x)+\frac{\gamma_x}{N}$. *We define TRIG to be the set of all trigger pairs, and denote* $P=|TRIG|$

Proposition 5.7. *[Properties of Sampler 5.4] With probability at least* 0.99 *over the choice of* h*, the following holds:*

1. *For every* $y\in[M]$ *such that* $X_D(y)=y$*, if we denote* $trig_y=\{(x,t)\in TRIG: h(x,t)=y\}$*, then:*
$$\frac{\Pr_{Y_D^{pair}}\left(trig_y\right)}{\Pr_Y\left(y\right)}\ge\frac{\gamma}{1100\log^4\left(N/\gamma\right)}$$

2. *For every* $y\in[M]$*:*
$$\Pr_Y\left(y\right)\in\left[\frac{1/\left(8\log\left(N/\gamma\right)\right)}{M},\frac{1100\log\left(N/\gamma\right)/\gamma}{N^f}\right]$$

3. *Assume that the pair* (x,t) *drawn by* Y_D^{pair} *is a trigger pair. Then, for every* $\rho\in(0,1)$*, with probability at least* $(1-\rho)$*,* t *satisfies:*
$$t\cdot\frac{\gamma}{N}\le D(x)\left(1-\rho\right)$$

5.1 Proofs of the Propositions in Section 5

5.1.1 Proving Proposition 5.2

We prove this proposition in stages. We first estimate the amount of trigger pairs $(x,t)\in\text{Supp}(D)\times\left[N^{1-f}/\gamma\right]$, i.e. pairs that satisfy $t\cdot\frac{\gamma}{N}\le D(x)+\gamma_x$. Note that by construction, for every entry $y\in[M]$ such that $X_D(y)=1$, there exists such a pair (x,t) for which $h(x,t)=y$, and every such pair satisfies $X_D\left(h(x,t)\right)=y$. We first prove that taking h_γ from a pairwise uniform hash family, the number of trigger pairs is strongly concentrated around N/γ. Therefore, if h had no collisions between trigger pairs, we'd expect $\text{wt}(X_D)=|\text{TRIG}|\approx N/\gamma$, where TRIG is the set of all trigger pairs. However, since M is not considerably larger that TRIG, we expect such collisions to appear. And so, in order to bound $\text{wt}(X_D)$, we are left to estimate how many collisions are there, and get an approximation for $\left|\text{Im}\left(h|_{\text{TRIG}}\right)\right|$. Concretely, we show that if $h:[N]\times\left[N^{1-f}/\gamma\right]\to[M]$ was drawn from a $10\log\left(N/\gamma\right)$-wise independent hash family, with high probability $\text{wt}(X_D)=\left|\text{Im}\left(h|_{\text{TRIG}}\right)\right|$ satisfies Inequality (6).

Claim 5.8. *Assume* $\left(\gamma_x\right)_x$ *were chosen by drawn* h_γ *from a* 2*-wise independent hash family, then with probability at least* 0.999 *over the choice of* h_γ*, the number of trigger pairs* P*, satisfies* $P\in\frac{N}{\gamma}\pm110\sqrt{\frac{N}{\gamma}}$.

Proof. Denote by $\mathrm{trig}_x = \left|\left\{t \in \mathbb{N} : t \cdot \frac{\gamma}{N} \leq D(x) + \frac{\gamma_x}{N}\right\}\right|$, the random variable that counts the number of trigger pairs associated with support element x. Note that this variable depends only on choice of γ_x, and takes either the value $\left\lfloor \frac{D(x)}{\gamma/N} \right\rfloor$ or the value $\left\lfloor \frac{D(x)}{\gamma/N} \right\rfloor + 1$.

trig_x assumes the value $\left\lfloor \frac{D(x)}{\gamma/N} \right\rfloor$ if $\frac{\gamma}{N} \cdot \left(\left(\left\lfloor \frac{D(x)}{\gamma/N} \right\rfloor\right) + 1\right) - D(x) > \gamma_x$. If γ_x were chosen uniformly from $[0, \gamma]$ this probability would be $\frac{\frac{\gamma}{N} \cdot \left(\left(\left\lfloor \frac{D(x)}{\gamma/N} \right\rfloor\right)+1\right) - D(x)}{\gamma/N} = \left\lfloor \frac{D(x)}{\gamma/N} \right\rfloor + 1 - \frac{D(x)}{\gamma/N}$. Since γ_x is chosen from a γ/N discretization of $[0, \gamma]$, the true probability is $\left\lfloor \frac{D(x)}{\gamma/N} \right\rfloor + 1 - \frac{D(x)}{\gamma/N} - \delta_x$ where $\delta_x \in [0, \gamma/N]$. And so, for every $x \in \mathrm{Supp}(D)$:

$$\begin{aligned}
\mathbb{E}_{\gamma_x}[\mathrm{trig}_x] &= \left\lfloor \frac{D(x)}{\gamma/N} \right\rfloor \cdot \left(\left\lfloor \frac{D(x)}{\gamma/N} \right\rfloor + 1 - \frac{D(x)}{\gamma/N} - \delta_x\right) + \left(\left\lfloor \frac{D(x)}{\gamma/N} \right\rfloor + 1\right) \cdot \left(\frac{D(x)}{\gamma/N} - \left\lfloor \frac{D(x)}{\gamma/N} \right\rfloor + \delta_x\right) \\
&= \frac{D(x)}{\gamma/N} + \delta_x
\end{aligned}$$

And so, by the linearity of expectation, we conclude that:

$$\mathbb{E}\left[P\right] = \mathbb{E}_{(\gamma_x)}\left[\sum_x \mathrm{trig}_x\right] = \frac{N}{\gamma} + \sum_x \delta_x$$

Since $\delta_x \in [0, \gamma/N]$ we get that $\mathbb{E}\left[P\right] \in \left[\frac{N}{\gamma}, \frac{N}{\gamma} + \gamma\right]$. Next, we show that P is well concentrated around its mean. Note that P is a sum of 2-wise independent Bernoulli random variables $P = \sum_x \mathrm{trig}_x$. For every x, $\mathrm{Var}\left[\mathrm{trig}_x\right] \leq 2$, since the difference between the largest and smallest value every variable trig_x assumes is 1.

The choice of (γ_x) is pairwise independent, and so $\mathrm{Var}[P] = \mathrm{Var}\left[\sum_x \mathrm{trig}_x\right] \leq \sum_x \mathrm{Var}\left[\mathrm{trig}_x\right] \leq 2N \leq \mathbb{E}\left[P\right]$, and by Chebichev's Inequality:

$$\Pr_{h_\gamma}\left(\left|P - \mathbb{E}\left[P\right]\right| > 100\mathbb{E}\left[P\right]\right) \leq \frac{1}{10000}$$

We thus conclude that with probability at least 0.999 over the choice of h_γ:

$$P \in \frac{N}{\gamma} \pm 110\sqrt{\frac{N}{\gamma}}$$

□

Recall that TRIG is the set of all trigger pairs, and observe that $\mathrm{wt}\left(X_D\right) = \left|\mathrm{Im}\left(h\big|_{\mathrm{TRIG}}\right)\right|$. For hash function h as described in Construction 5.1, and define:

$$S^k = \left|\{S \subseteq \mathrm{TRIG} : |S| = k, \ \ \exists y \in [M] \ \forall (x,t) \in S \ \ h(x,t) = y\}\right|$$

.

Claim 5.9. $\left|Im\left(h\big|_{TRIG}\right)\right| = P - \sum_{k=2}^{P} \left(-1\right)^k S^k$

Proof. Fix some $k_0 \leq P$, and function h. Assume that there are no hash collisions of size larger than k_0. That is:

$$k_0 = \max\left\{k : \exists\left((x_i, t_i)\right)_{i\in[k]}, \text{ s.t. } \forall i \neq j \in [k],\ (x_i, t_i) \neq (x_j, t_j), h(x_i, t_i) = h(x_j, t_j)\right\}$$

If we show that for all $k_0 \leq P$ it holds that $\left|\mathrm{Im}\left(h|_{\mathrm{TRIG}}\right)\right| = P - \sum_{k=2}^{k_0} (-1)^k S^k$, we are done. We do so by induction. First, note that this holds for $k_0 = 2$, since if there are only 2-collisions, and no 3-collisions, every element in $y \in [M]$ such that $X_D(y) = 1$ has either one trigger pair mapped to it, or two. Therefore, the size of the image would be $P - S^2$.

Next, fix some $k' \in \{2, \ldots P-1\}$, and assume that any function $h : \mathrm{TRIG} \to [M]$ satisfies $S^{k'} > 0$ and $S^{k'+1} = 0$, then: $\left|\mathrm{Im}\left(h|_{\mathrm{TRIG}}\right)\right| = P - \sum_{k=2}^{k'} (-1)^k S^k$. Consider the case that some function h satisfies the condition that $S^{k'+1} > 0$ but $S^{k'+2} = 0$.

First, define the set $B \subseteq \mathrm{TRIG}$ to be the set of all trigger pairs that collide with at least k' other pairs:

$$B = \bigcup \{S \subseteq \mathrm{TRIG} : |S| = k+1,\ \exists y \in [M]\ \forall (x,t) \in S\ h(x,t) = y\}$$

Next, since $M \gg P = |\mathrm{TRIG}|$, consider the function h' obtained from h by assigning each element in B a unique image under h', and define $S_{h'}^k$ to be the random variable that counts the number of k-collisions in h'. By the induction assumption, h' has at most k' collisions, and so:

$$\left|\mathrm{Im}\left(h'|_{\mathrm{TRIG}}\right)\right| = P - \sum_{k=2}^{k'} (-1)^k S_{h'}^k$$

Observe that $|B| = S^{k'+1} \cdot (k'+1)$. This is true since by definition it holds that $|B| \leq S^{k'+1} \cdot (k'+1)$, however, if $|B| < S^{k'+1} \cdot (k'+1)$ it would imply that $S^{k'+2} > 0$, against the assumption. Therefore, we need to consider every $(k'+1)$-collision under h, as a single element in the image, and so:

$$\left|\mathrm{Im}\left(h'|_{\mathrm{TRIG}}\right)\right| - \left|\mathrm{Im}\left(h|_{\mathrm{TRIG}}\right)\right| = S^{k'+1} \cdot k'$$

Next, observe that for every $k \le k'$, it holds that $S^k = S^k_{h'} + S^{k'+1} \cdot \binom{k'+1}{k}$. Therefore:

$$\begin{aligned}
\left|\mathrm{Im}\left(h|_{\mathrm{TRIG}}\right)\right| &= \left|\mathrm{Im}\left(h'|_{\mathrm{TRIG}}\right)\right| - S^{k'+1} \cdot k' \\
&= P - \sum_{k=2}^{k'} (-1)^k S^k_{h'} - S^{k'+1} \cdot k' \\
&= P - \sum_{k=2}^{k'} (-1)^k \left(S^k - S^{k'+1} \cdot \binom{k'+1}{k}\right) - S^{k'+1} \cdot k' \\
&= P - \sum_{k=2}^{k'} (-1)^k S^k - S^{k'+1} \sum_{k=2}^{k'} (-1)^k \binom{k'+1}{k} - S^{k'+1} \cdot k' \\
&= P - \sum_{k=2}^{k'} (-1)^k S^k - S^{k'+1} \left(\sum_{k=0}^{k'+1} (-1)^k \binom{k'+1}{k} - \left(1 - (k'+1) + (-1)^{k'+1}\right) + k'\right) \\
&= P - \sum_{k=2}^{k'} (-1)^k S^k - (-1)^{k'+1} S^{k'+1} \\
&= P - \sum_{k=2}^{k'+1} (-1)^k S^k
\end{aligned}$$

□

We thus approximate with high probability the random variables $\left(S^k\right)$. We divide this collection into two, and show that: S^k such that $k \le k_0 = \left\lceil \log\left(N/\gamma^{1/2}\right)\right\rceil + 1$, are well concentrated around their mean; while the sum of all S^k for which $k > k_0$ is with high probability very small.

Claim 5.10. *For $k_0 = \left\lceil \log\left(N/\gamma^{1/2}\right)\right\rceil + 1$, :*

- *For every $k \le k_0$, $\mathbb{E}\left[S^k\right] = \binom{P}{k}\left(\frac{1}{M}\right)^{k-1}$, and with probability at least 0.99 over the choice of h, for all $k \le k_0$: $\left|S^k - \mathbb{E}\left[S^k\right]\right| < \sqrt{\mathbb{E}\left[S^k\right]\left(400 \log\left(N/\gamma\right)\right)}$*
- $\sum_{k=k_0+1}^{P} \mathbb{E}\left[S^k\right] \le 1$.

Proof. For every $S \subseteq \mathrm{TRIG}$ denote by C_S the Bernoulli random variable that indicates that all elements in S were hashed to the same $y \in [M]$. By definition:

$$S^k = \sum_{S:|S|=k} C_S$$

For every $k \le T/2$, and for every C_S such that $|S| = k$, it holds that $\mathbb{E}\left[C_S\right] = \left(\frac{1}{M}\right)^{k-1}$, and so $\mathbb{E}\left[S^k\right] = \binom{P}{k}\left(\frac{1}{M}\right)^{k-1}$.

Since $k_0 \le T/2$ where h was drawn from a $T = 10 \log\left(N/\gamma\right)$-independent hash family, and $M \ge 16P$, we get that by choice of k_0, $\mathbb{E}\left[S^{k_0}\right] \le P \cdot 2^{4(k_0-1)} \le \left(\frac{\gamma}{N^2}\right)^4$. Since by definition the sequence S^k is monotonically decreasing, for every $k \ge k_0$, $\mathbb{E}\left[S^k\right] \le E\left[S^{k_0}\right]$. Therefore, by linearity of expectation, $\mathbb{E}\left[\sum_{k'=k_0}^{P} S^k\right] \le P \cdot \left(\frac{\gamma}{N^2}\right)^4 \le 1$, where the last inequality holds since $P = O\left(N/\gamma\right)$.

Next, in order to approximate S^k with high probability for $k \leq k_0$, we show that for all such k, $\mathrm{Var}\left[S^k\right] = O\left(\mathbb{E}\left[S^k\right]\right)$, and then use Chebichev's Inequality to argue that S^k is concentrated around its mean.

Fix such $k \in \{2, 3, \ldots, k_0\}$. Note that:

$$\mathrm{Var}[S^k] = \sum_{\substack{S_0, S_1 \subseteq \mathrm{TRIG}: \\ |S_0| = |S_1| = k}} \mathrm{Cov}\left[C_{S_0}, C_{S_1}\right]$$

Consider $\mathrm{Cov}\left[C_{S_0}, C_{S_1}\right]$ for some two sets S_0, S_1 of size k. Denote $|S_0 \cap S_1| = k' \leq k$. Note that the Bernoulli random variable $C_{S_0} \cdot C_{S_1}$ satisfies $\mathbb{E}\left[C_{S_0} \cdot C_{S_1}\right] = \left(\frac{1}{M}\right)^{2k-k'-1}$, and so $\mathrm{Cov}\left[C_{S_0}, C_{S_1}\right] \leq \left(\frac{1}{M}\right)^{2k-k'-1}$.

