Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 118 (2023)

Distribution-Free Proofs of Proximity

Hugo Aaronson[1], Tom Gur[1], Ninad Rajgopal[1], and Ron D. Rothblum[2]

[1]University of Cambridge, ha406@cam.ac.uk, tom.gur@cl.cam.ac.uk,nr549@cam.ac.uk
[2]Technion, rothblum@cs.technion.ac.il

Abstract

Motivated by the fact that input distributions are often unknown in advance, distribution-free property testing considers a setting in which the algorithmic task is to accept functions $f : [n] \to \{0, 1\}$ having a certain property Π and reject functions that are ε-far from Π, where the distance is measured according to an arbitrary and unknown input distribution $\mathcal{D} \sim [n]$. As usual in property testing, the tester is required to do so while making only a sublinear number of input queries, but as the distribution is unknown, we also allow a sublinear number of samples from the distribution $\mathcal{D}$.

In this work we initiate the study of *distribution-free interactive proofs of proximity* (df-IPPs) in which the distribution-free testing algorithm is assisted by an all powerful but untrusted prover. Our main result is that for any problem $\Pi \in \mathsf{NC}$, any proximity parameter $\varepsilon > 0$, and any (trade-off) parameter $\tau \leq \sqrt{n}$, we construct a df-IPP for Π with respect to ε, that has query and sample complexities $\tau + O(1/\varepsilon)$, and communication complexity $\tilde{O}(n/\tau + 1/\varepsilon)$. For τ as above and sufficiently large ε (namely, when $\varepsilon > \tau/n$), this result matches the parameters of the best-known general purpose IPPs in the standard uniform setting. Moreover, for such τ, its parameters are optimal up to poly-logarithmic factors under reasonable cryptographic assumptions for the same regime of ε as the uniform setting, i.e., when $\varepsilon \geq 1/\tau$.

For smaller values of ε (i.e., when $\varepsilon < \tau/n$), our protocol has communication complexity $\Omega(1/\varepsilon)$, which is worse than the $\tilde{O}(n/\tau)$ communication complexity of the uniform IPPs (with the same query complexity). With the aim of improving on this gap, we further show that for IPPs over specialised, but large distribution families, such as sufficiently smooth distributions and product distributions, the communication complexity can be reduced to $\tilde{O}(n/\tau^{1-o(1)})$. In addition, we show that for certain natural families of languages, such as symmetric and (relaxed) self-correctable languages, it is possible to further improve the efficiency of distribution-free IPPs.

ISSN 1433-8092

Contents

1 Introduction

Property Testing, initiated in [RS96, GGR98], is a rich and well-studied research field lying at the heart of many advancements in sublinear algorithms and complexity theory; see [Gol17, BY22] for a detailed introduction. Loosely speaking, a testing algorithm for a property Π is given oracle access to an input function $f : [n] \to \{0, 1\}$ and should decide whether $f \in \Pi$ using a small *sublinear* number of queries. As we cannot expect to do so exactly, the tester is required to distinguish between inputs that are in Π from those that are ε-far from every function in Π. Here, distance is typically measured using the relative Hamming distance – namely, the fraction of outputs of f that need to be changed to reach a member of Π.

While modeling distance using the relative Hamming distance is natural and convenient, in many settings it may not capture the underlying question (for example, when functions always satisfy a particular format or when some parts in the domain are more important than others). Following the Probably-Approximately-Correct (PAC) learning model, introduced by Valiant in his celebrated work in computational learning theory [Val84], *distribution-free* algorithms have widely been accepted as a closer abstraction of real-world computational tasks that are required to make decisions based on limited access to the input data. In this spirit, [GGR98] introduced *distribution-free property testing*, where the distance between two functions is with respect to a distribution $\mathcal{D}$ (over inputs to the function), which is *arbitrary* and *unknown* to the testing algorithm. Since $\mathcal{D}$ is unknown, in addition to the query oracle to the input $f : [n] \to \{0, 1\}$, the tester can draw independent identically distributed random labelled samples $(i, f(i))$ from a *sample oracle*, where each index i is generated independently from the distribution $\mathcal{D}$. The tester is required to reject any function that is ε-far[1] from Π along the unknown distribution $\mathcal{D}$, and the only access that the tester has to $\mathcal{D}$ is via the sample oracle.

The distribution-free model of testing naturally complements the PAC-learning model, and profound bidirectional connections are known between them.[2] Moreover, distribution-free testing is motivated by the fact that it captures the realistic setting where the tester is required to maintain its guarantees despite dealing with data from an unknown environment (i.e., via data samples from some unknown and arbitrary distribution $\mathcal{D}$). It also deals with situations where not all underlying data points are equally important, e.g., in graphs where certain edges or vertices are more important than others, and one would like to consider distributions that weigh them appropriately.

Following [GGR98], several distribution-free testing algorithms have been designed for function classes including monotone Boolean functions and low-degree polynomials over finite fields [HK07], k-juntas [LCS$^+$18, Bsh19, Bel19], conjunctions (monotone or non-monotone) and linear threshold functions [GS07, CX16], polynomial threshold functions and decision trees [BFH21], halfspaces [BFH21, CP22], and low-degree polynomials on $\mathbb{R}^n$ [FY20, ABF$^+$23]. Distribution-free testing has also been studied for graph properties including connectivity [HK08], bipartiteness [Gol19a], k-path and degree regularity [Gol19b], as well as for word problems like subsequence-freeness [RR22].

Despite such strides of progress, our understanding of distribution-free testing is much more limited than that of testing with respect to the uniform distribution. This is due to the multitude of challenges that arise in designing algorithms that need to deal with data samples that can come

[1] We say $f : [n] \to \{0, 1\}$ is ε-far from a (non-empty) property Π along $\mathcal{D}$, if for every $f' : [n] \to \{0, 1\}$ such that $f' \in \Pi$, it holds that $\mathbb{P}_{i \sim \mathcal{D}}[f(i) \neq f'(i)] > \varepsilon$.

[2] In particular, in [GGR98], it is shown that if a class of functions $\mathcal{C}$ has a *proper* PAC-learner using membership queries (where the learner outputs an approximate hypothesis that also belongs to $\mathcal{C}$), then $\mathcal{C}$ has a distribution-free tester that uses roughly the same number of queries and samples as the learner.

from any arbitrary distribution, which in turn, makes the model significantly more involved.

This paper aims to bridge the gap between testing over the uniform distribution and distribution-free testing by capitalising on the power of interactive proofs, and delegating the task of handling the challenges imposed by the distribution-free setting to a powerful, but untrusted, prover.

1.1 Distribution-free Interactive Proofs of Proximity

In this work, we initiate the study of *distribution-free interactive proofs of proximity* (distribution-free IPPs), which are distribution-free testers that are augmented with the help of a prover. In the rest of this paper, for convenience, rather than thinking of the input as a function, we view it as a string $x \in \{0,1\}^n$ (which can be similarly be viewed as a truth table of a function $f_x : [n] \to \{0,1\}$). Correspondingly, we view a property Π of functions as a language L over strings (which may be viewed as truth tables of the functions in Π).

Thus, distribution-free IPPs are protocols where a *sublinear* time, randomised algorithm, called the verifier, interacts with an untrusted prover to decide whether the given input $x \in \{0,1\}^n$ belongs to the language L or is far from such, where distance is measured with respect to a fixed, but unknown distribution $\mathcal{D}$ over $[n]$. The verifier is given access to the input x through a query oracle, as well as a sample oracle with respect to $\mathcal{D}$, while the prover can look at the input entirely. We assume that the prover does not know the queries that the verifier makes to either of its oracles.

We require that for any $x \in L$, there exists an honest prover that interacts with the verifier and convinces it to accept with high probability, while when x is ε-far from L with respect to the distribution $\mathcal{D}$, no cheating prover, even computationally unbounded, will make the verifier accept, except with low probability. Further, we require the distribution-free IPP to meet these requirements, with respect to the underlying (and unknown) distribution $\mathcal{D}$ from which the oracle draws samples.

In this setting, the verifier's *query complexity* and *sample complexity*, the number of bits exchanged in the protocol, i.e., the *communication complexity*, and the verifier's running time should all be sublinear in input length. Other complexity parameters of interest are the number of rounds of interaction, and the (honest) prover's running time.

Distribution-free IPPs capture the distribution-free property testing analogue of interactive proofs (for more information, see Section 1.4). As such, similar to uniform IPPs, distribution-free IPPs can be alternatively viewed as proof systems where the bounded verifier need only be convinced of the fact that the input is close to the language, by interacting with a more powerful prover. One of the main goals of distribution-free IPPs is to overcome the inherent limitations of distribution-free testing algorithms by showing that for certain properties, verifying proximity over arbitrary distributions is considerably faster with a prover than actually testing it. In particular, we want to design distribution-free IPPs (with sublinear query complexity) for rich families of properties that have no known distribution-free testers.

Of close relevance are the well-studied notion of IPPs over the uniform distribution, which we refer to in this work as Uniform IPPs, that were introduced in [EKR04, RVW13] (and are trivially generalised by distribution-free IPPs). Showcasing the power of interaction, [RVW13] constructed highly non-trivial uniform IPPs for every language that can be decided in bounded depth (e.g., NC), which was recently made near-optimal by [RR20] (see [KR15] for the conditional matching lower bound), and strengthened to encompass also bounded space languages [RRR21].

Motivated intrinsically and by natural applications to *delegation of computation*, the study of uniform IPPs has drawn much recent attention on its own right [RVW13, GR18, KR15, RRR21, GG21, GR22]. Moreover, their study has led to interesting models and applications of sublinear time verification, including non-interactive proofs of proximity (or MAPs) [GR18] (a related model was studied concurrently and independently by [FGL14]), arguments of proximity [KR15], testing properties of distributions [CG18, HR22], interactive oracle proofs of proximity [RRR21, BBHR18, RZR20, BLNR22], verifying machine learning tasks [GRSY21], batch verification for UP [RRR18, RR20], as well as variants involving zero-knowledge [BRV18] and quantum computation [DGMT22].

1.2 Our Results

Our main contribution is constructing distribution-free IPPs for any language in NC, which for any query vs communication trade-off parameter $\tau \leq \sqrt{n}$, matches the complexity of the best known IPPs for most settings of the proximity parameter ε – specifically, when $\varepsilon \geq \tau/n$. We further improve the efficiency of distribution-free IPPs for general ε (i.e., when $\varepsilon < \tau/n$), under specific distribution families such as "smooth" and "learnable" distributions, which are defined below.

In addition, for certain families of languages, such as symmetric and relaxed self-correctable languages, we construct distribution-free IPPs that improve on our general-purpose distribution-free IPPs, then use them to provide separation results that provide further insight into the distribution-free IPP model.

We elaborate on these results next.

1.2.1 Distribution-free IPPs for NC

Our first main result is a sublinear distribution-free IPP for any language computable by low-depth circuits. In more detail, let (logspace-uniform) NC be the set of languages computable by (logspace-uniform) Boolean circuits of polynomial size and poly-logarithmic depth. We show that every language in NC has a distribution-free IPP with sublinear complexity measures, for almost all values of the proximity parameter ε. We emphasize that this is in stark contrast to distribution-free testers, which are only known for a handful of languages based on their combinatorial or algebraic structure. Indeed, the following theorem shows that distribution-free IPPs capture a much richer class of languages that need not have such special structural properties.

Theorem 1.1 (**Distribution-Free IPP for NC**)**.** *For every language L in logspace-uniform NC and every trade-off parameter $\tau = \tau(n) \leq \sqrt{n}$, there exists a distribution-free IPP for L with proximity parameter $\varepsilon \geq \Omega\left(\frac{\log^3(n)}{n}\right)$, query complexity $\tau + O\left(\frac{1}{\varepsilon}\right)$, sample complexity $\tau + O\left(\frac{1}{\varepsilon}\right)$ and communication complexity $\tilde{O}\left(\frac{n}{\tau} + \frac{1}{\varepsilon}\right)$.*

Moreover, the verifier runs in time $\tilde{O}\left(\frac{n}{\tau} + \frac{1}{\varepsilon}\right)$, the prover runs in time $\mathsf{poly}(n)$ and the round complexity is $\mathsf{polylog}(n)$.

Here, τ denotes the parameter that trades-off between the query and communication complexities of the distribution-free IPP. Note that, for the above values of τ, our distribution-free IPP has sublinear query and communication complexity even for very small values of the proximity parameter ε of the form $1/n^{1-\delta}$, where $\delta > 0$. An interesting instantiation of our result is obtained by setting τ to $\sqrt{n}$, and thus, for every $\varepsilon \geq 1/\sqrt{n}$, the query complexity and sample complexities are $O(\sqrt{n})$, while the communication complexity and verifier running times are both $\tilde{O}(\sqrt{n})$.

It is worth noting that, for every $\varepsilon \geq \frac{1}{\tau}$ (and $\tau \leq \sqrt{n}$), this result is conditionally optimal up to poly-logarithmic factors, since [KR15] show a lower bound of $\Omega(n)$ on the product of the query and communication complexities of a uniform IPP for a language in NC^1, under a strong, but reasonable, cryptographic assumption. Furthermore, for any ε, the query complexity of $\Omega(1/\varepsilon)$ is necessary for any IPP over non-degenerate languages, even over the uniform distribution (see [RVW13, Remark 1.2]).

Remark 1. *While Theorem 1.1 refers to distribution-free* IPP*s over* NC *languages, the theorem is more general (see Theorem 4.5). In particular, it also yields distribution-free* IPP*s with sublinear query and communication complexities for languages computable by circuits of sub-exponential size and bounded polynomial depth.*

Likewise, in a similar fashion to the known literature on uniform IPP*s, we can combine our techniques directly with [RRR21] to get a constant-round distribution-free* IPP *for any language that is computable in* $\mathsf{poly}(n)$ *time and bounded polynomial space.*

Comparison to Uniform IPPs for NC [RVW13, RR20]: For any language in NC, Rothblum, Vadhan and Wigderson [RVW13] construct a uniform IPP for any $\tau = \tau(n)$ and proximity parameter $\varepsilon > 0$, with query complexity $\tau + O(1/\varepsilon)^{1+o(1)}$ and communication complexity $\frac{n}{\tau^{1-o(1)}}$. Rothblum and Rothblum [RR20] improve on this, by reducing the communication complexity to $\frac{n}{\tau} \cdot \mathsf{polylog}(n)$. In particular, the latter obtains an optimal trade-off, up to poly-logarithmic factors, between the query and communication complexities of a uniform IPP (conditionally, from [KR15]), for every value of τ and $\varepsilon \geq 1/\tau$. While these results are stated in [RVW13, RR20] by implicitly setting $\tau = O(1/\varepsilon)$, for any given ε, this IPP formulation parameterised by τ is obtained by inspection (see also [GG21, Theorem 6.3]). For comparison, in this setting, our distribution-free IPP has the same query (and sample) complexity, while the communication complexity and verifier running times are both $\tilde{O}(\varepsilon \cdot n + 1/\varepsilon)$.[3]

Theorem 1.1 gives a construction of a *distribution-free* IPP for any NC language that matches the query and communication complexities of the uniform IPP by [RR20], when $\varepsilon \geq \tau/n$. Moreover, this obtains the (conditionally) optimal trade-offs between query and communication complexities in the *same regime* of ε, but when $\tau \leq \sqrt{n}$. Indeed, when $\varepsilon \geq 1/\tau$, the product of the query and communication complexities of the distribution-free IPP from Theorem 1.1 is $\tilde{O}(n + \tau^2)$. Our protocol builds on [RVW13], introducing new ideas that allow us to construct IPPs in the more involved distribution-free setting.

Finally, when the proximity parameter ε is very small, Theorem 1.1 suffers a blow-up in the communication complexity compared to the uniform IPPs of [RVW13, RR20]. In more detail, when $\varepsilon \ll \tau/n$, the communication complexity in our distribution-free IPP is $\tilde{\Omega}\left(\frac{1}{\varepsilon}\right)$, whereas the communication complexity achieved by the uniform IPPs is $\tilde{O}\left(\frac{n}{\tau}\right)$ (the query complexity roughly remains the same across all three cases). Thus, our distribution-free IPP has communication complexity at least $\Omega(n/\tau)$ for every value of ε, whereas the communication complexity of the uniform IPPs is much lower when $\varepsilon \ll \tau/n$.

[3] In fact, we prove that for every value of the parameter τ and ε, the distribution-free IPP from Theorem 1.1 has communication complexity $\tilde{O}(\tau + n/\tau + 1/\varepsilon)$; thus, setting $\tau = O(1/\varepsilon)$ suffices. An additional point to note is that when $\tau > \sqrt{n}$, the IPP always has worse communication complexity than its uniform counterpart irrespective of the value of ε, and further, never meets the optimal [KR15] lower bound. As such, we only consider $\tau \leq \sqrt{n}$ as a more interesting regime of study.

1.2.2 IPPs for NC: The case of small ε

Following the discussion in the last section, we aim to construct distribution-free IPPs that achieve query and communication complexities that match the state-of-the-art uniform IPP for every value of ε. While we unable to do so in the most general case, we construct such IPPs over *specific families of distributions*, which match the complexities of [RVW13] and, in turn, differ from the complexities of [RR20] only by a factor of $n^{o(1)}$. For these IPPs, while the underlying distribution is still unknown, it is guaranteed to come from the specific family of distributions under consideration.

To describe our results, it will be convenient throughout this section to identify $[n]$ with the elements of an m-dimensional tensor of size $k \in \mathbb{N}$ in each dimension, such that $k^m = n$. In such a case, we refer to $[n]$ as $[k]^m$ (by fixing some canonical bijection between them).

ρ-Dispersed Distributions: Intuitively speaking, ρ-dispersed distributions capture the sense that for a smooth distribution over $[k]^m$, along any dimension, its probability mass on any element in $[k]^m$ is not much larger than the average of the probability masses of its neighbours. ρ-dispersed distributions relax this requirement by having the probability mass on any element bounded by ρ times the expected mass on any of its neighbours.[4] We refer to Section 5.1 for the formal definition of ρ-dispersed distributions and classification of well-studied distributions.

We show that for distributions that are reasonably smooth in this sense, i.e. for ρ-dispersed distributions for $\rho \leq k^{o(1)}$, we obtain IPPs for NC over such distributions for every $\tau = \tau(n) < n$ and $\varepsilon > 0$, with query complexity $O(\tau + 1/\varepsilon)^{1+o(1)}$, and communication complexity of $\tilde{O}\left(\frac{n}{\tau} \cdot \tau^{o(1)}\right)$, thus matching the bounds obtained by [RVW13]. It is worth noting that $k^{o(1)}$-dispersed distributions are still quite general, e.g. any distribution where the probability mass on any element in $[k]^m$ is in the range $\left[\frac{1}{an}, \frac{a}{n}\right]$, for some $a \leq k^{o(1)}$ is $k^{o(1)}$-dispersed.

Theorem 1.2 (IPP for NC over ρ-dispersed distributions)**.** *For every language in logspace-uniform* NC*, every $m, n, k \in \mathbb{N}$ such that $m = \log_k(n)$ (i.e., $k^m = n$) and $\rho \in \mathbb{R}$ such that $\rho \leq k^{o(1)}$, for every proximity parameter $\varepsilon > 0$ and trade-off parameter $\tau > 0$, there exists an* IPP *over ρ-Dispersed distributions over $[k]^m$ with query and sample complexities $O(\tau + 1/\varepsilon)^{1+o(1)}$ and communication complexity $\tilde{O}\left(\frac{n}{\tau^{1-o(1)}}\right)$.*

Moreover, the verifier runs in time $n^{o(1)} \cdot \left(\tau + \frac{n}{\tau} + \frac{1}{\varepsilon}\right)$, the prover runs in time $\mathsf{poly}(n)$ *and the round complexity is* $\mathsf{polylog}(n)$*.*

Theorem 1.2 also holds generally over ρ-dispersed distributions, for any ρ (see Theorem 5.2). The query complexity increases with ρ, while the communication complexity is *independent of* ρ. Theorem 1.2 builds on the ideas used for the distribution-free IPP from Theorem 1.1 while incorporating new technical insights into the analysis by [RVW13] to generalise over ρ-dispersed distributions. We leave the task of obtaining IPPs over ρ-dispersed distributions that match [RR20] as future work.

Product Distributions in the White-Box model: Note that in the IPPs of Theorems 1.1 and 1.2, the verifier does not learn the underlying distribution $\mathcal{D}$. Hence, we ask the following

[4] For example, the uniform distribution is the only 1-dispersed distribution, i.e., a maximally smooth distribution in this sense. On the other hand, every distribution over $[k]^m$ is trivially a k-dispersed distribution.

question: if we could gain more information about $\mathcal{D}$, or further, learn a reasonably good approximation for $\mathcal{D}$, can we improve the query complexity of the IPPs, over general values of ε? We answer this question in the affirmative for product distributions.

We consider the *white-box model* for distribution-free IPPs, where the verifier receives a succinct description of the unknown distribution $\mathcal{D}$ over $[k]^m$ via a *polynomial-sized* sampling circuit C, in addition to query access to the input string. It is worth noting that, for white-box IPPs, the sample complexity is irrelevant since the verifier has a succinct description of the entire distribution. Thus, the main complexity parameters here are the query complexity, communication complexity, and the verifier running time.

While white-box models have been widely studied in the setting of zero-knowledge proofs [SV97, Vad99, Vad06] and in distribution testing (see survey by [GV11]), we use this model to construct IPPs for languages in NC over a generalised family of product distributions over $[k]^m$, to get improved complexities for general values of ε, compared to the distribution-free IPP from Theorem 1.1. We call this family as *m-product distributions*, and denote any such distribution $\mathcal{D}$ as $\mathcal{D} = \mathcal{D}_1 \times \ldots \mathcal{D}_m$, where each $\mathcal{D}_j$ is supported on $[k]$ and is independent of any other coordinate distributions. In particular, $\mathcal{D}(i_1, \ldots, i_m)$ is defined as $\prod_{j=1}^{m} \mathcal{D}_j(i_j)$ (see Definition 3.3 for more details about white-box IPPs and Definition 6.1 for product distributions).

Theorem 1.3 (**IPPs for NC over m-product distributions**). *For every language in logspace-uniform* NC*, every $\tau = \tau(n)$, $\varepsilon > 0$, and $m, n, k \in \mathbb{N}$ such that $m \leq \log(n)$ and $k^m = n$, there exists a white-box* IPP *for L over m-product distributions over $[k]^m$. The* IPP *has query complexity $O(\tau + 1/\varepsilon)^{1+o(1)}$ and communication complexity $\left(\frac{n}{\tau^{1-o(1)}} \cdot k + k^2\right) \cdot \mathsf{polylog}(n)$. Moreover, the verifier runs in time $n^{o(1)} \left(\frac{n}{\tau} \cdot k + \tau + k^2 + \frac{1}{\varepsilon}\right)$ and the round complexity is* polylog(n).

Similar to the previous results, a general version of this IPP can be found in Theorem 6.6. In particular, when m is large enough (like $m = \log(n)$), then the query and communication complexity trade-off, as well as the verifier running time of the IPP from Theorem 1.3 match that of the uniform IPP from [RVW13], while working in this setting.[5] Theorem 1.3 builds on the framework of Theorem 1.1, and uses several new ideas in the construction of the IPP, as well as its analysis, to improve the complexity. Crucially, it uses that any product distribution has a succinct description to be able to *learn* it in the white-box-setting.

It is worth stressing that the IPPs from Theorems 1.2 and 1.3 are incomparable. Indeed, there exist m-product distributions $\mathcal{D} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_m$ that are poorly dispersed, for eg., $\mathcal{D}$ is no longer smooth when some $\mathcal{D}_j$ has a large probability mass over just one element (one row or more generally, a few rows). For such distributions, the IPP from Theorem 1.3 provides a much better query and communication trade-off than the IPP from Theorem 1.2, which is a more general result for smooth distributions.

1.2.3 On the power of distribution-free IPPs

Recall that Theorems 1.2 and 1.3 improve the query and communication complexity trade-off of our general distribution-free IPP in Theorem 1.1, by considering special families of distributions to design the IPPs over. A natural direction that complements this approach is to ask whether we can

[5] A subtle point here is that while Theorem 1.3 is over product distributions over $[k]^m$, when $m = 2$ (or a small constant), we get sublinear complexities only by considering distributions over biased matrices $[k_1] \times [k_2]$.

use additional information about the *language* L instead, to construct super-efficient distribution-free IPPs.

In turn, we study distribution-free IPPs for specific problems of interest. On one hand, for certain problems we can hope to improve the various associated complexity parameters over our general distribution-free IPP by capitalising on the structure of the language. On the other hand, this allows us to obtain complexity-theoretic separations between the power of standard, non-interactive, and interactive distribution-free testers.

Symmetric languages. We study the power of distribution-free testers and IPPs for symmetric languages, which are languages that are invariant under permutations. We show that there exist symmetric languages that are hard for distribution-free testers, yet, given interaction with a prover, the symmetrical structure can be leveraged to obtain exponentially faster distribution-free IPPs.

Theorem 1.4 (**Distribution-free IPPs for symmetric languages**). *The following statements hold.*

1. *Let L be a symmetric language. Then, there exist a distribution-free* IPP *for L with sample complexity $O(1/\varepsilon)$, communication complexity $O(\log^2(n)/\varepsilon)$ and $O(\log(n)/\varepsilon)$ round complexity.*

2. *There exists a symmetric language L' for every $\varepsilon > 0$ such that any distribution-free property tester for L' requires $\Omega(n^{1/3-0.0005})$ queries and labeled samples from the input.*

(Relaxed) self-correctable languages. Next, we show that for languages that admit self-correctability, we can transform any IPP into a distribution-free IPP at a negligible cost. In fact, we can deal with a far more general class of languages; namely, languages that are *relaxed locally correctable* [BSGH+04, GRR20]. Loosely speaking, these are languages that admit a correcting algorithm that is required to correct the symbol at every location of the codeword, by reading a small number of locations in it, but is allowed to abort if noticing that the given word is corrupted. This family of languages is of central importance in the interactive proofs and probabilistically checkable proofs literature, and in particular, it captures languages of low-degree polynomials, holographic IPPs, and various relaxed locally correctable and decodable languages that were used to prove complexity-theoretic separations (cf. [Gur17]).

Proposition 1.5 (**Generic Transformations for IPPs for RLCCs**). *For any subset L of a binary RLCC, $C \subseteq \{0,1\}^n$, if L has an* IPP *over the uniform distribution with query complexity q and communication complexity c for proximity $\varepsilon > 0$, then there exists a distribution-free* IPP *for L with the same round complexity, communication complexity and query complexity $q + O(\frac{t}{\varepsilon})$, where t is the query complexity of the corrector of C.*

A detailed statement can be found in Theorem 3.1. As a corollary of Proposition 1.5, we are able to lift complexity-theoretic results concerning uniform IPPs to the setting of distribution-free IPPs. In particular, we obtain strong separations between the power of distribution-free testers, distribution-free non-interactive proofs of proximity (MAPs), and distribution-free IPPs.

Corollary 1.6 (**Complexity separations**). *There exists a language L such the following hold true.*

1. *Property Testing: The query complexity of distribution-free testing L (without a proof) is* $\Theta(n^{0.999\pm o(1)})$.

2. MAP*: L has a distribution-free* MAP *with query and communication complexities* $\Theta(n^{0.499\pm o(1)})$. *Moreover, for every $p \geq 1$, the distribution-free* MAP *query complexity of L with respect to proofs of length p is* $\Theta\big(\frac{n^{0.999\pm o(1)}}{p}\big)$.

3. IPP*: L has a distribution-free* IPP *with query and communication complexities* $\mathsf{polylog}(n)$.

See Theorem 3.2 for a more detailed statement. Complementing this Corollary, we prove the existence of languages that can be tested under the uniform distribution with low query complexity (and thus, have a uniform IPP with low query complexity and no communication), but for which distribution-free IPPs require large query complexity or large communication complexity. This illustrates the difficulty of constructing distribution-free IPPs vs. standard uniform IPPs.

Proposition 1.7 (Distribution-free IPPs vs. uniform testing)**.** *The following hold true:*

1. *There exists $\varepsilon > 0$ and a language L such that L has a property tester over the uniform distribution with query complexity $O(1/\varepsilon)$ for proximity parameter ε. However, for any distribution-free* MAP *for L with proximity parameter ε, query complexity q, and proof length p, $\max(q,p) = \Omega(\varepsilon \cdot n)$.*

2. *Assuming the existence of exponentially hard pseudo-random generators, there exists $\varepsilon > 0$ such that for all $q = q(n) \leq n$, there exists a language L, such that for any distribution-free* IPP *for L with proximity parameter ε, communication complexity c, and query complexity q, $\max(c,q) = \Omega(\sqrt{\varepsilon \cdot n})$. However, L has a uniform property tester with query complexity $O(1/\varepsilon)$ for proximity parameter ε.*

See Section 3.3 for more details. Table 1 provides a comparison of some of these results with related testing models. It is an interesting open direction to exhibit distribution-free IPPs that improve on the query complexity lower bounds known for distribution-testing functional properties like monotonicity [HK07], monotone conjunctions [CX16], or k-juntas [LCS$^+$19].

1.3 Technical Overview

In this technical overview, we highlight the proofs of Theorems 1.1, 1.2, and 1.3. The general strategy for proving these theorems builds on the Uniform IPPs for NC from [RVW13, RR20]. However, the setting of distribution-free testing is more involved, and below, we highlight the key challenges encountered in this setting, and our ideas to overcome them. Our distribution-free IPPs are constructed through an interplay of various techniques and tools from interactive proofs, property testing, and distribution testing; see Section 3.2, for further details on the proof strategy of Theorem 1.4.

Note that, for convenience, we show the construction of the distribution-free IPP from Theorem 1.1 in the setting of $\tau = O(1/\varepsilon)$, for any proximity parameter ε, obtaining query complexity $O(1/\varepsilon)$ and communication complexity $\tilde{O}(\varepsilon \cdot n + 1/\varepsilon)$. This can be shown to be equivalent to the statement of Theorem 1.1 that is parameterised by τ; for more details see Section 4. Similarly, the IPPs for our other results are parameterised in terms of the proximity parameter ε.

	Property Testing	IPP	DF-Property Testing	DF-IPP
Languages in NC	$\Omega(n)$ (e.g., low-degree univariate polynomial)	$\tilde{O}(\sqrt{n})$ [RVW13, RR20]	$\Omega(n)$ similarly	$\tilde{O}(\sqrt{n})$ (arbitrary distributions, for $\varepsilon \geq 1/\sqrt{n}$); see Theorem 1.1 $n^{1/2+o(1)}$ (smooth distributions); see Theorem 1.2 $n^{1/2+o(1)}$ (product distributions); see Theorem 1.3
TensorSum	$\Omega(n^{0.99+o(1)})$ [GR18]	polylog(n) [GR18]	$\Omega(n^{0.99+o(1)})$ Trivially, from [GR18]	polylog(n); see Corollary 1.6
Symmetric Properties	$\Theta(1)$ ($\varepsilon = O(1)$) Folklore	polylog(n) [RVW13]	$\Omega(n^{\frac{1}{3}})$ Theorem 1.4	polylog(n); see Theorem 1.4

Table 1: This is a table of our main results (TensorSum is defined in Definition 3.6). The complexities shown here are those that minimise the sum of the query and communication complexity. Note that while the uniform property tester for symmetric properties is more efficient than the corresponding uniform IPP, this only holds for restricted (constant) values of ε.

1.3.1 Proof outline of Theorem 1.1

The [RVW13] protocol (as well as the follow-up work [RR20]) is centered around a parameterised problem called PVAL. Loosely speaking, the PVAL language contains all strings, whose encoding under a specific code, called the low degree extension, is equal to given values when projected on to the given coordinates. More precisely, the PVAL problem is parameterised by a (sufficiently large) finite field $\mathbb{F}$, integers k, m, n such that $k, m < |\mathbb{F}|$ and $k^m = n$, a set of vectors $J = (j_1, \ldots, j_t) \subset \mathbb{F}^m$ of size t and a t-length vector $\vec{v} \in \mathbb{F}^t$. An input $X \in \mathbb{F}^{k^m}$ is in PVAL$(J, \vec{v})$ if it holds that $P_X(j_i) = v_i$, for every $i \in [t]$, where $P_X : \mathbb{F}^m \to \mathbb{F}$ is the m-variate low-degree extension (LDE) of X.[6]

The interactive reduction from NC to PVAL. Let L be any language in NC and let $\varepsilon > 0$ be the input proximity parameter. Let $X \in \{0,1\}^n$ be the input to L and $\mathcal{D}$ be the unknown underlying distribution over which the verifier can access X through a sample oracle. The first step in [RVW13] is to show an interactive reduction Π_{NC} from L to (a parameterisation of) PVAL, where the verifier *does not access* the input $X \in \{0,1\}^n$.[7]

In more detail, let $B_{\mathcal{D}}(X)$ (respectively $B_{\mathcal{U}}(X)$) be the set of binary strings that are at a distance at most ε along the distribution $\mathcal{D}$ (respectively the uniform distribution $\mathcal{U}$) from X. In [RVW13], the verifier in Π_{NC} generates parameters $(\mathbb{F}, k, m, J, \vec{v})$ for PVAL, where J is a set of t points in $\mathbb{F}^m$, such that the following hold when t is sufficiently large.

- If $X \in L$, then $X \in$ PVAL$(J, \vec{v})$.

[6] Recall that the m-variate LDE P_X is the unique polynomial with individual degree $k-1$ such that P_X agrees with X on $[k]^m$, where we identify $[k]$ with a subset of field elements in some canonical way.

[7] Technically, an interactive proof is specified by a verifier and an honest prover. However, for the sake of exposition we refer to them both together as Π_{NC} in this section.

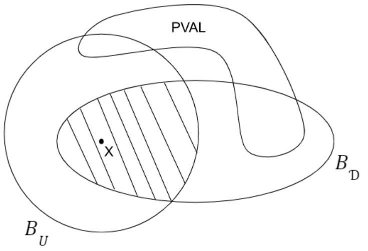

Figure 1: The shaded region ($B_{\mathcal{U}}(X) \cap B_{\mathcal{D}}(X)$) consists of the set of points in $\{0,1\}^n$ that are ε-close to X with respect to both $\mathcal{D}$ and $\mathcal{U}$. The soundness promise of the interactive reduction Π' ensures that any string in $\mathsf{PVAL}(J, \vec{v})$ is present in at most one of $B_{\mathcal{U}}(X)$ or $B_{\mathcal{D}}(X)$, but not in both (shaded region) (with high probability).

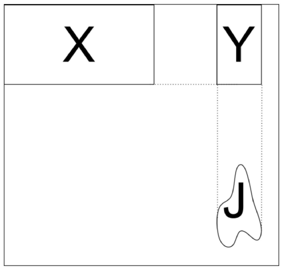

Figure 2: In the uniform IPP for PVAL, the prover sends the $(m-1)$-variate LDE of each row of X evaluated on J_2 (column indices of J), in the form of the purported matrix $Y' \in \mathbb{F}^{k \times t}$. However, to ensure consistency of Y' with respect to $\mathsf{PVAL}(J, \vec{v})$, for any $j = (a, b) \in J$, the univariate LDE of the b^{th}-column of Y' evaluated on a is required to be equal to $\vec{v}[j]$.

- If X is ε-far from L along $\mathcal{U}$ then, with high probability over the verifier's randomness, $B_{\mathcal{U}}(X)$ and $\mathsf{PVAL}(J, \vec{v})$ are disjoint. In other words, with high probability, X is ε-far from $\mathsf{PVAL}(J, \vec{v})$ along $\mathcal{U}$.

Furthermore, the points J output by the reduction Π_{NC} are *distributed uniformly at random* in $(\mathbb{F}^m)^t$. Crucially, [RVW13] show that the guarantees over the outputs of this reduction *only hold* when $t = O(\log(|B_{\mathcal{U}}(X)|)$ many points are picked in J.[8]

Since the size of the set $B_{\mathcal{U}}(X)$ is $\binom{n}{\varepsilon n} \leq O(2^{\varepsilon n \log(n)})$, following from the earlier discussion, by setting $t = O(\log(|B_{\mathcal{U}}(X)|) = \tilde{O}(\varepsilon n)$, we ensure that the guarantees of Π_{NC} hold. An immediate attempt would be try to extend this analysis verbatim to distribution-free testing, by setting t to $O(\log(|B_{\mathcal{D}}(X)|))$ instead, and thus having Π_{NC} guarantee that X is ε-far from $\mathsf{PVAL}(J, \vec{v})$ along the distribution $\mathcal{D}$, for soundness. However, for an arbitrary unknown distribution $\mathcal{D}$, the size of $B_{\mathcal{D}}(X)$ can be prohibitively large. For example, when $\mathcal{D}$ is supported over the first $\log(n)$ indices, for any value of ε, the size of $B_{\mathcal{D}}(X)$ blows up to at least $2^{n-\log(n)}$! Thus, for our choice of t, we already lose the sublinear time verification and communication complexity, and it is unclear if this reduction can achieve such soundness guarantees for PVAL.

