Electronic Colloquium on Computational Complexity, Report No. 209 (2026)

Interactive Proofs with Noisy Data

Noga Amit
MIT

Shafi Goldwasser
MIT

Guy N. Rothblum
Apple

Abstract

We study interactive proofs when the prover and the verifier access the same underlying data through independent noisy views. The noise is *persistent* for each party: each location is corrupted once, and repeated queries to the same location return the same corrupted value. We introduce two models in this setting: *noisy interactive proofs of proximity* (noisy IPPs), and *noisy PAC verification*, a noisy analogue of the PAC verification framework of Goldwasser et al. [ITCS 2021]. These are the first interactive-proof models in which the prover and verifier access the same data through separate noisy views, possibly with different noise rates, rather than sharing a common view of the input.

For noisy IPPs, we give a complete characterization of the four natural regimes determined by whether the honest prover is clean or noisy and whether the exact noise rates are known or only upper bounds are known. We show that the clean-prover, known-rate setting admits noisy IPPs for every language in NC, whereas, under a standard cryptographic assumption, in each of the other three regimes there is a language in NC^1 for which noisy IPPs are impossible. The positive result is obtained through a connection to *robust* IPPs: even subconstant robustness suffices to tolerate constant random noise. We also show that constant robustness is impossible in general, establishing a separation between random noise and worst-case local corruptions.

Beyond these general results, we construct noisy IPPs for natural languages, and we give a noisy PAC-verification protocol for the heavy Fourier coefficients of a Boolean function. These positive results are efficient for both the verifier and the prover, and hold in the general setting where the parties know only an upper bound on the noise rate and may experience different noise rates.

ISSN 1433-8092

Contents

1 Introduction

An interactive proof system is an interactive protocol between a verifier and an untrusted prover, in which the prover tries to convince the verifier of the validity of a computational statement about some underlying data. Verification is required to be substantially more efficient than solving the task from scratch. In the classical setting, the data is a string and the statement is that it belongs to a language; in the learning setting, the data is a function and the statement is that a given hypothesis approximates it well. In these standard models, both parties refer to the same underlying data. In many applications, however, the data may be available only through noisy measurements or imperfect storage, or, in the learning setting, may be corrupted by human annotation or by discretization of continuous signals. This raises a natural question: what is the right notion of interactive proof when the prover and verifier do not observe the same underlying data exactly, but instead access *different noisy views* of it?

We study a setting in which the communication channel between the parties is clean, while the data is corrupted by random faults. These corruptions are *independent* across the two parties and across different pieces of the data. A central feature of our model is that the noise is *persistent*: each location is corrupted once and for all for each party, and repeated queries to that location return the same corrupted value. Without persistence, repeated queries would recover the true value by majority vote, essentially collapsing the model to the standard noiseless setting. This persistent model is natural when each party accesses its own copy of the data, and each copy is produced only once by measurement, storage, or preprocessing. For example, the prover and verifier may obtain separate measurements of the same experiment, taken at different times or subject to independent measurement noise, or they may access separate annotated copies of a dataset.

We assume that the actual noise rate is unknown, and that the verifier and the honest prover are given only an upper bound on it, supplied for example by the data provider or by prior guarantees on the measurement process. Both the upper bound and the actual noise rate are independent of the input size. It is also natural to allow the two parties to have different noise rates, reflecting measurements or storage of different quality and cost, since higher precision is more expensive. The most natural regime is that the verifier, being the weaker party, has the noisier view: the prover may be able to use a higher-quality but more expensive measurement process, while the verifier is restricted to a cheaper and noisier one.

This leads to a new type of interactive-proof model. To the best of our knowledge, this is the first work on interactive proofs in which the two parties may access different versions of the same underlying data. This poses a substantial challenge, since many protocols rely on the prover and verifier sharing a common view of the input. Standard interactive proofs are too strong for dealing with such noise: a single corrupted bit may turn a no-instance into a yes-instance, but the verifier is required to reject every no-instance. Thus, the most basic model for interactive proofs with noise is that of interactive proofs of proximity (IPPs), in which the verifier is only required to reject inputs that are *far* from the language.

In this paper, we first define *noisy* IPP*s*. We then ask: what can one achieve in this model? It turns out that the model is hard in general: we prove lower bounds, showing hardness even for a relatively small complexity class. On the positive side, we give a general positive result for noisy IPPs in a restricted setting, and construct noisy IPPs for two natural problems. Finally, we turn to the learning analogue. We define *noisy PAC verification* and give a positive result for verifying the heavy Fourier coefficients of a Boolean function.

1.1 The Proposed Models

We introduce two new models of interactive proofs in the presence of noise.

1.1.1 Noisy Interactive Proofs of Proximity

Interactive proofs of proximity (IPPs) [EKR04, RVW13] are the sublinear analogue of standard interactive proofs. Instead of reading the entire input, the verifier is given oracle access to it and is restricted to making only a sublinear number of queries. One also requires the communication complexity to be small, so that the prover cannot simply transmit the entire input to the verifier. Thus, IPPs are strictly more restrictive than standard interactive proofs. They are well motivated in modern large-scale settings, where the input may be too large for the verifier to read in full. In this framework, soundness is relaxed: rather than rejecting *every* input outside the language, the verifier is only required to reject with high probability every input that is *far* from the language.

A long line of works studies IPPs and their variants, e.g., [GR17, GGR18, RR20, GLR21, GRS23]. The closest in spirit to our setting is the recent work of Amir et al. [AGR25], where the honest prover is also restricted to be sublinear. For related work on noisy property testing, the non-interactive analogue of IPPs, see Section 1.4.

Definition. We consider a noisy version of the IPP model in which each party accesses the input through its own persistent noisy oracle: each input bit is flipped independently with probability η, and the resulting corruption is then fixed for the entire protocol. Hence, repeated queries to the same location by the same party always return the same answer. Equivalently, for each party there is a fixed hidden set of input coordinates whose values are flipped. In this model, the actual noise rate η is unknown to the verifier $\mathcal{V}$ and the honest prover $\mathcal{P}$, which are given only an upper bound $\eta_b < 1/2$. Note that the communication channel between the parties is assumed to be clean.

Definition 1.1 (Noisy IPP, informal version of Definition 2.6)**.** *A* noisy IPP *for a language $\mathcal{L}$ is an interactive protocol $\langle \mathcal{P}, \mathcal{V} \rangle$ that satisfies the following conditions for every $\varepsilon > 0$, every upper bound $\eta_b < 1/2$ and every actual noise rate $\eta \leq \eta_b$:*

- ***Input Access.*** *Let $\xi^{\mathcal{V}}, \xi^{\mathcal{P}} \in \{0,1\}^n$ be two independent noise masks such that $\xi_i^{\mathcal{V}}, \xi_i^{\mathcal{P}} \leftarrow$ Bernoulli(η) independently across indices $i \in [n]$.*

  $\mathcal{V}$ only knows η_b and has query access to $x \oplus \xi^{\mathcal{V}}$.

  In completeness, the honest prover $\mathcal{P}$ only knows η_b and has access to $x \oplus \xi^{\mathcal{P}}$. In soundness, the unbounded cheating prover $\widetilde{\mathcal{P}}$ also knows η and has clean access to x.

- ***Completeness.*** *If $x \in \mathcal{L}$, then after interacting with $\mathcal{P}$, $\mathcal{V}$ accepts with probability $\geq 2/3$.*

- ***Soundness.*** *If x is ε-far from $\mathcal{L}$, then after interacting with any unbounded $\widetilde{\mathcal{P}}$, $\mathcal{V}$ accepts with probability $\leq 1/3$.*

All probabilities are over $\mathcal{V}$'s randomness and both noise masks, and distances are Hamming.

Complexity measures are query complexity, communication complexity, and running times, all depending on n, ε and η_b. As in the standard IPP model, we do not account for $\mathcal{P}$'s query complexity and allow it full access to $x \oplus \xi^{\mathcal{P}}$. One may also consider the stronger doubly-sublinear regime of [AGR25]; we return to this variant below. We also consider the simpler special case in which the exact noise rate η is known to the parties (this distinction will play an important role in our results).