For every $k' \leq k$, there are at most $\binom{P}{k} \cdot \binom{k}{k-k'} \cdot \binom{P-k}{k-k'}$ pairs of sets S_0, S_1 of size k that intersect on k' elements. Therefore:

$$\begin{aligned}
\mathrm{Var}\left[S^k\right] &= \sum_{\substack{S_0, S_1: \\ |S_0| = |S_1| = k}} \mathrm{Cov}\left[C_{S_0}, C_{S_1}\right] && (7)\\
&= \sum_{k'=0}^{k} \sum_{\substack{S_0, S_1: \\ |S_0| = |S_1| = k \\ |S_0 \cap S_1| = k'}} \mathrm{Cov}\left[C_{S_0}, C_{S_1}\right] && (8)\\
&\leq \sum_{k'=0}^{k} \binom{P}{k} \cdot \binom{k}{k-k'} \cdot \binom{P-k}{k-k'} \left(\frac{1}{M}\right)^{2k-k'-1} && (9)\\
&\leq \binom{P}{k} \left(\frac{1}{M}\right)^{k-1} \sum_{k'=0}^{k} \binom{k}{k-k'} \cdot \binom{P-k}{k-k'} \left(\frac{1}{M}\right)^{k-k'} && (10)\\
&\leq \binom{P}{k} \left(\frac{1}{M}\right)^{k-1} \sum_{k'=0}^{k} \binom{k}{k-k'} P^{k-k'} \left(\frac{1}{M}\right)^{k-k'} && (11)\\
&\leq \binom{P}{k} \left(\frac{1}{M}\right)^{k-1} \cdot P^k \cdot \sum_{k'=0}^{k} \binom{k}{k-k'} \left(\frac{1}{P}\right)^{k'} \left(\frac{1}{M}\right)^{k-k'} && (12)\\
&\leq \binom{P}{k} \left(\frac{1}{M}\right)^{k-1} \cdot P^k \cdot \left(\frac{1}{P} + \frac{1}{M}\right)^{k} && (13)\\
&\leq \binom{P}{k} \left(\frac{1}{M}\right)^{k-1} \cdot \left(1 + \frac{P}{M}\right)^{k} && (14)\\
&\leq 2\binom{P}{k} \left(\frac{1}{M}\right)^{k-1} \cdot e^{\frac{P}{M} \cdot k} && (15)\\
& && (16)
\end{aligned}$$

For every $k \leq k_0$, $\mathbb{E}\left[S^k\right] = \binom{P}{k}\left(\frac{1}{M}\right)^{k-1}$, and since $k \leq k_0 \leq \left\lceil \log\left(N/\gamma^{1/2}\right) \right\rceil \leq \frac{M}{P}$, we get $e^{\frac{P}{M} \cdot k} \leq 3$. Therefore we get that for every such k:

$$\mathrm{Var}\left[S^k\right] \leq 3\mathbb{E}\left[S^k\right] \cdot$$

By Chebichev's Inequality, for every $k \leq k_0$:

$$\Pr\left(\left|S^k - \mathbb{E}\left[S^k\right]\right| \geq \sqrt{\mathbb{E}\left[S^k\right](100T)/3}\right) \leq \frac{1}{100T} \tag{17}$$

Recall that $k_0 \leq T/2$, thus, taking the union bound over all $k \leq k_0$, with probability at least 0.99, it holds that for all for all $k \leq k_0$:

$$\left|S^k - \mathbb{E}\left[S^k\right]\right| < \sqrt{\mathbb{E}\left[S^k\right](100T)/3} \tag{18}$$

Plugging the value of T yields the desired result. □

Claim 5.11. *Let k_0 be as in Claim 5.10, then:*

- $\mathbb{E}[wt(X_D)] = P - \sum_{k=2}^{k=k_0} (-1)^k \binom{P}{k} \cdot \left(\frac{1}{M}\right)^{k-1} \pm 1$
- *With probability at least 0.98 over the choice of h:*

$$|wt(X_D) - \mathbb{E}[wt(X_D)]| \leq 170 \log^2(N/\gamma)\sqrt{P}$$

Proof. By the inclusion-exclusion principle, $\mathrm{wt}(X_D) = P - \sum_{k=2}^{P} (-1)^k S^k$. By Claim 5.10 and the linearity of expectation:

$$\mathbb{E}[X_D] = P - \sum_{k=2}^{P} (-1)^k \mathbb{E}\left[S^k\right] = P - \sum_{k=2}^{k=k_0} (-1)^k \binom{P}{k} \cdot \left(\frac{1}{M}\right)^{k-1} \pm 1$$

By the linearity of expectation as well as the triangle inequality we get:

$$\left|\sum_{k=2}^{k_0} (-1)^k S^k - \mathbb{E}\left[\sum_{k=2}^{k_0} (-1)^k S^k\right]\right| \leq \sum_{k=2}^{k_0} \left|(-1)^k \left(S^k - \mathbb{E}\left[S^k\right]\right)\right| \leq \sum_{k=2}^{k_0} \left|\left(S^k - \mathbb{E}\left[S^k\right]\right)\right| \tag{19}$$

By Claim 5.10 with probability at least 0.99, for all $k \leq k_0$, $\left|S^k - \mathbb{E}\left[S^k\right]\right| < \sqrt{\mathbb{E}\left[S^k\right](100T)/3}$. Assuming this holds, we get that:

$$\begin{aligned}\left|\sum_{k=2}^{k_0} (-1)^k S^k - \mathbb{E}\left[\sum_{k=2}^{k_0} (-1)^k S^k\right]\right| &\leq \sum_{k=2}^{k_0} \sqrt{\mathbb{E}\left[S^k\right](100T)/3} && (20)\\ &\leq \sqrt{64T \sum_{k=2}^{k_0} \mathbb{E}\left[S^k\right] \cdot \sqrt{k_0}} && (21)\\ &\leq 16\sqrt{T\log(N/\gamma)}\sqrt{\sum_{k=2}^{k_0} \mathbb{E}\left[S^k\right]} && (22)\end{aligned}$$

Where the second inequality is due to the Cauchy Schwarz Inequality. With probability at least 0.99 by Markov's Inequality and Claim 5.10:

$$\sum_{k'=k_0}^{P} S^k \leq 100\mathbb{E}\left[\sum_{k'=k_0}^{P} S^k\right] \leq 100 \tag{23}$$

Thus, assuming further that $\sum_{k=k_0}^{P} S^k \leq 100$, we get that:

$$|\mathrm{wt}\left(X_D\right) - \mathbb{E}\left[\mathrm{wt}\left(X_D\right)\right]| \leq \sum_{k=2}^{P} \left|S^k - \mathbb{E}\left[S^k\right]\right| \tag{24}$$

$$\leq 16\sqrt{T \log\left(N/\gamma\right)}\sqrt{\sum_{k=2}^{k_0} \mathbb{E}\left[S^k\right] + 100} \tag{25}$$

$$\leq 16\sqrt{T \log\left(N/\gamma\right)}\sqrt{k_0 \mathbb{E}\left[S^2\right] + 100} \tag{26}$$

$$\tag{27}$$

Where the last inequality is justified by the fact that $\mathbb{E}\left[S^k\right]$ is monotonically decreasing. Recall that $\mathbb{E}\left[S^2\right] = \binom{P}{2}\frac{1}{M} \leq P \cdot \frac{P}{M}$, and also $\frac{P}{M} \cdot T \cdot k_0 \leq 100 \log^3\left(N/\gamma\right)$. Plugging this in the inequality above (as well as assuming that $\sqrt{P} \geq 100$), we get:

$$|\mathrm{wt}\left(X_D\right) - \mathbb{E}\left[\mathrm{wt}\left(X_D\right)\right]| \leq 170 \log^2\left(N/\gamma\right)\sqrt{P} \tag{28}$$

By Claim 5.10, we get that with probability at least 0.98 the above inequality holds as required. □

We are now set to prove Proposition 5.2:

Proof of Proposition 5.2. By Claim 5.11, with probability at least 0.98 over the choice of h:

$$|\mathrm{wt}\left(X_D\right) - \mathbb{E}\left[\mathrm{wt}\left(X_D\right)\right]| \leq 170 \log^2\left(N/\gamma\right)\sqrt{P}$$

And from Claim 5.8, and since $110\sqrt{\frac{N}{\gamma}} \leq \frac{N}{\gamma}$ (which happens since we assume $\frac{N}{\gamma} \geq 110^2$):

$$|\mathrm{wt}\left(X_D\right) - \mathbb{E}\left[\mathrm{wt}\left(X_D\right)\right]| \leq 170 \log^2\left(N/\gamma\right) \cdot \sqrt{\frac{N}{\gamma} + 110\sqrt{\frac{N}{\gamma}}} \tag{29}$$

$$\leq 250 \log^2\left(N/\gamma\right) \cdot \sqrt{\frac{N}{\gamma}} \tag{30}$$

Denote denote $d = 110\sqrt{N/\gamma}$. Observe that for every $k \leq P/2$, :

$$\begin{align}
\mathbb{E}\left[\binom{P}{k}\right] &\leq \binom{\frac{N}{\gamma}+d}{k} \tag{31}\\
&= \binom{N/\gamma}{k}\prod_{i=1}^{d}\left(\frac{N/\gamma+i}{N/\gamma-k+i}\right) \tag{32}\\
&\leq \binom{N/\gamma}{k}\prod_{i=1}^{d}\left(\frac{N/\gamma+i}{N/\gamma-k+i}\right) \tag{33}\\
&= \binom{N/\gamma}{k}\prod_{i=1}^{d}\left(1+\frac{k}{N/\gamma-k+i}\right) \tag{34}\\
&\leq \binom{N/\gamma}{k}\left(1+\frac{k}{N/\gamma-k}\right)^{d} \tag{35}\\
&\leq \binom{N/\gamma}{k}\left(1+\frac{2kd}{N/\gamma-k}\right) \tag{36}\\
& \tag{37}
\end{align}$$

Where the last inequality is justified by the fact that $\frac{kd}{N/\gamma-k} \ll 1$, and so, $\left(1+\frac{k}{N/\gamma-k}\right)^{d} \leq e^{d\cdot\frac{2k}{N/\gamma-k}} \leq 1+\frac{2kd}{N/\gamma-k}$.

Similarly, we also get that:

$$\begin{align}
\mathbb{E}\left[\binom{P}{k}\right] &\geq \binom{\frac{N}{\gamma}-d}{k} \tag{38}\\
&= \binom{N/\gamma}{k}\prod_{i=1}^{d}\left(\frac{N/\gamma-k-(i-1)}{N/\gamma-(i-1)}\right) \tag{39}\\
&= \binom{N/\gamma}{k}\prod_{i=1}^{d}\left(1-\frac{k}{N/\gamma-(i-1)}\right) \tag{40}\\
&\geq \binom{N/\gamma}{k}\prod_{i=1}^{d}\left(1-\frac{k}{N/\gamma}\right) \tag{41}\\
&\geq \binom{N/\gamma}{k}\left(1-\frac{2kd}{N/\gamma}\right) \tag{42}
\end{align}$$

And so, we conclude that $\left|\mathbb{E}[\binom{P}{k}]-\binom{N/\gamma}{k}\right| \leq \binom{N/\gamma}{k}\cdot\frac{2kd}{N/\gamma-k} \leq \binom{N/\gamma}{k}\cdot\frac{4kd}{N/\gamma}$. And so:

$$\begin{align}
\left|\mathbb{E}[\mathrm{wt}\left(X_D\right)]-\left(\frac{N}{\gamma}-\sum_{k=2}^{k_0}\binom{N/\gamma}{k}\left(\frac{1}{M}\right)^{k-1}\right)\right| &\leq \left|\mathbb{E}[P]-\frac{N}{\gamma}\right|+\sum_{k=2}^{k_0}\binom{N/\gamma}{k}\cdot\frac{4kd}{N/\gamma}\cdot\left(\frac{1}{M}\right)^{k-1} \tag{43}\\
&\leq d+\frac{4k_0 d}{N/\gamma}\cdot\frac{1}{M} \leq 2d \tag{44}
\end{align}$$

Where the last inequality is justified since $k_0 \cdot = O(\sqrt{N/\gamma}\log^2(N/\gamma)) \ll M \cdot N/\gamma$, we conclude that:

$$\left|\mathbb{E}[\mathrm{wt}\,(X_D)] - \left(\frac{N}{\gamma} - \sum_{k=2}^{k_0}\binom{N/\gamma}{k}\left(\frac{1}{M}\right)^{k-1}\right)\right| \leq 2d = 220\sqrt{N/\gamma}$$

And so, by Inequality (29), with probability at least 0.98 over the choice of h:

$$\left|\mathrm{wt}\,(X_D) - \left(\frac{N}{\gamma} - \sum_{k=2}^{k_0}\binom{N/\gamma}{k}\left(\frac{1}{M}\right)^{k-1}\right)\right| \leq 220\sqrt{N/\gamma} + 250\log^2(N/\gamma)\cdot\sqrt{\frac{N}{\gamma}} \tag{45}$$

$$\leq 500\log^2(N/\gamma)\sqrt{N/\gamma} \tag{46}$$

□

5.1.2 Proof of Proposition 5.3

Algorithm 5.11.1: Distribution Reconstruction Algorithm

Input: string $X \in \{0,1\}^M$, parameter $\gamma \in (0,1)$, as well as hash function h, h_γ.
Output: if didn't reject, outputs $(D_x^X)_{x\in[N]}$.

1. For every $x \in [N]$, compute $t_x = \max\{t : X(h(x,t)) = 1\}$, by querying X in locations $h(x,1), h(x,2), \dots$, until reaching t such that $X(h(x,t)) = 0$ or $t = \lceil N^{1-f}/\gamma \rceil$, then set $t_x = t-1$, and $D_x^X = \frac{\gamma}{N}\cdot t_x - \frac{\gamma_x}{N}$. For every $y \in [M]$ keep a counter c_y that counts how many pairs (x,t) satisfy $h(x,t) = y$ and $t \leq t_x$. If there exists c_y such that $c_y \geq \log N$, reject.
2. For every $x \in [N]$, check that the set of locations $\{h(x,t) : t \in \{t_x+1, t_x+2, \dots, N^{1-f}/\gamma\}\}$ does not contain $\log N$ consecutive 1's: i.e. for every $i_0 \in \{t_x+1, \dots, N^{1-f}/\gamma - \log N\}$ and $V_{t_x+i_0} = \{(x, t_x+i) : i \in \{1,2,\dots\log N\}\}$, check that there exists $(x',t') \in V_{t_x+i_0}$ such that $y = h(x',t')$ and $X(y) = 0$. Reject otherwise.
3. Output $(D_x^X)_{x\in[N]}$

This section is devoted to showing that Algorithm 5.11.1 satisfies the conditions of Proposition 5.3.