Uniform IPP for PVAL is also "complete" for distribution-free IPPs for NC. Our key idea for constructing the distribution-free IPP for L, is in fact, an interactive reduction Π' to constructing a *uniform* IPP for PVAL (with a different parameterisation for PVAL than that obtained by Π_{NC}). Theorem 1.1 follows by using the ready-made uniform IPP for PVAL by [RR20].

Consider a NO input $X \in \{0,1\}^n$ to L, that is, an input that satisfies the soundness requirement $d_{\mathcal{D}}(X, L) > \varepsilon$, over the unknown distribution $\mathcal{D}$. To start with, Π' runs the interactive reduction

[8] Π_{NC} runs t parallel copies of the interactive reduction from L to PVAL over a single point by [GKR15], with the guarantee that if the input $X \notin L$, the probability that X is also in PVAL over t points, is at most 2^{-t}. Now, if X were instead ε-far from L, then a union bound over all the points in $B_{\mathcal{U}}(X)$ ensures a small probability for the event that there exists a point in $B_{\mathcal{U}}(X)$ that is also in PVAL over t points. We refer to Section 4.2.1 for more details.

Π_{NC} from L to $\mathsf{PVAL}(J, \vec{v})$ with the same value of $t = |J| = \tilde{O}(\varepsilon n)$.

Setting t to be $O(\log(|B_{\mathcal{D}}(X) \cap B_{\mathcal{U}}(X)|)) \leq O(\log(|B_{\mathcal{U}}(X)|)) = \tilde{O}(\varepsilon n)$, we can generalise the guarantees of Π_{NC} to show that the intersection of $\mathcal{B}_{\mathcal{U}}(X)$ and $\mathcal{B}_{\mathcal{D}}(X)$ is disjoint from $\mathsf{PVAL}(J, \vec{v})$, with high probability. Indeed, this builds on the earlier argument (and Footnote 8), but over $\mathcal{B}_{\mathcal{U}}(X) \cap \mathcal{B}_{\mathcal{D}}(X)$, alongside the fact that the size of this set is upper bounded by the size of $\mathcal{B}_{\mathcal{U}}(X)$. Thus, X cannot be ε-close to $\mathsf{PVAL}(J, \vec{v})$ along *both* $\mathcal{U}$ and $\mathcal{D}$, or in other words, X is ε-far from every element of PVAL along at least one of the two distributions (see Figure 1 and Section 4.2.1 for details).

Following this, assume that $d_{\mathcal{D}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon$. We construct the next stage of Π', based on a case analysis whether X is *additionally* ε-far from $\mathsf{PVAL}(J, \vec{v})$ under the uniform distribution or not. Indeed, suppose that X is ε-far from $\mathsf{PVAL}(J, \vec{v})$ under the uniform distribution. This is the easy case; we can catch this with the uniform IPP for $\mathsf{PVAL}(J, \vec{v})$ as usual.

On the other hand, suppose that instead, X is close to $\mathsf{PVAL}(J, \vec{v})$ under the uniform distribution, i.e., $d_{\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) \leq \varepsilon$. At this point, we observe (following [RR20]) that when J is distributed uniformly at random, with high probability $\mathsf{PVAL}(J, \vec{v})$ is a good error correcting code (i.e., with large minimal distance).[9] Since the output J of Π_{NC} is distributed uniformly at random, when X is ε-close to $\mathsf{PVAL}(J, \vec{v})$ over the uniform distribution, Π_{NC} guarantees that X is in fact close to a *unique* element X' in $\mathsf{PVAL}(J, \vec{v})$.

To summarize, so far we have that X is ε-close to $X' \in \mathsf{PVAL}(J, \vec{v})$ along $\mathcal{U}$, but by our soundness condition, X is ε-far from $\mathsf{PVAL}(J, \vec{v})$, and in particular from X', along $\mathcal{D}$. Now, the verifier uses the sample oracle to $\mathcal{D}$ to generate $O(1/\varepsilon)$ samples, which we denote by $I \subseteq [n]$, and the corresponding values in X given by $X|_I$. From the soundness assumption, with high probability there exists an index i in I such that $X_i \neq X'_i$. Combining this with the fact that every element in $\mathsf{PVAL}(J, \vec{v})$ other than X' is ε-far from X along the uniform distribution, X' is not in $\mathsf{PVAL}((J, I), (\vec{v}, X|_I))$, where PVAL is parameterised over a larger set. In other words, we see that X is ε-far from $\mathsf{PVAL}((J, I), (\vec{v}, X|_I))$ along the *uniform distribution* and a uniform IPP for $\mathsf{PVAL}((J, I), (\vec{v}, X|_I))$ suffices.

The argument for completeness trivially holds from the guarantees of Π_{NC} and definition of an LDE of X, since in this case $X \in \mathsf{PVAL}((J, I), (\vec{v}, X|_I))$. We end with a quick note on the complexity of the distribution-free IPP. The query complexity of $O(1/\varepsilon)$ is the same as that of the uniform IPP by [RR20], and the communication complexity is the sum of the number of bits used to send the $O(1/\varepsilon)$ samples in I in addition to the communication by the uniform IPP, which is $\tilde{O}(\varepsilon n)$. Overall the communication complexity is $\tilde{O}\left(\frac{1}{\varepsilon} + \varepsilon \cdot n\right)$ which matches that in [RR20] (up to poly-logarithmic factors) whenever $\varepsilon \geq 1/\sqrt{n}$.

1.3.2 Proof outlines of Theorems 1.2 and 1.3

Next, we describe the proof techniques of Theorems 1.2 and 1.3 that construct IPPs for NC over smooth distributions and product distributions, matching the complexities of [RVW13] for every value of ε. This improves over the communication complexity of the distribution-free IPP in Theorem 1.1 when $\varepsilon \ll 1/\sqrt{n}$ (with roughly the same query complexity). We follow the general strategy by [RVW13] and the main technical challenges arise during the analysis with respect to the new promise on the soundness of an IPP for PVAL. We assume some familiarity with the uniform IPP construction by [RVW13] for this section; see also Section 5.2.1 for more detailed intuition.

[9] It is worth emphasising that this does not hold for every choice of J, for eg., $\mathsf{PVAL}(J, \vec{v})$ is a bad error correcting code when J consists of t copies of the same point.

Uniform IPP for $\mathsf{PVAL}(J, \vec{v})$. We start with a summary of the Uniform IPP from [RVW13]. Let the input $X \in [k]^m$, for $k = \log n$ and $n = k^m$. Further, let $|J| = t$.

[RVW13] use a divide and conquer approach, by decomposing the t claims about X into new claims for each individual row instance $X_i \in \mathbb{F}^{k^{m-1}}$, for every $i \in [k]$. In more detail, let $J = (J_1, J_2)$, where the first component $J_1 \subset \mathbb{F}$ and $J_2 \subset \mathbb{F}^{m-1}$. The prover sends the matrix $Y' \in \mathbb{F}^{k \times t}$, where each row Y'_i is the purported set of evaluations of the $(m-1)$-variate LDE (of individual degree $k-1$) of X_i on J_2. By the definition of an m-variate LDE on X, the prover cannot lie about the consistency of Y' with $\vec{v}$, since for each $(a, b) \in J$ (where $b \in J_2$), the verifier can easily check if the univariate LDE of $Y'[\cdot, b]$ (the b^{th} column of Y) evaluated on the coordinate a equals $\vec{v}[(a, b)]$ (see Figure 2).

Thus, the initial PVAL instance is now reduced to k instances $X_i \in \mathbb{F}^{k^{m-1}}$ for $\{\mathsf{PVAL}(J_2, Y'_i)\}$. A natural idea now is for the verifier to send a random vector $z \in \mathbb{F}^k$ to the prover, and ask it back for a "folded" version $X' \in \mathbb{F}^{k^{m-1}}$, that is purported to be $z \cdot X$.[10] Now, the IPP could recurse on a *single input* $X' \in \mathbb{F}^{k^{m-1}}$ that has shrunk in size by a factor of k, to the problem $\mathsf{PVAL}(J_2, z \cdot Y')$. Completeness easily holds, since if X belonged to $\mathsf{PVAL}(J, \vec{v})$, then the honest prover will just send the "true" $Y' \in \mathbb{F}^{k \times t}$ and the verifier checks always pass.

Uniform Distance Preservation Lemma. However showing soundness is not straightforward. Suppose that X is ε-far from $\mathsf{PVAL}(J, \vec{v})$ under the uniform distribution. It turns out that the malicious prover has cheated in at least one row of the purported matrix Y' (if not, since X is not in PVAL, there would be at least one column in Y' which would be inconsistent with the corresponding value in $\vec{v}$ and the verifier would catch the prover in the checks made above).

For any row $X_i \in \mathbb{F}^{k^{m-1}}$ that is a lower-dimensional input instance, let ε_i be the distance between X_i and $\mathsf{PVAL}(J_2, Y'_i)$. To ensure that the verifier catches the cheating prover, the folded instance X' also needs to be reasonably far from PVAL on a lower dimension at the end of a recursive step. In order to capture this, [RVW13] (implicitly) use a *uniform distance preservation lemma*, which states that if X is ε-far from $\mathsf{PVAL}(J, \vec{v})$, then $\sum_{i=1}^{k} \varepsilon_i > k\varepsilon$.

Using the uniform distance preservation lemma, [RVW13] observe that if the prover ended up cheating (roughly) uniformly across all rows in Y', then any row X_i would be roughly ε-far from $PVAL(J_2, z \cdot Y'_i)$, and the IPP would recurse by picking a single row at random. However, the prover could have cheated across multiple rows of Y'_i and the verifier does not know these rows. To accommodate this, the verifier considers $\log(k)$ many random foldings of X, where the Hamming weight of the vectors z used to fold X, range across 1 to k (in powers of 2). In particular, this results in $O(\log(\log(n)))$ recursive instances in $\mathbb{F}^{k^{m-1}}$. Crucially, they use the uniform distance preservation lemma to generalise the intuition above and show that for at least one of these folded instances, the distance is roughly preserved. Moreover, for such a folded instance, the product of the new distance and the effective query complexity (the number of queries on X to compute the value at any index in $z \cdot X$) is $O(1/\varepsilon)$, along with small but super-constant multiplicative factors.

The IPP continues to recursively fold the instance dimension-wise by the above process, until the size of each final folded instance becomes $\tilde{O}(\varepsilon n)$, which happens after $\Omega(\log(n)/\log(\log(n)))$ steps. In such a case, the prover directly sends each final instance. Since there exists an instance $\tilde{X}^j$ at each level of recursion for which distance is preserved, there exists a final folded instance $\tilde{X}$, such that the verifier catches a cheating prover by uniformly *sampling* a few coordinates of $\tilde{X}$. Moreover, since the product of the distance and effective query complexities for each $\tilde{X}^j$ are

[10] The dot product $z \cdot X \in \mathbb{F}^{k^{m-1}}$ between $z \in \mathbb{F}^k$ and a matrix $X \in \mathbb{F}^{k \times k^{m-1}}$ is given by $\sum_{i=1}^{k} z_i X_i$.

roughly maintained to be small at each step of the recursion, making $O(1/\varepsilon^{1+o(1)})$ many queries to $\tilde{X}$ is sufficient to catch the cheating prover (since the total number of recursive instances after the stated number of steps is roughly $n^{o(1)} = 1/\varepsilon^{o(1)}$). The communication complexity is simply the number of bits used to send all the final folded instances, in addition to sending the matrices Y' of size $k \times t$, and thus is $\tilde{O}(\varepsilon^{1-o(1)}n)$.

IPPs for NC under specific distribution families. We now highlight some key ideas which help us construct IPPs over large distribution families like smooth distributions and product distributions. To begin with, on any input $X \in \{0,1\}^{k^m}$, we first reduce L to PVAL using Π_{NC}. Recall that in the distribution-free setting, Π_{NC} outputs $(J, \vec{v})$, such that for the soundness promise, with high probability X cannot be ε-close to $\mathsf{PVAL}(J, \vec{v})$ along both $\mathcal{U}$ and the unknown distribution from the given family, $\mathcal{D}$. In other words, X is ε-far from $\mathsf{PVAL}(J, \vec{v})$ along at least one of $\mathcal{U}$ or $\mathcal{D}$. Building on this observation, we design IPPs for $\mathsf{PVAL}(J, \vec{v})$ over these distribution families, using an intricate case analysis of the soundness condition.

In more detail, if X is ε-far from $\mathsf{PVAL}(J, \vec{v})$ under the uniform distribution, then we can directly use the uniform distance preservation lemma to catch a malicious prover as seen previously in the uniform IPP. If not, suppose that $d_{\mathcal{D}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon$. Next, we briefly describe the soundness analysis, using *specific distance preservation lemmas* for smooth distributions and product distributions. Given this, we build on the strategy of the uniform IPP above to construct an IPP for $\mathsf{PVAL}(J, \vec{v})$ over these distribution families, with the main technical work being that of simultaneously incorporating both the uniform and the respective distance preservation lemmas into the soundness analysis, across the recursive levels.

ρ-dispersed distributions. Recall that ρ-dispersed distributions over $[k]^m$ capture the smoothness of a distribution, by requiring that the probability mass on any element is bounded by ρ times the average mass on any of its neighbours. Adopting similar notation as above, let $\hat{\mathcal{D}}$ be the marginal distribution of $\mathcal{D}$ over $[k]^{m-1}$.

For any row $X_i \in \mathbb{F}^{k^{m-1}}$ that is a lower-dimensional input instance, let ε_i be the distance between X_i and $\mathsf{PVAL}(J_2, Y_i')$ over $\hat{\mathcal{D}}$. Here, we show a distance preservation lemma for ρ-dispersed distributions, such that for any distribution $\mathcal{D}$ that is ρ-dispersed, $\sum_{i=1}^{k} \varepsilon_i > (k\varepsilon)/\rho$.[11] The idea behind proving this is not obvious immediately; while ε_i measures the distance along marginal distributions, ε is the distance from each element of $\mathsf{PVAL}(J, \vec{v})$ over $\mathcal{D}$ (which could be a joint distribution). However, we crucially use properties about ρ-dispersed distributions to prove this distance preservation lemma.

Using the strategy described earlier, we get an IPP for NC over ρ-dispersed distributions, having query and sample complexities $\frac{\rho^{\log(1/\varepsilon)/\log\log(n)}}{\varepsilon^{1+o(1)}}$, while keeping communication complexity the same. In particular, for $\rho = k^{o(1)}$, the query complexity is $1/\varepsilon^{1+o(1)}$ and matches that of the uniform IPP for all $\varepsilon > 0$. We refer to Section 5.2.1 for further intuition about this.

Product distributions. Let $\mathcal{D}$ be an m-product distribution defined as $\mathcal{D} = \mathcal{D}_1 \times \ldots \mathcal{D}_m$ over $[k]^m$, where $k = \log(n)$, and each $\mathcal{D}_j$ is an independent distribution supported on $[k]$. In particular, $\mathcal{D}(i_1, \ldots, i_m)$ is defined as $\prod_{j=1}^{m} \mathcal{D}_j(i_j)$.

[11] Note that the uniform distribution is a 1-dispersed distribution and we thus generalise the uniform distance preservation lemma.

Our main approach here to construct IPPs over such distributions, is to first *learn* the underlying distribution and then use this as an aid to obtain near-optimal complexity parameters. For more context, consider the following k-dispersed distribution $\mathcal{D}$ over $[k]^m$, that is supported on the first row of the first dimension, i.e, exactly on the set of elements of the form $(1, i_2, \ldots, i_m)$ for every $(i_2, \ldots, i_m) \in [k]^{m-1}$.[12] We see that the IPP over k-dispersed distributions has query complexity $O(1/\varepsilon^2)$. However, if the verifier "learns" beforehand that $\mathcal{D}$ is only supported on the first row, then it can focus its attention on a smaller instance in $\mathbb{F}^{k^{m-1}}$ and potentially obtain much better query complexity, if $\mathcal{D}$ conditioned on the first row is ρ-dispersed, for a small ρ.

Our main technical idea here is to show a *learning-augmented* distance preservation lemma for product distributions. Let ε_i be the distance between X_i and $\mathsf{PVAL}(J_2, Y_i')$ over $\hat{\mathcal{D}} = \mathcal{D}_2 \times \cdots \times \mathcal{D}_m$. Based on an alternative analysis to that of ρ-dispersed distributions, we prove that for any product distribution $\mathcal{D}$, $\sum_{i=1}^{k} \varepsilon_i > C\varepsilon$, for $C > 1$ that *only depends on* $\mathcal{D}_1$. Using this key insight, if the verifier "transformed" $\mathcal{D}_1$ into the uniform distribution over $[a_0 \cdot k]$, where $a_0 \geq 1$ is a small constant, then we get a similar expression as the uniform distance preservation lemma, i.e., $C = O(k)$, despite still measuring distance according to $\hat{\mathcal{D}}$ for the lower dimensional instances.[13]

We briefly highlight the sequence of tools used to implement the latter idea. The verifier learns the probability vector of $\mathcal{D}_1$, into an approximation $\mathcal{P}_1$, using the *parallel set lower bound protocol* [BT06] which requires white-box access to $\mathcal{D}_1$. Following this, it runs a *"granularising"* algorithm taking $\mathcal{P}_1$ as input, that outputs the probability vector of a new $8k$-granular distribution $\mathcal{E}_1$ over $[k+1]$ (i.e., for every i, $\mathcal{E}_1(i)$ is $b_i/8k$), such that in the soundness case, the distance of the input over $\mathcal{E}_1$ is still ε (up to constant factors). Finally, this granularity set is used to "extend" X into a new input instance $X' \in \{0,1\}^{8k \times k^{m-1}}$, by making copies of each row according to it's granularity, and we can thus, equivalently consider the underlying row distribution as the uniform distribution over $[8k]$. The last two steps build on ideas from [Gol20] for testing unknown distributions, while our focus is on the setting of testing with an implicit input.

The details of adapting both distance preservation lemmas and the analysis of the IPP, to handle changing distributions and input sizes across different levels of recursion, is found in Section 6.

1.4 Related Work

Proofs of Proximity for Distributions. In a related model, [CG18, HR22] study proofs of proximity for *testing distributions.* In their setting, for a fixed property Π of distributions, the verifier receives samples from an unknown distribution $\mathcal{D}$, and interacts with the prover to decide whether $\mathcal{D} \in \Pi$ or $\mathcal{D}$ is ε-far from any distribution in Π along the total variation distance. While there are superficial similarities to our model regarding the use of sample oracle, we focus on testing properties (or languages) of strings, where the distribution oracle only provides a means of accessing the input string. In addition, the verifier also has oracle access to the input instance and the distance for the NO instance is measured with respect to the underlying distribution.

Sample-based IPPs. Another related model is that of Sample-based IPPs [GR22], where the verifier can *only* access the input through an oracle that provides labeled samples over the uniform distribution. They show that any language in logspace-uniform NC has an SIPP with $\tilde{O}(\sqrt{n})$ sample

[12] See Section 5.1; intuitively, for any $i_2, \ldots, i_m \in [k]^{m-1}$, $\mathcal{D}(1, i_2, \ldots, i_m)$ is the only element in the set $\{\ell, i_2, \ldots, i_m\}_{\ell \in [k]}$ with a non-zero probability mass and thus is k-times the average of the probability mass on its neighbourhood.

[13] For consistency, $a_0 = 1$, when $\mathcal{D}_1$ is just $\mathcal{U}_k$.

and communication complexities, by in fact constructing a reduction protocol from an SIPP to the query-based IPP by [RVW13]. Our model is more general conceptually, since any protocol in our model needs to be able to test for a language given access to labeled samples over any unknown distribution. On the other hand, to aid with this generality, we also provide the verifier with the more powerful oracle access to the input, which SIPPs do not.

That being said, we can use the uniform SIPP by [GR22] within the proof of Theorem 1.1 (instead of the query-based IPP by [RR20]) to obtain a distribution-free SIPP for NC where the verifier only accesses the input through labeled samples over $\mathcal{U}$ and the unknown distribution $\mathcal{D}$, for any $\varepsilon \geq \tau/n$.[14] It is unclear whether we can construct distribution-free SIPPs for general values of ε (even over smooth or product distributions) that match the complexities of the uniform IPPs and we leave it as future work.

Interactive Proofs for Agnostic Learning. [GRSY21] study the setting of verifying PAC-learners. There, the verifier has sampling access to an unknown distribution $\mathcal{D}$ over labeled examples of the form (i, x_i), where $i \sim \mathcal{D}$ and x is the underlying input. It's goal is to verify whether a hypothesis $h : \{0,1\}^{\log(n)} \to \{0,1\}$ given by the prover from a fixed hypothesis class, is the best approximation of $\mathcal{D}$. From the property testing perspective, the prover wants to convince the verifier that $\mathcal{D}'$ has the property that every hypothesis in the class has error larger than ε over $\mathcal{D}$, for some $\varepsilon > 0$ (i.e., the best possible approximation of $\mathcal{D}$ by the hypothesis class is at least ε).

Similar to the setting of SIPPs, their scenario focuses on the case where the verifier only has access to x via a labeled sample oracle, over an unknown distribution. Furthermore, they focus on testing specific properties pertaining to machine learning, such as closeness to an underlying hypothesis class, with the hope of getting very low sample complexity (with respect to the VC dimension of the hypothesis class). In contrast, we deal with verification of general classes of properties, and in some cases the sample and query complexities are both $\tilde{O}(\sqrt{n})$.

2 Preliminaries and definitions

We denote $[n] = \{1, 2, \cdots, n\}$. A language L is defined as $L = \bigcup_{i \in \mathbb{N}} L_n \subseteq \{0,1\}^*$, where each $L_n = L \cap \{0,1\}^n$.

Throughout this work, we consider languages computable by logspace-uniform Boolean circuits on n variables of size $S(n)$ (number of gates) and depth $d(n)$ (longest path from the output gate to some input), with XOR and AND gates of fan-in two. Of particular interest is the class logspace-uniform NC, which is the class of languages computable by logspace-uniform circuit families of size $\mathsf{poly}(n)$ and depth $O(\log^i(n))$ for some fixed $i \in \mathbb{N}$. In more detail, L belongs to logspace-uniform NC, if there exists $i \in \mathbb{N}$ and a logspace Turing machine M that takes input 1^n and outputs the description of an n-variate circuit of depth $O(\log^i(n))$, such that for each $x \in \{0,1\}^n$, $C(x) = 1$ if and only if $x \in L$.

2.1 Hybrid metrics

We denote by $\Delta(\Omega_n)$, the simplex of all possible distributions over a domain Ω_n of size $n \in \mathbb{N}$. Let $\mathbb{F}$ be a finite field and let x, y be vectors in $\mathbb{F}^n$. For any distribution $\mathcal{D} \in \Delta(\Omega_n)$, we define the

[14] The uniform SIPP by [GR22] has communication complexity $\tilde{O}\left(\frac{n}{\tau} + \frac{1}{\varepsilon}\right)$ (for tradeoff $\tau \leq \sqrt{n}$), and using this still gives us the same communication complexity as the query-based distribution-free IPP from Theorem 1.1.

distance between x and y as

$$d_{\mathcal{D}}(x,y) = \underset{i\sim D}{\mathbb{P}}\left[x_i \neq y_i\right].$$

We use $\mathcal{U}_n$ to denote the uniform distribution over the set $[n]$, where the size of the support is clear we denote the uniform distribution by $\mathcal{U}$. Note that if the distance is measured according to $\mathcal{U}$, then this is simply the normalised Hamming distance.

For any (non-empty) $L \subseteq \mathbb{F}^n$ and any vector $x \in \mathbb{F}^n$, we similarly define the distance between x and L as:

$$d_{\mathcal{D}}(x,L) = \min_{y\in L} d_{\mathcal{D}}(x,y).$$

If $d_{\mathcal{D}}(x,L) > \varepsilon$, we say that x is ε-far from L over the $\mathcal{D}$ distribution, otherwise we say it is ε-close. Furthermore, for any $n \in \mathbb{N}$, $\mathcal{D} \in \Delta(\Omega_n)$, $\varepsilon > 0$, and $X \in \mathbb{F}^n$, we denote by $B_{\mathcal{D},\varepsilon}(X)$ as the subset of $\mathbb{F}^n$ that is ε-close to X along $\mathcal{D}$. In other words,

$$B_{\mathcal{D},\varepsilon}(X) = \{Y \in \mathbb{F}^n | d_{\mathcal{D}}(X,Y) < \varepsilon\}. \tag{1}$$

The hybrid metric is the maximum over two distances, this increases the size of the set of elements ε-far from an input X. The notion of a hybrid metric is key to our proof of Theorem 1.1, see Section 4 for details. We define the hybrid metric as follows.

Definition 2.1 (Hybrid Metrics)**.** *For any pair of distributions $\mathcal{D}_1$, $\mathcal{D}_2$ over $[n]$, we define the ($\mathcal{D}_1$, $\mathcal{D}_2$)-Hybrid Metric $\mu_{\mathcal{D}_1,\mathcal{D}_2}$ as follows.*

$$\mu_{\mathcal{D}_1,\mathcal{D}_2}(x,y) = \underset{s\in\{\mathcal{D}_1,\mathcal{D}_2\}}{max}(d_s(x,y)).$$

Remark 2. *Note that taking the maximum over two metrics is also a metric, as the triangle inequality follows since for some $s \in \{\mathcal{D}_1, \mathcal{D}_2\}$, it holds that*

$$\mu_{\mathcal{D}_1,\mathcal{D}_2}(x,y) = d_s(x,y) < d_s(x,z) + d_s(z,y) \leq \mu_{\mathcal{D}_1,\mathcal{D}_2}(x,z) + \mu_{\mathcal{D}_1,\mathcal{D}_2}(z,y).$$

In addition, the definitions of distance of an input string to a language extend in a natural way with respect to $\mu_{\mathcal{D}_1,\mathcal{D}_2}$.

2.2 Interactive Proofs of Proximity (IPP)

We refer to the standard textbook [AB09] for the definition of an interactive proof (IP). IPPs [EKR04, RVW13] are interactive proofs that verify the "closeness" of an input string to the given language. In these interactive proofs, the verifier must accept if the input is in the language and reject when it is far with some computation performed by an untrusted prover. The goal is to achieve verification using sublinear queries and communication, by not having to read the input completely. Following previous literature on IPPs, we view the inputs given to the verifier as having two parts: an *implicit* input $X \in \mathbb{F}^n$ and an *explicit* input $w \in \mathbb{F}^*$ (w could be empty), for some finite field $\mathbb{F}$. The verifier can access X only via an oracle (query or sample), but can read w in its entirety. We then refer to $\{L_w\}_{w\in\mathbb{F}^*}$ as a family of *parameterised* languages, each language being parameterised by the explicit input.[15] At times, we will refer to this family of languages simply as L and take the implicit input as already given to the IPP.

[15] Equivalently, we can view L as a language over *pairs* (X,w) and define each $L_w = \{X \mid (X,w) \in L\}$. The closeness of a string to L is only measured with respect to X, the first string in the pair.

For any language L_w, we denote by $(P(X), V^X)(w, n, \varepsilon)$ as the output of the interaction between a verifier V having query access to an input X of length n and a prover P with explicit access to X, when both have full access to the shared inputs w, n, and ε. An IPP over the uniform distribution is defined as follows.

Definition 2.2 (IPPs **over the Uniform Distribution** [**EKR04, RVW13**])**.** *For any fixed string $w \in \mathbb{F}^*$, let $L_w \subseteq \mathbb{F}^*$ be a parameterised language. We say that L has an interactive proof of proximity (*IPP*) if there exists a proof system (P, V) with a (possibly computationally unbounded) prover P and a computationally bounded verifier V, such that for every n, input $X \in \mathbb{F}^n$ and proximity parameter $\varepsilon > 0$, the following hold.*

When P has full access to X, w, n, ε, and when V is given query access to X and full access to w, n, ε, the following hold:

- *Completeness: If $X \in L_w$, then*

$$\mathop{\mathbb{P}}_{V}\left[(P(X), V^X)(w, n, \varepsilon) = 1\right] \geq \frac{2}{3}.$$

- *Soundness: If $d_{\mathcal{U}_n}(X, L_w) > \varepsilon$, then for every computationally unbounded prover P^* we have*

$$\mathop{\mathbb{P}}_{V}\left[(P^*(X), V^X)(w, n, \varepsilon) = 0\right] \geq \frac{2}{3}.$$

Furthermore, we say that the IPP *has query complexity $q = q(n, |w|, \varepsilon)$, communication complexity $c = c(n, |w|, \varepsilon)$ and round complexity $R = R(n, |w|, \varepsilon)$, if P and V exchange at most c bits in at most R rounds of interaction (having 2 messages per round) and V makes at most q many queries during this process, for every w, $X \in \mathbb{F}^n$, and $\varepsilon > 0$.*

Additionally, we call this IPP *a Merlin-Arthur proof of proximity (*MAP*) if over the course of this protocol, the verifier does not send any messages to the prover.*

Below, we state the main result from [RVW13].

Theorem 2.1 (IPP **for Low Depth Languages over the Uniform Distribution** [**RVW13**])**.** *For every language $L \subseteq \{0,1\}^n$ and $\varepsilon \in (0, 1]$ computable by log-space-uniform circuits of depth $\Delta_L = \Delta_L(n)$ and size $S = S(n)$, there exists an interactive proof of proximity for L with perfect completeness and soundness at least $1/2$.*

This IPP *has query complexity $\frac{1}{\varepsilon^{1+o(1)}}$, communication complexity $\varepsilon \cdot n \cdot \left(\frac{1}{\varepsilon^{o(1)}}\right) + \varepsilon \cdot n \cdot \mathsf{poly}(\Delta_L)$ and round complexity $O\left(\frac{\log(\frac{1}{\varepsilon})}{\log\log(n)} + \Delta_L \cdot \log(S)\right)$. In addition, the honest prover runs in time $\mathsf{poly}(S, n)$ and the verifier runs in time $(\frac{1}{\varepsilon})^{1+o(1)} + (\varepsilon \cdot n)^{1+o(1)}\mathsf{poly}(\Delta_L)$.*

3 Distribution-free IPPs

In this section, we define the notion of distribution-free proofs of proximity and provide complexity theoretic insight regarding the power and limitation of the model. We start by defining the notion of a distribution-free proof of proximity and then extensions of this notion to the white-box model and polynomially-samplable distributions. From these definitions we explore some observations of this model. In Section 3.1, for certain structured languages we show that the existence of an IPP is equivalent to the existence of a distribution-free IPP. Following this, in Section 3.2, we demonstrate an exponential separation between property testing and IPPs in the distribution-free setting. In section 3.3, we use a lower bound for IPP from [KR15] to demonstrate a separation between uniform IPPs and distribution-free IPPs.

Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a distribution ensemble, where each $\mathcal{D}_n \in \Delta(\Omega_n)$. Whenever the context is clear, we abuse notation by dropping the support size in $\mathcal{D}_n$.

Distribution-free IPPs (DF-IPPs) are interactive proofs that verify the closeness of an input to a language L under any arbitrary distribution. If the input is in the language, the DF-IPP must accept and if it is far along this distribution, it must reject. Here, the verifier additionally has sample access to an input string $X \in \{0,1\}^n$ over an unknown (but fixed) distribution $\mathcal{D}$ over $[n]$, via a *sample oracle* $\mathcal{O}_{\mathcal{D}}(X)$. The oracle $\mathcal{O}_{\mathcal{D}}(X)$ returns the tuple (i, X_i), for an index i independently sampled from $\mathcal{D}$. The soundness condition now requires the algorithm to reject strings that are ε-far from the language *along the distribution* $\mathcal{D}$. Additionally, the cheating prover has full access to the distribution, i.e., the prover has access to all of the individual probabilities that constitute the distribution.

Definition 3.1 (**Distribution-Free** IPP)**.** *For any fixed string $w \in \mathbb{F}^*$, let $L_w \subseteq \mathbb{F}^*$ be a parameterised language. We say that L_w has a distribution-free* IPP *if there exists a proof system (P, V), where P is a (possibly computationally unbounded) prover and V is a computationally bounded verifier V, such that for every n, input $X \in \mathbb{F}^n$, proximity parameter $\varepsilon > 0$, and for any fixed (but unknown) distribution $\mathcal{D}_n \in \Delta(\Omega_n)$ from a distribution ensemble $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$, the following hold.*

When P has full access to X, w, n, ε and $\mathcal{D}_n$, and when V is given query access to X, as well as sample access to X via $\mathcal{O}_{\mathcal{D}_n}(X)$ and full access to w, n, ε, the following conditions hold.

- *Completeness: If $X \in L_w$, then*
$$\underset{V,\mathcal{O}_{\mathcal{D}_n}(X)}{\mathbb{P}}\left[\left(P(X,\mathcal{D}), V^{X,\mathcal{O}_{\mathcal{D}_n}(X)}\right)(w,n,\varepsilon) = 1\right] \geq \frac{2}{3}.$$
In other words, if $X \in L_w$ then the verifier accepts the input with probability at least $2/3$ over its own randomness and the samples from $\mathcal{O}_{\mathcal{D}_n}(X)$.

- *Soundness: If $d_{\mathcal{D}_n}(X, L_w) > \varepsilon$, then for any computationally unbounded prover P^* we have*
$$\underset{V,\mathcal{O}_{\mathcal{D}_n}(X)}{\mathbb{P}}\left[\left(P^*(X,\mathcal{D}), V^{X,\mathcal{O}_{\mathcal{D}_n}(X)}\right)(w,n,\varepsilon) = 0\right] \geq \frac{2}{3}.$$
In other words, if $d_{\mathcal{D}_n}(X, L_w) > \varepsilon$, the verifier rejects with all but $1/3$ probability over its own randomness and the samples from $\mathcal{O}_{\mathcal{D}_n}(X)$, regardless of the cheating prover strategy.

The query complexity $q = q(n, |w|, \varepsilon)$, communication complexity $c = c(n, |w|, \varepsilon)$ and round complexity $R = R(n, |w|, \varepsilon)$ of the interactive proof are as defined earlier. In addition, the IPP *has sample complexity $s = s(n, |w|, \varepsilon)$, if V invokes the sample oracle $\mathcal{O}_{\mathcal{D}}(X)$ at most $s(.)$ times during its interaction with P, for any w, X and $\varepsilon > 0$.*

Similar to the non-interactive form of an IPP, we can also define a distribution-free MAP. Moreover, analogous to the PAC-learning setting, we can also consider a fixed set of distributions $\mathcal{F}$ and define a distribution-free IPP over $\mathcal{F}$, by requiring the correctness of the IPP to hold only over the distributions in $\mathcal{F}$. It will be necessary for us in Sections 4 and 6 to consider IPPs with a soundness condition over the hybrid metric μ; for simplicity we will still refer to these as IPPs.

Remark 3. *It is worth noting that while Definition 3.1 provides the honest prover with a full description of $\mathcal{D}$, most of our protocols enjoy the property that the honest prover does not require the description.*

Remark 4. *It is worth noting that for typical properties that are testable given the entire input, both $q(n)$ and $c(n)$ have to be sublinear in $|X|$ for the* IPP *to be non-trivial. Indeed, if V sees all of X, it can directly check if X belongs to L or not. On the other hand, if P sends a string X' of length n (purported to be X), then V checks if $X' \in L$ and perform an equality test between X' and X over $O(1/\varepsilon)$ samples from $\mathcal{O}_{\mathcal{D}}(X)$. Completeness follows when $X' = X$, whereas soundness follows from the distance guarantee of the input X.*

White-Box Distribution-Free IPP. In contrast to Definition 3.1 where the verifier has sample access to the input only via $\mathcal{O}_{\mathcal{D}}(X)$, we define distribution-free IPPs in the *white-box model*, where the verifier now gets the sampling device to the distribution, in the form of a circuit (a notion explored by Sahai and Vadhan in [SV97]), in addition to oracle access to the input. The distribution is defined by the output of the circuit when it is fed with a random input of suitable length.