Discussion on the model. Our definition and results extend directly to the more general setting in which the verifier and the prover may have *different* noise rates.[1] In particular, the variant in which only the verifier sees a noisy input can be recovered by setting the prover's noise rate to 0, and is already non-trivial. In either case, the parties do not necessarily share the same view of the input, and this is one of the main difficulties of the model.

The noise is assumed to be *persistent*, namely, fixed once and for all throughout the protocol. As discussed earlier, if instead fresh noise were drawn independently for each query, then repeated queries to the same index would recover the underlying bit by majority vote, essentially collapsing the model to the standard noiseless setting. In contrast, persistence prevents one from applying standard protocol-repetition in a fully black-box manner: since the noise mask is fixed throughout the execution, repeating the protocol may reveal information about the verifier's noisy view to a dishonest prover, thereby enabling cheating strategies that do not arise in the standard model. Consequently, the standard amplification argument does not automatically go through.

In soundness, a lower noise rate of the prover only strengthens it. Thus, we fully strengthen the cheating prover by giving it a clean view of the input together with the actual noise rate η, and we formalize soundness accordingly. By contrast, noisy access only weakens an honest prover, even if it is computationally unbounded. Therefore, the prover's noise plays a role only in completeness.

The verifier's noise, on the other hand, affects both completeness and soundness: on yes-instances, it may cause the verifier's view to disagree with that of the honest prover, lowering the acceptance probability; on no-instances, it may mask violations, lowering the rejection probability. In particular, perfect completeness is impossible in the noisy setting: under persistent noise, any execution that can occur on a no-instance can also occur with positive probability on a yes-instance, and so perfect completeness would force the verifier to accept no-instances as well.

Finally, we emphasize that the verifier does not know the exact noise rate η, but only an upper bound η_b on the noise rate. We view this as a minimal assumption in our setting: since the prover is untrusted, the verifier cannot hope to obtain a reliable bound on the true noise rate through interaction. Note that, while our models and results are formulated for arbitrary ε and noise rates η (or upper bounds η_b), we find the constant-noise regime the most natural, since in many settings η is an external parameter of the measurement process or storage device and therefore should not depend on the input length.

1.1.2 Noisy PAC Verification

PAC learning studies the task of learning a hypothesis from labeled samples drawn from an unknown distribution. In the original model, the learner is given correctly labeled samples. In many applications, however, labels are obtained through imperfect processes, for example via sensors, human annotation, or discretization of continuous signals. The resulting corruptions are often persistent: once a sample is mislabeled by the data-collection process, repeated access to that same sample reproduces the same corrupted label. See, for example, [FV13] for a survey of common sources of label noise, and [GSS+25] for a broader discussion of labeling and measurement uncertainty in supervised learning. This motivates the persistent-noise formulation. The classical work of Angluin and Laird [AL88] introduced PAC learning with noise and already emphasized the upper-bound model, in which the learner is given only an upper bound on the unknown noise

[1] More broadly, asymmetric information and uncertainty between the two sides is natural in communication settings as well; see, for example, Guruswami and Smith [GS16].

rate. Learning under noisy labels has since been studied extensively in many variants; we refer the reader to Section 1.4 for a more complete discussion.

In our verification setting, Goldwasser et al. [GRSY21] introduced PAC verification, focusing on the task of verifying that a hypothesis provided by the prover is competitive with the best one in the class. Following [GRSY21], subsequent works [GJK+24, MS23, BDFPJ26] further developed PAC verification, giving protocols and lower bounds for natural concept classes, including Fourier-sparse functions and juntas. Beyond the classical setting, variants of PAC verification have also been studied in quantum models and related frameworks [CHI+24, MSD24, CEH+24].

Our noisy PAC verification model also addresses a direction suggested in [GRSY21], namely asymmetric-access models in which the prover's labels are "better" than the verifier's, capturing settings where $\mathcal{V}$ is an external auditor with limited or noisy measurements while $\mathcal{P}$ is the service provider with richer access to the underlying data. For exposition, we first focus on the symmetric regime in which both parties are corrupted at the same (unknown) noise rate, and then extend to the more general settings.

Definition. We begin by defining the two types of noisy access used in our model. Let $f : \{0,1\}^n \to \{-1,1\}$ be a target function[2] and let $\mathcal{D}$ be a distribution over $\{0,1\}^n$. Fix a noise rate $\eta \in [0, 1/2)$. We consider two independent *persistent* noise functions $b^{\mathcal{P}}, b^{\mathcal{V}} : \{0,1\}^n \to \{-1,1\}$, one for the prover $\mathcal{P}$ and one for the verifier $\mathcal{V}$, where for every $x \in \{0,1\}^n$ the bits $b^{\mathcal{P}}(x), b^{\mathcal{V}}(x)$ are independent and distributed so that $\Pr[b^{\mathcal{P}}(x) = -1] = \Pr[b^{\mathcal{V}}(x) = -1] = \eta$, and the collections $\{b^{\mathcal{P}}(x)\}_x$ and $\{b^{\mathcal{V}}(x)\}_x$ are mutually independent.

- *Noisy membership query access:* access to the membership query oracle $f_\eta : \{0,1\}^n \to \{-1,1\}$ defined by
$$f_\eta(x) := f(x) \cdot b^{\mathcal{P}}(x).$$
- *Noisy sample access:* access to noisy examples $(x, y^{\mathcal{V}})$ where $x \leftarrow \mathcal{D}$ and
$$y^{\mathcal{V}} := f(x) \cdot b^{\mathcal{V}}(x).$$

In both cases the noise is persistent: for each fixed input x, the same noise bit $b^{\mathcal{P}}(x)$ (resp., $b^{\mathcal{V}}(x)$) is used every time x is queried (resp., sampled).

In noisy PAC verification we consider an honest prover $\mathcal{P}$ that has noisy membership query access (we do not charge it for query complexity, but we do measure its running time), and a verifier $\mathcal{V}$ that only receives noisy samples (we measure both its sample complexity and running time). Both $\mathcal{P}$ and $\mathcal{V}$ are given the upper bound η_b on the noise rate, but the actual noise rate $\eta \le \eta_b$ is unknown to them. Importantly, in soundness, we allow an arbitrary (possibly unbounded) cheating prover $\widetilde{\mathcal{P}}$ who knows η and has clean access to f.

Recall that in a PAC verification protocol, $\mathcal{V}$ outputs a hypothesis h whose error is competitive with the best hypothesis in a prescribed class $\mathcal{H}$ (or else outputs reject). In the following definition, we use the 0–1 loss under $\mathcal{D}$ and write
$$\Delta_{\mathcal{D}}(f, h) := \Pr_{x \leftarrow \mathcal{D}}[f(x) \neq h(x)].$$
We also write $\Delta_{\mathcal{D}}(f, \mathcal{H}) := \inf_{h \in \mathcal{H}} \Delta_{\mathcal{D}}(f, h)$ for the optimal loss of $\mathcal{H}$ with respect to f under $\mathcal{D}$.

[2] We could alternatively work with range $\{0,1\}$; the distinction between the two conventions is immaterial for the analysis and complexities.