First, we show the following structural claim about the representation X_D obtained through Construction 5.1:

Claim 5.12. *For every D, with probability at least 0.999 over the choice of h drawn from a $10\log(N/\gamma)$-wise independent hash family, if X_D is the representation of D achieved through Construction 5.1 then: for every $x \in [N]$ and $t_0 \in [N^{1-f}/\gamma - \log N]$ there doesn't exist a tuple of trigger pairs $((x_j,t_j))_{j\in[\log N]}$ such that for all j, $x_j \neq x$ and:*

$$\forall j \in [\log N] \quad h(x, t_0 + j) = h(x_j, t_j) \tag{47}$$

Proof. Fix $x \in [N]$ and $t_0 \in [N^{1-f}/\gamma]$, and let K be some integer such that $K \leq 10\log(N/\gamma)$. For every tuple of trigger pairs $((x_j,t_j))_{j\in[K]}$ such that $x_j \neq x$ for all j, since h was drawn from a T-independent hash family, the probability that:

$$\forall j \in [K] \quad h(x_j,t_j) = h(x, t_x + i_0 + j) \tag{48}$$

Is at most $\left(\frac{1}{M}\right)^K$. The total number of different tuples of size K of trigger pairs is at most:

$$\binom{P}{K} \cdot K! \leq (P \cdot K)^K$$

Therefore, by the union bound, the probability that there exists one such tuple that satisfies Equations (48) is at most:

$$(P \cdot K)^K \cdot \left(\frac{1}{M}\right)^K = \left(\frac{P}{M} \cdot K\right)^K$$

Applying the union bound once more over all the possible values of t_0, for a fixed $x \in [N]$, the probability that exists some t_0 and tuple $\left((x_j, t_j)\right)_j$ for which Equation (23) holds, is at most:

$$\frac{N^{1-f}}{\gamma} \left(\frac{P}{M} \cdot K\right)^K \tag{49}$$

Recall that $\frac{P}{M} \leq \frac{1}{20 \log(N/\gamma)}$, and so, setting $K = \log N$ we get that $\left(\frac{P}{M} \cdot K\right)^K \leq \left(\frac{1}{20}\right)^{\log N} \leq \frac{\gamma}{N^4}$. Finally, taking the union bound over all x yields the desired result. □

We now show that Algorithm 5.11.1 satisfies the *first condition* of Proposition 5.3

Claim 5.13. *With probability at least* 0.99 *over the choice of* h, *it holds that given input* X_D *produced from* D *through Construction 5.1, with probability at least* 0.99 *over the choice of* h, *Algorithm 5.11.1 doesn't reject* X_D, *and outputs for every* x *the value* $D_x^{X_D}$ *that satisfies:*

$$\left|D_x^{X_D} - D(x)\right| \leq \frac{\gamma}{N} \cdot \log N$$

Proof. Denote:

$$t_x^{\text{true}} = \max\left\{t \in \left[N^{1-f}/\gamma\right] : t \cdot \frac{\gamma}{N} \leq D(x) + \frac{\gamma_x}{N}\right\}$$

By definition of X_D, it holds that t_x, as defined in Algorithm 5.11.1, satisfies $t_x \geq t_x^{\text{true}}$. We show that with high probability, for all x, $t_x \leq t_x^{\text{true}} + \log N$. By Claim 5.12, with probability at least 0.99 over the choice of h, there doesn't exist a tuple of trigger pairs $((x_j, t_j))$ such that:

$$\forall i \in [\log N], \quad h\left(x_i, t_i\right) = h\left(x, t_x^{\text{true}} + i\right) \tag{50}$$

And so, with probability at least 0.99 over the choice of h, for every x, and every $i_0 \in \left\{0, N^{1-f}/\gamma - \log N\right\}$ satisfies that the locations $\left(h\left(x, t_x^{\text{true}} + i_0 + j\right)\right)_{j \in [\log N]}$ contain some location y such that $X_D(y) = 0$, and by the construction of Algorithm 5.11.1 it holds that $\left|D(x) - D_x^{X_D}\right| \leq \log N \frac{\gamma}{N}$.

Moreover, with high probability over the choice of h, X_D isn't rejected by the algorithm. First, note that since the function h was drawn from a $10 \log(N/\gamma)$-independent family, the probability that there exist $\log N$ trigger pairs that are hashed to the same $y \in [M]$ is at most:

$$\binom{P}{\log N} \left(\frac{1}{M}\right)^{\log N - 1} \leq P \cdot \left(\frac{P}{M}\right)^{\log N - 1} \leq P \cdot \left(\frac{1}{\log(N/\gamma)}\right)^{\log N - 1} \leq \frac{1}{N^2}$$

From which we conclude that X_D isn't rejected in Step (1) of the algorithm. Next, by Claim 5.12 with high probability over the choice of h, also Step (2) doesn't result in rejection. □

Next, we prove that Algorithm 5.11.1 satisfies the second condition of Proposition 5.3

Lemma 5.14 (Distance Translation)**.** *Fix a distribution D, and $\delta \in (0,1)$. With high probability over the choice of h, any string X_0 that differs from X_D in at most $\frac{N}{\gamma \log^2 N} \cdot \delta$ locations, is either rejected by Algorithm 5.11.1, or the reconstructed output $\left((x, D_x^{X_0})\right)_{x\in[N]}$ satisfies:*

$$\sum_{x\in[N]} \left|D_x^{X_0} - D(x)\right| \leq \delta$$

Proof. Fix $X_0 \in \{0,1\}^M$ such that:

- X_0 isn't rejected by Algorithm 5.11.1.
- The output $D_x^{X_0}$ satisfies:
$$\sum_{x\in[N]} \left|D_x^{X_0} - D(x)\right| \geq \delta$$

Consider some $x \in [N]$. Denote $D_x^{X_0} - D(x) = \delta_x$, and define $\delta_x^+ = \max\{0, \delta_x\}$, and $\delta_x^- = -\min\{0, \delta_x\}$. Consider all x such that $D_x^{X_0} - D(x) = \delta_x^+ > 0$. Set $t_x = \max\left\{t : t \cdot \frac{\gamma}{N} \leq D(x) + \frac{\gamma_x}{N}\right\}$. By assumption, in X_0, the locations $y_i = h(x, t_0 + i)$ for $i \in \left\{1, 2, \ldots, \frac{\delta_x}{(\gamma/N)}\right\}$ satisfy $X_0(y_i) = 1$. However, in X_D, by Claim 5.12, with probability at least 0.999 over the choice of h, for every $x \in [N]$ and $t_0 > t_x$, the tuple $((x, t_0 + j))_{j\in[\log N]}$ contains entry $(x, t_0 + j_0)$ such that $X_D\left(h\left(x, t_0 + j_0\right)\right) = 0$. Therefore, for every $x \in [N]$ there are in X_D at least $\frac{\delta_x^+}{\gamma/N} \cdot \frac{1}{\log N}$ locations $y \in [M]$ such that $X_D(y) = 0$ and $X_0(y) = 1$. In total, denote the set of pairs (x,t) for which $X_D(h(x,t)) = 0$ and $X_0(h(x,t)) = 1$ by L, and by the above, we know that $|L| \geq \sum_x \frac{\delta_x^+}{\gamma/N} \cdot \frac{1}{\log N}$. Assuming that X_0 wasn't rejected by the circuit we know that every $y \in [M]$ such that $X_0(y) = 1$ has at most $\log N$ trigger pairs associated with it, which implies that $\left|\mathrm{Im}\left(h\big|_L\right)\right| \geq \sum_x \frac{\delta_x^+}{\gamma/N} \cdot \frac{1}{\log^2 N}$. We thus conclude that X_D and X_0 differ in at most $\frac{\sum_x \delta_x^+}{\gamma/N} \cdot \frac{1}{\log^2 N}$ locations.

Consider next $x \in [N]$ such that X_0 satisfies $D(x) - D_x^{X_0} \geq \delta_x > 0$, and denote t_{x,X_0}^{true} to be such that $D_x^{X_0} = \frac{\gamma}{N} \cdot t_{x,X_0}^{\text{true}}$. By assumption, X_0 wasn't rejected by the algorithm, therefore, for every $x \in [N]$, and any set of locations $\left(h\left(x, t_{x,X_0}^{\text{true}} + j\right)\right)_{j\in[\log N]}$ there exists a location y such that $X_0(y) = 0$, therefore, following the same line of argument as above, X_0 and X_D differ on at least $\sum_x \frac{\delta_x^-}{\gamma/N} \cdot \frac{1}{\log N}$ pairs. Since with high probability X_D doesn't fail the reconstruction algorithm, this implies, as above that the number of entries $y \in [M]$ for which they differ is at least $\sum_x \frac{\delta_x^-}{\gamma/N} \cdot \frac{1}{\log^2 N}$

Putting everything together, we get that:

$$M \cdot \mathrm{Ham}\left(X_0, X_D\right) \geq \sum_x \frac{\delta_x^+}{\gamma/N} \cdot \frac{1}{\log^2 N} + \sum_x \frac{\delta_x^-}{\gamma/N} \cdot \frac{1}{\log^2 N} = \delta \cdot \frac{N}{\gamma} \cdot \frac{1}{\log^2 N}$$

Finally, consider the counter-positive with high probability over the choice of h, if $M \cdot \mathrm{Ham}\left(X_0, X_D\right) \leq \frac{N}{\gamma \log^2 N} \cdot \delta$, then $\sum_x \left|D_x^{X_0} - D(x)\right| \leq \delta$. □

Finally, we argue that Algorithm 5.11.1 can indeed be implementable by a NC^1 circuit.

Claim 5.15. *Algorithm 5.11.1 can be implemented by a logspace uniform NC^1 circuit family.*

Proof. Observe that for every $x \in [N]$ computing t_x can be implemented as follows: for every $t \in [N^{1-f}/\gamma]$ add a gate G_t^x that checks for all $i < t$ $X_D(h(x,i)) = 1$ *AND* $X_D(h(x,t)) = 0$, then, have for every x a selection gate that selects the only gate G_t^x with value 1, and then computes the binary representation of t for the output. These are all AC^0 gadgets.

Next, we show that both checks can implemented by NC^1 circuits:

- In order to check $c_y \leq \log N$ for all $y \in [M]$ we do the following: for every $i \in [M]$, and for every $(x,t) \in [N] \times [N^{1-f}/\gamma]$, such that $h(x,t) = i$, define $T^i_{x,t,i}$ that to be the gate that checks $t \leq t_x$ (computed as explained above). Then, for every i, have a gate C_i that counts how many $T^i_{x,t}$ have value 1, and assumes the value 0 if there are more than $\log N$ such gates (counting and comparing can be done in NC^1). Then, the circuit outputs 0 if $\mathtt{AND}_{i\in[M]} C_i = 0$.
- In order to implement the check in Step (2): for every $x \in [N]$ and every $t \in \left[\left(N^{1-f}/\gamma\right) - \log N\right]$, define a gate $S_{x,t}$ if $t < t_x$, then $S_{x,t}$ assumes the value 1, otherwise, it checks that for $\mathtt{NOTAND}_{i\in[\log N]} h(x, t+i) = 1$. Finally, taking an $\mathtt{AND}$ gate over all $S_{x,t}$ yields the check in Step (2). Note again that all these are AC^0 gadgets and so can be implemented by an NC^1 gadget.

□

5.1.3 Proof of Proposition 5.7

This section is devoted for proving Proposition 5.7:

- Claim 5.16 and Claim 5.17 provide bounds on the probability that $y \in [M]$ such that $X_D(y) = 1$ was drawn by Y_D^{pair} by either a trigger pair, or *not* through a trigger, respectively. In combination with Claim 5.19 that gives an upper bound on the probability of y according to Y_U^{pair}, we prove the first condition of the proposition.
- Next, in order to bound the probability of every $y \in [M]$ according to Y, on top of the above-mentioned claims, we also show that for every y, we can bound from above the probability it was reached through Y_D^{pair} (Claim 5.18), and lower bound the probability it was reached through Y_U^{pair} (Claim 5.20). And so, we provide both an upper and lower bound for $\Pr_Y(y)$.
- Lastly, Claim 5.22 proves the third condition of Proposition 5.7.

The claims are proven individually, and are put together to prove the Proposition in Proof 5.1.3.

Claim 5.16. *For any trigger pair (x,t) it holds that:*

$$\Pr_{Y_D^{pair}}((x,t)) \geq \frac{1}{M \log N}$$

Proof. Fix trigger pair (x,t). The first coordinate x was drawn from D with probability $D(x)$. Since $t_{\max}$ is sampled from the set $\left\{\log N, \ldots, 2^i \log N, \ldots, N^{1-f}/\gamma\right\}$, then with probability at least $\frac{1}{\log(N/\gamma)}$, $t_{\max}$ was sampled such that: $D(x) \leq t_{\max} \cdot \frac{\gamma}{N} \leq D(x) \cdot \log N$, and $D(x_y) \geq \frac{\gamma}{N}(t_{\max} - 1)$.

Given that such $t_{\max}$ drawn, then, the probability that t was selected is at least $\left(\frac{D(x)\cdot\log N}{\gamma/N}\right)^{-1}$. Putting it all together, the probability that trigger pair (x, y) was drawn is at least:

$$D(x)\cdot\frac{1}{\log(N/\gamma)}\cdot\frac{\gamma/N}{D(x)\cdot\log N}=\frac{100}{100(N/\gamma)\log(N/\gamma)}\cdot\frac{1}{\log N}\geq\frac{1}{M\log N}$$

□

Claim 5.17. *With probability at least* 0.999 *over the choice of h, for every $y\in[M]$, if $X_D(y)=1$, it holds that the probability that a pair (x,t) was sampled by Y_D^{pair} such that $h(x,t)=y$ and (x,t)* isn't *a trigger pair is at most:*

$$\frac{1100\log^4(N/\gamma)}{M}$$

Proof. Consider the following division of all non-trigger tuples (x,t):

$$B_{i,j}=\left\{(x,t):D(x)\in\left[2^{i-1}\frac{\gamma}{N},2^i\frac{\gamma}{N}\right),t\in\left[2^{j-1},2^j\right),t>\frac{D(x)+\gamma_x}{\gamma/N}\right\}$$

Note that every non-trigger pair is contained in some set $B_{i,j}$, and that these sets contain *only* non-trigger pairs.

Observe that by definition of $B_{i,j}$ and Y_D^{pair}, for every $(x_0,t_0),(x_1,t_1)\in B_{i,j}$ it holds that $\frac{\Pr(Y=(x_0,t_0))}{\Pr(Y=(x_1,t_1))}\in[0.5,2]$. Moreover, by definition if $(x,t)\in B_{i,j}$, then: the probability that (x,t) was drawn by Y_D^{pair} is at most $D(x)\cdot\frac{1}{t}\leq\frac{\gamma}{N}$ due to the fact (x,t) is a non-trigger pair, and so $t\cdot\frac{\gamma}{N}>D(x)$. We thus conclude that the probability of each element in $B_{i,j}$ according to Y is at most $\min\left\{\frac{4}{|B_{i,j}|}\frac{\gamma}{N}\right\}$.