In more detail, C takes a uniformly random string as input and outputs an index in $[n]$, such that the probability of sampling the index using C is the same as that of $\mathcal{D}$. We consider distributions that are polynomially samplable, i.e., the circuit C takes $\mathsf{polylog}(n)$ many random bits, outputs an index in $[n]$, and its size is polynomial in the number of its inputs (i.e., the size of C is $\mathsf{polylog}(n)$). More formally,

Definition 3.2 (Polynomially samplable distributions)**.** *Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a distribution ensemble, where each $\mathcal{D}_n \in \Delta(\Omega_n)$. $\mathcal{D}$ is said to be polynomially-samplable, if there exists a family of circuits $C = \{C_{r(n)}\}_{n\in\mathbb{N}}$ of size $\mathsf{poly}(\log(n))$, where $C_{r(n)} : \{0,1\}^{r(n)} \to \{0,1\}^{\log(n)}$, such that for each $n \in \mathbb{N}$, the output distribution of $C_{r(n)}$ is the same as $\mathcal{D}_n$, i.e., for every $i \in [n]$, $\Pr_{x\sim U_{r(n)}}\{C(x) = i\} = \mathcal{D}_n(i)$.*[16]

Definition 3.3 (**White-Box Distribution-Free IPP over polynomially samplable distributions**)**.** *For any fixed string $w \in \mathbb{F}^*$, let $L_w \subseteq \mathbb{F}^*$ be a parameterised language. We say that L_w has a white-box* IPP *over polynomially samplable distributions, if there exists a proof system (P, V), where P is a (possibly computationally unbounded) prover and V is a computationally bounded verifier V, such that for every n, input $X \in \mathbb{F}^n$, proximity parameter $\varepsilon > 0$, and for any fixed (but unknown) distribution $\mathcal{D}$ over $[n]$ from a distribution ensemble that is samplable using a polynomial-sized circuit $C : \{0,1\}^{\mathsf{polylog}(n)} \to \{0,1\}^{\log(n)}$, the following hold.*

When P has full access to X, w, n, ε and $\mathcal{D}$, and when V is given the sampling circuit C, query access to X, and full access to w, n, ε, we have the following conditions.

- *Completeness: If $X \in L_w$, then*

$$\mathop{\mathbb{P}}_{V}\left[(P(X), V^X(w, n, C, \varepsilon) = 1\right] \geq \frac{2}{3}.$$

- *Soundness: If $d_{\mathcal{D}_n}(X, L_w) > \varepsilon$, then for any computationally unbounded prover P^* we have*

$$\mathop{\mathbb{P}}_{V}\left[(P^*(X), V^X(w, n, C, \varepsilon) = 0\right] \geq \frac{2}{3}.$$

[16] In particular, this implies that $r(n) \leq \mathsf{polylog}(n)$.

The query complexity $q = q(n, |w|, \varepsilon)$, communication complexity $c = c(n, |w|, \varepsilon)$ and round complexity $R = R(n, |w|, \varepsilon)$ of the IPP *are as defined earlier.*

More generally, we can define white-box IPPs on distributions over $[n]$ samplable by circuit families of size $S(n)$, where $S(n) = 2^{o(\log(n))}$. In this case, the running time of the verifier is given by $T(n, |w|, \varepsilon, S(n))$, and typically, we require $T(n)$ to be sublinear in n.

Remark 5. *Note that the sample complexity of the verifier is not a useful complexity parameter in the white-box model as the both the prover and the verifier get the entire sampling circuit. Indeed, the verifier can sample an index from the circuit and query the input value at this index, or it can possibly use the circuit to perform other computations or simulate input access via some other distribution. Any samples made using the circuit for querying X count towards the query complexity of the* IPP*. Of course, the verifier can go over all possible inputs to the sampler circuit to know the entire distribution exactly, but then its running time is no longer sublinear.*

3.1 DF-IPPs for (Relaxed) Correctable Languages

In this section, we show a generic transformation from an IPP to a distribution-free IPP for any subset of a relaxed locally correctable code (RLCC), while maintaining the round complexity, query complexity and communication complexity, in Theorem 3.1. This is an extension.

We show for this large and natural family of languages you can obtain a distribution-free IPP from a uniform IPP. This includes properties of polynomials or any locally correctable codes as well as many regularly studied problems in the literature for probabilistically checkable proof. These RLCCs have exponential better parameters that locally correctable codes and have had significant recent developments in complexity theory.

For any field $\mathbb{F}$, a code is a subset $C \subseteq \mathbb{F}^n$ with relative distance $\delta > 0$, if the relative minimum distance between any two elements in the code is at least δ, in other words

$$w_1, w_2 \in C \implies d_{\mathcal{U}}(w_1, w_2) \geq \delta.$$

Definition 3.4 (Locally Correctable Codes)**.** *For any field $\mathbb{F}$, let $C \subseteq \mathbb{F}^n$ be an error correcting code with relative distance δ. We say that C is locally correctable if there exists a correcting radius $\delta_r < \delta/2$ and an algorithm A, called the corrector, such that when A is give oracle access to an implicit input $w \in \mathbb{F}^n$ and an explicit input $i \in [n]$, the following hold.*

1. $w \in C \implies \mathbb{P}[A^w(i) = w_i] = 1$.
2. *if* $\exists c \in C$ *such that* $d_{\mathcal{U}}(c, w) \leq \delta_r \implies \mathbb{P}_A\left[A^w(i) = c_i\right] \geq \frac{2}{3}$.

We say A has query complexity t if it uses at most that many queries to perform the correction, on any inputs.

We now generalise this notion to relaxed locally correctable codes. In this setting, the corrector is allowed a third possible output "$\perp$", indicating that it has aborted.

Definition 3.5 (Relaxed Locally Correctable Codes (RLCCs))**.** *Let C be an error correcting code with relative distance δ. We say that C is locally correctable if there exists a $\delta_r < \delta/2$ and a corrector A, such that when A is given oracle access to an implicit input $w \in \{0, 1\}^n$ and explicit access to the input $i \in [n]$, the following hold.*

1. $w \in C \implies \mathbb{P}\left[A^w(i) = w_i\right] = 1$.

2. if $\exists c \in C$ *such that* $d_U(c,w) < \delta_r \implies \mathbb{P}_A\left[A^w(i) \in \{c_i, \perp\}\right] \geq \frac{2}{3}$.

$\perp$ *is a special abort symbol. We say* A *has query complexity* t *if it uses at most that many queries to perform the correction, on any inputs.*

Remark 6. *There is also a third condition that says that in the second case, there are a constant number of coordinates* $i \in [n]$ *for which* $\mathbb{P}_A[A^w(i) = c_i] > \frac{2}{3}$. *This follows from a transformation from [BGH+06] given the first two conditions and given that the algorithm requires only constant queries.*

The following theorem states that there exists an IPP for any subset of a relaxed locally correctable code, The proof follows by a reduction from relaxed correcting to distribution-free IPPs. This builds on a result from [HK07] which states that there is a distribution-free property tester for correctable languages that are testable.

Theorem 3.1. *For any* $n \in \mathbb{N}$, *let* $C \subseteq \{0,1\}^n$ *be an RLCC (i.e., a binary RLCC) with a corrector* C_{cor} *having query complexity* $t(n)$ *and correcting radius* δ_r. *Then for any language* $L \subseteq C$ *and every* $0 < \varepsilon \leq \delta_r$, *if* L *has an* IPP *over the uniform distribution with query complexity* $q(n)$, *communication complexity* $c(n)$ *and round complexity* $R(n)$ *on inputs of length* n, *there exists a distribution-free* IPP *for* L *with query complexity* $O(q(n) + \frac{t(n)}{\varepsilon})$, *communication complexity* $c(n)$ *and round complexity* $r(n)$.

Proof. Let (V_0, P_0) be the IPP for L over the uniform distribution. Recall that the corrector C_{cor} takes inputs $X \in \{0,1\}^n$ and an index $i \in [n]$, and returns the corrected value of X at i. Then we construct a distribution-free IPP (V_{df}, P_0) for L in the following way.

Protocol 1 Distribution-free IPP for a subset of an RLCC C, with corrector C_{cor}.

1. V_{df} runs the uniform IPP between V_0 and P_0 on the input $X \in \{0,1\}^n$. It rejects, if V_0 rejects.
2. Repeat O(1) times:
   (a) V_{df} samples $\frac{1}{\varepsilon}$ points from $\mathcal{O}_{\mathcal{D}}(X)$. Let S be this set.
   (b) V_{df} checks if there exists $i \in S$, $C_{\mathsf{cor}}(X,i) \neq X_i$. If so, it rejects.
3. V_{df} accepts otherwise.

For completeness of (V_{df}, P_0), if $X \in L$, by definition, the honest prover P_0 convinces V_0 to accept with probability at least $\frac{2}{3}$ and by the perfect completeness of the corrector, for every sample i, $C_{\mathsf{cor}}(X,i) = X_i$, therefore V_{df} must accept with probability at least $\frac{2}{3}$.

For soundness, suppose X is ε-far from L along $\mathcal{D}$. Either we have that X is ε-far from L along both $\mathcal{D}$ and $\mathcal{U}$, or it is far only along $\mathcal{D}$. In the first case, if $d_{\mathcal{U}}(X,L) > \varepsilon$, then X is rejected by the uniform IPP with probability at least $\frac{2}{3}$. For the other case, suppose that the uniform IPP accepts X since it is ε-close to L under the uniform distribution. Let c be the closest codeword to the input X along $\mathcal{U}$, i.e., $d_{\mathcal{U}}(X,c) \leq \varepsilon$. If V_{df} does not reject, either for each sampled point in each copy of S, c coincides with X, or there exists an i in some S such that $c_i \neq X_i$, but C_{cor} failed in correcting X_i to the value c_i.

Take any iteration of Step 2. The probability that there exists no sample $i \in S$ for which $X_i \neq c_i$ is at most $(1-\varepsilon)^{\frac{1}{\varepsilon}} \leq \frac{1}{e}$. On the other hand, the probability there exists an $i \in S$ on which the corrector fails (i.e., $C_{cor}(X, i) = X_i \neq c_i$) is at most $\frac{1}{3}$. By a union bound, the probability that V_{df} does not reject in any iteration of this step, is at most $\frac{1}{e} + \frac{1}{3}$ and we can achieve the required soundness by $O(1)$ repetitions.

Clearly, the communication and round complexities are unchanged as the only interactions with the prover are in the uniform IPP. Moreover, the query complexity is just $q(n) + \frac{O(t(n))}{\varepsilon}$. □

3.1.1 Complexity separations via correctability

A language of interest here is the parameterised sub-tensor sum property denoted as TensorSum, that was defined in [GR18]. This language is an example of a subset of a correctable code for which we have MAP lower and upper bounds, testing lower bounds and an IPP upper bound which means we can use this language to demonstrate separations between these complexity classes using these uniform results and our result for RLCCs. The protocols for this problem capture the sumcheck protocol which is one of the most important tools in the field of interactive proofs.

Definition 3.6 ($\mathsf{TensorSum}_{\mathbb{F},m,d,H}$)**.** *Let $\mathbb{F}$ be a finite field and let $H \subset \mathbb{F}$. Let $P : \mathbb{F}^m \to \mathbb{F}$ be a polynomial of individual degree d. Then, P belongs to $\mathsf{TensorSum}_{\mathbb{F},m,d,H}$ iff*

$$\sum_{x \in H^m} P(x) = 0.$$

We now state the query complexity gaps for TensorSum which as we observed earlier, is a subset of all low-degree polynomials, that in turn is an RLCC. We use various results from [GR18] to prove these gaps along with Theorem 3.1.

Theorem 3.2 (Query complexity gaps for distribution-free testing TensorSum)**.** *For any field $\mathbb{F}$, any $m, d \in \mathbb{N}$ such that $d < |\mathbb{F}|$, and any sub-field $H \subseteq \mathbb{F}$, the following hold true for the language $\mathsf{TensorSum}_{\mathbb{F},m,d,H}$.*

1. *For every $\varepsilon \in \left(0, 1 - \frac{dm}{|\mathbb{F}|}\right)$, if $d \geq 2(|H|-1)$, then every distribution-free MAP for TensorSum (with respect to proximity parameter ε) that has proof complexity $p \geq 1$ must have query complexity $q = \Omega\left(\frac{|H|^m}{p \log |\mathbb{F}|}\right)$.*
2. *If $dm < \frac{|\mathbb{F}|}{10}$, then, for every $\ell \in \{0, ..., m\}$, $\mathsf{TensorSum}_{\mathbb{F},m,d,H}$ has a distribution-free MAP with proof complexity $(d+1)^\ell \log(|\mathbb{F}|)$ and query complexity $|H|^{m-\ell}(dm^2 \log |H|) \cdot poly(1/\varepsilon)$.*
3. *If $dm < \frac{|\mathbb{F}|}{10}$, then there exists an m-round distribution-free IPP for $\mathsf{TensorSum}_{\mathbb{F},m,d,H}$ with communication complexity $O(dm \log |\mathbb{F}|)$, and query complexity $O(dm \cdot \mathsf{poly}(1/\varepsilon))$.*

The property testing lower bound here follows from the MAP lower bound from **Lemma 3.15** in [GR18] and from the fact that distribution-free testing is a more general setting which encompasses uniform testing and so will require at least as many queries.

The MAP and IPP upper bound come from **Lemma 3.14** and **Theorem 3.22** respectively from [GR18]. They also require Theorem 3.1 to reduce to the uniform setting.
Theorem 1.6 follows from this.

Proof Sketch of Theorem 1.6. We sketch the proof of the three claims as follows.

1. $\Theta(n^{0.999 \pm o(1)})$ distribution-free query complexity for testing $\mathsf{TensorSum}$:

   This holds from Item 1 of Theorem 3.2, by setting $p = 1$.

2. $\mathsf{TensorSum}$ df-MAP query and communication complexity $\Theta(n^{0.499 \pm o(1)})$ and for proof complexity $p \geq 1$, query complexity $\Theta(\frac{n^{0.999 \pm o(1)}}{p})$:

   For $d = \Theta(|H|) = n^{o(1)}$, $m = \log_{|H|}(n)$, $l = \log_{d+1}(\frac{p}{\log |\mathbb{F}|})$, and $\varepsilon = O(1)$, we obtain the upper bound for query complexity of the MAP in terms of the proof length, from Item 2 of Theorem 3.2 above. The corresponding lower bound is achieved by setting $|\mathbb{F}| = \sqrt{n}$ and $|H|^m = n$, in Item 1 of Theorem 3.2.

   In particular, setting $p = O(\sqrt{n})$ optimises the sum of proof and query complexities, thus proving Item 2.

3. Distribution-free IPP with query and communication complexity $\mathsf{polylog}(n)$:

   This IPP follows from Item 3 of Theorem 3.2, when $|\mathbb{F}| = \sqrt{n}$, $m = \log(n)$ and $d = 2$.

□

3.2 Symmetric Languages

In this section, we show an $\Omega(n^{1/3-0.0005})$ query complexity lower bound for any distribution-free testers for $\mathsf{HAM}(w(n))$, for some fixed $w(n)$. Following this, we construct a distribution-free IPP with $\mathsf{polylog}(n)$ query and communication complexity for any symmetric language. In fact, this is shown by a straight-forward interactive reduction to HAM, and constructing a distribution-free IPP for this problem. Put together, we prove Theorem 1.4, thus demonstrating an exponential advantage in the query complexity given interaction with a prover in the distribution-free setting. We define the problem as follows.

Definition 3.7 (The Hamming weight language)**.** *For any $X \in \{0,1\}^n$, let $\mathsf{Hwt}(X)$ be the Hamming weight (the number of non-zero entries) of X.*

Let $w : \mathbb{N} \to \mathbb{N}$ be a weight function such that for any $n \in \mathbb{N}$, $1 \leq w(n) \leq n$. We define the language $\mathsf{HAM}(w) = \{\mathsf{HAM}_n(w(n))\}_{n \in \mathbb{N}}$, such that for any $X \in \{0,1\}^n$, $X \in \mathsf{HAM}_n(w(n)) \iff \mathsf{Hwt}(X) = w(n)$.

The (parameterised) Hamming weight language is one of a general class of languages that are *symmetric*, i.e., those which depend only on the Hamming weight of the input string.

Definition 3.8 (Symmetric Languages)**.** *A language $L = \{L_n\}$ is called Symmetric if and only if for every $n \in \mathbb{N}$, there exists a predicate $\mathcal{S}_n : \{0, \ldots, n\} \to \{0,1\}$ such that*

$$L_n = \{X \in \{0,1\}^n \mid \mathcal{S}_n(\mathsf{Hwt}(X)) = 1\}$$

3.2.1 Testing Lower Bound

Theorem 3.3. *For any $\varepsilon > 0$, $n \in \mathbb{N}$, there exists $w = w(n) > 0$ such that any distribution-free property tester for $\mathsf{HAM}(w)$ requires $\Omega(n^{1/3-0.0005})$ queries.*

We prove this theorem via the following steps:

1. We construct two pairs $(\mathcal{D}_1, X), (\mathcal{D}_2, Y)$ of a distribution and an input to $\mathsf{HAM}(w)$, where $w = \mathsf{Hwt}(Y)$. The first pair is a NO instance, i.e., $d_{\mathcal{D}_1}(X, \mathsf{HAM}(w)) > \varepsilon$ and a tester should reject this. On the other hand, the second pair is a YES instance, where Y must be accepted irrespective of the access to the sample oracle with respect to any distribution.

2. We next show that the inputs X, Y are close along $\mathcal{U}$. In other words, with high probability, $o(n^{1/3-0.0005})$ queries made by the tester along the uniform distribution do not help it distinguish between X and Y.

3. Further, we show that $o(n^{1/3-0.0005})$ values in X along samples from $\mathcal{D}_1$ and those in Y along samples from $\mathcal{D}_2$ will be distributed in the same way ($\mathbb{P}_{i\sim\mathcal{D}_1}[X_i = 1] = \mathbb{P}_{i\sim\mathcal{D}_2}[Y_i = 1]$).

   By our construction, with high probability, there are no collisions between the samples from $\mathcal{U}$ and $\mathcal{D}_1$, or $\mathcal{U}$ and $\mathcal{D}_2$. This means that the tester can't distinguish between the 2 distribution-input pairs by sampling along any of these distributions, with high probability.

4. To conclude, we show a transformation from *any query-optimal tester* for HAM, to one that only uses uniformly sampled queries to the input, along with samples from the underlying distribution oracle (i.e., along $\mathcal{D}_1$ for testing X or $\mathcal{D}_2$ for testing Y). Putting this together with the fact that the uniform sampled queries are distinct (with high probability) from the samples along $\mathcal{D}_1$ or $\mathcal{D}_2$, we get query lower bound of $\Omega(n^{1/3-0.0005})$ for testing HAM.

We first define the pairs $(\mathcal{D}_1, X), (\mathcal{D}_2, Y)$. Consider the partition of $[n]$ into the following 3 intervals.

$$I_1 = [1, n - n^{\frac{2}{3}} - n^{\frac{2}{3}-0.001}].$$
$$I_2 = [n - n^{\frac{2}{3}} - n^{\frac{2}{3}-0.001} + 1, n - n^{\frac{2}{3}-0.001}].$$
$$I_3 = [n - n^{\frac{2}{3}-0.001} + 1, n].$$

The distributions $\mathcal{D}_1, \mathcal{D}_2 \in \Delta(\Omega_n)$ are defined as follows.

$$\begin{array}{rr}
\forall i_0 \in I_1 : \underset{i\sim\mathcal{D}_1}{\mathbb{P}}[i = i_0] = \frac{1-20\varepsilon}{|I_1|}. & \forall i_0 \in I_1 : \underset{i\sim\mathcal{D}_2}{\mathbb{P}}[i = i_0] = \frac{1-20\varepsilon}{|I_1|}. \\
\forall i_0 \in I_2 : \underset{i\sim\mathcal{D}_1}{\mathbb{P}}[i = i_0] = \frac{12\varepsilon}{|I_2|}. & \forall i_0 \in I_2 : \underset{i\sim\mathcal{D}_2}{\mathbb{P}}[i = i_0] = \frac{8\varepsilon}{|I_2|}. \\
\forall i_0 \in I_3 : \underset{i\sim\mathcal{D}_1}{\mathbb{P}}[i = i_0] = \frac{8\varepsilon}{|I_3|}. & \forall i_0 \in I_3 : \underset{i\sim\mathcal{D}_2}{\mathbb{P}}[i = i_0] = \frac{12\varepsilon}{|I_3|}.
\end{array}$$

Finally, we define $X, Y \in \{0,1\}^n$ as having a fixed proportion of bits in each interval set to 1, as follows.

- $\forall i \in I_1$, $X_i = Y_i = 1$.
- For I_2:
  - $\mathsf{Hwt}(X|_{I_2}) = \frac{|I_2|}{3}$, i.e. X takes value 1 for $\frac{1}{3}$ of these indices.

    In particular, $\forall i \in [n - n^{\frac{2}{3}} - n^{\frac{2}{3}-0.001} + 1, n - \frac{2}{3}n^{\frac{2}{3}} - n^{\frac{2}{3}-0.001} + 1]$, $X_i = 1$, for all other $i \in I_2$, $X_i = 0$.

  - $\mathsf{Hwt}(Y|_{I_2}) = \frac{|I_2|}{2}$, i.e. Y takes value 1 for $\frac{1}{2}$ of these indices.
    In particular, $\forall i \in [n - n^{\frac{2}{3}} - n^{\frac{2}{3}-0.001} + 1, n - \frac{1}{2}n^{\frac{2}{3}} - n^{\frac{2}{3}-0.001} + 1]$, $Y_i = 1$, for all other $i \in I_2$, $Y_i = 0$.

and

- For I_3:
  - $\mathsf{Hwt}(X|_{I_3}) = \frac{|I_3|}{2}$, X takes value 1 for $\frac{1}{2}$ of these indices
    That is, $\forall i \in [n - n^{\frac{2}{3}-0.001} + 1, n - \frac{1}{2}n^{\frac{2}{3}-0.001} + 1]$, $X_i = 1$, for all other $i \in I_3$, $X_i = 0$.
  - $\mathsf{Hwt}(Y|_{I_3}) = \frac{|I_3|}{3}$, Y takes value 1 for $\frac{1}{3}$ of these indices.
    That is, $\forall i \in [n - n^{\frac{2}{3}-0.001} + 1, n - \frac{2}{3}n^{\frac{2}{3}-0.001} + 1]$, $Y_i = 1$, for all other $i \in I_3$, $Y_i = 0$.

We first show that with $n^{\frac{1}{3}-0.0005}$ uniformly sampled indices, we expect to only receive elements of I_1 on which X and Y are identical.

Lemma 3.4. *With high probability, $t = o(n^{\frac{1}{3}})$ uniform samples will only return indices i for which $X_i = Y_i = 1$. In other words:*

$$\mathbb{P}_{i \sim \mathcal{U}^t}\left[\forall j \in [t] : X_{i_j} = Y_{i_j} = 1\right] = 1 - o(1) \tag{2}$$

Proof. The probability of each of these samples having this property is

$$\begin{aligned}\mathbb{P}_{i \sim \mathcal{U}_n^t}\left[\forall j \in [t] : X_{i_j} = Y_{i_j} = 1\right] &\geq \left(\mathbb{P}_{i \sim \mathcal{U}_n}\left[i_j \in I_1\right]\right)^t \\ &= \left(\frac{n - 2n^{-\frac{2}{3}}}{n}\right)^t \\ &= \left(1 - \frac{2}{n^{\frac{1}{3}}}\right)^t \\ &\geq 1 - o(1)\end{aligned}$$

□

Then we show that sampling along the corresponding distributions returns an index on which the probability that the input is 1 is the same for both instances.

Lemma 3.5. $\mathbb{P}_{i \sim \mathcal{D}_1}\left[X_i = 1\right] = \mathbb{P}_{i \sim \mathcal{D}_2}\left[Y_i = 1\right]$

Proof. We evaluate both probabilities by summing over each interval as follows

$$\begin{aligned}\underset{i \sim \mathcal{D}_1}{\mathbb{P}}\left[X_i = 1\right] &= \sum_{j \in \{1,2,3\}} \underset{i \sim \mathcal{D}_1}{\mathbb{P}}\left[X_i = 1 | i \in I_j\right] \underset{i \sim \mathcal{D}_1}{\mathbb{P}}\left[i \in I_j\right] \\ &= \frac{1 - 20\varepsilon}{|I_1|}|I_1| + \frac{12\varepsilon}{|I_2|}|I_2|\frac{1}{3} + \frac{8\varepsilon}{|I_3|}|I_3|\frac{1}{2} \\ &= 1 - 12\varepsilon\end{aligned}$$

$$\begin{aligned}\underset{i\sim\mathcal{D}_2}{\mathbb{P}}[Y_i=1] &= \sum_{j\in\{1,2,3\}} \underset{i\sim\mathcal{D}_2}{\mathbb{P}}[Y_i=1|i\in I_j]\underset{i\sim\mathcal{D}_2}{\mathbb{P}}[i\in I_j] \\ &= \frac{1-20\varepsilon}{|I_1|}|I_1| + \frac{8\varepsilon}{|I_2|}|I_2|\frac{1}{2} + \frac{12\varepsilon}{|I_3|}|I_3|\frac{1}{3} \\ &= 1-12\varepsilon.\end{aligned}$$

□

We then show that for each pair, the probability of sampling the same element twice from the distribution oracle and from uniform distribution is very small. The following notation is useful. For any $t\in\mathbb{N}$, by $\mathcal{U}^t\times\mathcal{D}^t$ we denote the $2t$-length tuple of indices that consists of t i.i.d. samples from $\mathcal{U}$ and t i.i.d. samples from $\mathcal{D}$, drawn independently of each other.

Lemma 3.6. *For each $j\in\{1,2\}$, sampling $t=o(n^{\frac{1}{3}-0.0005})$ times along $\mathcal{D}_j$ or along $\mathcal{U}$ results in distinct indices(no collisions) with high probability. In other words:*

$$j\in\{1,2\}\implies \mathbb{P}_{(i_1,\cdots,i_{2t})\sim\mathcal{U}^t\times\mathcal{D}_j^t}[\exists\ell,k\in[2t]:\ell\neq k\wedge i_\ell=i_k]\geq 1-o(1) \tag{3}$$

Proof. We prove this statement for $\mathcal{D}_1$ and the other case is analogous to this. For any $m>0$, $o(\sqrt{m})$ uniform samples from $[m]$ will be distinct with probability at least $1-o(1)$ (this follows from the birthday paradox). Each of $\sqrt{|I_1|}$, $\sqrt{|I_2|}$, and $\sqrt{|I_3|}=\Omega(n^{\frac{1}{3}-0.0005})$ and therefore with high probability, samples from $\mathcal{D}_1$ within any of these intervals will not produce a collision. This is because restricted to each of the intervals I_1,I_2,I_3, $\mathcal{D}_1$ is uniform and therefore even if all the samples were just on one of these intervals, then with high probability all of the indices sampled will be distinct.

Similarly, with high probability the t samples along the uniform distribution over $[n]$ won't collide with each other. Additionally the probability that they don't collide with the samples from $\mathcal{D}_1$ is at least $\left(1-\frac{o(n^{\frac{1}{3}})}{n}\right)^{o(n^{\frac{1}{3}})}\geq 1-o(1)$. □

Finally, we need the following result which shows that the only relevant queries for property testing HAM, are those that are uniformly sampled. Intuitively, this holds since HAM is a symmetric language: we can permute the indices of the input and the Hamming weight remains unchanged.

Lemma 3.7. *For any $w\in\mathbb{N}$ and any distribution-free testing algorithm $\mathcal{A}$ for $\mathsf{HAM}(w)$ with query complexity T and sample complexity S such that $T+S=o(\sqrt{n})$, there exists an equivalent distribution-free testing algorithm $\mathcal{A}'$ with $O(T)$ queries and $O(S)$ samples, such that $\mathcal{A}'$ only makes input queries along indices sampled from $\mathcal{U}$.*

Proof. Fix some $w\in\mathbb{N}$ and a distribution-free testing algorithm $\mathcal{A}$ for $\mathsf{HAM}(w)$ using T queries to X, some of which are possibly adaptive. Without loss of generality, suppose that $\mathcal{A}$ gets all its labelled samples from $\mathcal{O}_{\mathcal{D}}$ at the beginning. Following this, we know that $\mathcal{A}$ must have the same output (with high probability) for any pair of inputs which are consistent on these indices sampled from $\mathcal{D}$.

Let i be the last query that $\mathcal{A}$ makes, which is not a uniformly sampled index. Let I be the set of indices already queried and J be the set of indices sampled from $\mathcal{D}$. Let j be a uniformly random sample from $[n]$. In particular, $j \notin I \cup J$ with probability $\frac{n-o(\sqrt{n})}{n} = 1 - o(\frac{1}{\sqrt{n}})$.

For any $X \in \{0,1\}^n$, define $X^{(i,j)}$ as an input string with the values of X swapped at indices i and j, and similarly, $\mathcal{D}^{(i,j)}$ where the probabilities of sampling i and j are swapped. Let $\hat{\mathcal{A}}$ be the algorithm that follows the same queries as $\mathcal{A}$, however querying j instead of i, and whose outcome is the same as $\mathcal{A}$ on the input $X^{(i,j)}$ (i.e., $\mathcal{A}$ gets the value X_j when it queries i). This follows as long as $j \notin I \cup J$, which as we saw earlier, happens with high probability.

Next, we have the following fact for any $\mathcal{D} \in \Delta([n])$ and any $X \in \{0,1\}^n$:

$$\mathbb{P}_{\mathcal{D},\hat{\mathcal{A}}}\left[\hat{\mathcal{A}}(X) = 1\right] \geq \mathbb{P}_{\mathcal{D}^{(i,j)},\mathcal{A}}\left[\mathcal{A}(X^{(i,j)}) = 1\right] - o\left(\frac{1}{\sqrt{n}}\right). \tag{4}$$

Indeed, if $\mathcal{A}$ accepts $X^{(i,j)}$ when given a set of samples from $\mathcal{D}^{(i,j)}$ and coin flip outcomes, $\hat{\mathcal{A}}$ must accept X when sampling from $\mathcal{D}$ and having the same coin flip outcomes so long as no collision occurs which is accounted for by the $o\left(\frac{1}{\sqrt{n}}\right)$ term . In other words,

$$\begin{aligned}\mathbb{P}_{\mathcal{D},\hat{\mathcal{A}}}\left[\hat{\mathcal{A}}(X) = 1\right] &\geq \mathbb{P}_{\mathcal{D},\hat{\mathcal{A}}}\left[\hat{\mathcal{A}}(X) = 1 \wedge j \notin I \cup J\right] \\ &\geq \mathbb{P}_{\mathcal{D}^{(i,j)},\mathcal{A}}\left[\mathcal{A}(X^{(i,j)}) = 1\right] - o\left(\frac{1}{\sqrt{n}}\right).\end{aligned}$$

For each pair $(X^{(i,j)}, \mathcal{D}^{(i,j)})$, $\mathcal{A}$ returns the correct result with probability at least $\frac{2}{3}$. Therefore, from Equation 4, $\hat{\mathcal{A}}$ also returns the correct value on $(X, \mathcal{D})$ with the same probability up to an additive factor of $o\left(\frac{1}{\sqrt{n}}\right)$. In other words, repeating this at most T times for all arbitrary queries, results in a new tester $\mathcal{A}'$. This must be a distribution-free property tester for HAM using T queries which are all uniformly sampled and S samples from $\mathcal{D}$, with success probability at least $\frac{2}{3} - o\left(\frac{T}{\sqrt{n}}\right) = \frac{2}{3} - o(1)$. By repeating this tester $O(1)$ times, we can recover the original $\frac{2}{3}$ success probability. □

Given these lemmas, we now prove the main result of this section.

Proof of Theorem 3.3. Let $\mathsf{Hwt}(Y) = w$. By Lemmas 3.4, 3.5 and 3.6 we know that the labeled samples along $\mathcal{D}_1$ for X, or $\mathcal{D}_2$ for Y, along with queries made along U are not enough to distinguish whether X is the input tested against distribution $\mathcal{D}_1$, or Y is the input tested against distribution $\mathcal{D}_2$. Due to Lemma 3.7, we know that any distribution-free property tester can only use those queries to distinguish the two cases.

Therefore, it suffices to show that X is ε-far from $\mathsf{HAM}(w)$ along $\mathcal{D}_1$ as then it becomes impossible to distinguish $(X, \mathcal{D}_1)$ from $(Y, \mathcal{D}_2)$ with $o(n^{\frac{1}{3}-0.0005})$ queries to the input. The Hamming distance between X and Y is $\frac{1}{6}(|I_2| - |I_3|) = \frac{n^{2/3}}{6}(1 - n^{-0.001})$. This is the number of bits of X that need to be flipped from 0 to 1 to have Hamming weight same as Y. This can be done by flipping either the elements of I_2 or I_3 (note that X and Y have the same values in the set I_1). Furthermore, along $\mathcal{D}_1$, the weight of elements in I_2 is less than I_3; so in total the distance $d_{\mathcal{D}_1}(X, w) = \frac{n^{2/3}}{6}(1 - n^{-0.001})\frac{12\varepsilon}{|I_2|} > \varepsilon$. □

3.2.2 Upper Bound for Symmetric Languages

We next present a distribution-free IPP for $\mathsf{HAM}(w(n))$, for every weight function $w(n) \leq n$.

Theorem 3.8. *For every $n \geq 2$ and every $w \leq n$, there exists a distribution-free* IPP *for* $\mathsf{HAM}(w)$ *with perfect completeness and soundness $\frac{1}{3}$. This protocol has round complexity $O\left(\frac{2\log(n)}{\varepsilon}\right)$, communication complexity $O\left(\frac{\log(n)^2}{\varepsilon}\right)$ and sample complexity $\frac{1}{\varepsilon}$. Furthermore, the* IPP *does not make any queries to the input and only uses samples from the distribution.*

There are $O\left(\frac{1}{\varepsilon}\right)$ rounds in this IPP, in each round the verifier samples an index $i \sim \mathcal{D}$ and then performs a binary split across the indices in [n] as follows. Set $I = [n]$, $I_0 = [\lfloor n/2 \rfloor]$ and $I_1 = [\lfloor n/2 \rfloor + 1, n]$, the prover sends the Hamming weight of both I_0 and I_1. The protocol iterates by updating I to be set to whichever of I_0, I_1 contains i and iterate thereon. We iterate until $I = \{i\}$, the verifier queries X_i.

Protocol 2 Interactive Proof of ε-proximity for $\mathsf{HAM}(w)$

1. Repeat $O\left(\frac{1}{\varepsilon}\right)$ times:
   (a) The verifier samples (i, X_i) from $\mathcal{O}_{\mathcal{D}}$. Let $i = (i_0, \cdots i_{\lceil \log(n)-1 \rceil})_2$.
   (b) Set $I = [n]$, $v = w$, $L = 0$, $U = n$, $r = 0$ and $s = \emptyset$.
   (c) Repeat the following until $|I| = 1$.
      i. The verifier partitions I into two parts $I_0 = \left[L, \lfloor \frac{U+L}{2} \rfloor\right], I_1 = \left(\lfloor \frac{U+L}{2} \rfloor, U\right]$.
      ii. Let the Hamming weight of X over I_0, I_1 is h_{s0}, h_{s1} respectively. The prover sends h'_{s0}, h'_{s1} which is the purported value of h_{s0} and h_{s1} respectively.
      iii. The verifier rejects if $h'_{s0} + h'_{s1} \neq v$, or if either of $h'_{s0} \notin [0, |I_0|]$ or $h'_{s1} \notin [0, |I_1|]$.
      iv. The verifier sends i_r to the prover. If $i_r = 0$ then reassign $I = I_0$, $U = \lfloor \frac{U+L}{2} \rfloor$, $s = s0$, $v = h'_{s0}$ otherwise reassign $I = I_1$, $L = \lfloor \frac{U+L}{2} \rfloor + 1$, $s = s1$, $v = h'_{s1}$.
      In both cases set $r = r + 1$.
   (d) Finally, $X_i \neq v$, the verifier rejects.
2. The verifier accepts otherwise.

Proof. Fix some $w \leq n$. Let $X \in \{0,1\}^n$ be the input to $\mathsf{HAM}(2)$. For any $i \in [n]$, we write $i = (i_0, \cdots i_{\lceil \log(n-1) \rceil})_2$ as its binary expansion. The full IPP for $\mathsf{HAM}(w)$ is given in Protocol 2. Clearly, its sample complexity is $\frac{2}{\varepsilon}$ and it makes no additional queries to X. Below, we prove its correctness.

Completeness: If $X \in \mathsf{HAM}(w)$, completeness follows from the fact that the honest prover will only provide the genuine values for h_{s0}, h_{s1} in each iteration of the binary search. At the end of each of the $O\left(\frac{1}{\varepsilon}\right)$ rounds, the final value in v is the Hamming weight of X on the interval $I = \{i\}$ which is X_i, therefore the verifier will not reject.

Soundness: Suppose $d_{\mathcal{D}}(X, \mathsf{HAM}(w)) > \varepsilon$. For each round of Step 1, we show that the probability that the verifier catches the prover is at least ε. Let $Y = h'_{00\cdots0}h'_{00\cdots1}\cdots h'_{11\cdots1}$ be the

n-length Boolean string concatenating all possible values of h'_i that the prover sends the verifier in the final iteration of the binary search process (here the i in h'_i is represented in binary).