Definition 1.2 (Noisy PAC verification, informal version of Definition 5.21)**.** *A hypothesis class $\mathcal{H}$ is* Noisy PAC Verifiable *over a distribution $\mathcal{D}$ if there exists an interactive protocol $\langle \mathcal{P}, \mathcal{V} \rangle$ that satisfies the following conditions for every target function f, every $\varepsilon, \delta > 0$, every upper bound $\eta_b < 1/2$ and every actual noise rate $\eta \leq \eta_b$:*

- ***Input Access.*** *The verifier $\mathcal{V}$ only knows η_b and has noisy sample access as defined above. In completeness, the honest prover $\mathcal{P}$ only knows η_b and has noisy membership query access as defined above. In soundness, the unbounded cheating prover $\widetilde{\mathcal{P}}$ also knows η and has clean access to f.*
- ***Completeness.*** *After interacting with $\mathcal{P}$, $\mathcal{V}$ outputs h such that with probability at least $1 - \delta$, $h \neq \mathsf{reject}$ and $\Delta_{\mathcal{D}}(f, h) \leq \Delta_{\mathcal{D}}(f, \mathcal{H}) + \varepsilon$.*
- ***Soundness.*** *After interacting with any unbounded prover $\widetilde{\mathcal{P}}$, $\mathcal{V}$ outputs h such that with probability at least $1 - \delta$, $h = \mathsf{reject}$ or $\Delta_{\mathcal{D}}(f, h) \leq \Delta_{\mathcal{D}}(f, \mathcal{H}) + \varepsilon$.*

The probability is over the randomness of the samples, the internal randomness of both parties, and the randomness of the noise.

Discussion on the model. First, note that both the noiseless and noisy models are only interesting in the agnostic regime. In the realizable case, where $\mathcal{H}$ contains a perfect predictor for f, there is a trivial verification procedure whenever the noise upper bound is sufficiently smaller than the target error (say $\eta_b \leq \varepsilon/2$): the verifier estimates from noisy samples the disagreement rate of a proposed hypothesis h and accepts if it is below a threshold between η_b and ε. This uses only $O(1/\varepsilon^2)$ samples.

The following aspects of the model are analogous to those in noisy IPPs. A central modeling choice is that the noise is *persistent* for each party: each label $f(x)$ is corrupted by a fixed bit $b^{\mathcal{P}}(x)$ (respectively, $b^{\mathcal{V}}(x)$) that persists across repeated queries (respectively, repeated samples of the same point).[3] We also assume that the verifier and the honest prover are given only an upper bound η_b on the noise rate. In soundness, we consider a maximally powerful cheating prover that knows the true noise rate η and has clean access to f, but does not know the verifier's realized noise bits $b^{\mathcal{V}}(\cdot)$. Finally, the definition and our results extend directly to the more general setting in which the prover and verifier may have different noise rates.

1.2 Results

1.2.1 Impossibility Results for Noisy IPPs

As discussed above, IPPs are most interesting when both query and communication complexities are sublinear, and noisy IPPs are most interesting when the noise rate is constant. A useful benchmark for the limitations of (ordinary) IPPs is the lower bound of Kalai and Rothblum [KR15], who showed, assuming the existence of exponentially hard pseudorandom generators computable in

[3] Note that, however, for the verifier's sample access, persistence is largely immaterial in our main application, where $\mathcal{D} = \mathsf{Unif}(\{0,1\}^n)$: since the verifier uses only polynomially many samples, repeated sample points occur with negligible probability, and conditioned on no repetitions persistent noise is identical to fresh classification noise. We nevertheless use the persistent formulation for both parties; persistence is substantive on the prover side even when $\mathcal{D} = \mathsf{Unif}(\{0,1\}^n)$, since membership queries may revisit the same point.

NC^1, that there exists a language in NC^1 for which every IPP satisfies a query–communication tradeoff $q \cdot cc = \Omega(n)$.

In our noisy setting of Definition 1.1, there are two natural axes along which the model may become easier: whether the honest prover has a clean or noisy view of the input, and whether the verifier knows its exact noise rate or only an upper bound. The easiest nontrivial setting is that of a clean prover and a known verifier noise rate; indeed, as we show in the next section, this setting admits general positive results for NC. Our negative results show that relaxing either of these two advantages already makes sublinear communication impossible, even for a language in NC^1, and even if the verifier is allowed to read its entire noisy input.

Theorem 1.3 (Impossibility of noisy IPPs, informal version of Theorems 3.3 and 3.4)**.** *Under the cryptographic assumption of [KR15], for every pair of constants $0 < \eta \leq \eta_b < 1/2$, there exists a language $\mathcal{L} \in \mathsf{NC}^1$ for which both of the following hold.*

1. Noisy prover. *If both the honest prover and the verifier have noisy access to the input, then every noisy IPP for $\mathcal{L}$ has communication complexity $\Omega(n)$, even when the exact noise rate η is known.*

2. Unknown verifier noise rate. *If the verifier knows only that its noise rate is at most η_b, then every noisy IPP for $\mathcal{L}$ has communication complexity $\Omega(n)$, even if the honest prover has clean access to the input.*

Both statements already hold for a constant proximity parameter, regardless of the verifier's query complexity, the number of rounds, or the running time of the honest prover.

In the second setting, the lower bound is in fact stronger: it continues to hold even if the honest prover is told the verifier's exact noise rate. Moreover, both impossibility results extend to the general upper-bound model of Definition 1.1 and to the setting in which the prover and verifier experience different noise rates.

The two lower bounds isolate two distinct and complementary sources of hardness: uncertainty about the underlying data, and uncertainty about the noise channel. In the noisy-prover setting, even though the noise rate is known exactly, neither honest party has a clean reference point to the input, and even their joint noisy views may leave substantial uncertainty about the input and therefore about its membership in the language. In the unknown-verifier-noise setting, this source of difficulty disappears (since the honest prover knows the clean input), but the verifier faces a different challenge: it cannot tell whether discrepancies in its view originate in the input itself or were introduced by its noisy channel. Since the prover is untrusted, it cannot simply resolve this ambiguity by sending the true noise rate to the verifier. In both cases, the difficulty persists even if the verifier reads its entire noisy input, showing that the obstruction is not caused by sublinear access to the data.

The two impossibility results above concern *random* corruption of the parties' views of the input. It is natural to compare them with the related *worst-case* notion of robust IPPs, which was formalized in [HR24] and appears implicitly in earlier works such as [RR20]. Roughly speaking, a ρ-robust IPP is an IPP that satisfies a stronger soundness guarantee: once the interaction is fixed, the verifier's sublinear view of the input must remain ρ-far from any view that would make it accept. This notion of robustness is analogous to notions studied in the PCP and PCPP literature [BSGH+04], and has also been considered in the context of IOPs [RRR16, RZR25]. We prove a related, but separate, limitation for robust IPPs as well.

Theorem 1.4 (Impossibility of robust IPPs, informal version of Corollary 3.7)**.** *Under the cryptographic assumption of [KR15], for every sufficiently small constant robustness parameter, there is no uniform construction of robust* IPP*s for* NC^1 *that simultaneously achieves query complexity* $O(n^{0.99})$ *and sublinear communication complexity, even at a constant proximity parameter and regardless of the number of rounds or the running time of the prover.*

The formal statement additionally assumes robust-soundness error $O(1/\log n)$ and that the verifier's final decision predicate admits a succinct NC^1 description; these are mild requirements satisfied by the constructions motivating our comparison. The robustness parameter measures the fraction of the verifier's local view that may be corrupted adversarially, and hence we view constant robustness as the natural worst-case analogue of the constant-noise regime that we study. Importantly, however, our lower bound only rules out *constant* robustness.[4] As we show in the next section, subconstant robustness is already sufficient to obtain noisy IPPs tolerating a *constant* random noise rate, provided the honest prover has a clean view of the input and the verifier knows the exact noise rate. Thus, although constant robustness is impossible in general, substantially weaker robustness can still suffice for handling constant random noise, suggesting a separation between worst-case and random corruptions, in line with the familiar distinction between adversarial and random errors in coding theory.

1.2.2 Positive Results for Noisy IPPs

Having established that relaxing either the clean-prover assumption or exact knowledge of the verifier's noise rate leads to impossibility, we turn to the remaining, simplest nontrivial setting: the honest prover has a clean view of the input and the verifier knows its exact noise rate. In this setting, a trivial solution with linear communication is available: the prover can send the clean input, and the verifier can check both that it is a valid instance and that it is sufficiently consistent with its own noisy view. The question is whether this can be done with sublinear communication.