First, fix some i,j such that $|B_{i,j}|\leq M$, then, since the function h is $(10\log(N/\gamma))$-wise independent, we get that with probability at most $\frac{1}{100M}$ there exists $y\in[M]$ such that more than $10\log(N/\gamma)$ elements from $B_{i,j}$ were hashed to y, and so, the probability that y was sampled by Y through $(x,t)\in B_{i,j}$ is at most:

$$\frac{\gamma}{N}\cdot10\log(N/\gamma)\leq\frac{100\log(N/\gamma)}{M}\cdot10\log(N/\gamma)\leq\frac{1000\log(N/\gamma)^2}{M}$$

As there are at most $\mathsf{polylog}(N/\gamma)$ possible choices of (i,j), taking the union bound over all of them we get that with probability at least 0.999 over h, for all (i,j) such that $|B_{i,j}|\leq M$, the probability that an element y was sampled by Y through a non-trigger pair (x,t) such that $(x,t)\in B_{i,j}$ is at most

$$\frac{1000\log^4(N/\gamma)}{M}\tag{51}$$

Next, Fix i,j such that $|B_{i,j}|>M$. There are at most $\frac{N/\gamma}{2^{i-1}}$ elements in $\mathrm{Supp}(D)$ with probability in the range $\left[2^{i-1}\frac{\gamma}{N},2^i\frac{\gamma}{N}\right)$. Moreover, there 2^{j-1} possible values in the range $\left[2^{j-1},2^j\right)$. Therefore:

$$|B_{i,j}|=2^{j-1}\cdot\frac{N/\gamma}{2^{i-1}}=2^{j-i}\frac{N}{\gamma}$$

For every $y \in [M]$ consider the random variable $L^y_{i,j}$ which counts how many $(x,t) \in B_{i,j}$ were hashed to y (the *load* of y with respect to $B_{i,j}$). Observe that:

$$L^y_{i,j} = \sum_{(x,t)\in B_{i,j}} \mathbb{1}_{h(x,t)=y}$$

And since for all (x,t), $\Pr\left(h(x,t) = y\right) = \frac{1}{M}$, we get that $\mathrm{Var}\left[\mathbb{1}_{h(x,t)=y}\right] \le \mathbb{E}\left[\mathbb{1}_{h(x,t)=y}\right] = \frac{1}{M}$, and by the linearity of expectation: $\mathbb{E}\left[L^y_{i,j}\right] = \frac{|B_{i,j}|}{M}$. Since the function h is $(10\log(N/\gamma))$-wise independent, by Claim 3.17:

$$\Pr_h\left(\left|L^y_{i,j} - \frac{|B_{i,j}|}{M}\right| > \sqrt{60\log(N/\gamma)\cdot\frac{|B_{i,j}|}{M}}\right) \le \left(\frac{30\log(N/\gamma)\cdot\frac{1}{M}}{|B_{i,j}|\cdot\left(\sqrt{\frac{60\log(N/\gamma)}{M|B_{i,j}|}}\right)^2}\right)^{10\log\left(\frac{N}{\gamma}\right)} \tag{52}$$

$$\le \left(\frac{1}{2}\right)^{10\log\left(\frac{N}{\gamma}\right)} \tag{53}$$

$$\le \frac{1}{1000M^2} \tag{54}$$

Taking the union bound over all $y \in [M]$, we get that with probability at most $\frac{1}{1000M}$ there exists some $y \in [M]$ such that:

$$L^y_{i,j} > \frac{|B_{i,j}|}{M} + \sqrt{60\log(N/\gamma)\cdot\frac{|B_{i,j}|}{M}}$$

Next, observe that by definition of $B_{i,j}$ and Y, for every $(x_0,t_0),(x_1,t_1) \in B_{i,j}$ it holds that $\frac{\Pr(Y=(x_0,t_0))}{\Pr(Y=(x_1,t_1))} \in [0.5, 2]$. And so, we conclude that the probability of each element in $B_{i,j}$ according to Y is at most $\frac{4}{|B_{i,j}|}$, therefore, for a fixed i,j and every $y \in [M]$, we get that with probability at least $1 - \frac{1}{1000M}$ the probability that y was sampled according to Y through a tuple $(x,t) \in B_{i,j}$ is at most:

$$\left(\frac{|B_{i,j}|}{M} + \sqrt{60\log(N/\gamma)\cdot\frac{|B_{i,j}|}{M}}\right)\cdot\frac{4}{|B_{i,j}|} \le \frac{4}{M} + \sqrt{60\log(N/\gamma)\cdot\frac{1}{M\,|B_{i,j}|}} \le \frac{100\sqrt{60\log(N/\gamma)}}{M}$$

Again, taking the union bound with respect to all choices of i,j, we get that with probability at least 0.999 over the choice of h for all $y \in [M]$, the probability that y was sampled by Y through a trigger pair (x,t) such that $(x,t) \in B_{i,j}$ and $|B_{i,j}| > M$ is at most:

$$\frac{100\log^3(N/\gamma)}{M} \tag{55}$$

To conclude, we get that with probability at least 0.99 over the choice of h, for every $y \in [M]$ the probability it was sampled by Y through a non-trigger pair (x,t) such that $(x,t) \in B_{i,j}$ where $|B_{i,j}| \le M$ is at most $\frac{1000\log^4(N/\gamma)}{N}$, and the probability it was sampled by (x,t) such that $(x,t) \in B_{i,j}$ where $|B_{i,j}| > m$ is at most $\frac{100\log^3(N/\gamma)}{M}$. Summing up, the probability that y was sampled by Y through a non-trigger pair is at most:

$$\frac{1100\log^4(N/\gamma)}{M}$$

□

Claim 5.18. *With probability at least* 0.999 *over the choice of* h*, for every* $y \in [M]$*, if* $X_D(y) = 1$*, it holds that the probability that a pair* (x,t) *was sampled by* Y_D^{pair} *such that* $h(x,t) = y$ *and* (x,t) is *a trigger pair is at most:*

$$\frac{5\log(N/\gamma)}{N^f}$$

Proof. Recall that with probability at least 0.999 over the randomness of the representation, there are $P \le \frac{N}{\gamma} + 100\sqrt{\frac{N}{\gamma}} \le M/2$ trigger pairs, that are hashed into $[M]$ by a $10\log(N/\gamma)$-wise uniform hash function, therefore, the probability that for a given $y \in [M]$ there are at least $5\log(N/\gamma)$ trigger pairs that were hashed to it is at most:

$$\binom{P}{5\log(N/\gamma)}\left(\frac{1}{M}\right)^{5\log(N/\gamma)} \le \left(\frac{P}{M}\right)^{5\log(N/\gamma)} \le \left(\frac{1}{2}\right)^{5\log(N/\gamma)} \le \frac{1}{M^2}$$

Taking the union bound over all $y \in [M]$ we get that with probability at least 0.9999 over the randomness of the representation, it holds that every $y \in [M]$ has at most $5\log(N/\gamma)$ trigger pairs hashed to it. Since the probability of each trigger pair is at most $\left(\max_{x\in[N]} D(x)\right)\cdot\frac{1}{\log^N} \le \frac{1}{N^f \log^2(N)}$, we get that with high probability over the randomness of the representation, for every $y \in [M]$, the probability that a pair (x,t) was sampled by Y_D^{pair} such that $h(x,t) = y$ and (x,t) *is* a trigger pair is at most:

$$5\log(N/\gamma)\cdot\frac{1}{N^f\log^2(N)} \le \frac{5\log(N/\gamma)}{N^f}$$

□

Claim 5.19. *With probability at least* 0.99 *over the choice of* h*, for every* $y \in [M]$*:*

$$\Pr_{Y_U^{pair}}\left(\{(x,t) : h(x,t) = y\}\right) \le \frac{20\log^3(N/\gamma)/\gamma}{M}$$

Proof. Denote by Y_T^{pair} the marginal distribution of Y^{pair} with respect to the second coordinate. Divide the set $[N] \times \left[N^{1-f}/\gamma\right]$ into log subsets, in the following way:

$$\forall i \in \left\{1, 2, \ldots \log\left(N^{1-f}/\gamma\right)\right\} \quad B_i = \left\{(x,t) : t \in \left[2^{i-1}\log N, 2^i \log N\right]\right\}$$

And also define $B_0 = \{(x,t) : t \in [1, \log N]\}$. For every $i \in \left\{1, \ldots, \log\left(N^{1-f}/\gamma\right)\right\}$:

$$|B_i| = N \cdot 2^{i-1}\log N$$

If $|B_i| \le M$, then, the probability that there exists some $y \in [M]$ for which there are more than $\log N$ is at most:

$$M \cdot \binom{|B_i|}{\log N}\left(\frac{1}{M}\right)^{\log N} \le M\left(\frac{|B_i|}{M}\right)^{\log N} \cdot \left(\frac{1}{\log N}\right)^{(\log N)/2} \le M \cdot \left(\frac{1}{2}\right)^{16\log N} \le \frac{1}{N}$$

Taking the union bound over all such subsets i, we get that with probability at least 0.999 over the choice of h, it holds that for every $y \in [M]$, there are at most $(\log N)$ pairs (x,t) such that $h(x,t) = y$, and $(x,t) \in B_i$ such that $|B_i| \le M$. Note that every $\Pr_{Y_U^{\text{pair}}}(x,t) \le \frac{1}{N}\cdot\frac{1}{t}$. Therefore, for every $y \in [M]$ and every bucket i such that $|B_i| \leq M$ it holds that the probability that (x,t) was drawn by Y_U^{pair} such that $h(x,t) = y$ and $(x,t) \in B_i$ is at most:

$$(\log N) \cdot \frac{1}{N} \cdot \frac{1}{t} \leq \frac{1}{N} \leq \frac{10 \log\left(N/\gamma\right)/\gamma}{M}$$

Where the first inequality above is justified since by definition of Y_U^{pair}, the probability of sampling t is at most $\min\left\{\frac{1}{\log N}, \frac{1}{t}\right\}$. Taking the union over all B_i such that $|B_i| \leq M$, we get that with probability at least 0.999 over the choice of h, for all $y \in [M]$, the probability that (x,t) was sampled by Y_U^{pair} such that $h(x,t) = y$ and $(x,t) \in B_i$ for which $|B_i| \leq M$ is at most:

$$\frac{10 \log^3\left(N/\gamma\right)/\gamma}{M}$$

Next, fix some i such that $|B_i| > M$. For every $y \in [M]$, denote by $L_i^y = |\{(x,t) \in B_i : h(x,t) = y\}|$. Note that $L_i^y = \sum_{(x,t) \in B_i} \mathbb{1}_{h(x,t)=y}$. Observe that $\mathbb{E}\left[\mathbb{1}_{h(x,t)=y}\right] = \frac{1}{M}$, this implies that both $\mathbb{E}\left[L_i^y\right] = \frac{|B_i|}{M}$ and $\text{Var}\left[\mathbb{1}_{h(x,t)=y}\right] \leq \frac{1}{M}$. Since h is drawn from a $10\log\left(N/\gamma\right)$-wise uniform hash family, by Claim 3.17 it holds that:

$$\Pr_h\left(\left|L_i^y - \frac{|B_i|}{M}\right| > \sqrt{60 \log\left(N/\gamma\right) \cdot \frac{|B_i|}{M}}\right) \leq \left(\frac{30 \log\left(N/\gamma\right) \cdot \frac{1}{M}}{|B_i| \cdot \left(\sqrt{\frac{60 \log(N/\gamma)}{M|B_i|}}\right)^2}\right)^{10 \log\left(\frac{N}{\gamma}\right)} \tag{56}$$

$$\leq \left(\frac{1}{2}\right)^{10 \log\left(\frac{N}{\gamma}\right)} \tag{57}$$

$$\leq \frac{1}{M^2} \tag{58}$$

Taking the union over all $y \in [M]$, we get that with probability at least $\frac{1}{M}$ over the choice of h, for all $y \in [N]$:

$$L_i^y < \frac{|B_i|}{M} + \sqrt{60 \log\left(N/\gamma\right) \cdot \frac{|B_i|}{M}}$$

Note thatfor all i and $(x,t) \in B_i$, $\Pr_{Y_U^{\text{pair}}}\left((x,t)\right) \leq \frac{1}{|B_i|}$. And so, for every i such that $|B_i| > M$. We thus conclude that for every i such that $|B_i| > M$, with probability at least $1 - \frac{1}{M}$ over the choice of h, for all $y \in [M]$, the probability that (x,t) was sampled by Y_U^{pair} such that $h(x,t) = y$ and $(x,t) \in B_i$, is at most:

$$L_i^y \cdot \frac{1}{|B_i|} \leq \left(\frac{|B_i|}{M} + \sqrt{60 \log\left(N/\gamma\right) \cdot \frac{|B_i|}{M}}\right) \cdot \frac{1}{|B_i|} \leq \frac{1}{M} + \frac{1}{M}\sqrt{60 \log\left(N/\gamma\right)} \leq \frac{10 \log\left(N/\gamma\right)}{M}$$

Where the second to last inequality is due to the assumption that $|B_i| > M$. Taking the union bound over all B_i such that $|B_i| > M$, we get that with probability at least 0.999 for all i such that $|B_i| > M$, the probability that (x,t) was sampled by Y_U^{pair} such that $h(x,t) = y$ and $(x,t) \in B_i$ such that $|B_i| > M$ is at most:

$$\frac{10 \log^3\left(N/\gamma\right)}{M}$$

And we thus conclude that with probability at least 0.99 over the choice of h, for every $y \in [Y]$, the probability that (x, t) was sampled by Y_U^{pair} such that $h(x, t) = y$ is at most:

$$\frac{20 \log^3 (N/\gamma)/\gamma}{M}$$

□

Claim 5.20. *With probability at least* 0.999 *over the choice of h, for every $y \in [M]$, $\Pr_Y(y) \geq \frac{1}{8M \log(N/\gamma)}$*

Proof. Note that $\Pr_Y(y) \geq \frac{1}{2} \Pr_{Y_U^{\text{pair}}} (\{(x,t) : h(x,t) = y\})$. We therefore bound $\Pr_{Y_U^{\text{pair}}} (\{(x,t) : h(x,t) = y\})$ from below. First, note that with probability at least $\frac{1}{2\log(N/\gamma)}$, t_{max} was drawn to be the maximum value. Assume that this value was drawn. Then, (x, t) is drawn uniformly from the space $[N] \times [N^{1-f}/\gamma]$ of size N^{2-f}/γ. Denote L_y the random variable dependent on h that counts how many pairs (x, t) were hashed to y. Note that:

$$L_y = \sum_{(x,t) \in [N] \times [N^{1-f}/\gamma]} \mathbb{1}_{h(x,t)=y}$$

Observe that $\text{Var}\left[\mathbb{1}_{h(x,t)=y}\right] \leq \mathbb{E}\left[\mathbb{1}_{h(x,t)=y}\right] \leq \frac{1}{M}$, and since h is $10\log(N/\gamma)$-wise independent, by Claim 3.17 we get for a fixed y:

$$\Pr_h \left(\left| L_y - \frac{1}{M} \cdot \frac{N^{2-f}}{\gamma} \right| > 10\sqrt{\frac{1}{M} \cdot \frac{N^{2-f}}{\gamma}} \right) \leq \left(\frac{3\log(N/\gamma) \cdot \frac{1}{M}}{\frac{N^{2-f}}{\gamma} \left(10\sqrt{\frac{\frac{1}{M}}{\frac{N^{2-f}}{\gamma}}}\right)^2} \right)^{10\log(N/\gamma)} \leq \frac{1}{M^2}$$

Taking a union bound over all possible y, we get that with probability at least $1 - \frac{1}{M}$ over the choice of h, for every $y \in [M]$:

$$L_y > \frac{1}{M} \cdot \frac{N^{2-f}}{\gamma} - 10\sqrt{\frac{1}{M} \cdot \frac{N^{2-f}}{\gamma}}$$

Therefore, assuming that $t_{\max}$ was chosen to be the largest possible value, we get that the probability that y was drawn according to Y_U^{pair} is at least:

$$\left(\frac{1}{M} \cdot \frac{N^{2-f}}{\gamma} - 10\sqrt{\frac{1}{M} \cdot \frac{N^{2-f}}{\gamma}} \right) \cdot \frac{\gamma}{N^{2-f}} \geq \frac{1}{2M}$$

And since drawn the maximal $t_{\max}$ occurs with probability at least $\frac{1}{2\log(N/\gamma)}$, with probability at least 0.999 over h, it holds that for very $y \in [M]$:

$$\Pr_Y(y) \geq \frac{1}{2} \Pr_{Y_U^{\text{pair}}} (\{(x,t) : h(x,t) = y\}) \geq \frac{1}{8M \log(N/\gamma)}$$

□

Claim 5.21. *With probability at least* 0.999 *over the choice of* h*, for all* $y \in [M]$*, it holds that:*

$$\Pr_{Y_D^{pair}}\left(\{(x,t) : h(x,t) = y\}\right) \le \frac{10\log^4\left(N/\gamma\right)}{N^f}$$