It is sufficient to prove that an optimal dishonest prover strategy (which is rejected with the lowest probability at the latest possible point) will have the property that $\mathsf{Hwt}(Y) = w$. By our assumption, we have that $d_{\mathcal{D}}(X, Y) \geq d_{\mathcal{D}}(X, \mathsf{HAM}(w)) > \varepsilon$. Thus, with probability at least ε over the choice of the sample i from $\mathcal{D}$, the verifier rejects in a single round. In other words, repeating $O\left(\frac{1}{\varepsilon}\right)$ times will achieve the desired soundness probability.

Claim 3.9. *For all values of $n \geq 2$, if we assume in the soundness case that the cheating prover strategy maximises the probability the verifier accepts, $\mathsf{Hwt}(Y) = w$.*

Proof. We proceed by induction on the length of the binary representation of n, denoted by $R = \lceil \log(n) \rceil$.

For the base case, suppose $R = \lceil \log(n) \rceil = 1 \iff n = 2$. Here the input length $n = 2$ and thus, $w \leq 2$. In the one round of the binary split the prover sends h'_0, h'_1 for which $Y = h'_0 h'_1$ so that the value of $\mathsf{Hwt}(Y)$ has to be equal to w as if the prover sends values such that $h'_0 + h'_1 \neq w$, the verifier will reject before it checks its sample in step 1d with probability 1, so the optimal prover strategy will send values such that $\mathsf{Hwt}(Y) = w$.

For some $k \in \mathbb{N}$, suppose that $\mathsf{Hwt}(Y) = w$ for $R = k - 1$, let us take $R = k$. In the first round of the binary split: we have that $h'_0 + h'_1 = w$, as otherwise the verifier would immediately reject with probability 1, let $Y[0]$ be the restriction of Y to I_0, define $Y[1]$ similarly.

By the inductive assumption as $\lfloor \log(|I_0|) \rfloor = k - 1$, we have $\mathsf{Hwt}(Y[0]) = h'_0$, and similarly $\mathsf{Hwt}(Y[1]) = h'_1$. Therefore, $\mathsf{Hwt}(Y) = \mathsf{Hwt}(Y[0]) + \mathsf{Hwt}(Y[1]) = h'_0 + h'_1 = w$. □

Round complexity: This is the number of times the prover sends h_{s0}, h_{s1} which is $O\left(\frac{\log(n)}{\varepsilon}\right)$.

Communication complexity: The prover sends the values of h_{s0} and h_{s1} at each iteration of the binary search. This happens $\lceil \log(n) \rceil$ many times for each sample, and there are $O\left(\frac{\log(n)}{\varepsilon}\right)$ samples chosen. The verifier only sends the prover $O\left(\frac{\log(n)}{\varepsilon}\right)$ bits of information over the course of the IPP. Therefore in total, the communication complexity is $O\left(\frac{\log(n)^2}{\varepsilon}\right)$. □

Remark 7. *It is worth noting that the distribution-free* IPP *from Theorem 3.8 offers a quadratic improvement in the total "access" complexity (sample complexity and query complexity) over both its distribution-free tester [CEG95], as well as what we get by using the generic distribution-free* IPP *from Theorem 5.2 (since any symmetric language is computable in NC^1).*

Finally, we observe that the distribution-free IPP for $\mathsf{HAM}(w)$ easily extends to any symmetric language in the following corollary.

Corollary 3.10. *Any symmetric language L has a distribution-free* IPP *with round complexity $O\left(\frac{\log(n)}{\varepsilon}\right)$, communication complexity $O\left(\frac{\log(n)^2}{\varepsilon}\right)$ and sample complexity $O\left(\frac{1}{\varepsilon}\right)$.*

Proof. For any input $X \in \{0,1\}^n$, this is clear from the fact that by having the prover send the $w' = \mathsf{Hwt}(X)$, we reduce this problem to an instance of $\mathsf{HAM}(w')$, so that the completeness and soundness conditions follow from its distribution-free IPP. The round complexity only increases by 1 and the sample complexity is unchanged but the communication complexity increases by $\log(n)$

for the number of bits sent to specify w'. This leaves the communication complexity unchanged at $O(\frac{\log^2(n)}{\varepsilon})$. □

3.3 Separation between IPPs and Distribution-Free IPPs

In this section, we demonstrate that not all languages with a uniform IPP have a distribution-free IPP with the same query and communication complexity. We do so by showing the existence of a language L that has an efficient IPP but for which any distribution-free IPP requires a very large complexity. As a matter of fact, the result is even stronger as the IPP for L is essentially just a property tester - that is, the prover is not required.

The proof is rather straightforward - we construct a language L in which a very small portion of the input consists of a very hard property. A standard property tester can safely ignore this part of the input (since it is small), but for a distribution-free IPP, the distribution could be entirely concentrated on this small portion.

Proposition 3.11. *Suppose that there exists $\varepsilon > 0$ s.t. for all $q = q(n) \leq n$, there exists a language L and a function $\ell(n)$ s.t. for any uniform* IPP *(respectively uniform* MAP*) for L, with proximity parameter ε, query complexity $q = q(n)$, and communication complexity $c = c(n)$, $\max(q, c) \geq \ell(n)$, then the following is true.*

For all $q = q(n) < \frac{\varepsilon}{2} \cdot n$, there exists a language $L' = \bigcup_{n \in \mathbb{N}} L'_n$ which has the following properties:

1. *The property testing query complexity for L' with proximity parameter ε is $O(1/\varepsilon)$.*
2. *For any distribution-free* IPP *(respectively* MAP*) for L' with proximity parameter ε, query complexity $q(n)$, and communication complexity $c(n)$, it holds that $\max(q(n), c(n)) \geq \ell(n')$, where $n' = (\varepsilon/2) \cdot n$.*

Proposition 1.7 follows from Proposition 3.11 using known results. Specifically, [KR15, Theorem 4] shows a language for which the conditions of Proposition 3.11 hold for $\ell(n) = \Omega(\sqrt{n})$ (assuming the PRGs from the hypothesis of Proposition 1.7). The second item of Proposition 1.7 follows from [GR18, Theorem 4], which shows there exists a language with no efficient MAP satisfying the criterion for $\ell(n) = \Omega(n)$.

Proof of Proposition 3.11. We prove the theorem w.r.t. to IPPs (i.e., where both the assumption and conclusion are for IPPs). The result for MAPs is proved in exactly the same way.

Suppose there exists $\varepsilon > 0$ as in the statement of Proposition 3.11. Let $q = q(n) \leq \frac{\varepsilon}{2} \cdot n$ and $L = \bigcup_{n \in \mathbb{N}} L_n$ be a language such that every IPP for L with wrt proximity parameter ε with query complexity q has communication complexity c such that $\max(q, c) \geq \ell(n)$.

We construct a language L'_n as follows.

$$L'_n = \left\{(0^{(1-\varepsilon/2)\cdot n}, x) \mid x \in L_{(\varepsilon/2)\cdot n}\right\}$$

Thus, inputs to L'_n consists of an input of length $(\varepsilon/2) \cdot n$ of L, which is preceded by $(1 - \varepsilon/2) \cdot n$ zeros. The high level idea is that since the vast majority of the input is always 0, the language L' is easy to test. However, in the distribution-free setting, we can concentrate all of the “weight” of the distribution on the suffix. Details follow.

Indeed, in the uniform case, there is a trivial property tester for L' that queries the input $O(1/\varepsilon)$ times sampled uniformly over the first $\left(1 - \frac{\varepsilon}{2}\right) \cdot n$ bits and accepts if and only if all the queries return 0. For any $I \subseteq [n]$, we denote by $X|_I$ the substring of X restricted to I. Completeness follows as the tester will accept for $X \in L'$ as all queries return 0 by the definition of L'. Soundness follows as for $d_{\mathcal{U}}(X, L') > \varepsilon$, by the triangle inequality, $d_{\mathcal{U}}(X|_{[n\cdot(1-\frac{\varepsilon}{2})]}, 0^{n-(\varepsilon/2)n}) > \varepsilon - \varepsilon/2 \geq \varepsilon/2$ and the tester will query a non-zero entry with high probability after $O(1/\varepsilon)$ queries.

We proceed to the distribution-free setting. Let $\mathcal{D}$ be the distribution which is uniform over the last $(\varepsilon/2)\cdot n$ bits. Suppose there exists an IPP for L' with query complexity q and communication complexity c along $\mathcal{D}$ with proximity parameter ε. This IPP immediately yields yields a uniform IPP for L for input size $\frac{\varepsilon}{2}\cdot n$ and proximity parameter ε. This follows as the distance of X from L' is now the uniform distance from X restricted to $\left[n - \frac{\varepsilon}{2}\cdot n + 1, n\right]$ to L. In other words,

$$d_{\mathcal{D}}(X, L') = d_{\mathcal{U}}(X|_{\left[n-\frac{\varepsilon}{2}\cdot n+1,n\right]}, L).$$

Thus, for any distribution-free IPP for L'_n with query complexity $q(n)$ and communication complexity $c(n)$, there is a uniform IPP for inputs of length $n' = \frac{\varepsilon}{2}\cdot n$ for L with query complexity $q'(n') = q(n)$ and communication complexity $c'(n') = c(n)$. By the definition of the language L, $\max(q(n), c(n)) \geq \ell(n')$ by the property of uniform IPPs for L. □

4 Distribution-Free IPPs for NC

Recall that the class NC consists of languages computable by a sequence of Boolean circuits $\{C_n\}_{n\in\mathbb{N}}$ of polynomial size and poly-logarithmic depth. In this section, we construct a distribution-free IPP for any language in logspace-uniform NC that uses $O(1/\varepsilon)$ queries, as well as $O(1/\varepsilon)$ samples from the input, and $O\left(n\cdot\varepsilon\cdot\mathsf{polylog}(n) + \frac{\log(n)}{\varepsilon}\right)$ bits of communication, on inputs of length n and proximity parameter ε.

Our approach extends the strategy in [RVW13] to the more involved setting of distribution-free testing. In Section 4.2, we show that the polynomial evaluation problem PVAL, defined in Section 4.1, is not only 'complete' for constructing uniform IPPs for languages computable by low-depth circuits, but also for constructing *distribution-free* IPPs in the following sense: we reduce the task of proving proximity to any such language over an unknown (but fixed) distribution, to proving proximity to PVAL with respect to a new, *hybrid-metric* notion of soundness. This interactive reduction is used in all our distribution-free IPPs in this paper.

It is worth noting that analysis over this hybrid metric provides for a better exposition for Sections 5 and 6, and we also use it here to maintain overall consistency. In essence, it captures the case analysis explained in Section 1.3.1.

Finally, in Section 4.3 we prove that in fact, constructing distribution-free IPPs for any language computable in low-depth reduces to constructing an IPP for PVAL *only* over the *uniform distribution.* We restate the main result of this section.

Theorem 4.1 (**Distribution-Free IPP for languages computable in low depth**). *For every language L in logspace-uniform NC and every $\varepsilon > \frac{200\log^3(n)}{n}$, there exists a distribution-free IPP for L with proximity ε, having query complexity $O\left(\frac{1}{\varepsilon}\right)$, sample complexity $O\left(\frac{1}{\varepsilon}\right)$ and communication complexity $O\left(\frac{\log(n)}{\varepsilon}\right) + \varepsilon\cdot n\cdot\mathsf{polylog}(n)$. Moreover, the verifier runs in time $\tilde{O}\left(\frac{1}{\varepsilon} + \varepsilon\cdot n\right)$, the prover runs in time $\mathsf{poly}(n)$ and the round complexity is $\mathsf{polylog}(n)$.*

Note that this parameterisation is different from the one stated in Theorem 1.1 and we use it for more convenience in our analysis. The argument that Theorem 4.1 implies Theorem 1.1 follows. For any input length $n \in \mathbb{N}$, let the trade-off parameter τ be such that $\tau \leq \sqrt{n}$. Applying Theorem 4.1 with the proximity parameter $\varepsilon' = min(\varepsilon, 1/\tau)$, we obtain a distribution-free IPP that has query and sample complexities $O(1/\varepsilon') \leq \tau + O(1/\varepsilon)$, communication complexity $O\left(\frac{\log(n)}{\varepsilon} + \log(n) \cdot \tau\right) + (n/\tau) \cdot \mathsf{polylog}(n)$, and verifier runtime $\tilde{O}\left(\tau + \frac{n}{\tau} + \frac{1}{\varepsilon}\right)$, as stated in Theorem 1.1. On the other hand, the converse simply follows by setting $\tau = O(1/\varepsilon)$ in Theorem 1.1.

In particular, for the case when $\varepsilon \geq 1/\sqrt{n}$, we obtain a distribution-free IPP for NC that achieves the optimum of the trade-off between the sum of query and communication complexities. Formally,

Corollary 4.2. *For every language L in logspace-uniform* NC *and every $\varepsilon \geq 1/\sqrt{n}$, there exists a distribution-free* IPP *for L with proximity parameter ε, having query and sample complexities $O(\sqrt{n})$, with the communication complexity and verifier running time being at most $\tilde{O}(\sqrt{n})$.*

4.1 The Polynomial Evaluation Problem

Let $\mathbb{F}$ be a finite field, and $[k]$ be identified with a subset of $\mathbb{F}$ of size $k \in \mathbb{N}$ via some bijection. Let $m, n \in \mathbb{N}$ be integers such that $n = k^m$. We start by defining the low-degree extension of a string in $\mathbb{F}^{k^m}$.

Definition 4.1 (Low-degree Extension (LDE))**.** *For any $X \in \mathbb{F}^n$ $\left(\text{or alternatively } X \in \mathbb{F}^{k^m}\right)$, we define $P_X : \mathbb{F}^m \to \mathbb{F}$ as the unique m-variate polynomial over $\mathbb{F}$ with individual degree at most $k-1$, such that it evaluates to X on $[k]^m$.*

We next define the polynomial evaluation problem PVAL. For any fixed set of points $J \subset \mathbb{F}^m$ and a vector of values $\vec{v} \in \mathbb{F}^{|J|}$, PVAL parameterised by $J, \vec{v}$, is the problem of deciding whether the LDE of the given input $X \in \mathbb{F}^n$ is consistent with $\vec{v}$ on the points in J. Formally, the language PVAL is defined as follows.

Definition 4.2 (PVAL, [RVW13])**.** PVAL *is a language parameterised by $(\mathbb{F}, k, m, J, \vec{v})$, where $J \subset \mathbb{F}^m$ and $\vec{v} \in \mathbb{F}^{|J|}$, such that an input $X \in \mathbb{F}^{k^m}$ belongs to* $\mathsf{PVAL}(\mathbb{F}, k, m, J, \vec{v})$ *if and only if for every $j \in J$, $P_X(j) = \vec{v}_j$. Equivalently, this can be stated as saying that $P_X(J) = \vec{v}$.*

When it is clear from context, we drop $\mathbb{F}, k, m$ from the explicit input (i.e., the string of input parameters) and consider X as an instance of $\mathsf{PVAL}(J, \vec{v})$.

4.2 Distribution-Free Interactive reduction from NC to PVAL

We first demonstrate an interactive protocol that reduces the problem of constructing a distribution-free IPP for any low-depth computable language L to an instance of PVAL, but over a *hybrid* distance metric. In more detail, we depart from the interactive reduction by [RVW13] for the uniform setting, by showing that in the soundness case where the input is ε-far along $\mathcal{D}$ from L, this generalised reduction produces an instance for PVAL that only satisfies a *weaker promise* with respect to a new distance measure, instead of guaranteeing distance with respect to $\mathcal{D}$. Complementing this, our new task will be to design IPPs for PVAL which are powerful enough to reject a larger set of inputs that satisfy this weaker soundness promise, which we show in Section 4.3 (see also Sections 5.3 and 6.2).

The soundness condition in our case is based on a hybrid metric defined across the underlying distribution $\mathcal{D}_n$ and the uniform distribution $\mathcal{U}_n$. We say that on a given input X, if for every $Y \in$ PVAL: $d_{\mathcal{U}}(X, Y) \geq \varepsilon$ or $d_{\mathcal{D}}(X, Y) \geq \varepsilon$, then the IPP should reject the input X (this is exactly when PVAL does not intersect the shaded region of Figure 1). The set of inputs X ε-far along $\mathcal{D}$ from PVAL is a subset of this collection of inputs we need to reject. We explain the reason for reducing to this metric in the following section.

4.2.1 Protocol Intuition for the Interactive Reduction

Let L be a language in NC. For any input $X \in \{0,1\}^n$ for L, and any arbitrary but fixed distribution $\mathcal{D} \in \Delta([n])$, the interactive reduction runs $t = O(\varepsilon \cdot n \cdot \log(n))$ instances of the GKR protocol on L [GKR15]. Here we identify X as a (Boolean-valued) vector in $\mathbb{F}^{k^m}$, where $k^m = n$ as noted above.

The GKR protocol is an interactive protocol whose output is (j, v) on input X. If $X \in L$, then $P_X(j) = v$ with probability 1, and if $X \notin L$, then $P_X(j) = v$ with probability at most 1/2. This can be thought of as $X \in L$ being reducible to $X \in \mathsf{PVAL}(\{j\}, \{v\})$.

Each instance of the GKR protocol generates a pair (j, v), and all of these pairs are collected together into $(J, \vec{v})$ which defines an instance of PVAL. The completeness of this reduction follows immediately from that of the GKR protocol, as each pair (j, v) represents the true statement that $P_X(j) = v$. On the other hand, for any X', if $X' \notin L$, then for any malicious prover, the probability that $X' \in \mathsf{PVAL}(J, \vec{v})$ will be at most 2^{-t} by the soundness of the GKR protocol, i.e., $X' \notin \mathsf{PVAL}(J, \vec{v})$ with high probability. By taking a union bound over all X' considered close enough to X (all of which are not in L by the soundness condition of an IPP) and setting t to be large enough, with high probability, all such close element are not in PVAL implying distance between X and PVAL.

A natural idea is to now try to argue that, by setting t to be sufficiently large, the input X is also far along $\mathcal{D}$ from PVAL. However, as discussed in Section 1.3.1, the difficulty here is that while the size of the uniform ε-ball[17] $B_{\mathcal{U},\varepsilon}(X)$ around a single element is relatively small, a similar ball defined by $\mathcal{D}$ can be extremely large. In more detail, the uniform ball around a single element has size $O(n^{\varepsilon n})$, so we need to take $t = O(\log(n^{\varepsilon n})) = O(\varepsilon n \log(n))$ repetitions of the GKR protocol in order to apply the union bound in uniform setting. On the other hand, suppose that the underlying distribution $\mathcal{D}$ is supported only over the first $\log(n)$ indices, then $\mathcal{B}_{\mathcal{D},\varepsilon}(X)$ contains at least $2^{n-\log(n)}$ elements. We now need $\log(2^{n-\log(n)}) \approx n$ many instances of the GKR protocol, which already means that the resulting IPP no longer as sublinear time verification and communication complexity.

From hereon, it is not obvious how to proceed or even if there is a reduction to showing an IPP for PVAL. To this end, our solution is to instead, reduce to PVAL over a *new* soundness constraint. Beginning with the initial promise on the input, $d_{\mathcal{D}}(X, L) > \varepsilon$, we take the intersection of $B_{\mathcal{D},\varepsilon}(X)$ and $B_{\mathcal{U},\varepsilon}(X)$. This set is of course, bounded above by the uniform ball $B_{\mathcal{U},\varepsilon}(X)$, but also does not contain any elements of L (by the soundness condition) as shown in Figure 1. Taking a similar union bound over this intersection, we now have a weaker soundness condition for PVAL, $\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}) > \varepsilon$, signifying that no element in the intersection of those two balls is in PVAL. In other words, with high probability, we have

$$d_{\mathcal{D}}(X, L) > \varepsilon \implies \mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}) = \min_{Y \in \mathsf{PVAL}} (\max(d_{\mathcal{U}}(X, Y), d_{\mathcal{D}}(X, Y))) > \varepsilon.$$

[17] Recall that for a distribution $\mathcal{D}$, the ε-ball along $\mathcal{D}$ around a point X, $B_{\mathcal{D},\varepsilon}(X)$, is the set of elements ε-close to X along $\mathcal{D}$

4.2.2 Interactive Reduction from NC to PVAL

Below, we prove our reduction from verifying the proximity of an instance to a language in NC to verifying the proximity of an instance to PVAL, in the distribution-free sense.

Theorem 4.3. *For any $\varepsilon > 0$ and any language L computable by logspace-uniform Boolean circuits of depth $\Delta_L = \Delta_L(n)$, size $S = S(n)$, and fan-in 2, any $k \in \mathbb{N}$ and $m = \log_k(n)$, the following holds. Let $\mathbb{F}$ be a finite field of size $|\mathbb{F}| = \Omega(k \cdot \Delta_L \cdot \log(S))$.*

There exists an interactive protocol $(P_{\mathsf{NC}}, V_{\mathsf{NC}})$ with input $X \in \{0,1\}^n$, whose output is a coordinate set $J \subseteq \mathbb{F}^m$ of size $t = 4\varepsilon \cdot n \cdot \log(n)$, and a vector $\vec{v} \in \mathbb{F}^{|J|}$ defining an instance of PVAL, such that:

1. ***Completeness**: If $X \in L$, then $(J, \vec{v})$ is such that $X \in \mathsf{PVAL}(J, \vec{v})$ (with probability 1). In other words,*
$$X \in L \implies \underset{V_{\mathsf{NC}}}{\mathbb{P}}[X \in \mathsf{PVAL}(J, \vec{v})] = 1$$

2. ***Soundness**: $\forall \mathcal{D} \in \Delta([n])$, if $d_{\mathcal{D}}(X, L) > \varepsilon$, then for any cheating prover strategy with probability at least $1/2$ over the verifier's coins, $\mu_{\mathcal{D},\mathcal{U}_n}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon$. In other words,*
$$d_{\mathcal{D}}(X, L) > \varepsilon \implies \underset{V_{\mathsf{NC}}}{\mathbb{P}}[\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon] \geq \frac{1}{2}$$

The prover runs in time $\mathsf{poly}(S, \log|\mathbb{F}|)$, and the verifier runs in time $\varepsilon \cdot n \cdot \mathsf{poly}(k, \Delta_L, \log(S), \log|\mathbb{F}|)$ (the verifier does not need to access the input X). The communication complexity is $\varepsilon \cdot n \cdot \mathsf{poly}(k, \Delta_L, \log(S), \log|\mathbb{F}|)$, and the number of rounds is $O(\Delta_L \cdot \log(S))$. Moreover, J is a uniformly random set of t points from $\mathbb{F}^m$.

Remark 8. *Note that this interactive reduction does* not *transform the input X or even access it. Instead, we invoke an interactive protocol that outputs a concise description of a different language (namely, a parameterisation of the PVAL language) to which the distance from X is preserved (with high probability) over a different metric.*

The proof of Theorem 4.3 relies on a result by Goldwasser, Kalai and Rothblum [GKR15], which states that there is an interactive protocol reducing low depth languages to PVAL on a single point (i.e., $|J| = 1$).

Theorem 4.4 ([GKR15])**.** *Let L be a language computable by logspace-uniform Boolean circuits of depth $\Delta_L = \Delta_L(n)$, size $S = S(n)$, and fan-in 2, for any $k \in \mathbb{N}$ and $m = \log_k(n)$, then the following holds. Let $\mathbb{F}$ be a finite field of size $|\mathbb{F}| = \Omega(k \cdot \Delta_L \cdot \log(S))$.*

There exists an interactive protocol $(P_{\mathsf{GKR}}, V_{\mathsf{GKR}})$, with input $X \in \{0,1\}^n$, that outputs a coordinate $j \in \mathbb{F}^m$ and a value $v \in \mathbb{F}$, such that:

- ***Completeness** - if $X \in L$ then with probability 1, the m-variate low degree extension with individual degree $k-1$ evaluated at j is v: $P_X(j) = v$ with probability 1. In other words,*
$$X \in L \implies \underset{V_{\mathsf{GKR}}}{\mathbb{P}}[P_X(j) = v] = 1.$$

- ***Soundness*** *- if $X \notin L$ then for every prover strategy, the probability that $P_X(j) = v$ is at most $1/2$ over the verifier's randomness. In other words,*

$$X \notin L \implies \mathop{\mathbb{P}}_{V_{\mathsf{GKR}}}[P_X(j) = v] \leq \frac{1}{2}.$$

The verifier runs in time $\mathsf{poly}(k, \Delta_L, \log(S), \log|\mathbb{F}|)$, the prover runs in time $\mathsf{poly}(S, \log|\mathbb{F}|)$, the communication complexity is $\mathsf{poly}(k, \Delta_L, \log(S), \log|\mathbb{F}|)$, and the number of rounds is $O(\Delta_L \cdot \log(S))$. Moreover, the coordinate j is a uniformly random point from $\mathbb{F}^m$.

Equipped with Theorem 4.4, we are now able to prove our reduction.

Proof of Theorem 4.3. Let $(P_{\mathsf{NC}}, V_{\mathsf{NC}})$ be the protocol for which $t = 2\varepsilon n(\log(n)+\log|\mathbb{F}|) \leq 4\varepsilon n \log(n)$ iterations of $(P_{\mathsf{GKR}}, V_{\mathsf{GKR}})$ from Theorem 4.4, are run in parallel. This implies that the round complexity is the same as that of a single iteration of this protocol (see, e.g., [GR17, Appendix A] for additional details). This yields t pairs of the form $(j, v) \in \mathbb{F}^m \times \mathbb{F}$. We collect all of these terms into a set of coordinates $J \subseteq \mathbb{F}^m$ and a set of claims of values on the set $\vec{v} \in \mathbb{F}^t$.

The running times, round complexity and communication complexity follow from this construction as it is $t = O(\varepsilon n(\log(n) + \log|\mathbb{F}|))$ times the complexity of a single run of $(P_{\mathsf{GKR}}, V_{\mathsf{GKR}})$. By the perfect completeness of each run of $(P_{\mathsf{GKR}}, V_{\mathsf{GKR}})$, each invocation produces a pair (j, v) for which $P_X(j) = v$, we have that the collection of these t parallel runs that generate an instance of PVAL also satisfy that $P_X(J) = \vec{v}$.

It remains to prove the soundness of this protocol. Suppose that $d_{\mathcal{D}}(X, L) > \varepsilon$. Now, the probability that any $X' \notin L$ is in PVAL is at most $2^{-t} < 1/(2n \cdot |\mathbb{F}|)^{\varepsilon \cdot n}$; this is the probability that it is consistent with the outputs of each run of $(P_{\mathsf{GKR}}, V_{\mathsf{GKR}})$.

It suffices to prove that, with high probability, no element of PVAL is in the intersection $\mathcal{B}_{\mathcal{D},\varepsilon}(X)$ and $\mathcal{B}_{\mathcal{U},\varepsilon}(X)$. The number of elements in the intersection of these two balls is bounded by the size of the uniform ball, in other words,

$$|B_{\mathcal{D},\varepsilon}(X) \cap B_{\mathcal{U}_n,\varepsilon}(X)| \leq |B_{\mathcal{U}_n,\varepsilon}(X)| \leq n^{\varepsilon n}.$$

Therefore taking the union bound over the entire intersection we have the probability of any element in that intersection satisfying PVAL is less than $n^{\varepsilon n} \cdot \frac{1}{2^{4\varepsilon n \log(n)}} < 1/2$ as all of those elements are not in L, this implies that $\mu_{\mathcal{D},\mathcal{U}_n}(X, \mathsf{PVAL}) > \varepsilon$ with high probability. □

4.3 Proof of Theorem 4.1

In this section, we prove our main result that constructs distribution-free IPPs for any language computable by logspace-uniform circuits of low-depth.

High level sketch of the proof: Consider the (soundness) case where the input $X \in \{0,1\}^n$ to an NC-language L is such that $d_{\mathcal{D}}(X, L) > \varepsilon$. Our main goal is to reduce the construction of a distribution-free IPP for L to a uniform IPP for PVAL over a *larger* index set $(J \cup I)$, for which we can use a pre-existing IPP from [RR20] (which is a quantitative improvement over a prior IPP for PVAL from [RVW13]), and get the stated query and communication complexities.

To this end, for the output $(J, \vec{v})$ for which the interactive reduction from NC to PVAL from Theorem 4.3 guarantees that $\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon$. If X is far from PVAL along the uniform distribution, the uniform IPP will reject and so we can assume that X is close to $\mathsf{PVAL}(J, \vec{v})$ uniformly (i.e., $d_{\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) \leq \varepsilon$). At this point we observe that since PVAL is a good error correcting code (i.e., with large minimal distance), the input X must be close to a *unique* element $X' \in \mathsf{PVAL}(J, \vec{v})$. However, by our soundness assumption over μ, we know that $d_{\mathcal{D}}(X, X') > \varepsilon$.

Now, the verifier generates $O(1/\varepsilon)$ samples I from $\mathcal{D}$ (and the corresponding values in X). Let $\mathsf{PVAL}'(I, X|_I)$ be the set of strings in PVAL which agree with X on I. Alternatively, $\mathsf{PVAL}'(I, X|_I) = \mathsf{PVAL}((J, I), (\vec{v}, X|_I))$. Since $d_{\mathcal{D}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon$ from our initial assumption, we see that with high probability there exists an index in I on which X' and X disagree on. In other words, X' is not in $\mathsf{PVAL}'(I, X|_I)$ and using the above properties of X', we see that X is ε-far from $\mathsf{PVAL}((J, I), (\vec{v}, X|_I))$ along the *uniform distribution*. Thus, by applying the uniform IPP we can catch the cheating prover.

Theorem 4.5 (Theorem 4.1 restated)**.** *For every $n \in \mathbb{N}$, let $L \subseteq \{0,1\}^n$ be a language computable by logspace-uniform circuits with depth $\Delta_L = \Delta_L(n) \geq \log(n)$ and size $S = S(n)$. Then, for $\varepsilon > \frac{200 \log^3(n)}{n}$, there exists a distribution-free interactive proof of proximity for L with perfect completeness and soundness at least $1/2$.*

This protocol has query complexity $O\left(\frac{1}{\varepsilon}\right)$, sample complexity $O\left(\frac{1}{\varepsilon}\right)$, and communication complexity $O\left(\frac{\log(n)}{\varepsilon} + \varepsilon \cdot n \cdot \mathsf{poly}(\Delta_L)\right)$. In addition, the honest prover runs in time $\mathsf{poly}(n, S)$ and the verifier runs in time $\tilde{O}\left(\frac{1}{\varepsilon} + \varepsilon \cdot n \cdot \mathsf{poly}(\Delta_L)\right)$. Finally, the round complexity of the protocol is $\mathsf{polylog}(n) + O(\Delta_L \cdot \log(S))$.

To prove this, we need the following IPP for PVAL over the uniform distribution.

Theorem 4.6 (Uniform IPP for PVAL [RR20])**.** *Let $n \in \mathbb{N}$, $\varepsilon \geq \frac{200 \log^3(n)}{n}$, and $\mathbb{F}$ be a finite field of characteristic 2 of size $|\mathbb{F}| = \Theta(n^3 \varepsilon^2 \log^4(n))$.*

Then, for any set $J \in (\mathbb{F}^m)^t$ of size $O(n \cdot \varepsilon \cdot \log(n))$ and $\vec{v} \in \mathbb{F}^t$, there exists a uniform IPP, $(P_{\mathsf{Unif}}, V_{\mathsf{Unif}})$, for $\mathsf{PVAL}(J, \vec{v})$. This protocol has perfect completeness, soundness $1/2$, query complexity $O(1/\varepsilon)$, communication complexity $\tilde{O}(n \cdot \varepsilon)$. Moreover, the honest prover runs in $\mathsf{poly}(n)$ time, the verifier runs in time $\tilde{O}\left(\frac{1}{\varepsilon} + \varepsilon \cdot n\right)$, and the number of messages communicated between them is $\mathsf{polylog}(n)$.

Proof of Theorem 4.5. We construct a distribution-free IPP with perfect completeness and a constant soundness error (specifically $4/5$) which can reduced to, say $1/3$, by repetition. This distribution-free IPP for NC is given in Protocol 3.

The complexities follow from inspection for the chosen parameters. In particular, the query complexity comes from the IPP for $\mathsf{PVAL}(J, \vec{v})$ from Theorem 4.6 and the $O(1/\varepsilon)$ samples in Step 2 of the protocol make up the sample complexity. The running times of the prover and the verifier, along with the round complexity come from the sums of the respective values from Theorems 4.3 and 4.6. The communication complexity follows similarly, but also includes the T samples sent to the prover in addition.

The perfect completeness of Protocol 3, follows from the combination of the completeness guarantees of $(P_{\mathsf{NC}}, V_{\mathsf{NC}})$, as well as $(P_{\mathsf{Unif}}, V_{\mathsf{Unif}})$. On the other hand, for soundness, suppose $d_{\mathcal{D}}(X, L) > \varepsilon$. Firstly, we have the following result from [RR20].

Lemma 4.7 (Follows from Proposition 5.4 in [RR20])**.** *Let $\mathbb{F}$ be any field and $m, n \in \mathbb{N}$. Let $d_{\min}(\mathsf{PVAL}(J, \vec{v}))$ represent the relative minimum Hamming distance between any pair of n-length*

Protocol 3 Distribution-free IPP for any language L computable by circuits of size $S(n)$ and depth $\Delta_L(n)$.

Input: The verifier V_{df} gets implicit input $X \in \{0,1\}^n$ that is accessible through a query oracle, as well as the sample oracle $\mathcal{O}_{\mathcal{D}}(X)$, for some unknown distribution $\mathcal{D}$. The verifier also gets explicit access to $\varepsilon > 0$. The prover P_{df} gets direct access to X and ε.

The distribution-free IPP:

1. Let $(P_{\mathsf{NC}}, V_{\mathsf{NC}})$ be the interactive reduction from Theorem 4.3 with proximity parameter ε. P_{df} and V_{df} run $(P_{\mathsf{NC}}, V_{\mathsf{NC}})$ on X, to output a set $J \subset \mathbb{F}^m$ of size $t = 4\varepsilon \cdot n \cdot \log(n)$ and $\vec{v} \in \mathbb{F}^t$, using parameters $k = 2$ and $m = \log(S)$.

2. V_{df} sets $T = 3/\varepsilon$ and picks T fresh samples $I = ((i_1, X_{i_1}), \ldots, (i_t, X_{i_T}))$ from $\mathcal{O}_{\mathcal{D}}(X)$. Let $z \in \{0,1\}^T$ $z_j = X_{i_j}$, for every $j \in [T]$. The verifier V_{df} sends (I, z) to P_{df}.

3. P_{df} and V_{df} run the uniform IPP $(P_{\mathsf{Unif}}, V_{\mathsf{Unif}})$ from Theorem 4.6 for $\mathsf{PVAL}((J, I), (\vec{v}, \vec{z}))$ on input X, using parameters $m = \log(n)$ and $r = \log(1/\varepsilon)$.

4. V_{df} accepts if and only if V_{Unif} accepts.

strings in $\mathsf{PVAL}(J, \vec{v})$*. For any* $t \geq 2\varepsilon \cdot n(\log(n) + \log|\mathbb{F}|) + 4$*, we have*

$$\underset{J \sim \mathcal{U}_{(\mathbb{F}^m)^t}}{\mathbb{P}}[d_{\min}(\mathsf{PVAL}(J, \vec{v})) < 2\varepsilon \cdot n] < 2^{-4} < 1/10.$$

Using this, we prove the following lemma that establishes the constraints satisfied by the output $(J, \vec{v})$ of the protocol $(P_{\mathsf{NC}}, V_{\mathsf{NC}})$ from Theorem 4.3.

Lemma 4.8. *For any* $n \in \mathbb{N}$*,* $\varepsilon > 0$*, distribution* $\mathcal{D}$ *over* $[n]$*,* $X \in \{0,1\}^n$*, and language* $L \subseteq \{0,1\}^n$ *computable by logspace-uniform circuits with depth* $\Delta_L = \Delta_L(n)$ *and size* $S = S(n)$*, if* $d_{\mathcal{D}}(X, L) > \varepsilon$*, the output* $(J, \vec{v})$ *of the protocol* $(P_{\mathsf{NC}}, V_{\mathsf{NC}})$ *from Theorem 4.3 satisfies the following conditions:*

- $\underset{\mathcal{V}_{\mathsf{NC}}}{\mathbb{P}}[\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon] > 0.5.$
- $\underset{\mathcal{V}_{\mathsf{NC}}}{\mathbb{P}}[\exists X_1 \neq X_2,$ *such that* $X_1, X_2 \in \mathcal{B}_{\mathcal{U},\varepsilon}(X)$ *and* $X_1, X_2 \in \mathsf{PVAL}(J, \vec{v})] < 0.1.$

While the first condition maintains the soundness guarantee along the hybrid metric promised by Theorem 4.3, the second condition implies that there is at most one element close to X uniformly that is in PVAL.

Proof of Lemma 4.8. The first item is satisfied from the soundness guarantees of $(P_{\mathsf{NC}}, V_{\mathsf{NC}})$ by Theorem 4.3.