We show that robust IPPs provide exactly the structure needed to achieve this. Importantly, even subconstant robustness can suffice for tolerating a constant random noise rate; this is essential in light of Theorem 1.4, which rules out constant robustness. The reduction is straightforward: the parties execute the robust IPP without the verifier querying its input, the prover supplies the clean local view on the verifier's query set, and the verifier checks both that this view is accepting and that its disagreement with the verifier's noisy view is consistent with the known noise rate. Instantiating this transformation with the robust IPP of [RR20] yields the following.

Theorem 1.5 (Noisy IPPs for NC with a clean prover and known noise rate, informal version of Corollary 4.2)**.** *For every constant proximity parameter* $\varepsilon > 0$ *and constant known verifier noise rate* $\eta < 1/2$*, every language in* NC *has a noisy* IPP *with query and communication complexities* $\widetilde{O}(n^{2/3})$ *and an efficient prover.*

Together with the impossibility results of the previous subsection, this completes the picture for constant noise rates:

	Exact noise rate known	Only an upper bound known
Clean prover	Positive for NC (Theorem 1.5)	Impossible for NC^1 (Theorem 1.3)
Noisy prover	Impossible for NC^1 (Theorem 1.3)	Impossible for NC^1 (Theorem 1.3)

[4] Note that our impossibility result is consistent with the robust IPP of [RR20]: in their construction, the parameter regime yielding sublinear communication requires a subconstant proximity parameter, and since robustness cannot in general exceed proximity, its robustness is then necessarily subconstant as well.

Thus, with a clean prover, knowing the verifier's exact noise rate changes the picture completely: every language in NC admits a sublinear noisy IPP, whereas with only an upper bound this is impossible already for a language in NC^1. If the honest prover is noisy, impossibility holds already when the exact noise rates are known.

The general impossibility results above do not rule out efficient noisy IPPs for specific natural problems, even in the full noisy model. We therefore next turn to two such examples: Hamming Weight and Permutation. Both are obtained by adapting IPPs with perfect completeness, yet the resulting noisy protocols necessarily lose this property.[5] Both results hold in the fully noisy model, where the prover and verifier only know upper-bounds on their noise rates. For simplicity, we state them under a common upper bound η_b.

Hamming Weight. We study the Hamming Weight search problem (HW), where the goal is to output an additive εn-approximation to the Hamming weight of a string of length n. While testing HW requires $\Theta(1/\varepsilon^2)$ queries, in the noiseless setting it admits IPPs with the optimal linear dependence $O(1/\varepsilon)$ [RVW13, GGR18, AGR25]. We show that this optimal query complexity can be retained in the noisy setting.

Theorem 1.6 (Noisy IPP for HW, informal version of Theorem 4.4)**.** *Let $\varepsilon > 0$ be an accuracy parameter, and let $\eta_b < 1/2$ be a sufficiently small upper bound on the noise rate. Then, for every function $r : \mathbb{N} \to \mathbb{N}$ and every actual noise rate $\eta \leq \eta_b$, there exists an $r(n)$-round noisy* IPP *for* HW *such that the verifier makes $O(1/\varepsilon)$ queries, the honest prover runs in time $O(n)$, and the communication and verification time are $O((rn^{1/r} \log n)/\varepsilon)$.*

More generally, [GGR18] gives such a protocol for read-once branching programs (ROBPs), and our arguments for the noisy setting extend naturally to ROBPs. The HW protocol also naturally extends to the stronger doubly-sublinear setting of [AGR25]; see Remark 4.5.

Permutation. We next consider the Permutation problem (PERM), where the input is a function $f : [n] \to [n]$ and the goal is to verify that f is a permutation. In testing, PERM is governed by the birthday paradox and requires $\Omega(\sqrt{n})$ queries, whereas verification can be done with only $O(1/\varepsilon)$ queries. Indeed, one-round IPPs for PERM were constructed in [GGR18, GLR21, AGR25].[6] We show that a similar separation between testing and verification persists in the noisy setting.

Theorem 1.7 (Noisy IPP for PERM, informal version of Theorem 4.6)**.** *Let $\varepsilon > 0$ be a proximity parameter, and let $\eta_b < 1/2$ be a sufficiently small upper bound on the noise rate. Then, for every actual noise rate $\eta \leq \eta_b$, there exists a one-round noisy* IPP *for* PERM*, such that the verifier makes $O\left(1/(\varepsilon - \eta_b)\right)$ queries, the total communication is $O\left(\log n/(\varepsilon - \eta_b)\right)$, the verifier runs in time $O\left(1/(\varepsilon - \eta_b)\right)$, and the honest prover runs in time $O\left(n + 1/(\varepsilon - \eta_b)\right)$.*

[5] This is not accidental: noisy IPPs cannot achieve perfect completeness. See Claim 2.7 for the formal proof.

[6] In [GGR18], this is implicit, by applying their protocol to the ROBP that computes PERM, similarly to the way one derives an IPP for HW from their general framework.

1.2.3 Positive Results for Noisy PAC Verification

Fourier expansion and heavy coefficients. One convenient way to study the structure of a Boolean function $f : \{0,1\}^n \to \{-1,1\}$ is via its Fourier expansion

$$f(x) = \sum_{\gamma \in \{0,1\}^n} \widehat{f}(\gamma)\, \chi_\gamma(x), \qquad \text{where} \qquad \chi_\gamma(x) := (-1)^{\langle \gamma, x \rangle}.$$

Here $\gamma \in \{0,1\}^n$ is called a *Fourier coefficient*, and the corresponding *Fourier coefficient value* $\widehat{f}(\gamma) \in [-1,1]$[7] is the correlation

$$\widehat{f}(\gamma) := \mathop{\mathbb{E}}_{x \leftarrow \mathsf{Unif}(\{0,1\}^n)} \left[f(x) \chi_\gamma(x) \right].$$

The parity function $\chi_\gamma : \{0,1\}^n \to \{-1,1\}$ is called the *Fourier character of* γ. In many settings, f has a *sparse* Fourier spectrum: most of its Fourier mass is concentrated on a small set of characters. In such cases, identifying the *heavy* coefficients (those γ for which $\left|\widehat{f}(\gamma)\right|$ is relatively large) captures the main structure of f, since the remaining coefficients contribute only a small error. This is exactly the Fourier-sparse structure that our interactive protocol verifies, i.e., that f is well-approximated (over $\mathsf{Unif}(\{0,1\}^n)$) by a function whose Fourier expansion is supported on at most t characters.

Noisy interactive proof for Fourier-sparse functions. For any n-variate Boolean function f, an integer $t \in \mathbb{N}$ and $\varepsilon > 0$, we say that a set $\Lambda \subseteq \{0,1\}^n$ is ε-close to the set of the t heaviest Fourier coefficients of f if no vector in Λ whose coefficient has minimum absolute value can be replaced by a vector outside Λ whose coefficient has larger absolute value by more than ε. Equivalently, no element of Λ can be swapped with an element outside Λ to increase the minimum magnitude in Λ by more than ε. We show an interactive proof for verifying the t heaviest Fourier coefficients of a Boolean function up to accuracy ε, where the verifier only uses $\mathrm{poly}(t/\varepsilon)$ random noisy samples. We focus on $\mathcal{D} = \mathsf{Unif}(\{0,1\}^n)$, since Fourier coefficients are defined with respect to the uniform distribution.

Theorem 1.8 (Verifying heavy Fourier coefficients, informal version of Theorem 5.15)**.** *There exists an interactive proof such that for every f, every t, every accuracy parameter $\varepsilon > 0$, and every sufficiently small noise rate η, the verifier uses at most $\mathrm{poly}(t, 1/\varepsilon, 1/(1-2\eta))$ random noisy samples and outputs a set Λ that is ε-close to the set of the t heaviest Fourier coefficients of f with probability at least 0.9. Moreover, the prover is efficient in the sense that it is essentially as efficient as the learning algorithm it invokes, even though it only has access to noisy membership queries.*

As in the noiseless setting, verifying the heavy Fourier coefficients in this sense yields a PAC-verification protocol for Fourier-sparse functions. This theorem extends to the setting where both parties know only an *upper bound* on the noise rate (Definition 1.2), and also to the model in which the prover and verifier may experience different noise rates. In addition, we prove a lower bound on the required dependence between η and ε, giving partial evidence that a nontrivial η–ε tradeoff may be inherent in verification of heavy Fourier coefficients.