Proof. Since with probability at least 0.999 over the randomness of the representation, the set of trigger pairs TRIG contains at most $P \le \frac{N}{\gamma} + 100\sqrt{\frac{N}{\gamma}}$ pairs, and since $M = 100\frac{N}{\gamma}\cdot\log\left(\frac{N}{\gamma}\right)$, with probability at least 0.999 over the choice of h, since h is $10\log\left(N/\gamma\right)$-wise independent, the probability that there are more than $10\log\left(N/\gamma\right)$ trigger pairs that are hashed by h to the same $y \in [M]$ is:

$$\begin{aligned}
\binom{P}{10\log\left(N/\gamma\right)}\left(\frac{1}{M}\right)^{10\log(N/\gamma)} &\le \left(\frac{P}{M}\right)^{10\log(N/\gamma)}\cdot\left(\frac{1}{10\log\left(N/\gamma\right)}\right)^{10\log(N/\gamma)} && (59)\\
&\le \left(\frac{1}{10\log\left(N/\gamma\right)}\right)^{10\log(N/\gamma)} && (60)\\
&\le \frac{1}{1000} && (61)
\end{aligned}$$

And so, we conclude that every $y \in [M]$ contains at most $10\log\left(N/\gamma\right)$ trigger pairs. Since the probability under Y_D^{pair} of every trigger pair is at most $\frac{1}{N^f}\cdot\frac{1}{\log^2 N}$, we get that with high probability over h, for every $y \in [M]$, the probability that y was sampled by Y_D^{pair} through trigger pair (x,t) such that $h(x,t) = y$ is at most:

$$\frac{1}{N^f}\cdot\frac{10\log\left(N/\gamma\right)}{\log^2 N}\cdot \le \frac{10\log\left(N/\gamma\right)}{N^f}$$

Recall that by Claim 5.17, the probability that (x,t) was drawn according to Y_D^{pair} such that $h(x,t) = y$ and (x,t) *isn't* a trigger pairs, is at most $\frac{1100\log^4(N/\gamma)}{M}$. And so, we get that with high probability over the choice of h, for every $y \in [M]$: $\Pr_{Y_D^{\text{pair}}}(y) \le \frac{1100\log^4(N/\gamma)}{M} + \frac{10\log(N/\gamma)}{N^f} \le \frac{10\log^4(N/\gamma)}{N^f}$ □

Claim 5.22. *Assume that the pair* (x,t) *drawn by* Y_D^{pair} *is a trigger pair. Then, for every* $\rho \in (0,1)$*, with probability at least* $(1-\rho)$*,* t *satisfies:*

$$t\cdot\frac{\gamma}{N} \le D(x)\left(1-\rho\right)$$

Proof. Let t_1, t_2 be such that $t_1 < t_2$ and both (x,t_1) and (x,t_2) are trigger pairs, then, by definition of Y_D^{pair}, it holds that $\Pr_{Y_D^{\text{pair}}\big|_{X=x}}(t_2) \le \Pr_{Y_D^{\text{pair}}\big|_{X=x}}(t_1)$, since for every $t_{\max}$ drawn by Y_D^{pair} that satisfies $t_2 \le t_{\max}$, it also holds that $t_1 < t_{\max}$, and so, both t_1 and t_2 in this case will be equally likely to be sampled.

Therefore, the distribution T over $\left\{1, 2, \ldots, \frac{D(x)+\gamma_x}{\gamma/N}\right\}$ induced by the marginal distribution of t according to Y_D^{pair}, given that x was sampled and (x,t) is a trigger pair satisfies the condition that $T(i) \ge T(i+1)$. Observe that the probability that t_0 drawn according to T satisfies $t_0 > (1-\rho)\frac{D(x)+\gamma_x}{\gamma/N}$ is at most ρ. □

We are set to prove Proposition 5.7:

Proof of Proposition 5.7. For every y denote $\overline{\text{trig}_y} = \{(x,t) \notin \text{TRIG} : h(x,t) = y\}$. Observe that for every y:

$$\frac{\Pr_{Y_D^{\text{pair}}}\left(\text{trig}_y\right)}{\Pr_Y(y)} = \frac{\Pr_{Y_D^{\text{pair}}}\left(\text{trig}_y\right)}{\Pr_{Y_D^{\text{pair}}}\left(\text{trig}_y\right) + \Pr_{Y_D^{\text{pair}}}\left(\overline{\text{trig}_y}\right) + \Pr_{Y_U^{\text{pair}}}(y)}$$

Plugging in Claims 5.17 and 5.19 we get:

$$\frac{\Pr_{Y_D^{\text{pair}}}\left(\text{trig}_y\right)}{\Pr_Y(y)} \geq \left(\Pr_{Y_D^{\text{pair}}}\left(\text{trig}_y\right)\right) \cdot \left(\Pr_{Y_D^{\text{pair}}}\left(\text{trig}_y\right) + \frac{1100\log^4(N/\gamma)}{M} + \frac{20\log^3(N/\gamma)\,\gamma}{M}\right)^{-1}$$

Since by Claim 5.16 $\Pr_{Y_D^{\text{pair}}}\left(\text{trig}_y\right) \geq \frac{50}{M\log N}$, we get that:

$$\frac{\Pr_{Y_D^{\text{pair}}}\left(\text{trig}_y\right)}{\Pr_Y(y)} \geq \frac{50/\log N}{(50/\log N) + 1100\log^4(N/\gamma) + 20\log^3(N/\gamma)/\gamma} \geq \frac{\gamma}{1100\log^4(N/\gamma)}$$

This concludes the first part of Proposition 5.7.

Next, note that for every $y \in [M]$:

$$\Pr_Y(y) = \frac{1}{2}\Pr_{Y_D^{\text{pair}}}\left(\{(x,t) : h(x,t) = y\}\right) + \frac{1}{2}\Pr_{Y_U^{\text{pair}}}\left(\{(x,t) : h(x,t) = y\}\right) \tag{62}$$

$$= \frac{1}{2}\left(\Pr_{Y_D^{\text{pair}}}\left(\text{trig}_y\right) + \Pr_{Y_D^{\text{pair}}}\left(\overline{\text{trig}_y}\right)\right) + \frac{1}{2}\Pr_{Y_U^{\text{pair}}}\left(\{(x,t) : h(x,t) = y\}\right) \tag{63}$$

With probability at least 0.95 over the choice of h: by Claim 5.18, $(\Pr_{Y_D^{\text{pair}}}\left(\text{trig}_y\right) \leq \frac{5\log(N/\gamma)}{N^f}$; by Claim 5.17, $\Pr_{Y_D^{\text{pair}}}\left(\overline{\text{trig}_y}\right) \leq \frac{1100\log^4(N/\gamma)}{M}$; and by Claim 5.19, $\Pr_{Y_U^{\text{pair}}}\left(\{(x,t) : h(x,t) = y\}\right) \leq \frac{20\log^3(N/\gamma)/\gamma}{M}$. Plugging this in Equation 62:

$$\Pr_Y(y) \leq \frac{1100\log(N/\gamma)/\gamma}{N^f}$$

The lower bound for $\Pr_Y(y)$ is obtained through Claim 5.20. This conclude the proof of second clause of Proposition 5.7. Claim 5.22 provides the proof for the third clause. □

6 Full Protocol

Having established all the building blocks of the protocol in the previous sections, we show that Protocol 6.0.1 satisfies the conditions of Theorem 1.1.

Proposition 6.1. *Assume distribution D over $[N]$ satisfies the condition that for every $x \in [N]$, $D(x) \leq N^{1-\frac{1}{2(c+1)}}$, where c is the constant from Theorem 3.9. Let $\mathcal{P}$ be a distribution property that's ρ-approximately decidable by a logspace uniform family of NC^1 circuits. Let $0 \leq \varepsilon_c < \varepsilon_f \leq 1$ be such that $\frac{\varepsilon_f - \varepsilon_c}{3} > \rho$. With high probability over the randomness of Protocol 6.0.1:*

Protocol 6.0.1: General Distribution Property Verification Protocol

Input: Parameter $N \in \mathbb{N}$, black-box sample access to a distribution D over domain $[N]$ such that for all $x \in [N]$, $D(x) \leq \frac{1}{N^f}$, for $f = 1 - \frac{1}{2(c+1)}$, where c is the constant featured in Theorem 3.9; parameters $0 \leq \varepsilon_c(N) < \varepsilon_f(N) \leq 1$, satisfying $\varepsilon_f(N) - \varepsilon_c(N) > \mathsf{poly}(1/N)$ and a distribution property $\mathcal{P} = (\mathcal{P}_N)$ that is $\rho = \rho(N) = \frac{\varepsilon_f(N) - \varepsilon_c(N)}{3}$-approximately decidable by a family of logspace uniform NC^1 circuits $(C_N)_N$.

1. V: set $\gamma = \rho / \log N$ and $M = M(N)$ as defined in Construction 5.1. Draw functions $h : [N] \times \left[N^{1-f}/\gamma\right] \to [M]$, and $h_\gamma : [N] \to \left\{0, \frac{\gamma}{N}, \frac{2\gamma}{N}, \ldots, \gamma\right\}$ from a $10 \log(N/\gamma)$-wise independent hash family and a pairwise independent hash family, respectively. Send h, h_γ to P.
2. V-P: let X_D be the representation of D as described in Construction 5.1, and let C'_N be the circuit defined in Claim 5.21. Run the IPP protocol outlined in Figure 4.7.1 over the implicit input, X_D, the circuit C'_N, and the explicit input $\varepsilon_c, \varepsilon_f, h, h_\gamma$ (all serve as parameters to C'_N as explained in Claim 5.21), with parameter $r = N^f$, $u = N^{\frac{1}{4(c+1)}}$. Upon invoking the Y-sampler, the verifier draws a sample $((x,t), b, y)$ according to the Representation Sampler described in Construction 5.4, and uses $y \in [M]$ as the Y sample. At the end of this step, either the verifier rejects or it obtains:
   - Query set $Q \in [M]^{r \cdot c}$ as well as the alleged values of X_D on Q, $\bar{v} \in \{0,1\}^Q$.
   - The secret output of the verifier: I_Y, I_U, substrings of Q.
   - The output of the Representation Sampler $S^Y = (((x,t), b, y))$.
3. V: check that the number of 1's in $\bar{v}\left[I_U\right]$ satisfies Inequality (64), reject otherwise.
4. V-P: the verifier sets $S = \left(x : ((x,t), b, y) \in S^Y\right)$, secretly divided into $S_D = \left(x : ((x,t), 1, y) \in S^Y\right)$ and $S_U = \left(x : ((x,t), 0, y) \in S^Y\right)$. Then, the parties run the Verified Tagged-Sample Protocol (see Theorem 3.9) over distribution D with sample S, secret input S_D and S_U, and distance parameter $\sigma = \frac{1}{200}\left(\frac{\delta \cdot \gamma}{2000 \log^4(N/\gamma)}\right)^2$, for $\delta = \Theta\left(\frac{1}{u} \cdot \frac{1}{\log^3 M}\right)$. Reject if protocol rejects. Otherwise, obtain $\left(\widetilde{D}_x\right)_{x \in S_D}$.
5. V: for every $((x,t), 1, y) \in S^Y$, if $\bar{v}(y) = 0$, set $c_x = \frac{\gamma}{N} \cdot t$. Accept unless there exists x s.t. $\widetilde{D}_x \geq c_x$.

- *If $\delta_{TV}(D, \mathcal{P}_N) \leq \varepsilon_c$, the verifier accepts.*
- *If $\delta_{TV}(D, \mathcal{P}_N) \geq \varepsilon_f$, for every prover strategy P^*, with high probability, the verifier rejects.*

Moreover, ignoring polynomial dependence on $\left(\frac{1}{\rho}\right)$, Protocol 6.0.1 has verifier sample complexity of $\widetilde{O}\left(N^{1-\frac{1}{2(c+1)}}\right)$, its communication complexity and verifier runtime are both $\widetilde{O}\left(N^{1-\frac{1}{4(c+1)}}\right)$, the number of rounds is $O(\log N)$, and the protocol is doubly efficient.

Remark 6.2. *Observe that Protocol 6.0.1 assumes that for all $x \in [N]$, $D(x) \leq N^{-f} = N^{1-\frac{1}{2(c+1)}}$. In general, we can't assume a given distribution D satisfies this condition. However, in the vein of Herman and Rothblum [HR23], we can consider the following pre-processing phase to Protocol 6.0.1. First, the verifier takes a sample of size $\widetilde{O}\left(N^f\right)$. With high probability, this sample will contain all elements x such that $D(x) \geq N^{-f}$, if there are any. The verifier will classify all the elements sampled as* heavy, *and denote them as $\mathcal{H}$. Then, the verifier proceeds to first estimate $D(\mathcal{H})$ by* *taking* $\mathsf{poly}(1/\rho)$ *samples. If this quantity is smaller than* ρ*, the verifier can ignore the set* $\mathcal{H}$ *in its entirety, and think of all the elements inside as having probability* 0 *according to* D*. Otherwise, the verifier can estimate* $D\big|_{\mathcal{H}}$ *up to accuracy* ρ *with* $\widetilde{O}\left(N^f/\rho^2\right)$ *samples using the Folklore Distribution Learner from Theorem 3.3. Then, we consider the explicit description of* $D\big|_{\mathcal{H}}$ *as part of the implicit input to* C'_N*, in the form of hard-wired bits in the input string. Note that this separation of high-probability elements and low-probability elements requires only* $\widetilde{O}\left(N^f \cdot \rho^{-2}\right)$ *samples, and as such, does not exceed the sample complexity of the protocol.*

The rest of this section is devoted to proving Proposition 6.1. We first show the existence of the circuit C'_N used in the protocol:

Claim 6.3. *Let* $\mathcal{P}$ *be a distribution property that's* $\rho(N)$*-approximately decidable by a logspace uniform* NC^1 *circuit family. For every* $N \in \mathbb{N}$*: set* $\gamma = \gamma(N) = \rho(N)/\log N$*, and let* h *and* h_γ *be functions as described in Construction 5.1 drawn from* $(10\log(N/\gamma))$*-wise independent hash family and a pairwise independent hash family respectively (note that* h *and* h_γ *also depend on* N*). Then if we fix some distribution* D *over* $[N]$ *with representation* $X_D \in \{0,1\}^M$ *as described in Construction 5.1, there exists a circuit* C'_N *that takes as input* $X \in \{0,1\}^M$*,* h, h_γ*, as well as* $\varepsilon_c = \varepsilon_c(N), \varepsilon_f = \varepsilon_f(N) \in (0,1)$ *such that* $\varepsilon_f - \varepsilon_c > \rho(N) = \rho$*, and with probability at least* 0.99 *over the choice of* h *and* h_γ*:*

- *If* $\delta_{TV}(D, \mathcal{P}) < \varepsilon_c$*, then* $X_D \in \mathcal{L}(C'_N)$*.*
- *If* $\delta_{TV}(D, \mathcal{P}_N) > \varepsilon_f$*, then* $Ham(X_D, \mathcal{L}(C'_N)) \geq \frac{\rho}{\log^3 M}$*.*

Moreover, the family $(C'_N)_N$ *is a logspace-uniform* NC^1 *circuit family.*

Proof. Fix $\gamma = \rho/\log N$ and distribution D over $[N]$. Let (R_N) be the log-space uniform NC^1 circuit family that satisfies the conditions of Proposition 5.3.