To prove the second item, we first observe that the probability (over the internal randomness of V_{NC}) that there exist two distinct strings that are ε-close to X along the uniform distribution in PVAL is at most the probability that there exist two distinct strings in PVAL that are 2ε-close.

$$
\begin{aligned}
&\underset{\mathcal{V}_{\mathsf{NC}}}{\mathbb{P}}[\exists X_1 \neq X_2, \text{ such that } X_1, X_2 \in \mathcal{B}_{\mathcal{U},\varepsilon}(X) \text{ and } X_1, X_2 \in \mathsf{PVAL}(J, \vec{v})] \\
&\qquad \leq \underset{\mathcal{V}_{\mathsf{NC}}}{\mathbb{P}}[\exists X_1 \neq X_2, \text{ such that } d_{\mathcal{U}}(X_1, X_2) < 2\varepsilon \text{ and } X_1, X_2 \in \mathsf{PVAL}(J, \vec{v})] \\
&\qquad = \underset{\mathcal{V}_{\mathsf{NC}}}{P}[d_{\min}(\mathsf{PVAL}(J, \vec{v})) < 2\varepsilon \cdot n] \\
&\qquad < 0.1.
\end{aligned}
$$

The first transition follows from the triangle inequality for Hamming distances as the distance between such an X_1, X_2 is at most the sum of their distances to X. In turn, this probability is equal to that of the minimum distance of PVAL being less than 2ε, as seen in the next line. Since J is distributed uniformly at random and generated using V_{NC}'s internal randomness, and its size is $4\varepsilon \cdot n \log(n) \geq 2\varepsilon \cdot n \cdot (\log(n) + \log |\mathbb{F}|) + 4$, this probability can be upper bounded using Lemma 4.7. □

In the first step of each repetition of Protocol 3, we have $\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon$ with probability at least $1/2$ by the first item of Lemma 4.8. Suppose that $d_{\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon$, then V_{Unif} rejects with probability at least $1/2$ in Step 3.

On the other hand, suppose that $d_{\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) \leq \varepsilon$. Then, from the second item of Lemma 4.8, observe that with probability at least $9/10$, there exists at most one $W \in \mathsf{PVAL}(J, \vec{v})$ such that $d_{\mathcal{U}}(X, W) < \varepsilon$.

Further, using the guarantee from the first item of Lemma 4.8, we see that $d_{\mathcal{D}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon$, and in particular, $d_{\mathcal{D}}(X, W) > \varepsilon$. In turn, this implies that $W \notin \mathsf{PVAL}((J, I), (\vec{v}, \vec{z}))$ with probability at least $9/10$, since at least one of the entries of z will be an index on which X and W disagree. More precisely,

$$
\begin{aligned}
\mathbb{P}[W \notin \mathsf{PVAL}((J, I), (\vec{v}, \vec{z}))] &\geq \mathbb{P}[\exists i \in I : X_i \neq W_i] \\
&\geq 1 - (1 - \varepsilon)^t \\
&= 1 - (1 - \varepsilon)^{3/\varepsilon} \\
&\geq 1 - e^{-3} \\
&> 9/10.
\end{aligned}
$$

Put together, in each repetition, with probability at least $2/5$ (over the internal randomness of V_{NC} and the choice of I), $d_{\mathcal{U}}(X, \mathsf{PVAL}((J, I), (\vec{v}, \vec{z}))) > \varepsilon$. Indeed, this is the probability that both the items of Lemma 4.8 hold (more precisely, Item 1 and the complement of Item 2), times the probability that $W \notin \mathsf{PVAL}((J, I), (\vec{v}, \vec{z}))$.

Thus, at the end of each round, V_{df} rejects X with probability at least $1/5$, by the soundness guarantee of Theorem 4.6. Soundness error $1/2$ can now be achieved by standard soundness amplification. □

5 IPPs over Dispersed Distributions

In Section 4, we showed the construction of a distribution-free IPP for any NC language that uses $\tau + O\left(\frac{1}{\varepsilon}\right)$ queries and $\tilde{O}(\frac{n}{\tau} + \frac{1}{\varepsilon})$ bits of communication, for every $\tau \leq \sqrt{n}$ and ε. For $\varepsilon > \tau/n$ this matches the best IPPs for the uniform distribution. However, for $\varepsilon < \tau/n$, the communication complexity is at least $\tilde{\Omega}(n/\tau)$. Compare this to [RVW13], where for any $\varepsilon = o\left(\frac{\tau}{n}\right)$, the communication complexity is $\tilde{O}(n/\tau)$, still with $O(1/\varepsilon)$ query complexity.

While we do not know how to overcome this problem in general, in this section, we introduce a classification of distributions over $[k]^m$ (and $k^m = n$), which we call ρ-dispersed distributions (for $1 \leq \rho \leq k$), to capture the "closeness" of the behaviour of an underlying distribution to the Uniform distribution. Under such a definition, the larger the value of ρ, the lesser the distribution behaves like the uniform distribution in this sense. In particular, $\rho = 1$ captures the uniform distribution, while every distribution is k-dispersed. For IPPs over the set of ρ-dispersed distributions for a small enough ρ, we match the result by [RVW13].

For our main result of this section, we construct IPPs for NC languages over ρ-dispersed distributions, which give a smooth trade-off between ρ and the query complexity for fixed communication complexity. In particular, for small enough values of ρ (i.e., distributions behave like the uniform distribution), these IPPs achieve better communication complexity than Theorem 1.1, for roughly the same query complexity when the proximity parameter ε is less than $1/\sqrt{n}$.

Section Organization. First, we define ρ-Dispersed Distributions in Section 5.1 and then state the main theorem of this section which is an IPP for NC over such distributions. Subsequently, we show the construction of an IPP for PVAL over (hybrid metrics for) ρ-Dispersed Distributions in Sections 5.2 and 5.3, that builds on certain structural properties of such distributions. The final IPP from Theorem 5.2 is obtained by combining the reduction from Theorem 4.3 with this IPP for PVAL, and its formal details are provided in Section 5.3.

5.1 ρ-Dispersed Distributions

Below, we define ρ-Dispersed distributions. These are distributions over $[k]^m$ for some $k, m \in \mathbb{N}$ over which the probability of any element is at most ρ times the average probability taken over any single dimension. Formally,

Definition 5.1 (ρ-Dispersed Distributions)**.** *Let $\rho \in \mathbb{R}$ be such that $1 \leq \rho \leq k$. We say that a distribution $\mathcal{D} \in \Delta([k]^m)$ is ρ-Dispersed, if for every $j \in [m]$ and for every $(i_1, \ldots, i_m) \in [k]^m$, we have*

$$\mathcal{D}(i_1, \ldots, i_m) \leq \rho \cdot \mathop{\mathbb{E}}_{t \sim U_k}[\mathcal{D}(i_1, \ldots, t, \ldots, i_m)],$$

where $\mathcal{D}(\cdot)$ denotes the probability mass function.

In other words, for any element i in the support of a ρ-Dispersed distribution $\mathcal{D}$ and for every dimension $j \in [m]$, $\mathcal{D}(i)$ is at most ρ times the average of $\mathcal{D}$ along the j^{th}-dimension, keeping the rest of the coordinates of i fixed.

For any $1 \leq \rho \leq k$, we consider ρ-Dispersed distributions as a way of capturing distributions that behave closely to the uniform distribution. In particular, observe the following simple facts.

- The uniform distribution is a 1-Dispersed distribution as this implies that no index has weight which is greater that the average.
- Moreover, any distribution over $[k]^m$ is trivially a k-Dispersed distribution as k times the average is the total over that dimension for which every weight in that column is at most that value.

- Let $\mathcal{D}$ be a distribution, such that for some $i_1, \cdots, i_{m-1} \in [k]$, there is only one element in the set $\{(i_1, \cdots, i_{m-1}, s)\}_{s \in [k]}$ for which $\mathcal{D}$ has non-zero weight (and the rest of the distribution can behave arbitrarily). Then, $\mathcal{D}$ is k-dispersed but not $(k-\delta)$-dispersed for any $\delta > 0$. Indeed, suppose the element with non-zero weight in this set is $(i_1, \ldots, i_{m-1}, s_0)$. Then,

$$\mathcal{D}(i_1, \cdots, i_{m-1}, s_0) = \sum_{s=1}^{k} \mathcal{D}(i_1, \cdots, i_{m-1}, s) = k \mathop{\mathbb{E}}_{s \sim U_k}[\mathcal{D}(i_1, \cdots, i_{m-1}, s)].$$

- α-log Lipschitz distributions, introduced by [AFK13], define distribution families which are locally smooth, in the sense that if points that are close to each other in Hamming distance cannot have vastly different probability masses under $\mathcal{D}$. More formally, we define α-log Lipschitz distributions as follows:

Definition 5.2. *A distribution $\mathcal{D}$ is α-log Lipschitz if $\forall x, x' \in [k]^m$ that differ in only one value, the following holds.*

$$|\log(\mathcal{D}(x)) - \log(\mathcal{D}(x'))| \leq \log(\alpha)$$

Equivalently stated, we have for every $x, x' \in [k]^m$ that differ in only one value, $\frac{\mathcal{D}(x)}{\mathcal{D}(x')} \leq \alpha$.

This notion captures a wide variety of popularly studied distributions and has been studied in several different contexts (cf. [AFK13] for more references). Examples include the uniform distribution ($\alpha = 1$) or product distributions over $[k]^m$, where the probability of sampling each element is in the interval $\left[\frac{1}{k-1+\alpha}, \frac{\alpha}{k-1+\alpha}\right]$.

We observe that any α-log-Lipschitz distribution is also $\left(\frac{\alpha \cdot k}{\alpha+k-1}\right)$-dispersed.[18] This holds because, the log-Lipschitz condition of $\mathcal{D}$ implies that for the index $(i_1, \cdots, i_m) \in [k]^m$ and value $t \in [m]$ for which the ratio $\frac{\mathcal{D}(i_1, \cdots i_m)}{\mathop{\mathbb{E}}_{t \sim [k]}[\mathcal{D}(i_1, \cdots, t, \cdots, i_m)]}$ is maximised, the following is true.

$$\begin{aligned}
\max\left(\frac{\mathcal{D}(i_1, \cdots i_m)}{\mathop{\mathbb{E}}_{t \sim [k]}[\mathcal{D}(i_1, \cdots, t, \cdots, i_m)]}\right) &= \max\left(\frac{k\mathcal{D}(i_1, \cdots i_m)}{\sum_{t \in [k]} \mathcal{D}(i_1, \cdots, t, \cdots, i_m)}\right) \\
&\leq \frac{k\mathcal{D}(i_1, \cdots i_m)}{\mathcal{D}(i_1, \cdots i_m) + \frac{k-1}{\alpha}\mathcal{D}(i_1, \cdots i_m)} \\
&= \frac{\alpha \cdot k}{\alpha + k - 1}
\end{aligned}$$

and thus, it is $\left(\frac{\alpha \cdot k}{\alpha+k-1}\right)$-dispersed. In particular, we know that $k^{o(1)}$-log-Lipschitz distributions are $k^{o(1)}$-dispersed.

[18] Note that this inclusion is strict, since α-dispersed distributions need not be supported on all elements, unlike α-log-Lipschitz distributions, which by definition have non-zero measure everywhere.

- Let $\hat{\mathcal{D}}$ be an m-product distribution over $[k]^m$ defined as $\hat{\mathcal{D}} = \mathcal{D} \times \cdots \times \mathcal{D}$, where $\mathcal{D}$ is a distribution over $[k]$ with minimum weight p_{min} and maximum p_{max}, and the product is taken m times. Then, $\hat{\mathcal{D}}$ is $\left(\frac{kp_{max}}{p_{max}+(k-1)p_{min}}\right)$-dispersed. To see this, observe that, $p_{max} = \left(\frac{kp_{max}}{p_{max}+(k-1)p_{min}}\right) \cdot \mathop{\mathbb{E}}_{t\sim\mathcal{D}}[\mathcal{D}(t)]$ as $\mathop{\mathbb{E}}_{t\sim\mathcal{D}}[\mathcal{D}(t)] = \frac{p_{max}+(k-1)p_{min}}{k}$.

For any distribution $\mathcal{D}$ over $[k]^m$, for any $p \in [m]$ and for any $(i_1, \cdots, i_{m-p}) \in [k]^{m-p}$, we define the marginal distribution over p dimensions as $\mathcal{D}^{(m-p)} \in \Delta(\Omega_{k^{m-p}})$ as follows.

$$\mathcal{D}^{(m-p)}(i_1, \cdots, i_{m-p}) = \sum_{t\in[k]} \mathcal{D}(i_1, \cdots i_{m-p}, t).$$

Note that for $p_1, p_2 \in [m]$, $\mathcal{D}^{(m-p_1)^{(m-p_2)}} = \mathcal{D}^{(m-p_1-p_2)}$.

Lemma 5.1. *For any $m, k, \rho \in \mathbb{N}$, and any distribution $\mathcal{D}$ over $[k]^m$, $\mathcal{D}$ is a ρ-Dispersed distribution implies $\mathcal{D}^{(m-1)}$ is ρ-Dispersed.*

Proof. We first restrict our attention to the ρ-Dispersed condition on the first index and the same analysis extends to this condition on any index in $[m-1]$. For any $(i_1, \ldots, i_{m-1}) \in [k]^{m-1}$, we have

$$\begin{aligned}
\mathcal{D}^{(m-1)}(i_1, \cdots, i_{m-1}) &= \sum_{j\in[k]} \mathcal{D}(i_1, \cdots, i_{m-1}, j) \\
&\leq \sum_{j\in[k]} \rho \mathop{\mathbb{E}}_{l\sim[k]}[\mathcal{D}(l, i_2, \cdots, i_{m-1}, j)] \\
&\leq \rho \mathop{\mathbb{E}}_{l\sim[k]}\left[\sum_{j\in[k]} \mathcal{D}(l, i_2, \cdots, i_{m-1}, j)\right] \\
&\leq \rho \mathop{\mathbb{E}}_{l\sim[k]}[\mathcal{D}^{(m-1)}(l, i_2, \cdots, i_{m-1})]
\end{aligned}$$

The first and last lines follow by the definition of $\mathcal{D}^{(m-1)}$. □

We now state the main theorem of this section. Again, for convenience, we stick to the setting where the query vs communication complexity trade-off parameter is set to $O(1/\varepsilon)$.

Theorem 5.2 (Formal statement for Theorem 1.2)**.** *Let $n \in \mathbb{N}$, and set $k = \log(n)$ and $m = \log_k(n)$ (such that $k^m = n$), let $L \subseteq \{0,1\}^n$ be a language computable by logspace-uniform circuits with depth $\Delta_L = \Delta_L(n)$ and size $S = S(n)$. Then, for $\varepsilon > 0$, $\rho \in \mathbb{R}$, there exists an interactive proof of proximity over ρ-Dispersed distributions over $[k]^m$ for L with perfect completeness and soundness at least $1/2$.*

This protocol has query complexity $\frac{\rho^{\log(1/\varepsilon)/\log\log(n)}}{\varepsilon^{1+o(1)}}$, sample complexity $\frac{\rho^{\log(1/\varepsilon)/\log\log(n)}}{\varepsilon^{1+o(1)}}$, communication complexity $\varepsilon^{1-o(1)} \cdot n \cdot \log^2(n) + \varepsilon \cdot n \cdot \mathsf{poly}(\Delta_L)$ and round complexity $O\left(\frac{\log\left(\frac{1}{\varepsilon}\right)}{\log\log(n)} + \Delta_L \cdot \log(S)\right)$. In addition, the honest prover runs in time $\mathsf{poly}(S, n)$ and the verifier runs in time

$$n^{o(1)} \cdot \left(\frac{\rho^{\log(1/\varepsilon)/\log\log(n)}}{\varepsilon} + \varepsilon \cdot n \cdot \mathsf{poly}(\Delta_L)\right).$$

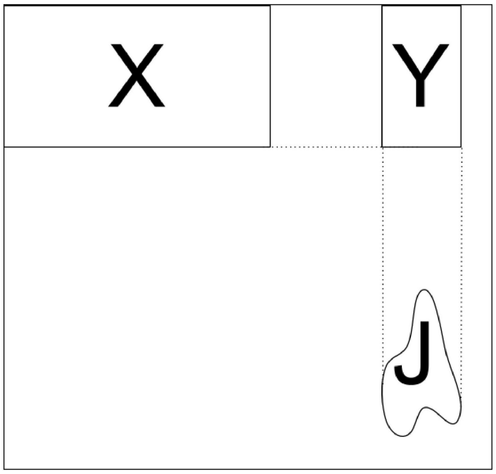

Figure 3: During the polynomial folding protocol, the prover sends the univariate LDE of each row of X evaluated on the columns of J, collected in the matrix $Y \in \mathbb{F}^{k_1 \times t}$. For any $j = (j_1, j_2) \in J$, the univariate LDE of the j_2^{th}-column of Y restricted to j_1 is equal to $\vec{v}[j]$.

It is worth noting that for $\rho = k^{o(1)}$, we have that $\rho^{\log(1/\varepsilon)/\log\log(n)} = k^{o(r)} = 1/\varepsilon^{o(1)}$ and so we match the query and communication complexities of the uniform IPP from [RVW13] (up to poly logarithmic factors). More importantly, ρ *does not contribute* to the communication complexity of the IPP.

5.2 Polynomial Folding Protocol

We now demonstrate an interactive protocol that accepts any input in $\mathsf{PVAL}(J, \vec{v})$, while rejecting any input that is far along the hybrid metric for any ρ-Dispersed $\mathcal{D}$ from Definition 5.1. The idea is to reduce an instance of PVAL to a set of PVAL instances, each on one lesser variable. We then generalise the protocol analysis in [RVW13], to be able to handle this new condition for soundness over the hybrid metric.

We require the following notation for this protocol intuition. We define column marginals over $[k_1] \times [k_2]$ for $k_1, k_2 \in \mathbb{N}$. For which we define the marginal as follows.

$$\forall j \in [k_2] : \mathcal{D}^c(j) = \sum_{i \in [k_1]} \mathcal{D}(i, j).$$

We define a hybrid metric $\mu_{\mathcal{D}^c, U_{k_2}}$ over the column marginals of $\mathcal{D}$ and the uniform distribution. For any $x, y \in \mathbb{F}^{k_2}$, we have

$$\mu_{\mathcal{D}^c, U_{k_2}}(x, y) = max(d_{\mathcal{D}^c}(x, y), d_{\mathcal{U}_{k_2}}(x, y)).$$

Whenever the usage of $\mathcal{D}$ is clear from the context, we refer to this metric as μ^c.

5.2.1 Protocol Intuition for Polynomial Folding

Let $\mathbb{F}$ be a finite field of size $\max\{\ell_1, \ell_2\} = \mathsf{poly}(k_1, k_2)$, where $k_1 < \ell_1$, $k_2 < \ell_2$. We start with the two dimensional case, by viewing the input X as an element in $\mathbb{F}^{k_1 \times k_2}$ and defining $P_X : \mathbb{F}^2 \to \mathbb{F}$

as its bivariate low-degree extension (LDE). We reduce the problem of checking proximity of X to PVAL on bivariate LDEs along $\mu_{\mathcal{D},\mathcal{U}_n}$ to checking proximity of strings in $\mathbb{F}^{k_2}$ to PVAL defined on *univariate* LDEs (of degree $k_2 - 1$) along $\mu_{\mathcal{D}^c,\mathcal{U}_{k_2}}$. This idea naturally extends to the m-variate case, where $X \in \mathbb{F}^{k^m}$ as we can reduce the dimensionality by 1 repeatedly, by taking X to be a $k \times k^{m-1}$ matrix of values over $\mathbb{F}$.

In more detail, PVAL is parameterised by $(J, \vec{v})$ with input X, where $J \subseteq \mathbb{F}^{l_1 \times l_2}$ is a set of t coordinates, and PVAL is satisfied if and only if the evaluations of P_X on these points is the vector $\vec{v}$. Let $J_2 = \{i_2 \mid \exists i \text{ s.t. } (i, i_2) \in J\}$, i.e., J_2 is the projection of the elements in J onto its second coordinate. For every $i_2 \in J_2$, we define a new set of coordinates $Q_{i_2} = \{(i, i_2) \mid i \in [k_1]\}$. Note that, we can interpolate the value at any coordinate $(i_1, i_2) \in J$ using the univariate LDE over Q_{i_2}, of degree $k_1 - 1$. The condition for completeness is for X to be in $\mathsf{PVAL}(J, \vec{v})$, whereas for soundness we would like to have that X is ε-far from $\mathsf{PVAL}(J, \vec{v})$ along the $(\mathcal{D}, \mathcal{U})$-hybrid metric, i.e., $\mu_{\mathcal{D},\mathcal{U}_n}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon$.

The protocol proceeds as follows, for each $i_2 \in J_2$, the prover sends the evaluations of $P_X(Q_{i_2})$. There will with high probability be t such values of i_2 (as $|J_2| \approx |J|$=t). Since each Q_{i_2} is of size k_1, in total, the honest prover sends a $k_1 \times t$ matrix Y of evaluations of P_X (as shown in Figure 3), the verifier receives Y'. The verifier then checks that these values are consistent with $\vec{v}$ at each point (i_1, i_2) in J using the univariate low degree extension over Q_{i_2}.

The [RVW13] protocol works as follows. For each $j \in [k_1]$, let X_j be the j^{th} row of X. We now have t new conditions on any row X_j; the low degree extension of X_j restricted to J_2 is the j^{th} row of Y', denoted by Y'_j. This corresponds to a new instance of $\mathsf{PVAL}(J_2, Y'_j)$ for each $j \in [k_1]$, which is defined on the univariate low degree extension of X_j.

If the previous check succeeds, the verifier sends a uniformly random vector $z \in \mathbb{F}^{k_1}$ to the prover and the new case of PVAL will be $\mathsf{PVAL}\bigg(J_2, z \cdot Y'\bigg)$ for which we want to test membership of $w = z \cdot X$.[19] Completeness of any such instance of PVAL follows from the linearity of polynomial interpolation.

On the other hand, we generalise the soundness analysis in the following way. It is worth emphasising that the distance of a vector in $\mathbb{F}^{k_2}$ from $\mathsf{PVAL}(J_2, Y'_j)$ for any j, is taken along the marginal distribution of columns in $\mathcal{D}$. This is the distribution of i_2 returned from sampling $(i_1, i_2) \sim \mathcal{D}$, so we can test against this distribution by sampling from $\mathcal{D}$.

At this point, to pass the verifier's checks and make it accept, the prover has to "lie" on a certain set of rows by pretending that the input is $X' \in \{0, 1\}^n$ which satisfies PVAL. We first look at the case that the prover lies in just one row and how a uniformly random z will assist the verifier in catching the prover. We then extend this intuition to the case where the prover lies on any number of rows and show how repeating this process with random z of varying Hamming weights will catch the prover.

Suppose first that the prover only lies about one row i^*. The distance of that row from satisfying $\mathsf{PVAL}(J_2, Y'_{i^*})$ is now ε along one of $\mathcal{U}_{k_2}$ or $\mathcal{D}^c$ because of the original soundness condition. When the verifier picks a uniformly random z from $\mathbb{F}^{k_1}$, with high probability z_{i^*} is non-zero. For some X' that belongs to $\mathsf{PVAL}(J, \vec{v})$, on every column that X_{i^*} differs from X'_{i^*}, w differs from $z \cdot X'$ (note that X' is consistent with X on all the other rows). Since X_{i^*} is ε-far from $\mathsf{PVAL}(J_2, Y'_{i^*})$ along μ^c, the LDE of the corresponding $z \cdot X$ is far from satisfying $z \cdot Y'$ along μ^c. This means that

[19] For any $z \in \mathbb{F}^{k_1}$ and any matrix $A \in \mathbb{F}^{k_1 \times k_2}$, the dot product $z \cdot A \in \mathbb{F}^{k_2}$ is the linear combination of the rows of A whose coefficients come from z.

w is far from this new folded instance of PVAL.

For when the prover cheats on multiple rows, we prove that there is some $m^* \in [\log(k_1)]$ such that by sampling a random set of $\frac{k_1}{2^{m^*}}$ rows, with high probability at least one of these rows will be $\Omega(\varepsilon \cdot k_1/\rho \cdot 2^{m^*})$-far from satisfying the corresponding row of PVAL along μ_c as $\mathcal{D}$ is ρ-Dispersed. Since the verifier does not know the value of m^*, the verifier looks at each $m \in [\log(k_1)]$ and uniformly samples a z_m of Hamming weight 2^m. Here, we use a lemma on distances between linear subspaces, which is a generalisation of an analogous lemma in [RVW13]. This lemma states that for our metric, if S and T are linear subspaces then a point in S far from T implies a uniformly random element of S will be far from T with high probability. This implies that $z_{m^*} \cdot X$ will be $(\varepsilon/\rho 2^{m^*})$-far from $\mathsf{PVAL}\bigg(J_2, z_{m^*} \cdot Y'\bigg)$ along μ^c with high probability.

There are $\log(k_1)$ different instances of PVAL and one of these is far from the corresponding $z_m \cdot X$. For each $m \in [\log(k_1)]$ the prover sends the verifier $z_m \cdot X$ and the verifier checks if each of this is consistent with $\mathsf{PVAL}\bigg(J_2, z_m \cdot Y'\bigg)$. The next stage is for the verifier to check that each w'_m that the prover purports to be $z_m \cdot X$ is close to the correct value. The verifier does this by sampling columns of X along the $\mathcal{D}^c$ and the $\mathcal{U}_{k_2}$ distributions and computing the projection of $z_m \cdot X$ onto these samples, then querying the entire corresponding columns of X. If either consistency checks fail then the verifier rejects. Completeness follows immediately, but for soundness we have that $\mu_{\mathcal{D}^c, \mathcal{U}_{k_2}}(z_{m^*} \cdot X, \mathsf{PVAL}(J_2, z_{m^*} \cdot Y')) > \varepsilon/2^{m^*}\rho$ and therefore sampling $z_m \cdot X$ will catch the cheating prover after $O(\frac{\rho}{2^{m^*}\varepsilon})$ samples from $z_m \cdot X$. Each query to $z_m \cdot X$ requires 2^m queries to X.

The total query complexity here is $\tilde{O}(\frac{\rho}{2^{m^*}\varepsilon}) \cdot 2^{m^*} = \tilde{O}(\rho/\varepsilon)$ and the total sample complexity is $O(\rho/\varepsilon)$, which is a blowup of ρ from the original uniform case in [RVW13]. This happens as the distance from X to X' that differ on a single element, (i,j) is $\mathcal{D}(i,j)$ originally but when we consider distance on a row vector, it becomes $\sum_{i' \in k_1} \mathcal{D}(i', j)$. In the uniform case this corresponds to multiplying by k_1, however, when the distribution is ρ-Dispersed, we multiply by k_1/ρ. The communication complexity is unchanged only sending Y' and $\log(k_1)$ different folded rows to total $O(|J|k_1 + k_2 \log(k_1))$. For example, we can still achieve sublinear complexity even for $\rho = k_1$ for $k_1 = n^{1/4}$, $k_2 = n^{3/4}$, $\varepsilon = n^{-1/2}$ and $|J| = n\varepsilon \log(n)$. In this case the communication and query complexity are both $\tilde{O}(n^{3/4})$. In this case, we require $k_1 \neq k_2$ as otherwise we do not have sublinear communication complexity and query complexity. In that case the query complexity would be $\tilde{O}(\sqrt{n}/\varepsilon)$ and the communication would be $\tilde{O}(n^{3/2}\varepsilon)$ where they can't both be sublinear.

5.2.2 Polynomial Folding Proof

For the following protocol, we take $X \in \mathbb{F}^{k \times k^p}$ for some $p \in \mathbb{N}$. For any $i \in [k]$, we define $X[i, \cdot] \in \mathbb{F}^{k^p}$ to be the i^{th} row of X such that

$$\forall (i_1, \cdots, i_p) \in [k]^p : X[i, \cdot]_{(i_1, \cdots, i_p)} = X_{i, i_1, \cdots, i_p}.$$

For $i \in [t]$, $Y \in \mathbb{F}^{|J| \times k^p}$, we define $Y[i, \cdot]$ and $Y'[i, \cdot]$ similarly. Note that for $j \in [k]^p$, $Y'[\cdot, j]$ refers to the j^{th} column of this object such that

$$\forall i \in [k] : Y'[\cdot, j]_i = Y_{i,j}.$$

P_X is the $p+1$-variate LDE of X to an $[\ell_1] \times [\ell_1]^p$ hypercube containing X for sufficiently large $\ell_1 = \mathsf{poly}(k)$. We sometimes identify this $p+1$-dimensional hypercube as a two dimensional $\ell_1 \times \ell_2$

matrix for $\ell_2 = \ell_1^p$). Additionally, we sometimes treat X as a $k \times k_2$ matrix of elements of a field $\mathbb{F}$ for $k_2 = k^p$, and J as a subset of a larger $l_1 \times l_2$ matrix and each $j \in J$ as $j = (j_1, j_2) \in [\ell_1] \times [\ell_2]$. We define J_2 to be the set of columns that contain elements of J in other words

$$J_2 = \{j_2 \in \mathbb{F}^{k^p} : (j_1, j_2) \in J\}.$$

Protocol 4 Polynomial Folding Protocol

The protocol, (P_1, V_1) has explicit input $(\mathbb{F}, k, p, J, \vec{v}, \kappa)$, for soundness amplification parameter $\kappa > 0$ and implicit input $X \in \mathbb{F}^{k \times k^p}$ that the prover has no access to. This protocol proceeds in two rounds:

1. Prover sends Verifier: for each row $i \in [k]$ of X, send its encoding by $P_{X[i,\cdot]}$ (the $(p-1)$-variate LDE of the i^{th} row of X) restricted to coordinates J_2. We call this matrix $Y \in \mathbb{F}^{k \times |J|}$.

   Verifier: receive $Y' \in \mathbb{F}^{k \times |J|}$, reject if for some $(j_1, j_2) \in J$, the univariate low degree extension of the j_2^{th} column of Y' (i.e., $P_{Y'[\cdot, j_2]}$) on j_1 is not equal to the correct value in $\vec{v}$. In other words reject if

$$\exists (j_1, j_2) \in J : P_{Y'[\cdot, j_2]}(j_1) \neq \vec{v}[j_1, j_2].$$

2. Verifier sends Prover: for each $a \in [log(k/\kappa) + 1]$, send a uniformly random vector $\vec{z_a} \in \mathbb{F}^k$ of Hamming weight $2^a \kappa$.

The output is $(log(k/\kappa) + 1)$ tuples $\{(a, \vec{z_a}, J_2, \vec{v_a} = \vec{z_a} \cdot Y')\}_{a \in [log(k/\kappa)+1]}$.

Theorem 5.3. *For any $\kappa > 0$, ρ-Dispersed distribution $\mathcal{D}^{(p+1)}$ over $[k] \times [k^p]$, the polynomial folding protocol (Protocol 4), (P_1, V_1) on shared input $(J, \vec{v})$ and prover input X produces an output of $(log(k/\kappa) + 1)$ tuples $\{(a, \vec{z_a}, J_2, \vec{v_a} = \vec{z_a} \cdot Y')\}_{a \in [\log(k/\kappa)+1]}$ and obeys the following conditions:*

***Completeness**: If X satisfies $\mathsf{PVAL}(J, \vec{v})$ and we have an honest prover, the verifier does not reject and $\forall a \in [\log(k/\kappa) + 1]$, $\vec{z}_a \cdot X \in \mathsf{PVAL}(J_2, \vec{v}_a)$.*

***Bounded Locality**: $\forall a \in [\log(k/\kappa)]$, in the a-th output of the interactive protocol $\{(a, \vec{z_a}, J_2, \vec{z_a} \cdot Y')\}_{a \in [log(k/\kappa)+1]}$, each coordinate of $\vec{z}_a \cdot X$ is a linear combination of $\tau_a = 2^a \cdot \kappa$ coordinates of X.*

***Soundness**: For $\mathcal{D}^{(p+1)} \in \Delta([n])$, if X is ε-far from $\mathsf{PVAL}(J, \vec{v})$ along $\mu_{\mathcal{D}^{(p+1)}, \mathcal{U}}$, then for any cheating prover P', with all but $((|\mathbb{F}| - 1)^{-1} + e^{-\kappa/(4\log(k))})$ probability over V's coins, either V rejects, or there exists some $a^* \in [\log(k/\kappa) + 1]$ s.t.*

$$\mu_{\mathcal{D}^{(p)}, \mathcal{U}_{k^p}}(\vec{z}_{a^*} \cdot X, \mathsf{PVAL}(J_2, \vec{v}_{a^*})) > \frac{\varepsilon \cdot 2^{a^*}}{4\rho}.$$

In other words,

$$\mathbb{P}_{V_0}\left[\mu_{\mathcal{D}^{(p)}, \mathcal{U}_{k^p}}(\vec{z}_{a^*} \cdot X, \mathsf{PVAL}(J_2, \vec{v}_a)) > \frac{\varepsilon \cdot 2^{a^*}}{4\rho}\right] \geq 1 - ((|\mathbb{F}| - 1)^{-1} + e^{-\kappa/(4\log(k))})$$

This protocol has communication complexity $O(|J| \cdot k \cdot \log(k) \cdot \log |\mathbb{F}|)$ and one round of communication. The honest prover runs in time $\mathsf{poly}(k^t, \log |\mathbb{F}|)$, and the verifier runs in time $\mathsf{poly}(|J|, k, \log |\mathbb{F}|)$.

Note that the verifier never accesses X in this protocol.

Now, $\mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}$ is the new distance measure between the individual row vectors. This distance is equivalently obtained under the process of sampling from $\mathcal{D}^{(p+1)}$ and ignoring the last part of the index, and doing the same for $\mathcal{U}$.

Let $\mathsf{PVAL}_i = \mathsf{PVAL}(J_2, Y'[i,\cdot])$ be the set of vectors $Z \in \mathbb{F}^{k^p}$, such that the low degree extension of Z restricted to the points in J_2 are equal to the values in $Y'[i,\cdot]$. Further, for each $i \in [k]$, define $\varepsilon_i = \mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}(X[i,\cdot], \mathsf{PVAL}_i)$.

The soundness of Theorem 5.3 now relies on the following sequence of lemmas. To begin with, we state the "distance preservation lemma" to prove that the distance from X to $\mathsf{PVAL}(J,\vec{v})$ is maintained for the sum of column marginal distances between $X[i,\cdot]$ and PVAL_i across all the k rows.

Lemma 5.4.

$$\mu_{\mathcal{D}^{(p+1)},\mathcal{U}_n}(X, \mathsf{PVAL}(J,\vec{v})) > \varepsilon \implies \sum_{i=1}^{k} \mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}(X[i,\cdot], \mathsf{PVAL}(J_2, Y'[i,\cdot])) > \frac{k}{\rho}\varepsilon$$

Proof. We proceed by choosing X' which allows us to relate the $\mathcal{D}^{(p+1)}$-distance between X and $\mathsf{PVAL}(J,\vec{v})$, with the distances between the individual rows of X to their corresponding lower-dimensional PVAL instances, but with respect to the marginal $\mathcal{D}^{(p)}$. We then see that this sum is maximal over both the $\mathcal{D}^{(p+1)}$ and the uniform distribution.

More precisely, for each $i \in [k]$, let $X'[i,\cdot]$ be the element of PVAL_i that minimises the hybrid distance, $\mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}(X[i,\cdot], X'[i,\cdot])$. We set X' such that for each $i \in [k]$, it's i^{th} row is $X'[i,\cdot]$. Put together with the fact that the verifier has not rejected at the end of step 1, we immediately observe that $X' \in \mathsf{PVAL}(J,\vec{v})$. Indeed, this holds as the univariate LDE of each column of Y' restricted to J_1, gives us exactly $\vec{v}$.