[7] In the Fourier-analysis literature, γ is often called a *coefficient*, a *character* or an *index*, and $\widehat{f}(\gamma)$ is also called a *coefficient* or a *coefficient value*. To avoid overloading the word "coefficient" in our setting, where it is crucial to distinguish between identifying heavy coefficients (the vectors γ) and merely estimating their values $\widehat{f}(\gamma)$, we deliberately refer to $\widehat{f}(\gamma)$ as the *coefficient value*, reserving the term *coefficient* for γ itself.

1.3 Technical Overview

1.3.1 Impossibility of Noisy IPPs with Constant Noise Rate

We sketch the two impossibility results of Theorem 1.3. Both follow a paradigm used in previous IPP lower bounds [RVW13, GR17, KR15]: construct two distributions on inputs that the protocol must treat differently, but that are hard to distinguish with limited resources. The first is tied to yes-instances, so that completeness forces high acceptance, while the second consists mostly of inputs far from the language, so soundness forces low acceptance. Thus, the protocol induces a distinguisher between the two distributions, contradicting the underlying hardness. In [RVW13, GR17] this hardness comes from unconditional circuit lower bounds, whereas [KR15] uses cryptographic pseudorandomness. We follow the latter approach, assuming an exponentially hard PRG G. In our setting, the standard bounded-communication simulation, which extends to noisy IPPs, approximates the maximum acceptance probability on a fixed input in time $2^{O(c)}\text{poly}(n)$; hence an acceptance gap between the two distributions contradicts pseudorandomness when $c = o(n)$.

Our new ingredient is constructing a suitable yes-distribution. A natural starting point is to build the language around a pseudorandom string $G(s)$. We want the language $\mathcal{L}$ to be decidable in a relatively small complexity class, however simply taking $\mathcal{L} = \{(G(s), s) : s \in \{0,1\}^k\}$ does not work, since the noisy views reveal information about s (while ordinary PRG security assumes that the seed is hidden). We therefore derive the seed from a longer random string w: the input contains w and the description a of a random pairwise-independent hash h_a, and the PRG seed is $h_a(w)$. Even after fixing one or two independent constant-noise views of w, the conditional distribution of w has linear min-entropy, meaning that a pairwise-independent hash (serving as a strong randomness extractor) can extract linearly many nearly uniform bits from it. Hence,

$$(a, h_a(w), \text{noisy view(s) of } w)$$

is statistically close to the same distribution with $h_a(w)$ replaced by an independent uniform seed s. We therefore define

$$\mathcal{L} := \{(a, G(h_a(w)), w)\}.$$

So the hash hides the PRG seed from the noisy views while keeping it determined by the clean input, and the PRG expands it into a long pseudorandom block; with sufficiently large constant stretch, uniform strings are far from $\mathcal{L}$. Consequently, either one or two noisy views of a random member of $\mathcal{L}$ are statistically close to the corresponding noisy view(s) of $(a, G(s), w)$, which is computationally indistinguishable from uniform by pseudorandomness of G.

For the noisy-prover lower bound (the first part of Theorem 1.3), consider the "fake yes-instance" distribution $z = (a, G(s), w)$, where a, w, s are independent and uniform. Although z need not belong to $\mathcal{L}$, a cheating prover can sample a noise mask, apply it to z and run the honest prover on it. Since the joint noisy views of this prover and the verifier are statistically close to their views in an honest execution on $(a, G(h_a(w)), w) \in \mathcal{L}$, the verifier accepts z with high probability by completeness. In contrast, soundness forces low acceptance on a uniform input, so the simulation above distinguishes $(a, G(s), w)$ from uniform, contradicting pseudorandomness of G when $c = o(n)$.

For the unknown-noise lower bound (the second part of Theorem 1.3), choose two possible verifier noise rates $\eta_0 < \eta_1$, and let θ satisfy $\eta_0 + \theta - 2\eta_0\theta = \eta_1$. Thus, θ-noise followed by independent η_0-noise is distributed exactly as η_1-noise. Starting from a random $x = (a, G(h_a(w)), w) \in \mathcal{L}$, define $z := x \oplus \xi$ for $\xi \leftarrow \mathsf{Bernoulli}(\theta)^n$. The verifier's noisy view of z at noise rate η_0 is therefore identical to its view of x at noise rate η_1. Since the verifier knows only an upper bound on the rate, a

cheating prover that emulates the honest prover on x makes it accept z with high probability by completeness. As we showed, $z = (a, G(h_a(w)), w) \oplus \xi$ is also statistically close to $(a, G(s), w) \oplus \xi$ for a uniform independent s, so the verifier accepts the latter with high probability as well. Once again, soundness forces low acceptance on a uniform input, so we obtain a distinguisher between $(a, G(s), w) \oplus \xi$ and uniform, contradicting pseudorandomness of G when $c = o(n)$.

1.3.2 Impossibility of Robust IPPs with Constant Robustness

We sketch the proof of Theorem 1.4. Recall that in a ρ-robust IPP, if the input is far from the language, then after the interaction the verifier's local view must remain ρ-far from every accepting view. Namely, if the verifier's query set is Q of size q and its decision predicate is $\phi : \{0,1\}^q \to \{0,1\}$, then

$$\Pr\left[\Delta\left(x|_Q, \{z \in \{0,1\}^q : \phi(z) = 1\}\right) \leq \rho\right] \leq \frac{1}{3}. \tag{1}$$

The formal definition is given in Definition 2.2.

The idea is to recursively compose the robust IPP with itself. After one communication phase, the verifier's decision predicate defines an accepting set over the queried coordinates. By succinctness, this accepting set is itself a language in NC^1, so we may recurse on the restricted input. Robustness guarantees that an input initially far from $\mathcal{L}$ induces, except with small probability, a restricted input that remains ρ-far from the new accepting set, and hence the same argument can be repeated. To control the error accumulation, at each stage we run several independent copies of the communication phase and let the prover select one copy on which to continue; this boosts completeness while increasing the robust-soundness error only by an amount that remains small over all stages. After $O(\log \log n)$ stages, the query complexity becomes constant while the total communication remains sublinear. The resulting IPP therefore satisfies $q \cdot \mathsf{cc} = o(n)$, contradicting the lower bound of [KR15]. Unlike our noisy-IPP lower bounds, which use the underlying cryptographic assumption directly, here we invoke the IPP lower bound of [KR15] itself.

1.3.3 Noisy IPPs for HW and PERM

We next explain how classical IPPs for Hamming Weight and Permutation can be adapted to our noisy setting (Theorems 1.6 and 1.7). Although the underlying protocols remain simple, the noisy model introduces new completeness and soundness effects that must be controlled without losing the sublinear verification guarantees. We highlight these issues below; the full formal statements and proofs appear in Sections 4.2 and 4.3.

Hamming Weight. Recall that in HW the goal is to output an additive εn-approximation to the Hamming weight of the input. The classical IPPs of [RVW13, GGR18, AGR25] recursively decompose a claimed total weight into claims on smaller blocks, while the verifier follows a random root-to-leaf path and checks consistency until reaching a single queried coordinate. If the initial claim is wrong by more than εn, a discrepancy survives along the random path with probability at least ε, yielding optimal query complexity $O(1/\varepsilon)$ after repetition. The same framework of [GGR18] applies more generally to ROBPs, and our noisy analysis extends to that setting as well.

The main issue in the noisy setting is completeness: the honest prover's top-level claim is computed from its own noisy view, and may therefore differ from the true Hamming weight, and moreover, at the end of the recursion, the prover's and verifier's noisy views may disagree on

the final queried bit. Soundness follows the same argument as in the noiseless protocol with one additional issue: verifier noise may hide a genuine inconsistency at the final queried coordinate. Thus the analysis controls the global error in the claimed weight, and the two local errors in the final consistency check arising in completeness and soundness.