Since we assumed distribution property $\mathcal{P}$ is $\rho(N) \leq \frac{\varepsilon_f - \varepsilon_c}{3}$-approximately decidable by some family of logspace uniform NC^1 circuits $(C_N)_N$, then, setting $\delta_c = \varepsilon_c + \rho$ and $\delta_f = \varepsilon_f - \rho$, we get that $\delta_f - \delta_c > \rho$. Therefore, with high probability over the choice of h and h_γ:

- If D satisfies $\delta_{\mathrm{TV}}(D, \mathcal{P}_N) \leq \varepsilon_c$, then by Proposition 5.3:

  $$\delta_{\mathrm{TV}}(R_N(X_D), \mathcal{P}) \leq \delta_{\mathrm{TV}}(R_N(X_D), D) + \delta_{\mathrm{TV}}(D, \mathcal{P}) \leq \varepsilon_c + \gamma \cdot \log N \leq \varepsilon_c + \rho \leq \delta_c$$

  And so $C_N(R_N(X_D)) = 1$.

- If D is ε_f-far from $\mathcal{P}$, then for every X_0 and for every $\delta \in (0, \rho)$, that differs from X_D in at least $\frac{N}{\gamma \log^2 N} \cdot \delta \leq M \cdot \frac{\delta}{\log^3(M)}$ locations, either R_N rejects, or $\delta_{\mathrm{TV}}(R_N(X_0), D) \leq \delta$. Therefore:

  $$\delta_{\mathrm{TV}}(R_N(X_0), \mathcal{P}) \geq \delta_{\mathrm{TV}}(D, \mathcal{P}) - \delta_{\mathrm{TV}}(R_N(X_0), D) \geq \varepsilon_f - \delta \geq \delta_f$$

  And so $C_N(R_N(X)) = 0$, and we conclude that X_D satisfies $\mathrm{Ham}(X_D, X) \geq \frac{\rho}{\log^3 M}$ for every $X \in \mathcal{L}(C_N(R_N(x)))$.

Therefore, it we set for every N, set the circuit $C'_N = C_N \circ R_N$, since (C_N) and (R_N) are logspace-uniform NC^1 circuit families, so is their composition. And we get that (C'_N) satisfies all the conditions of the above claim. □

Claim 6.4. *With probability at least* 0.999 *over the randomness of the verifier:*

$$\left| wt\left(X_D \mid Q[I_U]\right) - \frac{1}{M}\left(\frac{N}{\gamma} - \sum_{k=2}^{\log\left(N/\gamma^{1/2}\right)} \binom{N/\gamma}{k}\left(\frac{1}{M}\right)^{k-1}\right)\right| \leq \frac{400}{\sqrt{|I_U|}} \cdot \sqrt{\frac{N/\gamma}{M}} \tag{64}$$

Proof. Recall that I_U is the set of locations in Q, sampled in the following way: the verifier drew a pairwise random permutation of the domain, and according to it a partition, $\{P_i\}_{i\in[M/u]}$[4]. Then, the verifier samples r partition sets uniformly at random, and then adds the entire partition sets to the query set Q, while keeping in I_U the index of these entries in Q (we assume that when the verifier selects P_i to be added to Q it also chooses a random order to add it, in order to ignore the ordering inside each partition). Note that for every $y \in [M]$ and $i \in I_U$ the probability over the randomness of the partition and the randomness of the verifier that $Q\left(i\right) = y$ is $\frac{1}{M}$.

Next, define the random variable S_i to be the indicator that $y_i = Q(I_U(i))$ satisfies $X_D(y) = 1$. Since by definition for a uniformly chosen $y \in [M]$, $\Pr(X_D(y) = 1) = \mathrm{wt}(X_D)$ and since y_i is distributed uniformly over $[M]$, we get that $\mathbb{E}\left[S_i\right] = \mathrm{wt}(X_D)$. For any $i, i' \in |I_U|$, assume $S_i = 1$. Since the partition is taken from a pairwise independent permutation, the distribution of $y_{i'}$ given the value of y_i is uniform over $[M]$, and so $\mathbb{E}\left[S_{i'} = 1\big|_{S_i=1}\right] = \frac{M\mathrm{wt}(X_D)-1}{M-1}$.

By definition $\mathrm{wt}\left(X_D \mid Q[I_U]\right) = \frac{1}{|I_U|}\sum_{i\in[|I_U|]} S_i$. Note that by the above argument for every $i \neq i'$:

$$\mathrm{Cov}\left[S_i, S_{i'}\right] \leq \mathrm{wt}(X_D) \cdot \left(\frac{M\mathrm{wt}(X_D) - 1}{M-1}\right) - \left(\mathrm{wt}(X_D)\right)^2 \leq \frac{\mathrm{wt}^2(X_D)}{2M}$$

Therefore:

$$\mathrm{Var}\left[\sum_{i\in I_U} S_i\right] \leq \sum_{i\in I_U} \mathrm{Var}[S_i] + \sum_{i,i'\in I_U} \mathrm{Cov}[S_i, S_{i'}] \leq 2\sum_{i\in I_U} \mathbb{E}[S_i] = 2\mathbb{E}\left[\sum_{i\in I_U} S_i\right]$$

And so, by Chebichev's Inequality:

$$\Pr\left(\left|\mathbb{E}\left[\sum_{i\in I_U} S_i\right] - \sum_{i\in I_U} S_i\right| \geq 200\sqrt{\mathbb{E}\left[\sum_{i\in I_U} S_i\right]}\right) \leq \frac{1}{10000}$$

And so, with high probability if the prover is honest

$$\left||I_U|\,\mathrm{wt}\left(X_D \mid Q[I_U]\right) - |I_U|\,\mathrm{wt}(X_D)\right| \leq 200\sqrt{|I_U|\,\mathrm{wt}(X_D)}$$

Since $\mathrm{wt}\left(X_D\right)$ is well concentrated around its mean as shown in Proposition 5.2, we get that w.h.p.:

$$\left||I_U| \cdot \mathrm{wt}\left(X_D \mid Q[I_U]\right) - \frac{|I_U|}{M}\left(\frac{N}{\gamma} - \sum_{k=2}^{\log\left(N/\gamma^{1/2}\right)} \binom{N/\gamma}{k}\left(\frac{1}{M}\right)^{k-1}\right)\right| \tag{65}$$

$$\leq 200\sqrt{\frac{|I_U|\,N/\gamma}{M}} + \frac{200\log^3(M)\,|I_U|}{M} \cdot \sqrt{N/\gamma} \tag{66}$$

$$\leq 400\sqrt{|I_U|\,\frac{N/\gamma}{M}} \tag{67}$$

[4] see Section 3.1 for more on this. In particular, the verifier drew a k-wise independent γ-close partition, but we consider it in this section as a pair-wsie partition following Remark 3.16

□

Claim 6.5. *For any $\delta \in (0,1)$ such that $\delta = \omega\left(\frac{1}{\sqrt{r}} + \frac{\log^2 M}{\sqrt{M}}\right)$ and $\delta \leq \rho$, and if $u \geq 1100 \frac{N^{1-f}}{\gamma^2} \log(M)$: with high probability over the randomness of Protocol 4.7.1 and the choice of h and h_γ:*

- *If $\delta_{TV}(D, \mathcal{P}_N) \leq \varepsilon_c$, both Step (2) and Step (3) of Protocol 6.0.1 do not result in rejection.*
- *If $\delta_{TV}(D, \mathcal{P}_N) \geq \varepsilon_f$ for any prover strategy P^*, if both Step (2) and Step (3) passed, then:*

$$\frac{1}{|S_D|} \sum_{x:(x,t,i)\in S_D} \left(1 - \min\left\{\frac{c_x}{D(x)}, \frac{D(x)}{c_x}\right\}\right) \geq \frac{1}{200}\left(\frac{\delta \cdot \gamma}{2000 \log^4(N/\gamma)}\right)^2 \tag{68}$$

Proof. Assume first that the prover is honest and $\delta_{\mathrm{TV}}(D, \mathcal{P}_N) \leq \varepsilon_c$. If so, by Claim 6.3, with probability at least 0.99 over the randomness of the representation, $X_D \in \mathcal{L}(C'_N)$, and by Lemma 4.8, the verifier doesn't reject and $(X_D \mid Q) = \bar{v}$, and in particular,wt $(\bar{v} \mid Q[I_U]) = \mathrm{wt}(X_D \mid Q[I_U])$, by Claim 6.4, Step (3) passes with high probability.

Next, assume $\delta_{\mathrm{TV}}(D, \mathcal{P}_N) \geq \varepsilon_f$. Fix $\delta \leq \rho$. Then, by Claim 6.3, with high probability, X_D is differs from any $X \in \mathcal{L}(C'_N)$ in at least $M \cdot \delta / \log^3 M$ locations. This implies through Lemma 4.8, that with all but $\mathtt{exp}(-\theta(r))$ probability:

$$\Delta\left((X_D|Q[I_U]), \bar{v}[I_U]\right) = \Theta\left(\delta/\log^3 M\right).$$

In particular, assuming Step (3) of Protocol 6.0.1 passed, we know that:

$$|I_U| \cdot \mathrm{wt}\left(\bar{v}[I_U]\right) \in \frac{|I_U|}{M}\left(\frac{N}{\gamma} - \sum_{k=2}^{\log\left(N/\gamma^{1/2}\right)} \binom{N/\gamma}{k}\left(\frac{1}{M}\right)^{k-1}\right) \pm 400 \cdot \sqrt{|I_U|} \cdot \sqrt{\frac{N/\gamma}{M}} \tag{69}$$

And from Claim 6.4:

$$|I_U| \cdot \mathrm{wt}\left(X_D[I_U]\right) \in \frac{|I_U|}{M}\left(\frac{N}{\gamma} - \sum_{k=2}^{\log\left(N/\gamma^{1/2}\right)} \binom{N/\gamma}{k}\left(\frac{1}{M}\right)^{k-1}\right) \pm 400 \cdot \sqrt{|I_U|} \cdot \sqrt{\frac{N/\gamma}{M}}$$

Therefore, if we denote $R^U_{1\to 0} = \{y \in I_U : X_D(y) = 1, \bar{v}(y) = 0\}$ and $R^U_{0\to 1} = \{y \in I_U : X_D(y) = 1, \bar{v}(y) = 0\}$. By definition, it holds that $\left|R^U_{1\to 0}\right| + \left|R^U_{0\to 1}\right| = |I_U| \cdot \Delta\left((X_D|Q[I_U]), \bar{v}[I_U]\right)$, and so, it follows that:

$$\left|\left|R^U_{1\to 0}\right| - \left|R^U_{0\to 1}\right|\right| \leq 800 \cdot \sqrt{|I_U|} \cdot \sqrt{\frac{N/\gamma}{M}} \tag{70}$$

Since otherwise, Inequality (69) would not have held. And in particular:

$$\begin{aligned}
\left|R^U_{1\to 0}\right| &\geq \frac{1}{2} \cdot |I_U| \cdot \Delta\left((X_D|Q[I_U]), \bar{v}[I_U]\right) - 800 \cdot \sqrt{|I_U|} \cdot \sqrt{\frac{N/\gamma}{M}} && (71)\\
&\geq \Theta\left(|I_U|\left(\delta/\log^3 M - \frac{1}{\sqrt{|I_U| \log(N/\gamma)}}\right)\right) && (72)\\
&\geq |I_U|\,\Theta\left(\delta/\log^3 M\right) && (73)
\end{aligned}$$

Where the last inequality is justified by the assumption that $\delta = \Theta\left(\frac{1}{u} \cdot \frac{1}{\log^3(M)}\right)$, while $\frac{1}{\sqrt{|I_U|}} = \Theta\left(\frac{1}{\sqrt{r \cdot u}}\right)$, and since $r \gg u$, it follows that $\frac{1}{\sqrt{|I_U|}} = o(\delta/\log^3(M))$. Denote $R_{1\to 0} = \{y \in Q : X_D(y) = 1, \bar{v}(y) = 0\}$ the set of *all* queries where the prover claimed the value if 0, but the true value in X_D is 1, including locations from both I_U and I_Y. By Inequality (71) and since $R^U_{1\to 0} \subseteq R_{1\to 0}$, we conclude that:

$$|R_{1\to 0}| \geq |I_U| \, \Theta\left(\delta/\log^3 M\right) \tag{74}$$

By Proposition 5.7, for all $y \in [M]$:

$$Y(y) \in \left[\frac{1/\left(8\log\left(N/\gamma\right)\right)}{M}, \frac{1100\log\left(N/\gamma\right)/\gamma}{N^f}\right] \subseteq \left[\frac{1/\left(8\log\left(N/\gamma\right)\right)}{M}, \frac{1100\log\left(M\right) \cdot \frac{N^{1-f}}{\gamma^2}}{M}\right]$$

Also, by Lemma 4.8, we know that $|I_U| = \Theta(r \cdot u)$, and so, since we assumed $\delta = \omega\left(\frac{1}{\sqrt{r}}\right)$, it holds that $|R_{1\to 0}| \geq \Theta\left(\delta \cdot r \cdot u\right) \geq \Theta\left(\sqrt{r} \cdot u \cdot \log\left(M\right) \cdot 8\log\left(M\right)\right)$. Therefore, again by Lemma 4.8, it holds that:

$$|R_{1\to 0} \cap I_Y| \geq \Theta\left(\frac{1}{u \cdot \log M \cdot \log M}|R_{1\to 0}|\right) \geq \Theta\left(r \cdot \frac{\delta}{\log^3 M}\right)$$

Since by Proposition 5.7, $|I_Y| = \Theta(r)$, we get that $\frac{|R_{1\to 0} \cap I_Y|}{|I_Y|} \geq \Theta\left(\frac{\delta}{\log^3 M}\right)$. Finally, plugging $\eta = \Theta\left(\frac{\delta}{\log^3 M}\right)$ in Claim 6.6 concludes the proof.

□

Claim 6.6. *Let (c_x) be as defined in Protocol 6.0.1 and $R_{1\to 0} = \{y \in Q : X_D(y) = 1, \bar{v}(y) = 0\}$. With high probability over the samples drawn from Y, if $\frac{|R_{1\to 0} \cap I_Y|}{|I_Y|} = \eta$, then:*

$$\frac{1}{|S_D|}\sum_{x:(x,t,i)\in S_D}\left(1 - \min\left\{\frac{c_x}{D(x)}, \frac{D(x)}{c_x}\right\}\right) \geq \frac{1}{200}\left(\frac{\eta \cdot \gamma}{2000\log^4\left(N/\gamma\right)}\right)^2$$

Proof. Observe that all samples in I_Y by definition were first sampled by Y^{pair}, then were hashed to a value $y \in [M]$. For every $y \in [M]$ denote by $p^{\text{trig}}_y = \frac{\Pr_{Y^{\text{pair}}_D}(y)}{\Pr_Y(y)}$. Define:

$$I'_Y = \{(y, b) \in I_Y \times \{0,1\} : b = 1 \iff y \text{ drawn by trigger pair } (x,t)\text{, and } x \text{ by } D\}$$

Consider an alternative process of generating I'_Y: for every $y \in I_Y$, set $b_y = 1$ with probability p^{trig}_y, and 0 otherwise, then define:

$$I''_Y = \{(y, b_y) \in I_Y \times \{0,1\}\}$$

Note that only given I_Y, the distribution of I''_Y is identical to the distribution of I'_Y. Therefore, if we consider any prover strategy P^* that seeks to miss-classify $y \in I_Y$ for which $X_D(y) = 1$ and claim that $\bar{v}(y) = 0$, we can analyze the outcome according to I''_Y, where the value y_b can be chosen *after* the prover's response independently for each y.