In what follows, we focus our calculations to distance over the $\mathcal{D}^{(p+1)}$ distribution. The same calculations hold for distances over $\mathcal{U}$ as well ($\mathcal{U}$ is always 1-Dispersed).

$$\begin{aligned}
\sum_{i=1}^{k} \mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}(X[i,\cdot], \mathsf{PVAL}_i) &= \sum_{i=1}^{k} \mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}(X[i,\cdot], X'[i,\cdot]) \\
&\geq \sum_{i=1}^{k} d_{\mathcal{D}^{(p)}}(X[i,\cdot], X'[i,\cdot]) \\
&= \sum_{i=1}^{k} \mathop{\mathbb{P}}_{j' \sim \mathcal{D}^{(p)}} \left[X_{ij'} \neq X'_{ij'} \right]
\end{aligned}$$

This follows as the first expression is the maximum over the distances with respect to the two distributions under consideration. Further,

$$\begin{aligned}
\sum_{i=1}^{k} \underset{j' \sim \mathcal{D}^{(p)}}{\mathbb{P}} \left[X_{ij'} \neq X'_{ij'}\right] &= \sum_{i=1}^{k}\sum_{l=1}^{k} \underset{(i',j') \sim \mathcal{D}^{(p+1)}}{\mathbb{P}} \left[X_{ij'} \neq X'_{ij'} \wedge (i' = l)\right] \\
&\geq \frac{k}{\rho} \sum_{i=1}^{k} \underset{(i',j') \sim \mathcal{D}^{(p+1)}}{\mathbb{P}} \left[X_{ij'} \neq X'_{ij'} \wedge i' = i\right] \\
&= \frac{k}{\rho} \underset{(i',j') \sim \mathcal{D}^{(p+1)}}{\mathbb{P}} \left[X_{i'j'} \neq X'_{i'j'}\right] \\
&= \frac{k}{\rho} d_{\mathcal{D}^{(p+1)}}(X, X')
\end{aligned}$$

This follows from the definition of the marginal distribution and the definition of ρ-Dispersed distributions. We note that this inequality can be tight in certain cases.[20]

Therefore, as observed earlier, we have that $\sum_{i=1}^{k} \mu_{\mathcal{D}^{(p)},\mathcal{U}_{kp}}(X[i,\cdot], \mathsf{PVAL}_i) \geq \frac{k}{\rho} d_{\mathcal{D}^{(p+1)}}(X, X')$, as well as $\sum_{i=1}^{k} \mu_{\mathcal{D}^{(p)},\mathcal{U}_{kp}}(X[i,\cdot], \mathsf{PVAL}_i) \geq \frac{k}{\rho} d_{\mathcal{U}_n}(X, X')$. Thus,

$$\sum_{i=1}^{k} \mu_{\mathcal{D}^{(p)},\mathcal{U}_{kp}}(X[i,\cdot], \mathsf{PVAL}_i) \geq \frac{k}{\rho} \mu_{\mathcal{D}^{(p+1)},\mathcal{U}_n}(X, X') \geq \frac{k}{\rho} \mu_{\mathcal{D}^{(p+1)},\mathcal{U}_n}(X, \mathsf{PVAL}).$$

□

We next have the following lemma on the distance between subspaces on arbitrary metrics that satisfy certain invariance constraints.

Lemma 5.5. *Let $d : \mathbf{V} \times \mathbf{V} \to \mathbb{R}$ be a metric defined over a vector space $\mathbf{V}$ on a field $\mathbb{F}$, such that the following invariance conditions hold on d.*

1. *For every $X, Y \in \mathbf{V}, a \in \mathbb{F}$, $d(aX, aY) = d(X, Y)$.*
2. *For every $X, Y, Z \in \mathbf{V}$, $d(X + Z, Y + Z) = d(X, Y)$.*

Let S and T be two linear subspaces of $\mathbb{F}^n$ (for any finite field $\mathbb{F}$ and $n \in \mathbb{N}$). Suppose that there exists some point $s \in S$ such that $d(s, T) > \varepsilon$. Then, with all but $\frac{1}{|\mathbb{F}|-1}$ probability over the choice of a uniformly random point r from S, $d(r, T) > \frac{\varepsilon}{2}$.

Proof. We sample a uniformly random vector in S by first taking uniformly random $r \in S$ and then, if $r = s$ return s, and if not, take a uniformly random sample from the line between r and s, excluding s.

If $r = s$, then the condition is fulfilled, if not then we look at the line along r, s. We claim that there can be at most one element along this line whose distance from T is less than $\frac{\varepsilon}{2}$. If this is true, the theorem follows as the probability of sampling an element close to T is less than $\frac{1}{|\mathbb{F}|-1}$.

[20] For $\rho = k$, consider a distribution is only supported on one row i_0, if $i = i_0$ the following holds: $\underset{(i',j') \sim *}{\mathbb{P}} \left[X_{ij'} \neq X'_{ij'}\right] = \underset{(i',j') \sim \mathcal{D}^{(p+1)}}{\mathbb{P}} \left[X_{i_0 j'} \neq X'_{i_0 j'} \wedge (i' = i_0)\right]$. In the case that $i \neq i_0$, both sides of this equation are 0.

Suppose otherwise for contradiction and we have r_1, r_2 on the line and $t_1, t_2 \in T$ such that $d(r_1, t_1) < \frac{\varepsilon}{2}$ and $d(r_2, t_2) < \frac{\varepsilon}{2}$. As these points are on the same line, for some $a \in \mathbb{F}$, we have $s = r_1 + a(r_2 - r_1)$ and $t_s = t_1 + a(t_2 - t_1)$. Therefore:

$$\begin{aligned}
d(s, t_s) &= d(r_1 + a(r_2 - r_1), t_1 + a(t_2 - t_1)) \\
&= d((1-a)r_1 + ar_2, (1-a)t_1 + at_2) \\
&\leq d((1-a)r_1 + ar_2, (1-a)t_1 + ar_2) + d((1-a)t_1 + ar_2, (1-a)t_1 + at_2) \\
&\leq d((1-a)r_1, (1-a)t_1) + d(ar_2, at_2) \\
&\leq d(r_1, t_1) + d(r_2, t_2) \\
&< \frac{\varepsilon}{2} + \frac{\varepsilon}{2} \\
&< \varepsilon
\end{aligned}$$

This contradicts the initial assumption that s is far from T, therefore this lemma follows by contradiction. □

Note that $\mu_{\mathcal{D}^{(p)}, \mathcal{U}_{kp}}$ as the maximum of two metrics is a metric. We next prove that we can apply Lemma 5.5 with this distance measure, since it satisfies the constraints required.

Lemma 5.6. *The metric $\mu_{\mathcal{D}^{(p)}, \mathcal{U}_{kp}}$ satisfies both the invariance properties in Lemma 5.5.*

Proof.

- Invariance under vector addition:

$$\begin{aligned}
\mu_{\mathcal{D}^{(p)}, \mathcal{U}_{kp}}(X+Z, Y+Z) &= max(d_{\mathcal{D}^{(p)}}(X+Z, Y+Z), d_{U_{kp}}(X+Z, Y+Z)) \\
&= max(d_{\mathcal{D}^{(p)}}(X, Y), d_{U_{kp}}(X, Y)) \\
&= \mu_{\mathcal{D}^{(p)}, \mathcal{U}_{kp}}(X, Y)
\end{aligned}$$

- Invariance under scalar multiplication:

$$\begin{aligned}
\mu_{\mathcal{D}^{(p)}, \mathcal{U}_{kp}}(aX, aY) &= max(d_{\mathcal{D}^{(p)}}(aX, aY), d_{U_{kp}}(aX, aY)) \\
&= max(d_{\mathcal{D}^{(p)}}(X, Y), d_{U_{kp}}(X, Y)) \\
&= \mu_{\mathcal{D}^{(p)}, \mathcal{U}_{kp}}(X, Y)
\end{aligned}$$

□

We finish by proving the following set of important claims. In Claim 5.7, we prove that there is a set of rows $I \subseteq [k]$ for which for all $i \in I$, $X[i, \cdot]$ is sufficiently far from PVAL_i, given the size of I. Then, in Claim 5.8 we prove that with high probability $\exists a^* \in [\log(k/\kappa) + 1]$ such that a randomly chosen $z_{a^*} \in \mathbb{F}^k$ of Hamming weight roughly 2^{a^*} has some row from I. Finally, in Claim 5.9, we combine these with Lemma 5.5 to show that sampling $\log(k/\kappa) + 1$ many random vectors $\{z_a\}_{a \in [\log(k/\kappa)+1]}$ in the polynomial folding protocol, results in at least one folded instance of PVAL which is sufficiently far from the correspondingly folded instance $z_a \cdot X$.

Claim 5.7. *If the verifier does not reject in Step 1, then there exists an integer $b \in \{0, \cdots, \log(k)\}$, and a subset $I \subseteq [k]$, s.t. $\forall i \in I, \varepsilon_i \geq k\varepsilon/(2^{b+1}\rho)$ and $|I| \geq 2^b/4\log(k)$.*

Claim 5.8. *In Step 2 of protocol 4, for $a \in [log(k/\kappa)+1]$, let I_a be the set of non-zero coordinates in $\vec{z}_a$ (this set is of size $2^a \cdot \kappa$). Take b as guaranteed by Claim 5.7 and $a^* = min(log(k/\kappa), \log(k) - b)$. With all but $e^{-\kappa/4\log(k)}$ probability over the verifier's choice of z_{a^*}, there exists $i^* \in I_{a^*}$ s.t. $\varepsilon_{i^*} > \varepsilon \cdot 2^{a^*}/2\rho$.*

Claim 5.9. *Take a^* as guaranteed by Claim 5.8. With all but $((|\mathbb{F}|-1)^{-1} + e^{-\kappa/4\log(k)})$ probability over the verifier's choice of $\vec{z}_{a^*}$, it holds for $\vec{v}_{a^*} = z_{a^*} \cdot Y'$ that*

$$\mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}(\vec{z}_{a^*} \cdot X, \mathsf{PVAL}(J_2, \vec{v}_{a^*})) > \varepsilon \cdot 2^{a^*}/4\rho.$$

The proofs of the above claims are analogous to the uniform setting and can be found in Appendix A.

Proof of Theorem 5.3. Completeness follows as the only stage where the verifier can reject is in stage 1. There, the honest prover would send the true value of Y which is consistent with $\vec{v}$ on J. This also results in a valid folded version of PVAL, $\mathsf{PVAL}(J_2, \vec{z_a} \cdot Y)$, for which $\forall a \in [\log(k/\kappa)+1]$: $\vec{z}_a \cdot X \in \mathsf{PVAL}(J_2, \vec{z_a} \cdot Y)$, for any random linear combination $\vec{z}_a$ picked by the verifier.

Bounded locality follows from the fact that sampling an element of the folded vector will require queries to the input equal to the Hamming weight of the corresponding folding vector $\vec{z}_a$.

The soundness follows directly from the Claim 5.9 whereby we have that with all but $((|\mathbb{F}| - 1)^{-1} + e^{-\kappa/(4\log(k))})$ probability over the verifier's randomness, X will have $\mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}$ distance at least $\varepsilon \cdot 2^{a^*}/4\rho$ from satisfying at least one of the new instances of PVAL. □

5.3 IPP for PVAL over ρ-dispersed distributions

Now that we have that polynomial folding lemma, we can develop the overall interactive protocol for distinguishing between being in PVAL and being far from PVAL along the $\mu_{\mathcal{D},\mathcal{U}_n}$ distance. The IPP is presented in Protocol 5.

Note that since $\mathcal{D}^{(m-1)}$ is also a ρ-dispersed distribution by Lemma 5.1, we can iterate Theorem 5.3 on $[k]^{m-1}$.

Theorem 5.10. *For $n, k \in \mathbb{N}$, $r \leq \log_k(n)$, $k^r \leq \frac{1}{\varepsilon}$, $r \leq k$, $r = \omega(1)$, $m = \log_k(n)$, a field $\mathbb{F}$ for which $|\mathbb{F}| = \mathsf{polylog}(n)$ such that $10r \leq |\mathbb{F}| \leq 1/\varepsilon$, $\mathcal{D}$ a ρ-Dispersed distribution over $[k]^m$ and $(J, \vec{v})$ defining an instance of PVAL, Protocol 5, (P_0, V_0), satisfies the following properties:*

1. ***Completeness**: If $X \in \mathsf{PVAL}(J, \vec{v})$ then the verifier accepts with probability 1. In other words,*

$$X \in \mathsf{PVAL}(J, \vec{v}) \implies \underset{V_0,\mathcal{O}_\mathcal{D}(X)}{\mathbb{P}}\left[(P_0(X,\mathcal{D}), V_0^{X,\mathcal{O}_\mathcal{D}(X)})(n,\varepsilon) \text{ accepts }\right] = 1.$$

2. ***Soundness**: For $\mathcal{D} \in \Delta([n])$, if $\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}(J,\vec{v})) > \varepsilon$ then the verifier rejects with probability at least $\frac{1}{2}$.In other words,*

$$\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}(J,\vec{v})) > \varepsilon \implies \underset{V_0,\mathcal{O}_\mathcal{D}(X)}{\mathbb{P}}\left[(P_0^*(X,\mathcal{D}), V_0^{X,\mathcal{O}_\mathcal{D}(X)})(n,\varepsilon) \text{ rejects }\right] \geq \frac{1}{2}.$$

Protocol 5 IPP for PVAL over ρ-dispersed distributions

The implicit input is $X \in \mathbb{F}^{k^m}$, the explicit input is $(\mathbb{F}, k, m, J, \vec{v})$ and a round parameter $r \in \mathbb{N}$. Take $n = |X| = k^m$ to be the input message length, and set a soundness amplification parameter $\kappa = 8\log(r) \cdot \log(k)$.

1. Set $\mathcal{W}_0 \longleftarrow (\lambda, \lambda, J, \vec{v})$, where λ is the empty string. For $s \in 1, ..., r$, $\mathcal{W}_s \longleftarrow \Phi$.
2. Proceed in phases $s \longleftarrow 0, ..., r-1$:

   For each $((\vec{z}_1, ..., \vec{z}_s), (a_1, ..., a_s), J, \vec{v})$ in $\mathcal{W}_s$, in parallel, the prover and the verifier run Protocol 4 (the Polynomial Folding Protocol) with $k = k$, $l_1 = \mathsf{poly}(k)$, $p = m - s - 1$, $\kappa = 8 \cdot \log(k)\log(r)$. Taking $X_s = \vec{z}_s \cdot (... \cdot (\vec{z}_1 \cdot X))$, the instance is $(X_s, J, \vec{v})$.

   The output of each run is a collection of tuples $\{(a, \vec{z_a}, J_2, \vec{z_a} \cdot Y')\}_{a \in [log(k/\kappa)+1]}$. For each $a \in [log(k/\kappa) + 1]$, add $((\vec{z}_1, ..., \vec{z}_s, \vec{z}_{s+1}, a), (a_1, ..., a_s, a), J_{s+1}, \vec{v}_{s+1}, a)$ to $\mathcal{W}_{s+1}$.
3. For each $((\vec{z}_1, ..., \vec{z}_r), (a_1, ..., a_r), J_r, \vec{v}_r) \in \mathcal{W}_r$, do the following in parallel:
   (a) Prover sends Verifier: $X_r = \vec{z}_r \cdot (... \cdot (\vec{z}_1 \cdot X))$ where $X_r \in \mathbb{F}^{k^{m-r}}$.
   (b) Verifier: receive X'_r and check if $P_{X'_r}|_{J_r} = \vec{v}_r$, else reject immediately.
   (c) Verifier: set $\varepsilon_r = \varepsilon \cdot \prod_{s=1}^{r} \rho^{-1}(2^{a_s}/4)$. Pick $(10/\varepsilon_r)$ uniformly random coordinates in X'_r and then the same number of coordinates along the $\mathcal{D}$ distribution. For each coordinate j that was picked, verify that $X'_r[j] = (\vec{z}_r.(....(\vec{z}_1.X)))[j]$ by querying the appropriate coordinates in the original input message X. If any of these checks fail, then reject immediately.
4. If the verifier did not reject so far then it accepts.

This protocol has query complexity $\rho^r(1/\varepsilon)^{1+o(1)}$, sample complexity $\rho^r(1/\varepsilon)^{1+o(1)}$, communication complexity $(n/k^r + |J| \cdot k)(1/\varepsilon)^{o(1)}$, and the number of messages is $(2r+1)$ (round complexity is $r+1$). The honest prover runs in $\mathsf{poly}(n)$ time, and the verifier runs in $((\rho^r/\varepsilon) + n/k^r + |J|k)n^{o(1)}$ time.

Proof. The honest prover runs in time $\mathsf{poly}(n)$, this follows from construction, as all the prover sends the verifier is a series of matrices Y' and inner products of the various values of z with X. The communication complexity will be the total communication from the polynomial folding protocol for each iteration of step 2 $(1 \leq s < r)$ along with the complexity of sending each value of X_r in step 3a.

$$\begin{aligned}
&\sum_{s=1}^{r} \log(k/\kappa+1)^s \cdot O(|J| \cdot k \cdot \log(k) \cdot \log|\mathbb{F}|) + \log(k/\kappa+1)^r \frac{n}{k^r} \log|\mathbb{F}| \\
&= (\log(k/\kappa)+1)^r \cdot O(|J| \cdot k \cdot \log(k) \cdot \log|\mathbb{F}|) + \log(k/\kappa+1)^r \frac{n}{k^r} \log|\mathbb{F}| \\
&= O(\log(k))^r \cdot \log|\mathbb{F}| \cdot O(n/k^r + |J| \cdot k \cdot \log(k)) \\
&= \frac{1}{\varepsilon^{o(1)}}(n/k^r + |J| \cdot k) \\
&= (n/k^r + |J| \cdot k)\frac{1}{\varepsilon^{o(1)}}.
\end{aligned}$$

This follows from the fact that $k^r = O\left(\frac{1}{\varepsilon}\right)$ and $|\mathbb{F}| \leq 1/\varepsilon$.

The completeness of this protocol follows as the only parts it can reject are the polynomial folding protocols which are perfectly complete and step 3b which says that X'_r is consistent with $\vec{v}_r$. The latter holds since $X'_r = X_r$ is consistent with $\vec{v}_r$.

Additionally, the verifier runs in time $n^{o(1)} \cdot ((\rho^r/\varepsilon) + n/k^r + k|J|)$, this follows from construction. The first part from sampling X, the second from processing the X_r sent by the prover and the last part from processing the Y' sent by the prover in the polynomial folding protocol. Note also that $\log|\mathbb{F}|$, the number of rounds and the number of folded instances of PVAL are encompassed by the $n^{o(1)}$ term.

The query complexity only has contributions from step 3c. At this step for each tuple $(a_1, \cdots, a_r)$, the verifier takes $10/\varepsilon_r = 10/\varepsilon \cdot \prod_{i=1}^{r} \frac{4\rho}{2^{a_s}}$ samples and uniform queries to X_r. This results in total sample complexity

$$\sum_{(a_1,\cdots,a_r)\in[\log(k/\kappa)+1]^r} \frac{10}{\varepsilon} \prod_{s=1}^{r} 4\rho/2^{a_s} \leq \rho^r/\varepsilon^{1+o(1)}.$$

Furthermore, the number of queries taken will be the number of samples and queries to X_r times the number of queries from X to obtain a query from X_r which, by bounded locality of the polynomial folding protocol is equal to

$$\tau_r = \prod_{s=1}^{r} 2^{a_s}\kappa.$$

Therefore the total query complexity is

$$\begin{aligned}
\sum_{(a_1,\cdots,a_r)\in[\log(k/\kappa)+1]^r} \frac{10\tau_r}{\varepsilon_r} &= \sum_{(a_1,\cdots,a_r)\in[\log(k/\kappa)+1]^r} \frac{10}{\varepsilon}\left(\prod_{s=1}^{r} 4\rho/2^{a_s}\right)\left(\prod_{s=1}^{r} 2^{a_s}\kappa\right) \\
&= \sum_{(a_1,\cdots,a_r)\in[\log(k/\kappa)+1]^r} \frac{10}{\varepsilon}(4\rho\kappa)^r \\
&= (\log(k/\kappa)+1)^r \frac{10 \cdot (4\rho\kappa)^r}{\varepsilon} \\
&= \rho^r/\varepsilon^{1+o(1)}
\end{aligned}$$

This follows since $\log^r(k) = 1/\varepsilon^{o(1)}$, $4^r = 1/\varepsilon^{o(1)}$, and $\kappa^r = (\log(k)\log(r))^r < \log^{2r}(k) = 1/\varepsilon^{o(1)}$ as $r < k$.

The total set of messages are those from r iterations of the polynomial folding protocol along with 1 message from step 3a, in total this amounts to $2r+1$ messages and round complexity r.

Soundness follows as given the initial input, X that is ε-far from PVAL along the $\mu_{\mathcal{D},\mathcal{U}_n}$ distance. By soundness of the polynomial folding protocol (Theorem 5.3) and with a union bound over r such rounds, with all but $r \cdot ((|\mathbb{F}|-1)^{-1} + e^{-\kappa/4\log(k)})$ probability, there is some resulting tuple: $U_r^* = ((\vec{z}_1^*, ..., \vec{z}_r^*), (a_1^*, ..., a_r^*), J_r^*, \vec{v}_r^*)$, such that the instance $(X_r^*, J_r^*, \vec{v}_r^*)$ specified by U_r^* is ε_r^*-far from $\mathsf{PVAL}(J_r^*, \vec{v}_r^*)$ on $P_{X_r'}$ along the $\mu_{\mathcal{D}^{(m-r)},\mathcal{U}_{n/k^r}}$ distance, where:

$$\varepsilon_r^* \geq \varepsilon \cdot \prod_{s=1}^{r} \cdot 2^{a_s^*}/4\rho.$$

As we will assume that the verifier does not reject in Step 3, the corresponding $X_r'^*$ has to satisfy this instance of PVAL, therefore it must be far from X_r^* along $\mu_{\mathcal{D}^{(m-r)},\mathcal{U}_{n/k^r}}$. In Step 3c, the verifier picks $10/\varepsilon_r^*$ uniformly random coordinates and then the same number of coordinates along the folded $\mathcal{D}$ distribution. As the $\mu_{\mathcal{D}^{(m-r)},\mathcal{U}_{n/k^r}}$-distance between X_r^* and $X_r'^*$ is at least ε_r^* far along one of these distributions then the verifier will reject with probability at least $9/10$.

A union bound over all the r-many polynomial foldings, ensures that the soundness error is at most:

$$r \cdot (1/(|\mathbb{F}|-1) + e^{-\kappa/4\log(k)}) + 1/10 \leq r \cdot (1/(10r-1) + e^{-2\log(r)}) + 1/10 < 1/2$$

□

Remark 9. *The only step in Protocol 5 that is altered from the proof of the* IPP *under uniform distribution (Theorem 2.1) is in Step 3c, which involves sampling indices from the folded indices along $\mathcal{U}$, as well as $\mathcal{D}$. Thus, the number of queries increase by a factor of ρ^r by the fact that we have ρ-Dispersed distributions, but the prover run-time and communication complexity remain unchanged.*

Finally, the IPP for languages computable by low-depth circuits over ρ-dispersed distributions is provided in Protocol 6 and the proof of Theorem 5.2 is provided in Appendix B.

6 IPPs over Product Distributions

In Section 4, we show the existence of distribution-free IPPs for NC languages with query complexity $O\left(\frac{1}{\varepsilon}\right)$ and communication complexity $\tilde{O}\left(\varepsilon \cdot n + \frac{1}{\varepsilon}\right)$. Motivated by matching the communication complexity of $\varepsilon^{1-o(1)} \cdot n$ from [RVW13], in Section 5 we construct IPPs for ρ-dispersed distributions that achieve this communication complexity, while having query complexity $\frac{\rho^{\log(1/\varepsilon)/\log\log(n)}}{\varepsilon^{1+o(1)}}$. This, however, does not provide optimal query complexity for certain product distributions, eg., distributions that are concentrated on a small set of rows along some dimension, as they could be $\Omega(k)$-dispersed.

As such, in this section, we construct IPPs over product distributions that match the complexities of the uniform IPP from [RVW13]. In what follows, we prove Theorem 1.3 to show a white-box IPP for NC over any m-product distribution samplable using polynomial-sized circuits. The theme

Protocol 6 IPP over ρ-dispersed distributions, (P, V), for a language L with a circuit of size $S(n)$ and depth $\Delta_L(n)$.

Let $(P_{\mathsf{NC}}, V_{\mathsf{NC}})$ be the interactive reduction from Theorem 4.3 and let (P_0, V_0) be Protocol 5 from Theorem 5.10.
The input is $X \in \{0,1\}^n$ and we set $k = \log(n)$, $m = \log_k(n)$, $r = \log(1/\varepsilon)/\log(k)$ and $|\mathbb{F}| = \mathsf{poly}(n)$.

1. (P, V) run $(P_{\mathsf{NC}}, V_{\mathsf{NC}})$ on X. The output of this protocol is $J \subset \mathbb{F}^m$ for which $|J| = 4\varepsilon n \cdot \log(n)$ and $\vec{v} \in \mathbb{F}^t$.
2. (P, V) run (P_0, V_0) to verify membership of X, identified as an element in $\mathbb{F}^{k^m}$, in $\mathsf{PVAL}(J, \vec{v})$. V rejects, if V_0 rejects.
3. V accepts otherwise.

of this section is to use the sample oracle to learn the distribution $\mathcal{D}$ and then use the learned distribution to design an IPP over a distribution family $\mathcal{F}$, improving the query complexity of the IPP from Theorems 4.5 and 5.2 which does not acquire any information about the distribution. In Appendix D, we show a framework for translating any interactive proof that can learn a distribution family $\mathcal{F}$ using only samples, into a black-box IPP for NC over any distribution in $\mathcal{F}$, that generalises the intuition for Theorem 1.3. We start by defining the relevant family of distributions.

Definition 6.1 (m-Product Distributions). *Let $\mathbb{F}$ be any field. For any $n \in \mathbb{N}$ and any integral function $m = m(n)$, let k be an integer such that $n = k^m$. Then $\mathcal{D} = \{\mathcal{D}_n\}$ is called an m-product distribution ensemble, where $\mathcal{D}_n$ is a distribution over $[k]^m$ (by fixing some canonical bijection from $[k]^m$ to $[n]$), if there exists distributions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_m \in \Delta([k])$, such that for any index $(i_1, \ldots, i_m) \in [k]^m$, $\mathcal{D}(i_1, \ldots, i_m) = \prod_{r=1}^{m} \mathcal{D}_r(i_r)$.*

To understand this better, consider a product distribution $\mathcal{D}$ over $[n]$ where the probability of picking each index $i \in \{0,1\}^{\log(n)}$ is given by $\log(n)$ independent random variables over $\{0,1\}$, where the j^{th}-bit in the index is 1 with probability p_j. Alternatively, we can view the input as a $(\log(n))$-dimensional tensor, having 2 elements in each dimension (by fixing some enumeration over the cells into input indices). Now, $\prod_j p_j$ represents the probability of sampling a cell in the tensor. m-product distributions generalise this by considering product distributions over m-dimensional tensors, having k elements in each dimension, such that $k^m = n$.

Given this, we can define the oracle $\mathcal{O}_{\mathcal{D}}$ that provides labeled samples to the verifier. For example, any 2-product distribution $\mathcal{D}_n$ over $[\sqrt{n}] \times [\sqrt{n}]$ can be defined as a pair $\mathcal{D}_n = \mathcal{D}_1 \times \mathcal{D}_2$, where $\mathcal{D}_1$ and $\mathcal{D}_2$ are distributions over $[\sqrt{n}]$. In such a case, the implicit input $X \in \{0,1\}^n$ can be viewed as a matrix in $\{0,1\}^{\sqrt{n} \times \sqrt{n}}$ (by fixing a bijection between the indices of X and cells in the matrix), and $\mathcal{O}_{\mathcal{D}}$ can be viewed as an oracle that provides a sample $((i,j), X_{ij})$, where (i,j) are the row and column indices of this matrix sampled from $\mathcal{D}$. Similarly, for a $\left(\frac{\log(n)}{\log\log(n)}\right)$-product distribution $\mathcal{D}$, we can view the $\mathcal{O}_{\mathcal{D}}$ as being defined over $\left(\frac{\log(n)}{\log\log(n)}\right)$-dimensional tensors with $\log(n)$ elements in each dimension.

We next state the following parallel variant of the Set Lower Bound protocol [GS89] (observed in [BT06]) that will be required in our IPP.

Lemma 6.1 (Corollary 2.7 of [BT06])**.** *For any circuit $C : \{0,1\}^\ell \to \{0,1\}^{\log(k)}$, any $\tau \in (0,1)$, define the promise problem $\Pi = \{\Pi_{\ell,k}\}$*

$$\Pi^Y_{\ell,k} := \{(C, \tau, y_1, p_1, \cdots, y_k, p_k) : \forall i \in [k] : |C^{-1}(y_i)| \geq p_i k\}$$
$$\Pi^N_{\ell,k} := \{(C, \tau, y_1, p_1, \cdots, y_k, p_k) : \exists i \in [k] : |C^{-1}(y_i)| \leq (1-\tau) p_i k\}$$

Then, for any $\delta > 0$, $\ell, k \in \mathbb{N}$, there exists a constant-round interactive proof for $\Pi_{\ell,k}$ with completeness probability $1-\delta$ and soundness probability δ. The verifier runs in time $O\left(\frac{\mathsf{poly}(|C|)\cdot k^2}{\delta\tau^2}\right)$ and communication complexity of the interactive proof is $O\left(\frac{\ell\cdot k^2}{\delta\tau^2}\right)$, where $|C|$ is the size of the circuit C in terms of its input size ℓ. Moreover, the honest prover runs in time at most $2^\ell \cdot \mathsf{poly}(k, |C|)$.[21] *In particular, if C is a polynomial sized-circuit then $\ell = \mathsf{polylog}(k)$ and thus, $|C|$ is $\mathsf{polylog}(k)$ as well.*

We next define the notion of concatenated languages.

Definition 6.2 (Concatenated languages)**.** *For any fixed k, m, define $g^{\mathsf{cat}} : \mathbb{F}^{k^m} \to \mathbb{F}^{[k+1]\times[k]\cdots\times[k]}$, as the map that concatenates $0^{k^{m-1}}$ to the first dimension of a tensor in $[k]^m$. In other words, for any $X \in \mathbb{F}^{[k]^m}$, we have*

$$g^{\mathsf{cat}}(X) = \begin{cases} X_{i_1,\ldots i_m}, & \text{if } 1 \leq i_1, \ldots, i_m \leq k, \\ 0, & \text{if } i_1 = k+1 \text{ and, } \forall 1 < \ell \leq m, 1 \leq i_\ell \leq k \end{cases}$$

Moreover, for any language L, we define $L_0 \subseteq \mathbb{F}^{[k+1]\times[k]\ldots[k]}$ as $L_0 = \{g^{\mathsf{cat}}(X) \mid X \in L \cap \mathbb{F}^{[k]^m}\}$.

For example, when $X \in \{0,1\}^{[\sqrt{n}]^2}$, $g^{\mathsf{cat}}(X)$ denotes the matrix obtained by concatenating the vector $0^{\sqrt{n}}$ as the last row. When $X \in \{0,1\}^n$, g^{cat} appends a 0 to the end of X. In this case, for any language $L \in \{0,1\}^*$, L_0 is defined as $\{(X \circ 0) \mid X \in L_n\}$. Note that, one can easily generalise Definition 6.2 to concatenating any $w \in \mathbb{F}^{k^{m-1}}$ along the j^{th}-dimension of X, for some $j \leq m$, but for the purposes of this section this definition suffices.

In Section 6.1, we show a reduction from testing for a language L over a *known* m-product distribution, to testing a closely related language L' over a *granular distribution*.

6.1 Granularisation

Definition 6.3 (m-grained distributions)**.** *We say that a distribution $\mathcal{D}$ in $\Delta([n])$ is m-grained, for some $m \in \mathbb{N}$, if for every $i \in [n]$, there exists an $1 \leq a_i \leq m$ such that $\mathcal{D}(i) = \frac{a_i}{m}$.*

Definition 6.4 ($\mathcal{D}$-extending a matrix)**.** *Let $\mathcal{D}$ be a grained distribution over $[k]$ and let $B = \{b_1, \ldots, b_k\}$ be it's granularities. Let X be a matrix in $\mathbb{F}^{k\times k}$. We call another matrix $M \in \mathbb{F}^{(k+r)\times k}$, where $r = \sum_j (b_j - 1)$, as a $\mathcal{D}$-extension of X if the following hold for any row M_i, $1 \leq i \leq k+r$.*

$$M_i = \begin{cases} X[i,\cdot], & \text{if } 1 \leq i \leq k \\ X[1,\cdot], & \text{if } k < i \leq k + b_1 - 1 \\ X[j,\cdot], & \text{for } j \in [k], \text{ such that } k + \sum_{\ell=1}^{j-1}(b_\ell - 1) < i \leq k + \sum_{\ell=1}^{j}(b_\ell - 1) \end{cases} \tag{5}$$

[21] The honest prover may have to go over all possible inputs to C to find one which is mapped to $0^{\log(p_i k)}$ by a pairwise independent hash function in the Goldwasser-Sipser set lower bound protocol, for each $i \in [k]$. This implies the honest prover running time stated in Lemma 6.1.

This definition naturally generalises to $\mathcal{D}$-extend a tensor $X \in \mathbb{F}^{k^m}$ to another tensor $M \in \mathbb{F}^{(k+r)\cdot k^{m-1}}$.

Intuitively, the $\mathcal{D}$-extension $M \in \mathbb{F}^{(k+r)\times t}$ is just the matrix X with $b_1 - 1$ repetitions of row X_1 appended to its last row, and so on in sequential order, until $b_k - 1$ repetitions of row X_k are appended at the end. In the case where $m = 1$, the $\mathcal{D}$-extension just appends the relevant bits to the end of the string.

Lemma 6.2 (The granularisation algorithm)**.** *Let L be any language and $\varepsilon > 0$. Then the following hold true.*

- *Let $X \in \{0,1\}^n$ be an input to L. Let $\mathcal{D}$ be any distribution over $[n]$ such that $\mathcal{D}(i) = p_i$ for every $i \in [n]$.*

  Then, there exists an algorithm $\mathcal{A}_{\mathsf{gran}}$ that takes as input $\{p_1, \ldots, p_n\}$ and outputs the granularities $\{a_1, \ldots, a_{n+1}\}$ of an $8n$-grained distribution $\mathcal{D}'$ over $[n+1]$ (i.e., for every $j \in [n+1]$, $\mathcal{D}'(j) = a_j/8n$), that runs in time $O(n)$, such that:

  - *If $X \in L$, then for $X' = g^{\mathsf{cat}}(X)$, $X' \in L_0$.*
  - *If $d_{\mathcal{D}}(X, L) > \varepsilon$, then $d_{\mathcal{D}'}(X', L_0) > \varepsilon/2$.*

- *Let $X \in \{0,1\}^{k^m}$ be an input to L (take $n = k^m$). Let $\mathcal{D} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_m$ be an m-product distribution, where each $\mathcal{D}_i \in \Delta([k])$. Let $\mathcal{D}_1$ be described by the probability distribution vector $\{p_{11}, \ldots, p_{1k}\}$.*

  Then, $\mathcal{A}_{\mathsf{gran}}$ takes as input $\{p_{11}, \ldots, p_{1k}\}$ and outputs $\{a_{11}, \ldots, a_{1(k+1)}\}$ as the granularities of an $8k$-grained distribution $\mathcal{D}'_1$ over $[k+1]$, running in time $O(k)$, such that:

  - *If $X \in L$, then for $X' = g^{\mathsf{cat}}(X)$, $X' \in L_0$.*
  - *Let $\mathcal{D}' = \mathcal{D}'_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_m$ be defined over $[k+1] \times [k] \times \cdots \times [k]$. If $d_{\mathcal{D}}(X, L) > \varepsilon$, then $d_{\mathcal{D}'}(X', L_0) > \varepsilon/2$.*

While Lemma 6.2 is inspired from [Gol20], their work focuses on the reduction from testing whether an *unknown* input distribution (via samples) equals some fixed distribution $\mathcal{D}$, to testing whether an unknown distribution equals the uniform distribution (via granular distributions). In our case, firstly the input distribution is known, and further, our focus is on the property testing setting where an implicit input string is provided. The proof of this Lemma is provided in Appendix C.

6.2 The White-box IPP for PVAL

We start with a white-box IPP for the PVAL problem over polynomially-samplable m-product distributions.

Theorem 6.3. *For any $m, n \in \mathbb{N}$, let $\mathcal{F}$ be a set of polynomially samplable m-product distributions over $[k]^m$, such that $n = k^m$. Let $\mathbb{F}$ be a field such that $|\mathbb{F}| = \mathsf{polylog}(k)$. Let $J \subset \mathbb{F}^m$ of size t and $\vec{v} \in \mathbb{F}^t$. Let $r \in \mathbb{N}$ be the round parameter such that $10 < r \leq \log(1/\varepsilon)/\log(k)$ and $|\mathbb{F}| > 10r$.*

Then, for every $\varepsilon > 0$ and $\mathcal{D} \in \mathcal{F}$, Protocol 8 is a white-box IPP for $\mathsf{PVAL}(\mathbb{F}, k, m, J, \vec{v})$ over $\mathcal{F}$ with proximity parameter ε, and completeness and soundness probabilities $2/3$, where the soundness promise is over the $\mu_{\mathcal{D},\mathcal{U}}$ metric.