Permutation. Recall that in PERM the input is a function $f : [n] \to [n]$, and the goal is to verify that f is a permutation. We adapt the IPP of [GLR21]: the verifier samples a uniformly random value $y \in [n]$, asks the prover for a preimage x, and checks that $f(x) = y$. If f is ε-far from PERM, then at least an ε-fraction of values have no preimage, giving a completeness–soundness gap of ε which is then amplified using $O(1/\varepsilon)$ repetitions.

In our noisy function-oracle model, each oracle value is independently corrupted with probability η to an arbitrary value in $[n]$, persistently across repeated queries. Noise affects both sides of this ε gap. In soundness, a corrupted verifier answer may create an apparent preimage for a value that has none. In completeness, the honest prover may fail to see the true preimage or may instead see a spurious one created by its noisy oracle, while the verifier's own noisy view may reject an otherwise correct answer. Controlling these events shows that the completeness–soundness gap decreases from ε to roughly $\varepsilon - O(\eta)$, and hence $O(1/(\varepsilon - \eta_b))$ repetitions suffice whenever η_b is a sufficiently small constant fraction of ε.

1.3.4 Noisy PAC Verification of Fourier-Sparse Functions

The proof of Theorem 1.8 follows the high-level template of the noiseless setting ([GRSY21, GJK+24]): constructing a PAC-verification protocol in which the verifier has access to membership queries, and then converting it into a samples-only verifier via a query-to-samples transformation.

Noisy query-to-samples transformation. In the noiseless query-to-samples transformations of [GRSY21, GJK+24], the verifier uses a random labeled sample to construct a membership-query set having the correct distribution, sends the unlabeled query set to the prover, and uses the sampled point to check one of the prover's returned labels. Since the prover does not know which query came from the verifier's sample, consistently mislabeling the query set risks being detected. With noise, this check becomes a biased-coin test rather than a one-sided certificate, and consequently our noisy transformation incurs a quadratic overhead in the number of repetitions; see Theorem 5.4 and Remark 5.6.

When the prover is noisy as well, the transformation can only simulate access to the prover's noisy oracle f_η, so the underlying query-based verifier must remain complete under random noise. Moreover, since the verifier must tolerate the mismatches caused by honest noise, a cheating prover can use this tolerance to hide *adversarial* corruptions. Thus, for soundness, the underlying verifier must also tolerate an adversarially corrupted query oracle; this yields the fully noisy transformation of Theorem 5.8.

Noisy Fourier verification with queries. We adapt the query-based Fourier-verification protocol of [GJK+24]. In the noiseless protocol, the prover sends a candidate set $\Lambda = \{\gamma_1, \ldots, \gamma_t\}$ of heavy coefficients; the verifier estimates the corresponding values $|\widehat{f}(\gamma_i)|$ from random samples, and independently estimates the values of the true top t Fourier coefficients without identifying their locations. It accepts only if these two collections of values are sufficiently close.

There are three ingredients in extending this protocol to the noisy setting. First, the honest prover can still find the heavy coefficients from noisy membership queries using a noise-tolerant Goldreich–Levin (or Kushilevitz–Mansour [KM93]) procedure; see Theorem C.1. Second, the verifier can estimate the values of the claimed coefficients directly from its own noisy random samples, correcting for the known noise bias. The main difficulty is the third ingredient: estimating the values of the true top Fourier coefficients in a way that remains sound when an η-fraction of the query answers may be corrupted adversarially (so that the query-to-samples transformation can be applied). This is done in Theorem 5.12 and is the most technical part of this section.

As in [GJK+24], the verifier randomly partitions the Fourier domain into affine subspaces and estimates an L_4-based quantity for each subspace using four-query tuples. The key new idea is to estimate all of these quantities simultaneously from the same collection of randomly generated four-query tuples. Because every subspace estimate uses the same queried points, an adversary cannot concentrate its corruption budget on a particular subspace containing a heavy coefficient: corrupting an η-fraction of the queries changes every estimate by at most $O(\eta)$. Taking fourth roots then recovers the desired coefficient values. This is also where the condition $\eta = O(\varepsilon^4)$ arises, since a coefficient of magnitude $\Omega(\varepsilon)$ contributes $\Omega(\varepsilon^4)$ to the relevant L_4-based estimate, and in Theorem 5.18 we give partial evidence that some nontrivial dependence between η and ε is necessary, by showing that tolerating $\eta = \Omega(\varepsilon)$ for a natural class of such protocols would contradict the Learning Parity with Noise (LPN) assumption. Together, these ingredients yield Theorem 1.8.

Finally, the protocol extends to the setting where the parties know only an upper bound on the noise rate, by calibrating the tests to the worst-case rate and absorbing the resulting slack into the adversarial-corruption budget.

1.4 Related Work on Noisy Testing and Learning

In this section, we survey prior work on noisy models of property testing and learning, which can be viewed as the two simpler analogues of our noisy IPPs and noisy PAC-verification settings.

Property Testing with noise. Property testing can be viewed as a special case of IPPs where there is no prover and the verifier alone tests the input to check if it satisfies a property. The motivation remains the same: the input is often too large for a computationally constrained verifier to read entirely.

Several models extend property testing to settings where the input is not only large but also noisy. The two primary models involve *adversarial erasures* (introduced in [DRTV16]) and *adversarial corruptions* (mostly referred to as *tolerant testing*, introduced in [PRR06]). In both models, an adversary manipulates the input before the verification starts, either by replacing values with $\perp$ (erasures) or by flipping bits (corruptions). More recently, [KRV22] initiated the study of online adversarial manipulations, where the input may deteriorate during the execution itself; later works considered more general online models with both erasures and corruptions.

In our model, the corruptions are *random* rather than adversarial: each location is corrupted independently at rate η, instead of being chosen by an adversary subject to a budget. In this sense, the upper bound on the noise rate in our model plays a role analogous to the adversarial budget in noisy-testing models: both quantify the amount of noise the verifier must tolerate.

Our lower bounds are qualitatively different from previous impossibility results in the adversarial erasure model (e.g., [DRTV16, BEFLR20]), in two ways. First, we rule out a more powerful verification model (IPP, where the verifier has help from a prover) rather than a tester. Second, we

show impossibility under *random* corruptions instead of adversarial erasures. On the other hand, our result is weaker in that it relies on a cryptographic assumption.

Learning with noise. Learning with random (fresh) classification noise in labeled samples was first formalized by Angluin and Laird [AL88], who also emphasize the role of an upper bound on the noise rate.[8] Later, Kearns [Kea98] introduced the statistical query (SQ) framework and showed that SQ-learnability implies tolerance to random (fresh) classification noise up to rates approaching $1/2$; this gives a general route to noise-robust learning in the non-persistent setting.

To rule out the trivial denoising-by-repetition phenomenon when membership queries are available, Goldman, Kearns, and Schapire [GKS93] introduced persistent noise in queries (in an exact-identification setting, not PAC learning), including a slightly more general asymmetric flip model. There, they assume the learner gets an upper bound on the noise rate. Later, Jackson, Shamir, and Shwartzman [JSS99] develop general tools for learning with membership queries corrupted by persistent classification noise by extending SQ-style noise-compensation techniques, with applications including a noise-tolerant "Weak Parity" (Goldreich–Levin / Kushilevitz–Mansour).

Note that in learning settings where the noise rate is unknown, we may often, without loss of generality, assume it is known by employing a standard search trick over the allowed range, even in agnostic regimes: one runs the learner for a sequence of candidate noise parameters and selects an output using an appropriate validation criterion. While an honest prover could in principle perform such an optimization, it is irrelevant to our setting: the verifier cannot afford to run the learner (otherwise verification gives no efficiency gain), and the prover cannot simply transmit a claimed noise rate to the verifier in a sound way, as discussed above.

For broader context, *Massart noise* allows an input-dependent flip probability bounded by η and has been studied extensively, but it differs from our model in that the corruption is not persistent and may vary adversarially with the input. Another standard alternative is *malicious/adversarial noise* (and related contamination models), where an η-fraction of labels/examples is corrupted arbitrarily, typically also in a non-persistent, per-sample manner.