Concretely, fix I_Y and some prover's claim $\bar{v}$, which induces a set $R_{1\to 0}$. Assume that $\frac{|R_{1\to 0} \cap I_Y|}{|I_Y|} = \eta$, i.e. $|R_{1\to 0} \cap I_Y| \geq \eta \cdot |I_Y|$. Then, for every $y \in I_Y$ denote $\mathbb{1}_{b_y = 1}$ to indicate $b_y = 1$. Then, if we denote $R^{Y,\text{trig}}_{1\to 0} \subseteq R_{1\to 0} \cap I_Y$ to be the random set that contains all entries in $y \in R_{1\to 0} \cap I_Y$

such that $(y,1) \in I''_Y$, then, by definition $\left|R^{Y,\text{trig}}_{1\to 0}\right| = \sum_{y\in R_{1\to 0}\cap I_Y} \mathbb{1}_{b_y=1}$ is a sum of independent Bernoulli variables, so by the Chernoff Inequality:

$$\Pr_{I''_Y}\left(\left|\left|R^{Y,\text{trig}}_{1\to 0}\right| - \mathbb{E}\left[\left|R^{Y,\text{trig}}_{1\to 0}\right|\right]\right| > \sqrt{30\mathbb{E}\left[\left|\left|R^{Y,\text{trig}}_{1\to 0}\right|\right|\right]}\right) \le 2e^{\frac{30}{3}} \le \frac{1}{10000}$$

And since by Proposition 5.7, for every $p^{\text{trig}}_y \ge \frac{\gamma}{1100\log^4(N/\gamma)} \ge \frac{\gamma}{1100\log^4(M)}$, it holds that:

$$\mathbb{E}\left[\left|R^{Y,\text{trig}}_{1\to 0}\right|\right] = \mathbb{E}\left[\sum_{y\in R_{1\to 0}\cap I_Y} \mathbb{1}_{b_y=1}\right] \ge \left|R_{1\to 0}\cap I_Y\right| \cdot \frac{\gamma}{1100\log^4\left(N/\gamma\right)} \ge \frac{\eta\cdot\gamma}{1100\log^4\left(N/\gamma\right)}\left|I_Y\right|$$

And so, we conclude that with high probability over the randomness of Y, for any prover strategy P^*, if $\frac{|R_{1\to 0}\cap I_Y|}{|I_Y|} = \eta$, then:

$$\left|R^{Y,\text{trig}}_{1\to 0}\right| \ge \frac{\eta\cdot\gamma}{1100\log^4\left(N/\gamma\right)}\left|I_Y\right| - \sqrt{\frac{30\eta\cdot\gamma}{1100\log^4\left(N/\gamma\right)}\left|I_Y\right|} \ge \frac{\eta\cdot\gamma}{1500\log^4\left(M\right)}\left|I_Y\right|$$

Define $\eta' = \frac{\eta\cdot\gamma}{2000\log^4(N/\gamma)}$. Next, for every $y \in R^{Y,\text{trig}}_{1\to 0}$ denote (x_y, t_y) the (trigger) pair drawn by Y^{pair} by which y was sampled. By Proposition 5.7 we get that the *expected* number of $y \in R^{\text{trig}}_{1\to 0}$ such that trigger pairs (x_y, t_y) satisfy $\frac{\gamma}{N}\cdot t_y \ge (1-\eta'/200)D(x)$ is $\frac{\eta'}{200}$. Therefore, by Markov's Inequality, with probability at most least 0.99, the fraction of $y \in [M]$ for which (x_y, t_y) satisfies $\frac{\gamma}{N}\cdot t_y \ge (1-\eta'/200)D(x)$ is at most $\frac{\eta'}{2}$. Therefore, at least $(1-\frac{\eta'}{2})$-fraction of $R^{Y,\text{trig}}_{1\to 0}$ satisfies $\frac{\gamma}{N}\cdot t_y < (1-\eta'/200)D(x)$. However, since for every $y \in R^{\text{trig}}_{1\to 0}$ the prover claimed that $\bar{v}(y) = 0$, the prover effectively claims that the probability of x according to D is smaller that $(1-\frac{\eta'}{200})D(x)$. And so, setting c_x as in Protocol 6.0.1, we conclude that for at least $(1-\eta/2)$-fraction of $y \in R^{Y,\text{trig}}_{1\to 0}$, it holds that (x_y, t_y) satisfies $\left(1-\min\left\{\frac{c_x}{D(x)}, \frac{D(x)}{c_x}\right\}\right) \ge \frac{\eta'}{200}$. Therefore, let S be as defined in Protocol 6.0.1, then:

$$\frac{1}{|S_D|}\sum_{x:(x,t,i)\in S_D}\left(1-\min\left\{\frac{c_x}{D(x)}, \frac{D(x)}{c_x}\right\}\right) \ge \frac{1}{|S_D|}\cdot\left|R^{Y,\text{trig}}_{1\to 0}\right|\cdot\frac{\eta'}{200} = \frac{1}{|S_D|}\cdot\left|I_Y\right|\cdot\eta'\cdot\frac{\eta'}{200} \ge \frac{\eta'^2}{200}$$

Where the final equality stems from the fact that $|S_D| < |I_Y|$.

□

We are now set to Proposition 6.1:

Proof of Proposition 6.1. Assume first $\delta_{\text{TV}}\left(D, \mathcal{P}_N\right) \le \varepsilon_c$. By Claim 6.5, with high probability, Steps (2) and Step (3) of Protocol 6.0.1 don't result in rejection. Next, from the completeness of the protocol satisfying the conditions of Theorem 3.9, Step (4) also passes with high probability. Since the prover is honest, for all x $D(x) \le c_x$, and Step (5) also passes.

Assume next that $\delta_{\text{TV}}\left(D, \mathcal{P}_N\right) \ge \varepsilon_f$. By Claim 6.5, since $\delta = \frac{1}{u} = \frac{1}{N^{14(c+1)}} = \omega\left(\frac{1}{\sqrt{r}} + \frac{1}{\sqrt{M}}\right)$ with high probability over the randomness of the protocol, it holds that for any prover strategy P^*, if both Steps (2) and (3) passed, then Inequality (68) holds. Assume Step (4) doesn't result in rejection. This means that with high probability over the randomness of the Tagged-Sample-Protocol, according to Theorem 3.9, it holds that:

$$\Delta_{S,S_D}((\widetilde{D}_x),D) = \frac{1}{|S_D|}\cdot\sum_{i\in S_D}\left(1-\min\left\{\frac{\widetilde{D}_x}{D(z_i)},\frac{D(z_i)}{\widetilde{D}_x}\right\}\right)\leq\sigma^2$$

However, by Claim 6.5 we know that with high probability Inequality (68) holds. And since in Step (5) we check that $c_x \geq \widetilde{D}_x$, and by assumption we know that $c_x < D(x)$, we get that these two inequalities contradict. More concretely, Inequality (68) implies $\Delta_{S,S_D}((\widetilde{D}_x),D) = \frac{1}{|S_D|}\cdot\sum_{i\in S_D}\left(1-\min\left\{\frac{\widetilde{D}_x}{D(z_i)},\frac{D(z_i)}{\widetilde{D}_x}\right\}\right)\geq\sigma$. And so, we conclude that assuming Steps (2) and (3) passed, then for every prover strategy P^*, with high probability either Step (4) or Step (5) fails, and the verifier rejects.

Next, we analyze the complexity of the protocol:

- **V's sample complexity.** The verifier takes samples as part of running the IPP protocol from Lemma 4.8, and in the Verified-Tagged-Sample Protocol from Theorem 3.9. In total, the first subprotocol requires $r=\Theta\left(N^f\right)=\Theta\left(N^{\frac{1}{2}\left(1+\frac{c}{c+1}\right)}\right)=\Theta\left(N^{1-\frac{1}{2(c+1)}}\right)$ samples, and the second requires $\widetilde{O}\left(N^{1/2}\cdot\sigma^{-c}\right)=\widetilde{O}\left(N^{1/2}\cdot\left(\frac{1}{u^2}\right)^{-c}\right)=\widetilde{O}\left(N^{1/2}\cdot N^{\frac{c}{2(c+1)}}\right)=\widetilde{O}\left(N^{1-\frac{1}{2(c+1)}}\right)$.

- **Communication complexity and V's runtime.** The communication is dominated by the communication of the two sub-protocols. By Lemma 4.8, the communication complexity of the first subprotocol is $\widetilde{O}\left(r\cdot u+M/u\right)=\widetilde{O}\left(N^{1-\frac{1}{4(c+1)}}\right)$, which also dominates the communication complexity of the second protocol. The verifier runtime matches the communication complexity (the most computationally taxing task is to read the prover's message).

- **Round count.** The number of rounds is dominated by the IPP subprotocol, and is $\mathsf{polylog}(N)$

- **Double Efficiency.** The honest prover in Protocol 6.0.1 has to run the prover of Protocol 4.7.1 as well as the prover in the Verified Tagged Sample protocol from Theorem 3.9, both of which are doubly efficient. This makes Protocol 6.0.1 doubly efficient as well.

□

7 Domain Reduction

We show a domain reduction technique that takes a distribution D over a huge domain $\mathcal{U}$, whose support is of size at most N, and produces a distribution D' over domain $[M]$ for $M\approx(N\cdot\log|\mathcal{U}|)$, where D' "encodes" most of the information about D. Indeed, a complete representation of D can be (approximately) reconstructed from the complete representation of D' by a uniform low-depth circuit. This reconstruction procedure is distance-preserving, in the sense that any distribution that is close to D' is reconstructed to a distribution that is close to D (or rejected). Finally, we can generate a sample from D' by post-processing a (single) sample from D. Thus, we can reduce from verifying a property of D (over a huge domain) to verifying a property of D' (over a manegable domain), and use the protocol of Theorem 1.1 for this latter task.

Domain Reduction

Parameters. a data universe $\mathcal{U} = \{0,1\}^u$, support size $S \in \mathbb{N}$, error parameter $\varepsilon \in (0,1)$. Take $\mathcal{V} = \{0,1\}^v$ to be a hash range of size $O(S/\varepsilon)$, and $\mathrm{ECC} : \{0,1\}^u \to \{0,1\}^{u/\eta}$ to be an error-correcting code with constant rate η. The reduction's target domain is of size $M = O(S \cdot \log|\mathcal{U}|/\varepsilon)$

Key generator. $\mathrm{Gen}(u,s)$ outputs a hash function $h : \mathcal{U} \to \mathcal{V}$ from a pairwise independent hash family.

Domain reducer. Reduce_h gets as input an element $x \in \mathcal{U}$ and h generated by Gen. It outputs $y \in [M]$:

1. draw a uniformly random $j \in [(u/\eta)]$.
2. output $y = (h(x), j, \mathrm{ECC}(x)_j)$.

We think of $\mathrm{Reduce}_h(D)$ as a distribution over tuples in $\mathcal{V} \times [(u/\eta)] \times \{0,1\}$.

Reconstruction. $\mathrm{Reconst}_h$ gets as input a complete description of a distribution F over $[M]$ with support size at most S (as a list of the elements and their probabilities) and either rejects or outputs a complete description of a distribution over $\mathcal{U}$ (also of support size at most S).

1. For an element $z \in \mathcal{V}$ that appears in a tuple in F's support, let $p_z > 0$ be the total probability that F assigns to tuples that begin with z. For each such z, verify that for all $j \in [(u/\eta)]$, the probabilities of the prefixes (z,j) are identical, i.e. that $\{F(z,j,0) + F(z,j,1)\}$ are all the same. Reject otherwise.
2. For each $z \in \mathcal{V}$ in F's support say that z has a *unique answer* $w \in \{0,1\}^{(u/\eta)}$ if for each $j \in [(u/\eta)]$, F assigns non-zero probability to (z,j,w_j) and zero probability to $(z,j,(1-w_j))$. Verify that the sum of probabilities p_z for z's that do not have unique answers is at most $\Theta(\varepsilon)$. Reject otherwise.
3. For each $z \in \mathcal{V}$ in F's support that has a unique answer w, decode the vector $w \in \{0,1\}^{u/\eta}$ using the code ECC and let $x \in \mathcal{U}$ be the resulting decoded element, reject if the decoding failed. Assign probability p_z to x in the list of elements in the support of the reconstructed distribution.

Protocol 7.0.1: Domain Reduction

Taking ε to be an error parameter, the construction hashes the elements of $\mathcal{U}$ into a domain $\mathcal{V}$ of size (N/ε) using a pairwise independent hash function h. For $x \sim D$, taking $z = h(x)$ we "encode" the information about x as a collection of tuples with prefix z in the support of D'. To do so, we take $w = \mathrm{ECC}(x)$ to be an error correcting encoding of x, and output (z,j,w_j), where j is a random index in the encoding. This defines the distribution D', and note that it is over a domain of size $O((N/\varepsilon) \cdot \log(|\mathcal{U}|))$ (we take ECC to be a good error correcting code, with constant rate and distance). Observe that D' assigns to the set of tuples with prefix $z = h(x)$ total probability exactly $D[x]$, and we can recover x from these tuples. Further, changing the distribution of D' over elements with prefix z so that the reconstructed element is $x' \neq x$ requires making large changes to the probabilities of such tuples in D' (because of the encoding). The full statement and details are below.

Lemma 7.1. *Let $\mathcal{U} = \mathcal{U}(n)$ be a data universe, $S = S(n) \leq |\mathcal{U}|$ a bound on the support size, and $\varepsilon = \varepsilon(n), \sigma = \sigma(n) \in (0,1)$ be error parameters. Let* ECC *be a binary error correcting code with constant rate η and constant distance. The construction of Figure 7.0.1 reduces a domain of size $|\mathcal{U}|$ to a range of size $M = O(S \cdot \log|\mathcal{U}|/\varepsilon)$. For an appropriate choice of the code, the algorithms* Gen*,* Reduce *and* Reconst *can all be implemented by logspace uniform* NC^1 *circuits. For every distribution D over $\mathcal{U}$ they guarantee the following:*

- ***Reconstructability:*** *with 0.99 probability over $h \leftarrow \mathrm{Gen}(1^n)$,*

$$\Delta\left(D, \mathrm{Reconst}_h(\mathrm{Reduce}_h(D))\right) \leq O\left(\varepsilon\right)$$

- ***Distance preservation:*** *with 0.99 probability over $h \leftarrow \mathrm{Gen}(1^n)$, for every distribution F over the range $[M]$ of* Reduce *simultaneously , if $\Delta\left(\mathrm{Reduce}_h(D), F\right) \leq \sigma$ then either* $\mathrm{Reconst}_h(F)$ *rejects or it holds that*

$$\Delta\left(D, \mathrm{Reconst}_h(F)\right) = O(\sigma + \varepsilon).$$

Proof. We show that the domain reduction in Figure 7.0.1 satisfies the conditions of Lemma 7.1. We begin by showing *reconstructability*: fix a distribution D over $\mathcal{U}$ such that $|\mathrm{Supp}(D)| \leq S$. Let $x \in \mathrm{Supp}(D)$ be such that for all $x' \in \mathrm{Supp}(D)$ such that $x \neq x$, $h(x) \neq h(x')$. Then, by construction, for every $j \in [(u/\eta)]$, the probability of $(h(x), j, \mathrm{ECC}(x)_j) = \frac{D(x)}{u/\eta}$, and by assumption $(h(x), j, 1 - \mathrm{ECC}(x)_j) = 0$. Therefore, $\mathrm{Reconst}_h(\mathrm{Reduce}_h(D))(x) = D(x)$. We are left to show that the mass of D lost to collisions under h is small. Indeed, for every $x \neq x' \in \mathrm{Supp}(D)$, since h was drawn from a pairwise uniform family, it holds that $\Pr_h\left(h(x) = h(x')\right) = O\left(\frac{1}{S/\varepsilon}\right)$, and so, for every x, id we denote I_x be the indicator of the event that there exists some x' for which $h(x) = h(x')$, the $\mathbb{E}\left[I_x\right] \leq |\mathrm{Supp}(D)| \cdot O\left(\frac{\varepsilon}{S}\right) = O(\varepsilon)$, where the last inequality is due to the assumption that $|\mathrm{Supp}(D)| \leq S$. Consider $L = \sum_{x \in \mathrm{Supp}(D)} D(x) \cdot I_x$ to be the random variable representing the mass of D on elements colliding under h. From the above it holds that $\mathbb{E}\left[L\right] = O(\varepsilon)$, therefore, by Markov, with probability at most 0.99, it holds that $L = O(\varepsilon)$. We thus conclude that with high probability over h, the algorithm $\mathrm{Reconst}_h$ does not reject $\mathrm{Reduce}_h(D)$ and for all $x \in \mathrm{Supp}(D)$, but for an $O(\varepsilon)$-fraction of the mass, it holds that $D(x) = \mathrm{Reconst}_h(\mathrm{Reduce}_h(D))(x)$.