This IPP *has query complexity* $1/\varepsilon^{1+o(1)}$*, communication complexity* $\mathsf{polylog}(n)\cdot\left(k^2+\frac{k}{\varepsilon^{o(1)}}\cdot\varepsilon\cdot n\right)$*, and the verifier runs in* $n^{o(1)}(\varepsilon\cdot n\cdot k+k^2+\frac{1}{\varepsilon})$ *time. Moreover, the* IPP *has* $O(r)$ *many rounds and the honest prover runs in* $2^{\mathsf{polylog}(n)}$ *time.*

Proof. Let $\mathcal{D}$ be a fixed (but unknown) m-product distribution given as $\mathcal{D}_1\times\cdots\times\mathcal{D}_m$, where each $\mathcal{D}_i$ is supported on $[k]$. Let $C:\{0,1\}^{\mathsf{polylog}(n)}\to\{0,1\}^{\log(n)}$ be the polynomial-sized circuit that samples $\mathcal{D}$ (i.e., for every $i\in[n],\mathbb{P}_{x\sim U_{\mathsf{polylog}(n)}}[C(x)=i]=\mathcal{D}(i)$).

Let $X\in\{0,1\}^{k^m}$ be the implicit input to $\mathsf{PVAL}(J,\vec{v})$. We view X as an m-dimensional tensor with length k in each dimension.[22] Similar to the setting in Section 4, the input X either has the promise that it belongs to $\mathsf{PVAL}(J,\vec{v})$ or that $\mu_{\mathcal{D},\mathcal{U}_n}(X,\mathsf{PVAL}(J,\vec{v})>\varepsilon$.

Notation: For any $0\le\ell\le r-1$ and every $j=(j_0,\ldots,j_{m-1})$ in J, let $J_\ell\subset\mathbb{F}^{k^{m-\ell}}$ be the set of points specified by the last $(m-\ell)$ coordinates of any point in J. Following a similar process as Protocol 5 (in Section 5.3), we build a $(\log(8k/\kappa))+1)$-arity tree of depth r, before making the queries in the resulting folded instances in each leaf of this tree.

Any node in layer ℓ has a label $(a_1,\ldots,a_\ell)$ that specifies the tuple $(z_1,\ldots,z_\ell,a_1,\ldots,a_\ell,J_\ell,\vec{v}_\ell)$ (note that $J_0=J$ and $\vec{v}_0=\vec{v}$). Here, each $z_i\in\mathbb{F}^{8k},a_i\in[\log(8k/\kappa)+1]$, such that $\mathsf{Hwt}(z_i)=2^{a_i}\kappa$, and $\vec{v}_\ell$ is some vector in $\mathbb{F}^t$. $\mathcal{S}_\ell$ maintains the set of nodes in any layer ℓ, where $\mathcal{S}_0=\{\lambda,\lambda,J,\vec{v}\}$ is just the original instance $(X,J,\vec{v})$ and is written this way for technical reasons. Any tuple in $\mathcal{S}_\ell$ determines the corresponding folded instance X_ℓ after ℓ rounds and the corresponding PVAL claim of satisfying v_ℓ with respect to J_ℓ.

It is worth defining the folded instance $X_\ell\in\mathbb{F}^{k^{m-\ell}}$ in more detail. Let $\mathcal{E}_i$ supported on $[k+1]$ be the granular approximation to $\mathcal{D}_i$ obtained from $\mathcal{A}_{\mathsf{gran}}$. We abuse the dot product notation, to define X_ℓ as the *ℓ-wise dot product* of $z_1,\ldots,z_{\ell-1}\in\mathbb{F}^{8k}$ with X, as $z_\ell\cdot(\cdots(z_1\cdot X))$, where at each stage the dot product is in fact, computed between z_i and the $\mathcal{E}_i$-extension of $g^{\mathsf{cat}}(X_{i-1})$ (with X_0 set to X). For eg., if $U_1\in\mathbb{F}^{8k\times k^{m-1}}$ is the $\mathcal{D}'_1$-extension of $g^{\mathsf{cat}}(X_0)$, then $z_1\cdot X\in\mathbb{F}^{k^{m-1}}$ is in fact, the dot product, $z_1\cdot U_1=\sum_{j=1}^{8k}z_{1j}\cdot U_1[j,\cdot]\in\mathbb{F}^{k^{m-1}}$.

The extended polynomial protocol is similar to Protocol 5, except that it outputs tuples with respect to extended matrices. Using that we construct white-box IPP over $\mathcal{F}$ in Protocol 8.

Completeness: In any invocation of the extended polynomial folding protocol, if the prover sends the matrix Y, the verifier always accepts. Moreover, by the definition of matrix extensions, we see that $X'\in\mathbb{F}^{8k\times k^{s-1}}$, which is the extension of $g^{\mathsf{cat}}(X)$ using B, also obeys similar consistency claims, i.e., for each $i\in[8k],P_{X'[i,\cdot]}$ evaluate on every point in J_2 is equal to $U[i,\cdot]$.[23] This means that for each $a\in[\log(8k/\kappa)+1]$ and $z_a\in\mathbb{F}^{8k}$, by the linearity of computing the LDE on $z_a\cdot X'$, we see that $z_a\cdot X'\in\mathsf{PVAL}(J_2,z_a\cdot U)$.

Now, for the completeness of IPP itself, we first see that, since all the extended polynomial folding instances are YES instances with probability 1, so are the ones in $\mathcal{S}_r$. Thus, the honest prover would send the correct X_r and the verifier will not reject. In fact, the only place that the verifier may not accept is during the learning step, with probability at most $1/20r$; taking a union bound over the r rounds, we see that Protocol 8 accepts with probability at least $19/20$.

[22] For simplicity, we fix a bijection between the indices of the string X and the cells of a tensor in $\mathbb{F}^{k^m}$, e.g., in the lexicographic order of enumerating the cells of a tensor.

[23] In particular, when $X'_i=0^{k^{s-1}}$, we see that $P_{X'_i}$ is identically zero over $\mathbb{F}^{s-1}$.

Protocol 7 Extended Polynomial Folding Protocol

Explicit Inputs: The granularity set $B = \{b_1, \ldots, b_{k+1}\}$ and $(\mathbb{F}, k, s, \hat{J}, \vec{v}, \kappa)$, where $\hat{J} \subset \mathbb{F}^s$ of size t and $\vec{v} \in \mathbb{F}^t$.

Prover Input: The prover input is $X \in \mathbb{F}^{k^s}$ (note that this is the implicit input to the verifier, but it is unused).

Let $\hat{J} = (J_1, J_2)$, where $J_2 \subset \mathbb{F}^{s-1}$.

1. For each $i \in [k]$, let $X[i, \cdot]$ be the i^{th} row of X. The prover computes $P_{X[i,\cdot]}$ on every point in J_2. It sends these values in the matrix $Y \in \mathbb{F}^{k \times t}$.
2. The verifier receives $\tilde{Y} \in \mathbb{F}^{k \times t}$, and rejects if there exists $j = (j_1, j_2) \in \hat{J}$ where $j_2 \in \mathbb{F}^{s-1}$, such that the univariate LDE of degree at most $(k-1)$ of the j_2^{th}-column in $\tilde{Y}$, $P_{\tilde{Y}[\cdot, j_2]}$ evaluated on j_1 is not equal to $\vec{v}[j]$.
3. The verifier uses B to extend the matrix $g^{\mathsf{cat}}(\tilde{Y})$ into $U \in \mathbb{F}^{8k \times t}$.
4. For each $a \in [\log(8k/\kappa) + 1]$, the verifier samples a uniformly random vector $z_a \in \mathbb{F}^{8k}$ of Hamming weight $2^a \cdot \kappa$ and sends it to the prover. It then outputs $\log(8k/\kappa)+1$ such tuples $(z_a, a, J_2, z_a \cdot U))$ (the usual dot product).

Soundness: The proof of soundness follows a similar chain of steps as that of Protocol 5. Towards this end, suppose that $\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}) > \varepsilon$. We establish a new distance preservation lemma for analysing the extended polynomial folding protocol. The idea here is to use the distribution extended instances to provide a tighter analysis of the query complexity.

For any ℓ, such that $0 \le \ell \le r-1$, consider the ℓ^{th} round of the extended polynomial folding protocol. With high probability, from Lemma 6.1, we see that for each $i \in [k]$, $\mathcal{P}_{\ell+1}(i) \ge (1-\tau) \cdot \mathcal{D}_{\ell+1}(i)$. Moreover, let $\mathcal{E}_{\ell+1}$ be the granularised distribution on $[k+1]$ output by $\mathcal{A}_{\mathsf{gran}}$ on input $\mathcal{P}_{\ell+1}$. Furthermore, we define the following truncated product distributions, $\widehat{\mathcal{D}}_{\ell+1} = \mathcal{D}_{\ell+1} \times \cdots \times \mathcal{D}_m$ supported over $[k]^{m-\ell}$ and $\widehat{\mathcal{U}}_{\ell+1}$ as the uniform distribution over $[k]^{m-\ell}$ (note that $\widehat{\mathcal{D}}_1 = \mathcal{D}$ and $\widehat{\mathcal{U}}_1$ is the uniform distribution over $[k]^m$).

Finally, define $X'_\ell \in \mathbb{F}^{8k \times k^{m-\ell-1}}$ as the $\mathcal{E}_{\ell+1}$-extension of $g^{\mathsf{cat}}(X_\ell)$ and $U_\ell \in \mathbb{F}^{8k \times t}$ as the $\mathcal{E}_{\ell+1}$-extension of $g^{\mathsf{cat}}(\widetilde{Y}_\ell)$, where $\widetilde{Y}_\ell$ is the proof in the ℓ^{th} round of Protocol 7.

Lemma 6.4 (Distance preservation lemma for product distributions)**.** *For any $0 \le \ell \le r-1$ and for any $\gamma > 0$,*

$$\mu_{\widehat{\mathcal{D}}_{\ell+1},\widehat{\mathcal{U}}_{\ell+1}}(X_\ell, \mathsf{PVAL}(J_\ell, \vec{v}_\ell)) > \gamma \implies \sum_{i=1}^{8k} \mu_{\widehat{\mathcal{D}}_{\ell+2},\widehat{\mathcal{U}}_{\ell+2}}(X'_{\ell,i}, \mathsf{PVAL}(J_{\ell+1}, U_{\ell,i})) > 2k(1-\tau)\gamma$$

Proof. For ease of exposition, we prove the statement for the case where $\ell = 0$. The lemma statement follows from the same calculations for any $\ell \le r-1$.

Recall that $X_0 = X, J_0 = J, \vec{v}_0 = \vec{v}$ and when $\ell = 0$, γ equals ε. Let $B = \{b_1, \ldots, b_{k+1}\}$ be granularities of the distribution $\mathcal{E}_1$ returned by $\mathcal{A}_{\mathsf{gran}}$ on input $\mathcal{P}_1$. Define $X' \in \mathbb{F}^{8k \times k^{m-1}}$ and $U \in \mathbb{F}^{8k \times t}$ to be the $\mathcal{E}_1$-extensions of $g^{\mathsf{cat}}(X)$ and $g^{\mathsf{cat}}(\widetilde{Y})$ respectively.

Protocol 8 White-box IPP for PVAL over m-product distributions

Implicit Input: The implicit input is $X \in \mathbb{F}^{k^m}$, where $|X| = k^m = n$. The verifier can access X using a query oracle. The verifier is also given a circuit $C : \{0,1\}^{\mathsf{polylog}(n)} \to \{0,1\}^{\log(n)}$, that samples a fixed (but unknown) m-product distribution $\mathcal{D} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_m$ over $[k]^m$, which the verifier may use to simulate the sample oracle $\mathcal{O}_{\mathcal{D}}$ (or some other distribution), by querying X.

Explicit Inputs: $(\mathbb{F}, k, m, J, \vec{v})$, $\varepsilon > 0$ and a round parameter $r \leq \frac{\log(1/\varepsilon)}{\log(k)}$.

Prover Access: The prover can access X in its entirety and also has access to C.

Input Promise: Either $X \in \mathsf{PVAL}(J, \vec{v})$ or $\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v}) > \varepsilon$.

The Protocol:
Set $\mathcal{S}_0 = (\lambda, \lambda, J, \vec{v})$. Further, set $B_0 = \emptyset, X_0 = X, J_0 = J$, and $\vec{v}_0 = \vec{v}$.
For each round $0 \leq \ell \leq r-1$, do the following:

1. **Learn $\mathcal{D}_{\ell+1}$:** The prover sends the $(\ell+1)^{\text{th}}$ marginal distribution $\mathcal{P}_{\ell+1} = \{\tilde{p}_{(\ell+1),1}, \ldots, \tilde{p}_{(\ell+1),k}\}$. The verifier and the prover run the interactive proof from Lemma 6.1 with inputs $C, \mathcal{P}_{\ell+1}, \tau = 1/1000$ and $\delta = 1/20r$, such that the verifier rejects if the interactive proof rejects.

2. **Granularise $\mathcal{P}_{\ell+1}$:** The verifier then runs $\mathcal{A}_{\mathsf{gran}}$ (Item 2 of Lemma 6.2) on input $\mathcal{P}_{\ell+1}$ to get the granularities $B_{\ell+1} = \{b_{(\ell+1),1}, \ldots, b_{(\ell+1),(k+1)}\}$ of an $8k$-grained distribution over $[k+1]$.

3. **Extended Polynomial Folding:** For each tuple $((z_1, \ldots, z_\ell), (a_1, \ldots, a_\ell), J_\ell, \vec{v}_\ell)$ in $\mathcal{S}_\ell$, in parallel, the prover and verifier run the extended polynomial folding protocol, Protocol 7, with explicit inputs $(B_\ell, \mathbb{F}, k, m-\ell, J_\ell, \vec{v}_\ell, \kappa = 32 \cdot \log(8k)\log(r))$. The implicit input instance is $X_\ell \in \mathbb{F}^{k^{m-\ell}}$, which is the ℓ-wise dot product of $z_1, \ldots, z_\ell \in \mathbb{F}^{8k}$ with X_0 (obtained from the tensor extensions given $B_0, \ldots, B_\ell$). In essence, the underlying PVAL input instance is $(X_\ell, J_\ell, \vec{v}_\ell)$.

   In return, Protocol 7 outputs a collection of tuples $\{(a, z^a_{\ell+1}, J_{\ell+1}, \vec{v}^a_{\ell+1})\}$, one for each $a \in [\log(8k/\kappa) + 1]$, where $z^a_{\ell+1} \in \mathbb{F}^{8k}$ is a uniformly random vector of weight $2^a \cdot \kappa$, and $(J_{\ell+1}, \vec{v}^a_{\ell+1})$ is the new claim corresponding to the folding obtained using $z^a_{\ell+1}$. Thus, for each a, the tuple $((z_1, \ldots, z^a_{\ell+1}), (a_1, \ldots, a_\ell, a), J_{\ell+1}, \vec{v}^a_{\ell+1})$ is added to $\mathcal{S}_{\ell+1}$.

Verify the folded instances: For each $((z_1, ..., z_r), (a_1, ..., a_r), J_r, \vec{v}_r) \in \mathcal{S}_r$, in parallel:

1. The prover sends X_r, the r-wise dot product of X with $z_1, \ldots, z_r$. The verifier receives $\widetilde{X}_r$ and checks if $P_{\widetilde{X}_r}|_{J_r} = \vec{v}_r$, else it rejects immediately.

2. The verifier sets $\varepsilon_r = \varepsilon\left(\prod_{s=1}^{r}(2^{a_s}/16)\right)$. It picks $(10/\varepsilon_r)$ uniformly random coordinates in $\widetilde{X}_r$, as well as along $\widehat{\mathcal{D}}_{r+1}$ (i.e., $\mathcal{D}^{r+1} \times \cdots \times \mathcal{D}^m$) using the sampler C.

   For each coordinate j that was sampled, verify that $\tilde{X}_r[j] = X_r[j]$ by using the sets $B_1, \ldots, B_r$ and the vectors $z_1, \ldots, z_r$ to compute the appropriate queries to the original input message X (this can be done because any extension only works with rows of X or the all 0's row). If any of these checks fail, then the verifier rejects immediately.

If the verifier did not reject so far, it accepts.

Let $W_i \in \mathbb{F}^{k^{m-1}}$ be the instance that minimises the distance along $\mu_{\widehat{\mathcal{D}}_2,\widehat{\mathcal{U}}_2}$ between $X[i,\cdot]$ and $\mathsf{PVAL}(J_1, \widetilde{Y}[i,\cdot])$, and let W be the instance in $\mathbb{F}^{k \times k^{m-1}}$ composed of these W_i's as rows. Observe that $W \in \mathsf{PVAL}(J, \vec{v})$ and thus, $\mu_{\mathcal{D},\mathcal{U}}(X, W) > \varepsilon$. Let $W' \in \mathbb{F}^{8k \times k^{m-1}}$ be the $\mathcal{E}_1$-extension of $g^{\mathsf{cat}}(W)$. Since X' and W' are both extended using the same distribution $\mathcal{E}_1$, for each $i \in [8k]$, the copies of the closest instance $W[i,\cdot]$ in W' correspond exactly to the copies of the row $X[i,\cdot]$ in X'. Thus, for each $i \in [8k]$, $W'[i,\cdot]$ is the closest instance along $\mu_{\widehat{\mathcal{D}}_2,\widehat{\mathcal{U}}_2}$ to $X'[i,\cdot]$.

We next study the distance between X' and W' along $\mu_{(\mathcal{U}_{8k}\times\widehat{\mathcal{D}}_2),(\mathcal{U}_{8k}\times\widehat{\mathcal{U}}_2)}$, by firstly computing the distance over the $\widehat{\mathcal{D}}_1$ distribution.

$$\begin{aligned}
\mu_{(\mathcal{U}_{8k}\times\widehat{\mathcal{D}}_2),(\mathcal{U}_{8k}\times\widehat{\mathcal{U}}_2)}(X', W') &\geq d_{\mathcal{U}_{8k}\times\widehat{\mathcal{D}}_2}(X', W') \\
&= \sum_{u=1}^{8k} \mathop{\mathbb{P}}_{(i,j)\sim\mathcal{U}_{8k}\times\widehat{\mathcal{D}}_2}\left[X'[i,j] \neq W'[i,j] \mid i = u\right] \cdot \mathop{\mathbb{P}}_{i\sim\mathcal{U}_{8k}}[i = u] \\
&= \sum_{u=1}^{k+1} \mathop{\mathbb{P}}_{(i,j)\sim\mathcal{E}_1\times\widehat{\mathcal{D}}_2}\left[g^{\mathsf{cat}}(X)[i,j] \neq g^{\mathsf{cat}}(W)[i,j] \mid i = u\right] \cdot \frac{b_u}{8k} \\
&= \sum_{u=1}^{k+1} \mathop{\mathbb{P}}_{(i,j)\sim\mathcal{E}_1\times\widehat{\mathcal{D}}_2}\left[g^{\mathsf{cat}}(X)[i,j] \neq g^{\mathsf{cat}}(W)[i,j] \mid i = u\right] \cdot \mathop{\mathbb{P}}_{i\sim\mathcal{E}_1}[i = u] \\
&= d_{\mathcal{E}_1\times\widehat{\mathcal{D}}_2}(g^{\mathsf{cat}}(X), g^{\mathsf{cat}}(W))
\end{aligned}$$

Here, the third expression holds from the definition of $\mathcal{D}_1$-extension, where the u^{th} row appears b_u many times in the extension with probability $1/8k$ each. Thus, (abusing the index u) we see that the probability of sampling any row $u \in [k+1]$ of $g^{\mathsf{cat}}(X) \in \mathbb{F}^{(k+1)\cdot k^{m-1}}$ is $b_u/8k$, which is the same as $\mathcal{E}_1$.

From Item 2 of Lemma 6.2, $\mathcal{A}_{\mathsf{gran}}$ guarantees us that

$$\begin{aligned}
d_{\mathcal{E}_1\times\widehat{\mathcal{D}}_2}(g^{\mathsf{cat}}(X), g^{\mathsf{cat}}(W)) &> \frac{1}{2} d_{\mathcal{P}_1\times\widehat{\mathcal{D}}_2}(X, W) \\
&= \frac{1}{2}\sum_{L=1}^{k} \mathop{\mathbb{P}}_{(i,j)\sim\mathcal{P}_1\times\widehat{\mathcal{D}}_2}[X[i,j] \neq W[i,j] \mid i = L] \cdot \mathcal{P}_1(L) \\
&\geq \frac{(1-\tau)}{2}\sum_{L=1}^{k} \mathop{\mathbb{P}}_{(i,j)\sim\mathcal{D}_1\times\widehat{\mathcal{D}}_2}[X[i,j] \neq W[i,j] \mid i = L] \cdot \mathcal{D}_1(L) \\
&= \frac{(1-\tau)}{2} d_{\widehat{\mathcal{D}}_1}(X, W)
\end{aligned}$$

The third expression comes from the guarantees provided by the parallel set lower bound protocol (with high probability) on $\mathcal{P}_1$.

We next look the same calculations for distances over $\mathcal{U}_{8k} \times \widehat{\mathcal{U}}_2$. It is worth noting from the proof of Lemma 6.2, that when the underlying distribution is $\mathcal{U}_k$, $\mathcal{A}_{\mathsf{gran}}$ outputs the distribution $\mathcal{E}_1$ over $[k+1]$ with granularities $b_1 = \cdots = b_k = 8$ and $b_{k+1} = 0$ (the distribution is still uniform over $[k]$). In such a case, we maintain the caveat since the concatenated row $0^{k^{m-1}}$ has probability 0 under $\mathcal{P}_1$, it can be eliminated from the extension altogether and thus, the $\mathcal{E}_1$-extended instance is still over $\mathbb{F}^{8k \times k^{m-1}}$. Thus, by defining X' and W' in a similar fashion,

$$\begin{aligned}
\mu_{(\mathcal{U}_{8k} \times \widehat{\mathcal{D}}_2),(\mathcal{U}_{8k} \times \widehat{\mathcal{U}}_2)}(X', W') &\geq d_{\mathcal{U}_{8k} \times \widehat{\mathcal{U}}_2}(X', W') \\
&= \sum_{u=1}^{8k} \underset{(i,j)\sim \mathcal{U}_{8k} \times \widehat{\mathcal{U}}_2}{\mathbb{P}} \left[X'[i,j] \neq W'[i,j] \mid i = u\right] \cdot \underset{i \sim \mathcal{U}_{8k}}{\mathbb{P}}[i = u] \\
&= \sum_{u=1}^{k} \underset{(i,j)\sim \mathcal{E}_1 \times \widehat{\mathcal{U}}_2}{\mathbb{P}} [X[i,j] \neq W[i,j] \mid i = u] \cdot \frac{b_u}{8k} \\
&= \sum_{u=1}^{k} \underset{j \sim \widehat{\mathcal{U}}_2}{\mathbb{P}} [X[u,j] \neq W[u,j]] \cdot \frac{b_u}{8k} \\
&\geq \sum_{u=1}^{k} \underset{j \sim \widehat{\mathcal{U}}_2}{\mathbb{P}} [X[u,j] \neq W[u,j]] \cdot \frac{2}{8k} \\
&= \frac{1}{4} d_{\widehat{\mathcal{U}}_1}(X, W)
\end{aligned}$$

Put together, we have that

$$\mu_{(\mathcal{U}_{8k} \times \widehat{\mathcal{D}}_2),(\mathcal{U}_{8k} \times \widehat{\mathcal{U}}_2)}(X', W') > (1-\tau)\mu_{\widehat{\mathcal{D}}_1, \widehat{\mathcal{U}}_1}(X, W) > \frac{(1-\tau)\gamma}{4} \tag{6}$$

On the other hand, we have the following upper bound

$$\begin{aligned}
&\mu_{(\mathcal{U}_{8k} \times \widehat{\mathcal{D}}_2),(\mathcal{U}_{8k} \times \widehat{\mathcal{U}}_2)}(X', W') \\
&= \max \left\{ \left(\frac{1}{8k} \sum_{u=1}^{8k} \underset{j \sim \widehat{\mathcal{D}}_2}{\mathbb{P}} \left[X'[u,j] \neq W'[u,j]\right] \right), \left(\frac{1}{8k} \sum_{u=1}^{8k} \underset{j \sim \widehat{\mathcal{U}}_2}{\mathbb{P}} \left[X'[u,j] \neq W[u,j]\right] \right) \right\} \\
&= \frac{1}{8k} \max \left\{ \sum_{u=1}^{8k} \underset{j \sim \widehat{\mathcal{D}}_2}{\mathbb{P}} \left[X'[u,j] \neq W'[u,j]\right], \sum_{u=1}^{8k} \underset{j \sim \widehat{\mathcal{U}}_2}{\mathbb{P}} \left[X'[u,j] \neq W'[u,j]\right] \right\} \\
&\leq \frac{1}{8k} \sum_{u=1}^{8k} \max \left\{ \underset{j \sim \widehat{\mathcal{D}}_2}{\mathbb{P}} \left[X'[u,j] \neq W'[u,j]\right], \underset{j \sim \widehat{\mathcal{U}}_2}{\mathbb{P}} \left[X'[u,j] \neq W'[u,j]\right] \right\} \\
&= \frac{1}{8k} \sum_{u=1}^{8k} \mu_{\widehat{\mathcal{D}}_2, \widehat{\mathcal{U}}_2}(X'_u, W'_u)
\end{aligned} \tag{7}$$

Here, the third expression follows from the fact that for any $c_1, \ldots, c_k > 0$ and $d_1, \ldots, d_k > 0$, we have that $\max\{\sum_i c_i, \sum_i d_i\} \leq \sum_i \max\{c_i, d_i\}$.

Recall that W'_i is the closest element in $\mathsf{PVAL}(J_1, U_i)$ to X'_i, for each $i \in [8k]$. Combining this with equations 6 and 7, we get the following expression, from which the lemma follows.

$$\frac{1}{8k} \sum_{i=1}^{8k} \mu_{\widehat{\mathcal{D}}_2, \widehat{\mathcal{U}}_2}(X'_i, \mathsf{PVAL}(J_1, U_i)) \geq \mu_{(\mathcal{U}_{8k} \times \widehat{\mathcal{D}}_2),(\mathcal{U}_{8k} \times \widehat{\mathcal{U}}_2)}(X', W') > \frac{(1-\tau)\gamma}{4}$$

□

Applying the distance preservation lemma from Lemma 6.4 in the proofs of Claims 5.7, 5.8, and 5.9 as before, we have the following claim about the extended polynomial folding protocol.

Claim 6.5. *For any $0 \le \ell \le r-1$ and $\gamma > 0$, suppose that $\mu_{\widehat{\mathcal{D}}_{\ell+1},\widehat{\mathcal{U}}_{\ell+1}}(X_\ell, \mathsf{PVAL}(J_\ell, \vec{v}_\ell)) > \gamma$. Let X'_ℓ be the $\mathcal{E}_{\ell+1}$-extension of $g^{\mathsf{cat}}(X_\ell)$ and let U_ℓ be the $\mathcal{E}_{\ell+1}$-extension of $\tilde{Y}_\ell \in \mathbb{F}^{k\times t}$, which is the proof sent in the $(\ell+1)^{th}$ round of Protocol 7.*

Then, there exists an $a^* \in [\log(8k/\kappa)+1]$, such that for a uniformly random $z_{a^*} \in \mathbb{F}^{8k}$ of Hamming weight $2^{a^*}\kappa$, with probability all but $(1/(|\mathbb{F}|-1)+e^{-\kappa/16\log(k)})$ probability over the verifier's choice of $\vec{z}_{a^*}$, it holds for $\vec{v}_{\ell+1} = z_{a^*}\cdot U_\ell$ that

$$\mu_{\widehat{\mathcal{D}}_{\ell+2},\widehat{\mathcal{U}}_{\ell+2}}(z_{a^*}\cdot X'_\ell, \mathsf{PVAL}(J_{\ell+1}, \vec{v}_{\ell+1})) \ge \gamma \cdot 2^{a^*}/16.$$

Therefore, given that $\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon$, then by the distance preservation lemma and taking a union bound over all rounds, we get that with probability all but $r\cdot(1/(|\mathbb{F}|-1)+e^{-\kappa/16\log(k)})$, there exists tuple $((z_1,\dots,z_r),(a_1,\cdots,a_r),J_r,\vec{v}_r) \in \mathcal{S}_r$ and corresponding folded instance X_r, such that

$$\mu_{\widehat{\mathcal{D}}_{r+1},\widehat{\mathcal{U}}_{r+1}}(X_r, \mathsf{PVAL}(J_r, \vec{v}_r)) > \varepsilon \cdot \prod_{s=1}^{r} k\cdot 2^{a_s}/16$$

Thus, the overall probability of accepting X (by a union bound over the learning step, the extended polynomial folding step, and the folded instance verification step) for $r > 10$ is at most

$$r\delta + r(1/(|\mathbb{F}|-1)+e^{-\kappa/16\log(k)}) + (1-\varepsilon_r)^{\frac{10}{\varepsilon_r}} < \frac{1}{20} + \frac{r}{10r-1} + \frac{r}{r^2} + \frac{1}{10} \le \frac{1}{3}$$

Query Complexity: The input is queried only in the final verification stage. For each of the $\frac{10}{\varepsilon_r}$ indices sampled on each X_r (over $\mathcal{U}$ and $\mathcal{D}$), the verifier makes $\prod_{s=1}^{r} 2^{a_s}\kappa$ queries to X by the *bounded locality* of the extended polynomial folding protocol. There are at most $O(\log^r(k)) = 1/\varepsilon^{o(1)}$ many such instances, for every tuple in $\mathcal{S}_r$. By a similar calculation as Theorem 5.10, the query complexity is given by

$$\frac{1}{\varepsilon^{o(1)}}\cdot\frac{10}{\varepsilon_r}\cdot\prod_{s=1}^{r} 2^{a_s}\kappa = \frac{1}{\varepsilon^{o(1)}}\cdot\frac{10}{\varepsilon}\cdot\prod_{s=1}^{r}\frac{16}{2^{a_s}}\cdot\prod_{s=1}^{r} 2^{a_s}\kappa = \frac{1}{\varepsilon^{1+o(1)}}$$

This follows from the fact that $k^r = \frac{1}{\varepsilon}$, $O(\log(k))^r = \frac{1}{\varepsilon^{o(1)}}$ and $\kappa^r = O(\log(k)\log(r))^r = \frac{1}{\varepsilon^{o(1)}}$.

Communication Complexity: The communication complexity from r iterations of the learning step is $O(rk)$ for sending the r marginals $\mathcal{P}_1,\dots,\mathcal{P}_r$ and $O(\mathsf{polylog}(n)\cdot k^2)$, for our setting of δ and τ^2 in Lemma 6.1, and observing that the input length of C is at most $\mathsf{polylog}(n)$ and the number of rounds r is at most $\log(n)$.

Next, from a similar analysis as Theorem 5.10, we see that r iterations of the extended polynomial folding have $r\cdot O(\log(k))^r\cdot kt\cdot\log|\mathbb{F}| = \frac{1}{\varepsilon^{o(1)}}\cdot k\cdot n\cdot\varepsilon\cdot\mathsf{polylog}(n)$ communication complexity, from sending all the matrices Y. This calculation follows from the fact that r, $O(\log(k))^r$ and

$k = \frac{1}{\varepsilon^{o(1)}}$ along with that $t = n\varepsilon\log(n)$ and $|\mathbb{F}| = \mathsf{polylog}(k)$. Further, communicating the folded instances X_r in the final step uses $\frac{n}{k^r\varepsilon^{o(1)}}$ many bits.

Therefore the total communication complexity is

$$\mathsf{polylog}(n)\cdot k^2 + \frac{1}{\varepsilon^{o(1)}}\cdot n\cdot\varepsilon\mathsf{polylog}(n) + \frac{n}{k^r\cdot\varepsilon^{o(1)}} = \mathsf{polylog}(n)\cdot\left(k^2 + \frac{k}{\varepsilon^{o(1)}}\cdot n\cdot\varepsilon\right).$$

Number of Messages: The number of messages (and the round complexity) is $O(r)$, since the learning phase is done in constantly many rounds from Lemma 6.1 and the polynomial folding protocols are performed in parallel.

Honest Prover Running Time: The running time of the honest prover in the ℓ^{th} round is $O\left(2^{\mathsf{polylog}(n)}\right)$ to compute the probability distribution vector of $\mathcal{D}_{\ell+1}$ by going over all possible inputs to C and $O\left(2^{\mathsf{polylog}(n)}\right)$ in the IP from Lemma 6.1. As seen earlier, a $\mathsf{poly}(n)$-time honest prover is sufficient for the other stages in the protocol. In total, the honest prover runs in $O\left(2^{\mathsf{polylog}(n)}\right)$ time.

Verifier running time: In total, the verifier runs in $O\left(\frac{r\cdot\mathsf{poly}(|C|)k^2}{\delta\tau^2}\right) = k^2\mathsf{polylog}(n)$ for learning, and $O(rk) = O(k\log(n))$ for granularisation. The rest follows from a similar analysis as Theorem 5.10, and put together, the verifier running time is $n^{o(1)}(k\varepsilon n + k^2 + 1/\varepsilon)$. □

We end this section with the white-box IPP for low-depth circuits over polynomially-samplable m-product distribution families. This IPP builds on the framework from Section 5.

Theorem 6.6. *For any $m, n \in \mathbb{N}$, let $\mathcal{F}$ be a set of polynomially samplable m-product distributions over $[k]^m$, such that $n = k^m$. Moreover, for every $n \in \mathbb{N}$, let $L \subseteq \{0,1\}^n$ be a language computable by circuits of depth $\Delta(n)$ and size $S = S(n)$.*

Then, for every large enough input $n \in \mathbb{N}$ and every $\varepsilon > 0$, there exists a white-box IPP for L over $\mathcal{F}$ with proximity parameter ε, and completeness and soundness probabilities $2/3$.

The query complexity of the white-box IPP is $1/\varepsilon^{1+o(1)}$, the communication complexity is $\mathsf{polylog}(n)\cdot\left(k^2 + k\cdot\varepsilon^{1-o(1)}\cdot n\right) + \varepsilon\cdot n\cdot\mathsf{poly}(\Delta)$, and the verifier running time is $n^{o(1)}(k\cdot\varepsilon\cdot n + k^2 + 1/\varepsilon) + \varepsilon\cdot n\cdot\mathsf{poly}(\Delta)$. Moreover, the number of messages is $O(\log(1/\varepsilon)/\log(k)) + \Delta\cdot\log(S))$ and the honest prover running time is $2^{\mathsf{polylog}(n)} + \mathsf{poly}(S)$.

The theorem follows by combining the interactive reduction from Theorem 4.3 (which doesn't access the implicit input), with the white-box IPP over $\mathcal{F}$ for PVAL in Protocol 8.

Acknowledgements

We are grateful to Oded Goldreich for his insightful comments, some of which led to rephrasing Theorem 1.1 to indicate the trade-off between the query and communication complexities with better clarity. We are also thankful to Marcel Dall'Agnol for many helpful discussions.

Tom Gur and Ninad Rajgopal are supported by the Tom Gur's UKRI Future Leaders Fellowship MR/S031545/1. Tom Gur is also supported in part by EPSRC New Horizons Grant EP/X018180/1 and EPSRC RoaRQ Grant EP/W032635/1. Ron Rothblum is funded by the European Union (ERC, FASTPROOF, 101041208). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach.* Cambridge University Press, 2009.

[ABF+23] Vipul Arora, Arnab Bhattacharyya, Noah Fleming, Esty Kelman, and Yuichi Yoshida. Low degree testing over the reals. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 738–792. SIAM, 2023.

[AFK13] Pranjal Awasthi, Vitaly Feldman, and Varun Kanade. Learning using local membership queries. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, volume 30 of *JMLR Workshop and Conference Proceedings*, pages 398–431. JMLR.org, 2013.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon interactive oracle proofs of proximity. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[Bel19] Aleksandrs Belovs. Quantum algorithm for distribution-free junta testing. In René van Bevern and Gregory Kucherov, editors, *Computer Science - Theory and Applications - 14th International Computer Science Symposium in Russia, CSR 2019, Novosibirsk, Russia, July 1-5, 2019, Proceedings*, volume 11532 of *Lecture Notes in Computer Science*, pages 50–59. Springer, 2019.

[BFH21] Eric Blais, Renato Pinto Jr Ferreira, and Nathaniel Harms. VC dimension and distribution-free sample-based testing. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 504–517, 2021.

[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.

[BLNR22] Sarah Bordage, Mathieu Lhotel, Jade Nardi, and Hugues Randriam. Interactive oracle proofs of proximity to algebraic geometry codes. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 30:1–30:45. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[BRV18] Itay Berman, Ron D. Rothblum, and Vinod Vaikuntanathan. Zero-knowledge proofs of proximity. 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 19:1–19:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[BSGH$^+$04] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of proximity, shorter PCPs and applications to coding. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 1–10, 2004.

[Bsh19] Nader H. Bshouty. Almost optimal distribution-free junta testing. In Amir Shpilka, editor, *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 2:1–2:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[BT06] Andrej Bogdanov and Luca Trevisan. Average-case complexity. *Found. Trends Theor. Comput. Sci.*, 2(1), 2006.

[BY22] Arnab Bhattacharyya and Yuichi Yoshida. *Property Testing - Problems and Techniques.* Springer, 2022.