Organization. Section 2 introduces the formal definitions of noisy and robust IPPs. Section 3 proves impossibility results for noisy IPPs in the noisy-prover and unknown-noise settings, and establishes a related impossibility result for robust IPPs. Section 4 gives our positive results for noisy IPPs, including a transformation from robust IPPs, and protocols for HW and PERM. Finally, Section 5 develops noisy PAC verification.

2 Interactive Proofs of Proximity (IPPs)

We begin with the standard definition of an IPP.

Definition 2.1 (Interactive Proof of Proximity (IPP))**.** *Fix a language $\mathcal{L} \subseteq \{0,1\}^n$ and a proximity parameter $\varepsilon \in (0,1]$. An ε-*Interactive Proof of Proximity (ε-IPP) *for $\mathcal{L}$ is an interactive protocol between a verifier $\mathcal{V}$ and a prover $\mathcal{P}$, both taking explicit inputs $1^n, \varepsilon$ and oracle access to the input $x \in \{0,1\}^n$, such that the following hold:*

[8] They also describe how one may estimate a valid upper bound in the *realizable* setting by iteratively searching for a hypothesis with sufficiently low disagreement; in general this amounts to solving an ERM problem, which may be computationally infeasible. Moreover, since PAC verification is inherently *agnostic*, this estimation regime is not relevant here, and we instead take η_b as an externally given parameter.

- ***Completeness.*** *If $x \in \mathcal{L}$, then after interacting with $\mathcal{P}$, $\Pr[\mathcal{V}$ accepts$] \geq \frac{2}{3}$.*
- ***Soundness.*** *If x is ε-far from $\mathcal{L}$, then for every unbounded cheating prover $\widetilde{\mathcal{P}}$, after interacting with $\mathcal{V}$, $\Pr[\mathcal{V}$ accepts$] \leq \frac{1}{3}$.*

All probabilities are over $\mathcal{V}$'s randomness, and distances are relative Hamming.

We say the protocol has query complexity $q = q(n, \varepsilon)$ if, for every input $x \in \{0,1\}^n$ and any prover, $\mathcal{V}$ makes at most $q(n, \varepsilon)$ oracle queries to x; the prover may read arbitrarily many locations of the input (up to n queries). The protocol has communication complexity $\mathsf{cc} = \mathsf{cc}(n, \varepsilon)$ if the total number of bits exchanged between $\mathcal{V}$ and $\mathcal{P}$ is at most cc. Its running times are $\mathcal{V}_{\mathsf{time}} = \mathcal{V}_{\mathsf{time}}(n, \varepsilon)$ for the verifier and $\mathcal{P}_{\mathsf{time}} = \mathcal{P}_{\mathsf{time}}(n, \varepsilon)$ for the honest prover.

2.1 Robust IPPs

For $u, v \in \{0,1\}^m$, $\Delta(u, v)$ denotes relative Hamming distance and $\mathsf{wt}(u \oplus v)$ denotes the Hamming weight (absolute Hamming distance).

Next, we state the definition of robust IPPs, following [HR24].

Definition 2.2 (Robust IPP)**.** *Fix a language $\mathcal{L} \subseteq \{0,1\}^n$, a proximity parameter $\varepsilon \in (0, 1]$, and a robustness parameter $\rho \in (0, 1]$. An (ε, ρ)-*robust IPP *for $\mathcal{L}$ is an interactive protocol between a verifier $\mathcal{V}$ and a prover $\mathcal{P}$, both taking explicit inputs $1^n, \varepsilon$ and oracle access to the input $x \in \{0,1\}^n$, such that the following hold.*

The interaction between the two parties produces a communication transcript. After the interaction (and before any oracle queries), the transcript and the verifier's randomness fix a query set $Q \subseteq [n]$ and a predicate $\phi : \{0,1\}^q \to \{0,1\}$, where $q := |Q|$. The verifier then reads $x|_Q$ and accepts iff $\phi(x|_Q) = 1$. Let $A := \{z \in \{0,1\}^q : \phi(z) = 1\}$.

- ***Completeness.*** *If $x \in \mathcal{L}$, then after interacting with $\mathcal{P}$, $\Pr[\phi(x|_Q) = 1] \geq \frac{2}{3}$.*
- ***Robust soundness.*** *If x is ε-far from $\mathcal{L}$, then for every unbounded cheating prover $\widetilde{\mathcal{P}}$, after interacting with $\mathcal{V}$,*
$$\Pr[\Delta(x|_Q, A) \leq \rho] \leq \tfrac{1}{3}.$$

All probabilities are over $\mathcal{V}$'s internal randomness, and distances are relative Hamming.

Complexity measures are as in Definition 2.1 (query complexity q, communication complexity cc, and running times $\mathcal{V}_{\mathsf{time}}, \mathcal{P}_{\mathsf{time}}$), with parameter dependence $q = q(n, \varepsilon, \rho)$, $\mathsf{cc} = \mathsf{cc}(n, \varepsilon, \rho)$, $\mathcal{V}_{\mathsf{time}} = \mathcal{V}_{\mathsf{time}}(n, \varepsilon, \rho)$, and $\mathcal{P}_{\mathsf{time}} = \mathcal{P}_{\mathsf{time}}(n, \varepsilon, \rho)$.

We note that robustness cannot be amplified by simply repeating the query phase, since the guarantee is stated in terms of *relative* Hamming distance: repeating the same local test may increase the absolute distance from the accepting set, but it increases the number of queried coordinates by the same factor, and hence need not improve the relative distance.

Remark 2.3 (Optimal robustness)**.** *As pointed out in [HR24, Remark 4.4], robustness cannot in general exceed ε. Indeed, if x is ε-close to $\mathcal{L}$, a cheating prover can act as if the input were some $x' \in \mathcal{L}$ with $\Delta(x, x') = \varepsilon$. It might be the case that x differs from x' in a random ε-fraction of the queried locations. Thus, on accepting transcripts for x', the local view $x|_Q$ is only about ε-far from the accepting set A. Hence one cannot in general guarantee robustness $\rho > \varepsilon$.*

Succinct decision predicates. We use NC^1 for the class of logspace-uniform Boolean circuit families of logarithmic depth and polynomial size. Namely, $\mathcal{L} \in \mathsf{NC}^1$ if there exists a logspace Turing machine M that, on input 1^n, outputs a description of a circuit $C : \{0,1\}^n \to \{0,1\}$ of depth $O(\log n)$ and size $\mathrm{poly}(n)$ such that $C(x) = 1$ if and only if $x \in \mathcal{L}$.

For robust IPPs, we define succinctness of decision predicates. Various notions of succinct representation have appeared in prior works (e.g., [RRR18, RR20, AR23, HR24]); the notion we use here is the weakest among them, and should therefore be viewed as a very mild requirement. It is also satisfied by the IPP theorem of [RR20], and implicitly in other constructions (e.g., [RVW13, GGR18, GLR21]).

The $\widetilde{O}$ notation hides polylogarithmic factors.

Definition 2.4 (Succinct decision predicates)**.** *A robust* IPP *has a succinct decision predicate if the decision predicate $\phi : \{0,1\}^q \to \{0,1\}$ that the verifier generates at the end of the interaction phase has a description $\langle\phi\rangle$ of length $\widetilde{O}(q)$, such that $\langle\phi\rangle$ can be expanded to a full description of an NC^1 circuit computing ϕ. That is, there exists a logspace Turing machine that on input 1^q, outputs a description of an NC^1 circuit C such that for every $z \in \{0,1\}^q$, it holds that $C(\langle\phi\rangle, z) = \phi(z)$.*

2.2 Noisy IPPs

Next, we state the definition of a noisy IPP. We begin with Definition 2.5, which gives the simpler definition in the setting where both parties know the exact noise rate η. We then present in Definition 2.6 the more general definition, in which the parties know only an upper bound $\eta_b \geq \eta$. Both definitions will be useful in the positive and negative results.