Next, we show that the *distance preservation* property holds as well. Let F be a distribution over $[M]$ such that $\Delta\left(\mathrm{Reduce}_h(D), F\right) \leq \sigma$. Assume distribution F is not rejected by the $\mathrm{Reconst}_h$ algorithm. This means that: (i) for every $z \in \mathcal{V}$, and every $j, j' \in [(u/\eta)]$, the probability that $F(z, j, 0) + F(z, j, 1) = F(z, j', 0) + F(z, j', 1)$; (ii) for all but $O(\varepsilon)$ of the mass of F, $z \in \mathcal{V}$ has a *unique answer* as defined in the reconstruction algorithm.

Let $z \in \mathcal{V}$ be such that z has a *unique answer*, denote this by u_z^F. Assume first that for some j, b, $(z, j, b) \in \mathrm{Supp}(D)$ such that z has a *unique answer* according to D as well, denoted by u_z^D. Note that:

$$\mathrm{Reduce}_h(D)\left(\left\{(z, j, b)\right\}_{j,b}\right) = \mathrm{Reconst}_h(\mathrm{Reduce}_h(D))(u_z^D)$$

$$F\left(\left\{(z, j, b)\right\}_{j,b}\right) = \mathrm{Reconst}_h(F)(u_z^F)$$

If $u_z^D = u_z^F = u$:

$$|\mathrm{Reconst}_h(F)(u) - \mathrm{Reconst}_h(\mathrm{Reduce}_h(D))(u)| = \sum_{j \in (u/\eta), b \in \{0,1\}} |\mathrm{Reduce}_h(D)\left(z, j, b\right) - F\left(z, j, b\right)|$$

Otherwise, it holds that $u_z^D \neq u_z^F$. Denote the distance of the ECC by constant δ, then, by construction, $\mathrm{Ham}\left(\mathrm{ECC}(u_z^D), \mathrm{ECC}(u_z^F)\right) \geq \delta$, and so, for at least $\delta \cdot (u/\eta)$ entries in the set $\left\{(z, j, b) : j \in \left[(u/\eta)\right], b \in \{0, 1\}\right\}$ it holds that $F(z, j, b) > 0$ and $\mathrm{Reduce}_h(D)(z) = 0$ or otherwise. Moreover, since z an a unique answer for both distributions, it holds that that if $F(z, j, b) > 0$, then

$F(z,j,b) = \frac{p^F(z)}{u/\eta}$, and if $\mathrm{Reduce}_h(D)(z,j,b) > 0$ then $\mathrm{Reduce}_h(D)(z,j,b) = \frac{p^D(z)}{u/\eta}$. This implies that:

$$\begin{aligned}\left|\mathrm{Reconst}_h(F)(u_z^F) - \mathrm{Reconst}_h(\mathrm{Reduce}_h(D))(u_z^F)\right| &+ \left|\mathrm{Reconst}_h(F)(u_z^D) - \mathrm{Reconst}_h(\mathrm{Reduce}_h(D))(u_z^D)\right| \\ &= \mathrm{Reconst}_h(F)(u_z^F) + \mathrm{Reconst}_h(\mathrm{Reduce}_h(D))(u_z^D) \\ &\leq \frac{1}{\delta} \sum_{j \in (u/\eta), b \in \{0,1\}} \left|\mathrm{Reduce}_h(D)\left(z,j,b\right) - F\left(z,j,b\right)\right|\end{aligned}$$

For every other $z \in \mathcal{V}$ such that it's outside the support of one distribution and has a unique answer in the other it follows that:

$$\begin{aligned}\left|\mathrm{Reconst}_h(F)(u_z^F) - \mathrm{Reconst}_h(\mathrm{Reduce}_h(D))(u_z^F)\right| &+ \left|\mathrm{Reconst}_h(F)(u_z^D) - \mathrm{Reconst}_h(\mathrm{Reduce}_h(D))(u_z^D)\right| \\ &= \mathrm{Reconst}_h(F)(u_z^F) + \mathrm{Reconst}_h(\mathrm{Reduce}_h(D))(u_z^D) \\ &= \sum_{j \in (u/\eta), b \in \{0,1\}} \left|\mathrm{Reduce}_h(D)\left(z,j,b\right) - F\left(z,j,b\right)\right|\end{aligned}$$

Therefore, setting $\mathtt{BAD}$ to be the set of (z,j,b) that don't have a *unique answer* according to either $\mathrm{Reconst}_h(D)$ or F, then:

$$\begin{aligned}\sum_{z:(z,j,b)\notin \mathtt{BAD}} \Big(\left|\mathrm{Reconst}_h(F)(u_z^F) \right. &- \left.\mathrm{Reconst}_h\left(\mathrm{Reduce}_h(D)\right)\left(u_z^F\right)\right| \quad &(75)\\ &+ \left|\mathrm{Reconst}_h(F)(u_z^D) - \mathrm{Reconst}_h\left(\mathrm{Reduce}_h(D)\right)\left(u_z^D\right)\right| \Big) \quad &(76)\\ &\leq \frac{1}{\delta} \sum_{(z,j,b)\notin \mathtt{BAD}} \left|\mathrm{Reduce}_h(D)\left(z,j,b\right) - F\left(z,j,b\right)\right| \quad &(77)\\ & &(78)\end{aligned}$$

We are thus left to bound the sum where $z : (z,j,b) \in \mathtt{BAD}$. Note that for these sets if we denote $B_D \subseteq \mathcal{V}$ the set of all elements z with unique answer according to $\mathrm{Reduce}_h(D)$ but not according to F, and B_F in the same vein w.r.t. to F, then:

$$\begin{aligned}\sum_{z \in B_D} \mathrm{Reconst}_h(\mathrm{Reduce}_h(D))(u_z^D) &+ \sum_{z \in B_F} \mathrm{Reconst}_h(\mathrm{Reduce}_h(D))(u_z^F) \quad &(79)\\ &\leq \sum_{(z,j,b)\in \mathtt{BAD}} \left|\mathrm{Reduce}_h(D)\left(z,j,b\right) - F\left(z,j,b\right)\right| + 2\varepsilon \quad &(80)\end{aligned}$$

We thus conclude that:

$$\Delta\left(D, \mathrm{Reconst}_h(F)\right) \leq \frac{1}{\delta}\sigma + 2\varepsilon = O(\sigma) + 2\varepsilon).$$

□

The following claim is follows directly from Lemma 7.1.

Claim 7.2. *Let $\mathcal{P} = (\mathcal{P}_N)$ be a distribution property over a large domain $\mathcal{U} = (\mathcal{U}_N)$. Assume $\mathcal{P}$ is $\rho = \rho(N)$-approximately decidable by a logspace uniform NC^1 family of circuits. Let* Gen*,* Reduce*,* Reconst *be as in the construction of Figure 7.0.1 with parameters as in Lemma 7.1, taking* $\varepsilon = \Theta(\rho)$*. There exists a logspace uniform* NC^1 *family of circuits* $(C'_N)_N$ *that take as input an explicit description of a distribution over* $[M]$ *and the function* h*. For a fixed distribution* D *over* $\mathcal{U}$ *with support of size at most* N*, with high probability over* $h \leftarrow \mathrm{Gen}$*:*

- *If* $\delta_{TV}(D, \mathcal{P}_N) \leq \varepsilon_c$*, then* $\mathrm{Reduce}_h(D)$ *satisfies the circuit* $C'_N(h, \cdot)$ *(the circuit with the function* h *hardwired into it).*
- *If* $\delta_{TV}(D, \mathcal{P}_N) \geq \varepsilon_f$*, then* $\mathrm{Reduce}_h(D)$ *is* $\left(\varepsilon_f - \frac{\rho}{2}\right)$*-far from any distribution* D' *that satisfies* $C'_N(h, \cdot)$*.*

Moreover, black box sample access to D *can be used to simulate black box sample access to* $\mathrm{Reduce}_h(D)$ *(each sample from* D *generates a single sample from* $\mathrm{Reduce}_h(D)$*).*

Remark 7.3. *In Theorem 1.1 we require the property over the small domain to be approximately decidable: the approximate decision circuit should work for* any *distribution. In Claim 7.2, we show that the approximate decision condition holds (w.h.p) for the specific distribution* $\mathit{Reduce}_h(D)$ *under consideration. This suffices for guaranteeing that running the protocol of Theorem 1.1 will result in* $\mathrm{Reduce}_h(D)$ *being accepted (in the completeness case) or rejected (in the soundness case). We obtain a complete and sound protocol for the original property* $\mathcal{P}$ *over the huge domain.*

References

[ABR16] Maryam Aliakbarpour, Eric Blais, and Ronitt Rubinfeld. Learning and testing junta distributions. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 19–46. JMLR.org, 2016.

[AGRR23] Hugo Aaronson, Tom Gur, Ninad Rajgopal, and Ron Rothblum. Distribution-free proofs of proximity. *Electron. Colloquium Comput. Complex.*, TR23-118, 2023.

[BGH$^+$06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust pcps of proximity, shorter pcps, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006.

[BKR04] Tugkan Batu, Ravi Kumar, and Ronitt Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 381–390. ACM, 2004.

[CG18] Alessandro Chiesa and Tom Gur. Proofs of proximity for distribution testing. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 53:1–53:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[EKR04] Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate probabilistically checkable proofs. *Inf. Comput.*, 189(2):135–159, 2004.

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.

[GJK$^+$24] Tom Gur, Mohammad Mahdi Jahanara, Mohammad Mahdi Khodabandeh, Ninad Rajgopal, Bahar Salamatian, and Igor Shinkar. On the power of interactive proofs for learning. In *STOC '24: 56th Annual ACM SIGACT Symposium on Theory of Computing (to appear), Vancouver, Canada, June 24-28, 2024*. ACM, 2024.

[GKR15] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *J. ACM*, 62(4):27:1–27:64, 2015.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In Robert Sedgewick, editor, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 291–304. ACM, 1985.

[Gol17] Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.

[GR18] Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. *Comput. Complex.*, 27(1):99–207, 2018.

[GR21] Oded Goldreich and Dana Ron. A lower bound on the complexity of testing grained distributions. *Electron. Colloquium Comput. Complex.*, page 129, 2021.

[GR22] Guy Goldberg and Guy N. Rothblum. Sample-based proofs of proximity. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 77:1–77:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[GRS23] Oded Goldreich, Guy N. Rothblum, and Tal Skverer. On interactive proofs of proximity with proof-oblivious queries. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 59:1–59:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[GRSY21] Shafi Goldwasser, Guy N. Rothblum, Jonathan Shafer, and Amir Yehudayoff. Interactive proofs for verifying machine learning. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 41:1–41:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[HR22] Tal Herman and Guy N. Rothblum. Verifying the unseen: interactive proofs for label-invariant distribution properties. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1208–1219. ACM, 2022.

[HR23] Tal Herman and Guy N. Rothblum. Doubley-efficient interactive proofs for distribution properties. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 743–751. IEEE, 2023.

[HV06] Alexander Healy and Emanuele Viola. Constant-depth circuits for arithmetic in finite fields of characteristic two. In Bruno Durand and Wolfgang Thomas, editors, *STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science, Marseille, France, February 23-25, 2006, Proceedings*, volume 3884 of *Lecture Notes in Computer Science*, pages 672–683. Springer, 2006.

[KNR09] Eyal Kaplan, Moni Naor, and Omer Reingold. Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica*, 55(1):113–133, 2009.

[LR88] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.

[NR99] Moni Naor and Omer Reingold. On the construction of pseudorandom permutations: Luby-rackoff revisited. *J. Cryptol.*, 12(1):29–66, 1999.

[PRR06] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *J. Comput. Syst. Sci.*, 72(6):1012–1042, 2006.

[RR20] Guy N. Rothblum and Ron D. Rothblum. Batch verification and proofs of proximity with polylog overhead. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part II*, volume 12551 of *Lecture Notes in Computer Science*, pages 108–138. Springer, 2020.

[RR22] Noga Ron-Zewi and Ron D. Rothblum. Proving as fast as computing: succinct arguments with constant prover overhead. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1353–1363. ACM, 2022.

[RRR16] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 49–62. ACM, 2016.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

[RS05] Ronitt Rubinfeld and Rocco A. Servedio. Testing monotone high-dimensional distributions. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 147–156. ACM, 2005.

[RV20] Ronitt Rubinfeld and Arsen Vasilyan. Monotone probability distributions over the boolean cube can be learned with sublinear samples. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 28:1–28:34. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[RVW13] Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 793–802. ACM, 2013.

A Bounded-Space Distribution Properties

We also get a result in the spirit of Theorem 1.1 for distribution properties that can be approximately decided by a polynomial-time and bounded-polynomial-space Turing machine (the machine gets as input an explicit description of the distribution, as in Theorem 1.1).

Theorem A.1 (IPs for space-bounded properties)**.** *There exist constants $\alpha, \sigma > 0$ s.t. for every approximation parameter $\rho = \rho(N) \in (0,1)$, and every property that can be $\rho(N)$-approximately decided by a polynomial-time Turing machine that runs in space $O(N^{\sigma})$, there is an interactive proof system as follows. The prover and the verifier both get as input an integer N and proximity parameters $\varepsilon_c, \varepsilon_f \in [0,1]$ s.t. $\varepsilon_f - \varepsilon_c \geq \Theta(\rho)$, as well as sampling access to a distribution D over the domain $[N]$, where*

- *Completeness: if D is ε_c-close to the property and the prover follows the protocol, then the verifier accepts w.h.p.*
- *Soundness: if D is ε_f-far from the property, then, no matter how the prover cheats, the verifier rejects w.h.p.*
- *Efficient verification: the verifier's sample complexity is $\widetilde{O}(N^{1-\alpha} \cdot \mathsf{poly}(1/\rho))$. The communication complexity and verifier runtime are $(\widetilde{O}(N^{1-\alpha}) \cdot \mathsf{poly}(1/\rho))$. The protocol has a constant number of rounds.*
- *Doubly-efficient prover: the honest prover's sample complexity is $\widetilde{O}(N) \cdot \mathsf{poly}(1/\rho)$ and its runtime is $\mathsf{poly}(N, 1/\rho)$.*

The protocol is identical to the protocol underlying Theorem 1.1. The only difference is that instead of using an IPP that builds on the GKR protocol [GKR15] for bounded-depth computations (i.e., the IPP described in Theorem 2.1), we use an IPP that builds on the RRR protocol [RRR16]. See the statements about these IPP in the work of Rothlbum and Rothblum [RR20] for further details. The exponent α that bounds the sample and communication complexities is identical to Theorem 1.1. The exponent σ in the bound on the space of the Turing machine is derived from α to ensure that the $\mathsf{poly}(N^{\sigma})$ complexity of the RRR protocol is $O(N^{1-\alpha})$.