[CEG95] Ran Canetti, Guy Even, and Oded Goldreich. Lower bounds for sampling algorithms for estimating the average. *Inf. Process. Lett.*, 53(1):17–25, 1995.

[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.

[CP22] Xi Chen and Shyamal Patel. Distribution-free testing for halfspaces (almost) requires pac learning. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1715–1743. SIAM, 2022.

[CX16] Xi Chen and Jinyu Xie. Tight bounds for the distribution-free testing of monotone conjunctions. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 54–71. SIAM, 2016.

[DGMT22] Marcel Dall'Agnol, Tom Gur, Subhayan Roy Moulik, and Justin Thaler. Quantum proofs of proximity. *Quantum*, 6:834, 2022.

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

[FGL14] Eldar Fischer, Yonatan Goldhirsh, and Oded Lachish. Partial tests, universal tests and decomposability. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 483–500, 2014.

[FY20] Noah Fleming and Yuichi Yoshida. Distribution-free testing of linear functions on $\mathbb{R}^n$. 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 22:1–22:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[GG21] Oded Goldreich and Tom Gur. Universal locally verifiable codes and 3-round interactive proofs of proximity for CSP. *Theoretical computer science*, 878:83–101, 2021.

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

[GKR15] Shafi Goldwasser, Yael Tauman Kalai, and Guy N Rothblum. Delegating computation: interactive proofs for muggles. *Journal of the ACM (JACM)*, 62(4):1–64, 2015.

[Gol17] Oded Goldreich. *Introduction to property testing.* Cambridge University Press, 2017.

[Gol19a] Oded Goldreich. Testing bipartitness in an augmented VDF bounded-degree graph model. *CoRR*, abs/1905.03070, 2019.

[Gol19b] Oded Goldreich. Testing graphs in vertex-distribution-free models. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 527–534, 2019.

[Gol20] Oded Goldreich. The uniform distribution is complete with respect to testing identity to a fixed distribution. In Oded Goldreich, editor, *Computational Complexity and Property Testing - On the Interplay Between Randomness and Computation*, volume 12050 of *Lecture Notes in Computer Science*, pages 152–172. Springer, 2020.

[GR17] Tom Gur and Ron D. Rothblum. A hierarchy theorem for interactive proofs of proximity. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 39:1–39:43. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

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

[GR22] Guy Goldberg and Guy N Rothblum. Sample-based proofs of proximity. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022).* Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[GRR20] Tom Gur, Govind Ramnarayan, and Ron Rothblum. Relaxed locally correctable codes. *Theory of Computing*, 16(1):1–68, 2020.

[GRSY21] Shafi Goldwasser, Guy N Rothblum, Jonathan Shafer, and Amir Yehudayoff. Interactive proofs for verifying machine learning. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021).* Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[GS89] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. *Adv. Comput. Res.*, 5:73–90, 1989.

[GS07] Dana Glasner and Rocco A. Servedio. Distribution-free testing lower bounds for basic boolean functions. In Moses Charikar, Klaus Jansen, Omer Reingold, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 10th International Workshop, APPROX 2007, and 11th International Workshop, RANDOM 2007, Princeton, NJ, USA, August 20-22, 2007, Proceedings*, volume 4627 of *Lecture Notes in Computer Science*, pages 494–508. Springer, 2007.

[Gur17] Tom Gur. *On locally verifiable proofs of proximity.* PhD thesis, The Weizmann Institute of Science (Israel), 2017.

[GV11] Oded Goldreich and Salil P Vadhan. On the complexity of computational problems regarding distributions. *Studies in Complexity and Cryptography*, 6650:390–405, 2011.

[HK07] Shirley Halevy and Eyal Kushilevitz. Distribution-free property-testing. *SIAM J. Comput.*, 37(4):1107–1138, 2007.

[HK08] Shirley Halevy and Eyal Kushilevitz. Distribution-free connectivity testing for sparse graphs. *Algorithmica*, 51:24–48, 2008.

[HR22] Tal Herman and Guy N Rothblum. Verifying the unseen: interactive proofs for label-invariant distribution properties. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1208–1219, 2022.

[KR15] Yael Tauman Kalai and Ron D Rothblum. Arguments of proximity. In *Advances in Cryptology–CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 422–442. Springer, 2015.

[LCS$^+$18] Zhengyang Liu, Xi Chen, Rocco A Servedio, Ying Sheng, and Jinyu Xie. Distribution-free junta testing. *ACM Transactions on Algorithms (TALG)*, 15(1):1–23, 2018.

[LCS$^+$19] Zhengyang Liu, Xi Chen, Rocco A. Servedio, Ying Sheng, and Jinyu Xie. Distribution-free junta testing. *ACM Trans. Algorithms*, 15(1):1:1–1:23, 2019.

[RR20] Guy N Rothblum and Ron D Rothblum. Batch verification and proofs of proximity with polylog overhead. In *Theory of Cryptography: 18th International Conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings, Part II*, pages 108–138. Springer, 2020.

[RR22] Dana Ron and Asaf Rosin. Optimal distribution-free sample-based testing of subsequence-freeness with one-sided error. *ACM Transactions on Computation Theory (TOCT)*, 14(1):1–31, 2022.

[RRR18] Omer Reingold, Guy N Rothblum, and Ron D Rothblum. Efficient batch verification for UP. In *33rd Computational Complexity Conference (CCC 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[RRR21] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. *SIAM J. Comput.*, 50(3), 2021.

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

[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.

[RZR20] Noga Ron-Zewi and Ron D Rothblum. Local proofs approaching the witness length. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 846–857. IEEE, 2020.

[SV97] Amit Sahai and Salil P. Vadhan. Manipulating statistical difference. In Panos M. Pardalos, Sanguthevar Rajasekaran, and José Rolim, editors, *Randomization Methods in Algorithm Design, Proceedings of a DIMACS Workshop, Princeton, New Jersey, USA, December 12-14, 1997*, volume 43 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 251–270. DIMACS/AMS, 1997.

[Vad99] Salil Pravin Vadhan. *A study of statistical zero-knowledge proofs.* PhD thesis, Massachusetts Institute of Technology, 1999.

[Vad06] Salil P Vadhan. An unconditional study of computational zero knowledge. *SIAM Journal on Computing*, 36(4):1160–1214, 2006.

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

A Proofs of Claims 5.7, 5.8, and 5.9

Claim 5.7. *If the verifier does not reject in Step 1, then there exists an integer $b \in \{0, \cdots, \log(k)\}$, and a subset $I \subseteq [k]$, s.t. $\forall i \in I, \varepsilon_i \geq k\varepsilon/(2^{b+1}\rho)$ and $|I| \geq 2^b/4\log(k)$.*

Proof. We have by Lemma 5.4 that $\sum_{i\in[k]} \varepsilon_i > k\varepsilon/\rho$. We suppose by contradiction that for all $b \in \{0, \cdots, \log(k)\}$, the number of rows i s.t. $\varepsilon_i \in [k\varepsilon/(2^{b+1}\rho), k\varepsilon/(2^b\rho))$ is less than $2^b/4\log(k)$, it follows that:

$$\begin{aligned}\sum_{i=1}^{k} \varepsilon_i &< \left(\sum_{b=0}^{\log(k)-1} (k\varepsilon/(2^b\rho)) \cdot (2^b/4\log(k))\right) + (\varepsilon/2\rho)k \\ &= (\log(k)+1)(k\varepsilon/4\rho\log(k)) + (k\varepsilon/2\rho) \\ &< k\varepsilon/\rho.\end{aligned}$$

Here the left summand on the first line are the contributions from where $\varepsilon_i > \varepsilon/2\rho$, the right are the rest for which there can be at most k. □

Claim 5.8. *In Step 2 of protocol 4, for $a \in [log(k/\kappa)+1]$, let I_a be the set of non-zero coordinates in $\vec{z}_a$ (this set is of size $2^a \cdot \kappa$). Take b as guaranteed by Claim 5.7 and $a^* = min(log(k/\kappa)$, $\log(k) - b)$. With all but $e^{-\kappa/4\log(k)}$ probability over the verifier's choice of I_{a^*}, there exists $i^* \in I_{a^*}$ s.t. $\varepsilon_{i^*} \geq \varepsilon \cdot 2^{a^*}/2\rho$.*

Proof. We know by Claim 5.7 that there is some $b \in \{0, \cdots, \log(k)\}$, and a set I of at least $2^b/4\log(k)$ rows each of which has $\varepsilon_i > k\varepsilon/(2^{b+1}\rho)$.

When we pick $min(k, \kappa k/2^b)$ random rows to include in I_{a^*}, with all but $\left(1 - \frac{|I|}{k}\right)^{\kappa k/2^b} \leq e^{-\kappa/4\log(k)}$ probability, there will be non zero intersection between I and I_{a^*} (via a "birthday Paradox" argument). The cardinality of I_{a^*} is equal to $\kappa k/2^b$, but if this is greater than k, then setting it to k suffices. The latter holds true, because the total number of rows is k, and in particular we end up picking every row in I. Now, we set $a^* = min(\log(k/\kappa), \log(k) - b)$ (as $|I_{a^*}| = 2^{a^*}\kappa$), to ensure that the size of our set I_{a^*} is large enough but has size no greater than k. □

Claim 5.9. *Take a^* as guaranteed by Claim 5.8. With all but $((|\mathbb{F}|-1)^{-1}+e^{-\kappa/4\log(k)})$ probability over the verifier's choice of $\vec{z}_{a^*}$, it holds for $\vec{v}_{a^*} = z_{a^*} \cdot Y'$ that*

$$\mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}(X_{a^*}, \mathsf{PVAL}(J_2, \vec{v}_{a^*})) \geq \varepsilon \cdot 2^{a^*}/4\rho.$$

Proof. Let T be the linear subspace of messages in $\mathbb{F}^{k^p}$, whose encodings are 0 on J_2(the set of column coordinates of the elements of J and therefore the coordinates of the row of Y). Also let A_i be the set of vectors that when you add them to the i^{th} row of X, they evaluate to the corresponding row of y. Observe that for any $\vec{s} \in A_i$: $A_i = \vec{s} + T$. Take any such vector $\vec{s}_i$ for each row i.

By the Claim 5.8, with all but $e^{-\kappa/4\log(k)}$ probability over the choice of non-zero coordinates of I_{a^*}, there is some $i^* \in I_{a^*}$ s.t. $\varepsilon_{i^*} \geq 2^{a^*}\varepsilon/2\rho$.

In this case, we give the non-zero values of $\vec{z}_{a^*}$(the values in I_{a^*}) uniformly random elements of $\mathbb{F}$. We now have a value for $X_{a^*} = \vec{z}_{a^*} \cdot X$ and a corresponding $\vec{v}_{a^*} = \vec{z}_{a^*} \cdot Y'$, we want to lower bound the $\mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}$ distance from this X_{a^*} and any satisying X'.

Let A be the set of shift vectors that when added to X_{a^*} are consistent with $\vec{v}_{a^*}$. $A = \vec{s} + T$ as before for any shift vector $\vec{s} \in A$, the minimal $\mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}$ weight of A (what we are now trying to minimise) is the same as the $\mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}$ distance from $\vec{s}$ to T.

$\vec{s}_{a^*} = \sum_{i\in[k]} \vec{z}_{a^*}[i]\vec{s}_i$ is a uniformly random vector in the linear span of $\{\vec{s}_i\}_{i\in I_{a^*}}$. There is some $i^* \in I_{a^*}$ s.t. $\varepsilon_{i^*} > \varepsilon 2^{a^*}/2\rho$. Therefore by lemma 5.5, with a union bound we have that with probability at least all but $((|F|-1)^{-1}+e^{-\kappa/4\log(k)})$, we have for some a^*, $\mu_{\mathcal{D}^{(p)},\mathcal{U}_{k^p}}(X_{a^*}, \mathsf{PVAL}(J_2, \vec{v}_{a^*})) \geq \varepsilon \cdot 2^{a^*}/4\rho$. □

B Proof of Theorem 5.2

Proof of Theorem 5.2. The IPP for any such L over ρ-dispersed distributions is specified in Protocol 6. This protocol has perfect completeness and soundness error 1/4, to achieve the required soundness we simply repeat $O(1)$ times. Let $X \in \{0,1\}^n$ be the input to L. The properties and the complexity of the protocol are proved below.

Completeness: Both the protocols used in this IPP have perfect completeness and therefore $X \in L$ implies that the verifier accepts with probability 1.

In the protocol from Theorem 4.3:

$$X \in L \implies \underset{V_{\mathsf{NC}}}{\mathbb{P}}[X \in \mathsf{PVAL}(J, \vec{v})] = 1.$$

In protocol 5:

$$X \in \mathsf{PVAL}(J, \vec{v}) \implies \underset{V_0,\mathcal{O}_{\mathcal{D}}(X)}{\mathbb{P}}\left[(P_0(X,\mathcal{D}), V_0^{X,\mathcal{O}_{\mathcal{D}}(X)})(n,\varepsilon) = 1\right] = 1.$$

Therefore, in the overall IPP, if $X \in L$, V will accept with probability 1.

Soundness: Assume that $d_{\mathcal{D}}(X, L) > \varepsilon$. For each repetition of this protocol, the probability that the verifier outputs $(J, \vec{v})$ such that $\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon$ is at least $\frac{1}{2}$ by the soundness condition of Theorem 4.3. Similarly, by Theorem 5.10, there is also probability at least $\frac{1}{2}$ that given $\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon$, V_0 rejects.

In the protocol from Theorem 4.3:

$$d_{\mathcal{D}}(X, L) > \varepsilon \implies \underset{V_{\mathsf{NC}}}{\mathbb{P}}[\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon] \geq \frac{1}{2}.$$

In protocol 5:

$$\mu_{\mathcal{D},\mathcal{U}}(X, \mathsf{PVAL}(J, \vec{v})) > \varepsilon \implies \underset{V_0, \mathcal{O}_{\mathcal{D}}(X)}{\mathbb{P}}\left[(P_0^*(X, \mathcal{D}), V_0^{X, \mathcal{O}_{\mathcal{D}}(X)})(n, \varepsilon) = 0\right] \geq \frac{1}{2}.$$

Therefore V rejects with probability $\frac{1}{4}$ each repetition of this protocol. In total this means that the verifier rejects with probability at least $\left(1 - \frac{3}{4}\right) = \frac{1}{4}$. We can achieve our soundness condition by standard soundness amplification.

The complexities of this protocol are achieved by summing the complexities of the component protocols.

Communication Complexity: The communication complexity from step 1 is $\varepsilon n \cdot \mathsf{poly}(\Delta_L)$, and is $(n\varepsilon + 4\varepsilon n \log^2(n))\frac{1}{\varepsilon^{o(1)}}$ from step 2. In total, $c(n) = (\varepsilon \cdot n + 4\varepsilon n \log^2(n))\frac{1}{\varepsilon}^{o(1)} + \varepsilon \cdot n \cdot \mathsf{poly}(\Delta_L)$.

Query Complexity: Note that the reduction to PVAL has no access to the input or the distribution so do not contribute to either the query or sample complexity. Therefore, we only need to consider step 2 for which the query complexity is $\frac{\rho^{\log(1/\varepsilon)/\log\log(n)}}{\varepsilon^{1+o(1)}}$.

Sample Complexity: Similarly, we only need consider step 2 for which the sample complexity is $\frac{\rho^{\log(1/\varepsilon)/\log\log(n)}}{\varepsilon^{1+o(1)}}$.

Prover Running Time: The prover running time from step 1 is $\mathsf{poly}(S)$, and is $\mathsf{poly}(n)$ from step 2. In total, the running time is $\mathsf{poly}(n, S)$.

Verifier Running Time: The verifier running time from step 1 is $\varepsilon \cdot n \cdot \mathsf{poly}(\Delta_L)$, and is $\left(\frac{\rho^{\log(1/\varepsilon)/\log\log(n)}}{\varepsilon} + n\varepsilon + 4n\varepsilon\log^2(n)\right) n^{o(1)}$ from step 2. In total, the verifier runs in $n^{o(1)}\left(\frac{\rho^{\log(1/\varepsilon)/\log\log(n)}}{\varepsilon} + \varepsilon \cdot n \cdot \mathsf{poly}(\Delta_L)\right)$ time.

Round Complexity: The round complexity from step 1 is $O(\Delta_L \cdot \log(S))$, and is $\frac{\log(1/\varepsilon)}{\log\log(n)} + 1$ from step 2. In total, the round complexity is $O\left(\frac{\log(1/\varepsilon)}{\log\log(n)} + \Delta_L \cdot \log(S)\right)$. □

C Proof of the Granularisation Lemma

Proof of Lemma 6.2. We prove the first item of the lemma here and this extends analogously to the second item as well.

We show that the set $\{a_1, \cdots, a_{n+1}\}$ returned by $\mathcal{A}_{gran}$ form the granularities of a distribution $\mathcal{D}'$ over $[n+1]$. For this, we first show that for any $\forall i \in [n+1] : \mathcal{D}'(i) = \frac{a_i}{8n} \in [0, 1]$ and then that $\sum_{i\in[n+1]} \mathcal{D}'(i) = 1$.

$$\forall i \in [n] : a_i = \lfloor 6n \cdot p(i) \rfloor + 2 \in [2, 6n+2] \subseteq [0, 8n].$$

Therefore $\forall i \in [n] : \mathcal{D}'(i) = \frac{a_i}{8n} \in [0, 1]$.

Algorithm 9 $\mathcal{A}_{gran}$: Algorithm for granularising an input distribution.

The input is the distribution $\mathcal{D}$ over $[n]$ presented as the list of values $\{p_1, \cdots, p_n\}$.

```
1. Set $t = 0$.
2. For each $i \in [n]$
    (a) return $a_i = \lfloor 6n \cdot p(i) \rfloor + 2$.
    (b) assign $t = t + a_i$.
3. return $a_{n+1} = 8n - t$.
```

Let $q = \sum_{i \in [n]} \frac{\lfloor 6np_i \rfloor + 2}{8n}$. Observe that $q \in (0, 1]$, since

$$0 < \sum_{i \in [n]} \frac{\lfloor 6np_i \rfloor + 2}{8n} \leq \sum_{i \in [n]} \frac{6np_i + 2}{8n} \leq \frac{1}{4} + \sum_{i \in [n]} \frac{3}{4} p_i \leq 1$$

where the we use the fact that $\{p_1, \ldots, p_n\}$ form a probability distribution. Moreover, observe that $\mathcal{D}'(n+1) = 1 - q \in [0, 1)$. This implies that every element of the distribution $\mathcal{D}'$ is within the range $[0, 1]$ and therefore $\mathcal{D}'$ is a distribution as

$$\sum_{i \in [n+1]} \mathcal{D}'(i) = \sum_{i \in [n+1]} \frac{a_i}{8n} = \sum_{i \in [n]} \frac{a_i}{8n} + a_{n+1} = \left(\sum_{i \in [n]} \frac{a_i}{8n} \right) + \frac{8n - \sum_{i \in [n]} a_i}{8n} = 1.$$

The linear running time of $\mathcal{A}_{\mathsf{gran}}$ follows from inspection of Algorithm 9.

Next, we see that if $X \in L$, by the definition of L_0, $X' = g^{\mathsf{cat}}(X) \in L_0$. On the other hand, to show that this transformation of $\mathcal{D}$ maintains the distance for any input NO instance, we prove the following claim.

Claim C.1. $\forall i \in [n]$, $\frac{a_i}{8n} \geq \frac{p_i}{2}$

Proof. Suppose $p(i) \leq \frac{1}{3n}$, then

$$\frac{a_i}{8n} \geq \frac{2}{8n} \geq \frac{1}{3n} \cdot \frac{1}{2} \geq \frac{p_i}{2}$$

Suppose instead that $p_i \geq \frac{1}{3n}$, let $p_i = \frac{r}{6n} + s$ for $r \in \mathbb{N}, s \in [0, \frac{1}{6n})$:

$$\begin{aligned} \frac{a_i}{8n} &= \frac{r+2}{8n} \geq \frac{r}{12n} + \frac{s}{2} \\ &\geq \frac{p_i}{2} \end{aligned}$$

□

Let $Y' \in L_0$, this implies that there is some $Y \in L$ such that $Y' = g^{cat}(Y)$.

$$\begin{aligned} d_{\mathcal{D}'}(X', Y') &= \sum_{i\in[n+1]} |X'_i - Y'_i|\mathcal{D}'(i) \\ &= \sum_{i\in[n]} |X'_i - Y'_i|\mathcal{D}'(i) \\ &\geq \sum_{i\in[n]} |X_i - Y_i|\frac{\mathcal{D}(i)}{2} \\ &\geq \frac{1}{2}d_{\mathcal{D}}(X, Y) > \varepsilon/2. \end{aligned}$$

This follows for every value of $Y' \in L_0$ including the minimiser, therefore $d_{\mathcal{D}'}(X', L_0) > \varepsilon/2$. □

D IPPs for Efficiently Learnable Distribution Families

In this section, we demonstrate an IPP for NC over logspace-uniform low-depth circuit classes, given that the distribution we test against is efficiently learnable.

For any pair of distributions $\mathcal{D}_1, \mathcal{D}_2 \in \Delta(\Omega_n)$, define the total variation distance (d_{TV}) as follows

$$d_{TV}(\mathcal{D}_1, \mathcal{D}_2) = \sum_{i\in[n]} \left|\mathbb{P}_{j\sim\mathcal{D}_1}[j = i] - \mathbb{P}_{j\sim\mathcal{D}_2}[j = i]\right|.$$

Definition D.1 (Efficiently Learnable Class of Distributions). *Let $\mathcal{C} \subseteq \{\Delta(\Omega_n)\}_{n\in\mathbb{N}}$ be a class of distributions. We say that $\mathcal{C}$ is learnable by an interactive proof, if there exists (P, V) where the verifier V is given sample access to an unknown (but fixed) $\mathcal{D} = \{\mathcal{D}_n\} \in \mathcal{C}$ (i.e., V gets samples from $[n]$ according to $\mathcal{D}_n$) and the honest prover has full knowledge of $\mathcal{D}$, along with common inputs n, ε, such that the proof system satisfies the following properties for every large enough n.*

- *Completeness: For every $\mathcal{D} \in \mathcal{C}$, the verifier outputs $\tilde{\mathcal{D}} = \left(P(\mathcal{D}), V^{\mathcal{D}}\right)(n, \varepsilon)$ as a distribution vector of probabilities or the value 'reject' ($\perp$), such that*

$$\mathbb{P}_{V,\mathcal{D}}\left[(\tilde{\mathcal{D}} \neq \perp) \wedge (d_{TV}(\mathcal{D}, \hat{\mathcal{D}}) < \varepsilon)\right] = 1.$$

- *Soundness: For every $\mathcal{D} \in \mathcal{C}$ and for any computationally unbounded prover P^*, the output $\tilde{\mathcal{D}} = \left(P(\mathcal{D}), V^{\mathcal{D}}\right)(n, \varepsilon)$ is either a distribution vector of probabilities or the value 'reject' ($\perp$), such that*

$$\mathbb{P}_{V,\mathcal{D}}\left[(\tilde{\mathcal{D}} \neq \perp) \wedge (d_{TV}(\mathcal{D}, \hat{\mathcal{D}}) \geq \varepsilon)\right] \leq 0.1.$$

The sample complexity $s(n, \varepsilon)$, proof complexity $p(n, \varepsilon)$, verifier running time $t(n, \varepsilon)$ are as defined earlier. Furthermore, we specify the honest prover running time here to be $\mathsf{poly}(n)$.

We next state the following lemma on the closeness of an input with respect to a distribution $\mathcal{D}'$ that is close to the underlying distribution $\mathcal{D}$, in total variation distance.

Lemma D.1. *For any strings $X, Y' \in \{0, 1\}^n$ and distributions $\mathcal{D}, \mathcal{D}'$ over $[n]$,*

$$d_{\mathcal{D}}(X, Y') \leq d_{TV}(\mathcal{D}, \mathcal{D}') + d_{\mathcal{D}'}(X, Y').$$

Proof. The proof consists of the following sequence of calculations.

$$\begin{aligned}
d_{\mathcal{D}}(X, Y') &= \mathop{\mathbb{P}}_{i \sim \mathcal{D}} \left[X_i \neq Y'_i\right] \\
&= \sum_{i \in [n] \wedge X_i \neq Y'_i} \mathop{\mathbb{P}}_{j \sim \mathcal{D}} [j = i] \\
&= \sum_{i \in [n] \wedge X_i \neq Y'_i} \mathop{\mathbb{P}}_{j \sim \mathcal{D}'} [j = i] + \left(\mathop{\mathbb{P}}_{j \sim \mathcal{D}} [j = i] - \mathop{\mathbb{P}}_{j \sim \mathcal{D}'} [j = i] \right) \\
&\leq \sum_{i \in [n] \wedge X_i \neq Y'_i} \mathop{\mathbb{P}}_{j \sim \mathcal{D}'} [j = i] + \left| \mathop{\mathbb{P}}_{j \sim \mathcal{D}} [j = i] - \mathop{\mathbb{P}}_{j \sim \mathcal{D}'} [j = i] \right| \\
&\leq \left(\sum_{i \in [n] \wedge X_i \neq Y'_i} \mathop{\mathbb{P}}_{j \sim \mathcal{D}'} [j = i] \right) + d_{TV}(\mathcal{D}, \mathcal{D}') \\
&= d_{\mathcal{D}'}(X, Y') + d_{TV}(\mathcal{D}, \mathcal{D}')
\end{aligned}$$

□

Corollary D.2. *For any string $X \in \{0,1\}^n$, language $L \subseteq \{0,1\}^n$ and distributions $\mathcal{D}, \mathcal{D}'$ over $[n]$,*

$$d_{\mathcal{D}}(X, L) \leq d_{TV}(\mathcal{D}, \mathcal{D}') + d_{\mathcal{D}'}(X, L).$$

Proof. There exists $Y' \in L$ such that $d_{\mathcal{D}'}(X, L) = d_{\mathcal{D}'}(X, Y')$. Therefore, by Lemma D.1

$$d_{\mathcal{D}}(X, L) \leq d_{\mathcal{D}}(X, Y') \leq d_{TV}(\mathcal{D}, \mathcal{D}') + d_{\mathcal{D}'}(X, Y') = d_{TV}(\mathcal{D}, \mathcal{D}') + d_{\mathcal{D}'}(X, L).$$

□

The following lemma demonstrates a reduction from proving IPPs over learnable distributions to uniform IPPs given that we know the distribution.

Lemma D.3. *Let L be any language computable by logspace-uniform circuits of size $S(n)$ and depth $D(n)$, and let $\varepsilon > 0$. Let $\mathcal{D}$ be any distribution over $[n]$ such that $\mathcal{D}(i) = p_i$ for every $i \in [n]$, where each $p_i \in [0,1]$.*

Then, there exists an algorithm $\mathcal{B}_{\mathsf{gran}}$ that given explicit inputs $\{p_1, \ldots, p_n\}$, as well as oracle access to a string $X \in \{0,1\}^n$, outputs a vector $\vec{Q} \in \{0,1\}^{8n \log(n)}$, such that for a (parameterised) language L'_Q computable by logspace-uniform circuits of size $S(n) + \tilde{O}(n)$ and depth $D(n) + O(\log(n))$, there exists $X' \in \{0,1\}^{8n}$ for which the following holds.

- *If $X \in L$, then $X' \in L'_Q$.*
- *If $d_{\mathcal{D}}(X, L) > \varepsilon$, then $d_{U_{8n}}(X', L'_Q) > \varepsilon/2$.*

This algorithm runs in time $\tilde{O}(n)$. Additionally, any query to X' can be implemented using a single query to X and $O(1)$ running time, given explicit access to $\vec{Q}$.

In other words, Lemma D.3 says that given explicit access to a distribution $\mathcal{D}$, $\mathcal{B}_{\mathsf{gran}}$ provides an "implicit" reduction between L and a parameterised language L'_Q computable by log-space uniform circuits of similar size and depth. By this we mean that $\mathcal{B}_{\mathsf{gran}}$ reduces a testing problem for L over $\mathcal{D}$ to another testing problem for L' over the uniform distribution, by simulating oracle access to the input $X' \in \{0,1\}^{8n}$ to L' using the oracle to the original input $X \in \{0,1\}^n$.

Proof Sketch. $\mathcal{B}_{\mathsf{gran}}$ first runs $\mathcal{A}_{\mathsf{gran}}$ from Lemma 6.2 on input $\mathcal{D}$ to obtain a set A of granularities of a distribution $\mathcal{D}'$ over $[n+1]$ for which the following hold.

$$X \in L \implies g^{\mathsf{cat}}(X) \in L_0$$

and

$$d_{\mathcal{D}}(X, L) > \varepsilon \implies d_{\mathcal{D}'}(g^{\mathsf{cat}}(X), L_0) > \varepsilon/2.$$

Given granularities $A = \{a_1, \cdots, a_{n+1}\}$, in $\tilde{O}(n)$ running time we can obtain a vector $\vec{Q}$ defined as follows.

$$\vec{Q}_i = \begin{cases} i, & i \in [n] \\ 1, & i \in [n+1, n+a_1-1] \\ 2, & i \in [n+a_1, n+a_1+a_2-2] \\ \vdots & \\ n+1 & i \in [n+1+\sum_{j=1}^{n-1} a_j - 1, 8n] \end{cases}$$

Let $X' \in \{0,1\}^{8n}$ be the extension of $g^{cat}(X)$ using A. From this, each query i to X' becomes the query $\vec{Q}_i$ to X as $X_{\vec{Q}_i} = X'_i$ by the definition of extensions. Therefore in total this algorithm runs in time $\tilde{O}(n)$ and a query to the oracle for X' makes a single query to X and has $O(1)$ running time.

Define a parameterised language L'_Q as the set of strings that are the extensions of L_0 using A (i.e., $\mathcal{D}'$-extensions of L_0). Formally,

$$L'_Q = \left\{W \in \{0,1\}^{8n} \mid \exists Y \in L_0 \cap \{0,1\}^{n+1} \text{ such that } W \text{ is the extension of } Y \text{ using } A\right\}$$

Let X' be the $\mathcal{D}'$-extension of $g^{\mathsf{cat}}(X)$. From this, Item 1 follows as

$$X \in L \implies g^{\mathsf{cat}}(X) \in L_0 \implies X' \in L'_Q$$

On the other hand, for any $Y, \widetilde{Y} \in \{0,1\}^n$, with Y' and $\widetilde{Y}'$ being the $\mathcal{D}'$-extensions of $g^{\mathsf{cat}}(Y)$ and $g^{\mathsf{cat}}(\widetilde{Y})$ respectively, we have

$$\begin{aligned} d_{\mathcal{D}'}(g^{\mathsf{cat}}(Y), g^{\mathsf{cat}}(\widetilde{Y})) &= \sum_{i=0}^{n+1} \frac{a_i}{8n} |g^{\mathsf{cat}}(Y)_i - g^{\mathsf{cat}}(\widetilde{Y})_i| \\ &= \sum_{i=0}^{8n} \frac{1}{8n} |Y'_i - \widetilde{Y}'_i| \\ &= d_{\mathcal{U}_{8n}}(Y', \widetilde{Y}') \end{aligned}$$

Therefore,

$$d_{\mathcal{D}}(X, L) > \varepsilon \implies d_{\mathcal{D}'}(g^{\mathsf{cat}}(X), L_0) = d_{\mathcal{U}_{8n}}(X', L'_Q) > \varepsilon/2.$$

The last thing we need to prove is that L'_Q is computable by log-space uniform circuits of size $S(n)+8n$ and depth $D(n)+\log(8n)$. Thus, a log-space Turing machine can first generate a circuit for L on the first n indices and then generate an $O(\log(n))$-depth circuit of AND gates at the top to check $\bigwedge_{j\in[n+1,8n]} X_{\vec{Q}_j} = X_j$. This is performed using additional circuitry of size $O(n\log(n))$ (n AND statements each requiring $\log(n)$ bitwise comparisons) and depth $O(\log(n))$. □

With this lemma, we can extend our framework from Theorem 5.2 to construct IPPs for languages computable by low-depth circuits over the class of efficiently learnable distributions (using interactive proofs) in the sense of Definition D.1.

Theorem D.4. *Let $\mathcal{C} \subseteq \{\Delta(\Omega_n)\}_{n\in\mathbb{N}}$ be a class of distributions that is learnable using an interactive proof that has sample complexity $s(n,\varepsilon)$, proof complexity $p(n,\varepsilon)$, verifier running time $T_V(n,\varepsilon)$ and honest prover running time $\mathsf{poly}(n)$. Moreover, for every $n\in\mathbb{N}$, let $L\subseteq\{0,1\}^n$ be a language computable by circuits of depth $\Delta(n)$ and size $S=S(n)$.*

Then, for every large enough input length $n\in\mathbb{N}$ and every $\varepsilon>0$, there exists an IPP for L over $\mathcal{C}$ with proximity parameter ε, and perfect completeness and soundness error $2/3$.

This IPP has query complexity $O(1/\varepsilon)$, sample complexity $s(n,\varepsilon/4)$ and communication complexity $p(n,\varepsilon/4)+\tilde{O}(\varepsilon n)+\varepsilon\cdot n\cdot\mathsf{poly}(\Delta,\log(n)))$. In addition, the honest prover runs in $\mathsf{poly}(S,n)$ time and the verifier runs in $t(n,\varepsilon/2)+O(1/\varepsilon)+\varepsilon\cdot n\cdot\mathsf{poly}(\Delta,\log(n))+\tilde{O}(n)$.

Proof Sketch. This protocol proceeds by first applying the learning algorithm for $\mathcal{C}$ with proximity parameter $\varepsilon/2$ to learn a distribution $\tilde{\mathcal{D}}$ which is $\varepsilon/2$-close to $\mathcal{D}$ (it rejects if the learner outputs $\perp$). Next, it runs the granularisation algorithm $\mathcal{B}_{\mathsf{gran}}$ from Theorem D.3 to reduce testing X along the distribution $\tilde{\mathcal{D}}$ for membership of L to testing X' along the uniform distribution for membership of L'_Q, for which we can simulate oracle access to X' using X and the output of $\mathcal{B}_{\mathsf{gran}}$, $\vec{Q}$. Finally, it runs the IPP from Theorem 4.6 for testing X''s membership in L'_Q with proximity parameter $\varepsilon/4$.

Perfect completeness follows as in the learning stage the verifier learns a valid distribution $\tilde{\mathcal{D}}$ from the honest prover, the completeness of $\mathcal{B}_{\mathsf{gran}}$ and of Theorem 2.1 ensures that if $X\in L$ then $X'\in L'_Q$ and therefore that the IPP from Theorem 2.1 accepts with probability 1.

For soundness, we show that the transformation to verifying L'_Q preserves distance.

$$\begin{aligned} d_{\mathcal{D}}(X,L)>\varepsilon &\implies d_{\tilde{\mathcal{D}}}(X,L)+d_{TV}(\mathcal{D},\tilde{\mathcal{D}})>\varepsilon \\ &\implies d_{\tilde{\mathcal{D}}}(X,L)+\frac{\varepsilon}{2}>\varepsilon \\ &\implies d_{\tilde{\mathcal{D}}}(X,L)>\frac{\varepsilon}{2} \\ &\implies d_{\mathcal{U}_{8n}}(X',L'_Q)>\frac{\varepsilon}{4} \end{aligned}$$

The first expression comes from Corollary D.2, the second comes from the guarantees on $\tilde{\mathcal{D}}$ of the distribution learner, and the fourth from the distance preservation from Lemma D.3. This means that with probability at least 0.9, the verifier either rejects (because the learner outputs $\perp$) or $\tilde{\mathcal{D}}$ is a good approximation of $\mathcal{D}$, and with probability 2/3, the IPP will reject. Put together, the verifier rejects the input X with probability at least 1/2.

The query complexity follows from the fact that each query to X' is simulated by 1 query to X and from the query complexity from Theorem 4.6. The verifier running time follows as these queries also cost an additional $\tilde{O}(n)$ to generate (via $\mathcal{B}_{\mathsf{gran}}$) and the additional running time from the new oracle can only introduce at most a multiplicative factor of $O(1)$. The other complexities follow from construction. □

From Theorem D.4, we get the following IPP over learnable distributions for NC languages that matches the parameters of the uniform IPP on every ε.

Corollary D.5. *Let $\mathcal{F}$ be a class of distributions that can be learnt by an interactive protocol $\mathcal{A}$ that takes sample access to some $\mathcal{D}$ in $\mathcal{F}$, using $O(1/\varepsilon)$ samples and $\tilde{O}(\varepsilon n)$ communication complexity, where the verifier runs in time $T_V(n)$ and the honest prover runs in* poly(n) *time.*

Then, there exists an IPP *for* NC *over $\mathcal{F}$, with sample complexity $O(1/\varepsilon)$, query complexity $O(1/\varepsilon)$ and communication complexity $\tilde{O}(\varepsilon n)$. Moreover, the honest prover runs in time* poly(n)*.*