Recall our notations for the relative Hamming distance $\Delta(u, v)$ and the absolute Hamming distance (Hamming weight) $\mathsf{wt}(u \oplus v)$.

Definition 2.5 (Noisy IPP with known noise rate)**.** *Fix a language $\mathcal{L} \subseteq \{0,1\}^n$, a proximity parameter $\varepsilon \in (0,1]$, and a noise rate $\eta \in [0, \frac{1}{2})$. An (ε, η)-*noisy IPP with known noise rate *for $\mathcal{L}$ is an interactive protocol between a verifier $\mathcal{V}$ and a prover $\mathcal{P}$, both taking explicit inputs $1^n, \varepsilon, \eta$ and each getting oracle access to a (possibly different) noisy copy of the input $x \in \{0,1\}^n$, such that the following hold:*

- ***Input access.*** *Each party $A \in \{\mathcal{P}, \mathcal{V}\}$ has a private persistent-noise oracle defined by a noise mask $\xi^A \in \{0,1\}^n$ with i.i.d. bits $\xi_i^A \leftarrow \mathsf{Bernoulli}(\eta)$, independent across indices and across parties. On the first query to index $i \in [n]$, A receives*

$$\tilde{x}_i^A := x_i \oplus \xi_i^A,$$

  and every subsequent query to the same i returns the same value $\tilde{x}_i^A$.

- ***Completeness.*** *If $x \in \mathcal{L}$, then after interacting with $\mathcal{P}$, $\Pr[\mathcal{V}$ accepts$] \geq \frac{2}{3}$.*

- ***Soundness.*** *If x is ε-far from $\mathcal{L}$, then for every unbounded cheating prover $\widetilde{\mathcal{P}}$ (that knows the clean x), after interacting with $\mathcal{V}$, $\Pr[\mathcal{V}$ accepts$] \leq \frac{1}{3}$.*

All probabilities are over $\mathcal{V}$'s internal randomness and both noise masks, and distances are Hamming.

Complexity measures are as in Definition 2.1 (query complexity q, communication complexity cc, and running times $\mathcal{V}_{\mathsf{time}}, \mathcal{P}_{\mathsf{time}}$), with parameter dependence $q = q(n, \varepsilon, \eta)$, $\mathsf{cc} = \mathsf{cc}(n, \varepsilon, \eta)$, $\mathcal{V}_{\mathsf{time}} = \mathcal{V}_{\mathsf{time}}(n, \varepsilon, \eta)$, and $\mathcal{P}_{\mathsf{time}} = \mathcal{P}_{\mathsf{time}}(n, \varepsilon, \eta)$.

We now turn to the more general definition, in which the parties know only an upper bound on the noise rate.

Definition 2.6 (Noisy IPP with unknown noise rate)**.** *Fix a language $\mathcal{L} \subseteq \{0,1\}^n$, a proximity parameter $\varepsilon \in (0,1]$, and an* upper bound $\eta_b \in [0, \frac{1}{2})$*. An (ε, η_b)-*noisy IPP with unknown noise rate *for $\mathcal{L}$ is an interactive protocol between a verifier $\mathcal{V}$ and a prover $\mathcal{P}$, both taking explicit inputs $1^n, \varepsilon, \eta_b$ and each getting oracle access to a (possibly different) noisy copy of the input $x \in \{0,1\}^n$, such that the following hold for every* actual *noise rate $\eta \in [0, \eta_b]$ which is not given to $\mathcal{V}$ or $\mathcal{P}$:*

- ***Input access.*** *Each party $A \in \{\mathcal{P}, \mathcal{V}\}$ has a private persistent-noise oracle defined by a noise mask $\xi^A \in \{0,1\}^n$ with i.i.d. bits $\xi_i^A \leftarrow \mathsf{Bernoulli}(\eta)$, independent across indices and across parties. On the first query to index $i \in [n]$, A receives*

$$\tilde{x}_i^A := x_i \oplus \xi_i^A,$$

  and every subsequent query to the same i returns the same value $\tilde{x}_i^A$.

- ***Completeness.*** *If $x \in \mathcal{L}$, then after interacting with $\mathcal{P}$, $\Pr[\mathcal{V}$ accepts$] \geq \frac{2}{3}$.*

- ***Soundness.*** *If x is ε-far from $\mathcal{L}$, then for every unbounded cheating prover $\widetilde{\mathcal{P}}$ (that knows the clean x and η), after interacting with $\mathcal{V}$, $\Pr[\mathcal{V}$ accepts$] \leq \frac{1}{3}$.*

All probabilities are over $\mathcal{V}$'s internal randomness and both noise masks, and distances are Hamming.

Complexity measures are as in Definition 2.5 (query complexity q, communication complexity cc, and running times $\mathcal{V}_{\mathsf{time}}, \mathcal{P}_{\mathsf{time}}$), with parameter dependence $q = q(n, \varepsilon, \eta_b)$, $\mathsf{cc} = \mathsf{cc}(n, \varepsilon, \eta_b)$, $\mathcal{V}_{\mathsf{time}} = \mathcal{V}_{\mathsf{time}}(n, \varepsilon, \eta_b)$, and $\mathcal{P}_{\mathsf{time}} = \mathcal{P}_{\mathsf{time}}(n, \varepsilon, \eta_b)$.

Note that when $\eta_b = \eta$ is exact, Definition 2.6 coincides with Definition 2.5.

Next, we formalize the intuition that a noisy IPP cannot have perfect completeness. Since we are proving an impossibility result, it suffices here to work with the simpler definition in which the exact noise rate η is known to both parties (Definition 2.5). Informally, because the noise is persistent, every consistent pattern of oracle answers on the verifier's queried locations can arise from a fixed yes-instance under a suitable noise mask. Perfect completeness would therefore force the verifier to accept every such local view, and a cheating prover can ignore the actual input and simply emulate the honest prover on a fixed yes-instance, causing the verifier to accept even no-instances and thereby violating soundness. This argument is similar in spirit to [Tel20], where it is shown that tolerant testers cannot have one-sided error (that is, perfect completeness or perfect soundness) unless they are trivial.

Claim 2.7 (Perfect completeness is impossible)**.** *Fix $\eta \in (0, 1/2)$ and let $\langle \mathcal{P}, \mathcal{V} \rangle$ be an (ε, η)-noisy IPP for a nonempty language $\mathcal{L}$. If the protocol has perfect completeness, i.e. for every $x \in \mathcal{L}$ the verifier accepts with probability 1 (over all randomness of the protocol and the noise masks), then the protocol cannot satisfy soundness for any $\varepsilon \in (0,1]$ for which there exists an input that is ε-far from $\mathcal{L}$. This remains true even in the verifier-only-noise model.*

Proof. Fix any $x^\star \in \mathcal{L}$, and fix any realization of the honest prover's persistent-noise mask that occurs with positive probability. This induces a deterministic honest-prover strategy $\mathcal{P}^\star$ on $x^\star$. By perfect completeness, every positive-probability execution of $\langle \mathcal{P}^\star, \mathcal{V} \rangle$ on $x^\star$ is accepting.

Now consider the cheating prover $\widetilde{\mathcal{P}}$ that, on every input x, ignores x and emulates $\mathcal{P}^\star$. Fix any execution of $\langle \widetilde{\mathcal{P}}, \mathcal{V} \rangle$ on an arbitrary $x \in \{0,1\}^n$, including the verifier's randomness and persistent-noise mask. Because the noise is persistent, this execution determines a consistent answer a_i for every coordinate i queried by the verifier. However, the same execution can occur on $x^\star$: consider a verifier's noise mask that satisfies $\xi_i^{\mathcal{V}} = x_i^\star \oplus a_i$ on every queried coordinate. Since $\eta \in (0, 1/2)$, such a mask has positive probability. With the same verifier randomness and the same prover strategy $\mathcal{P}^\star$, the verifier therefore sees exactly the same messages and oracle answers, makes the same subsequent choices, and reaches the same decision.

Thus every execution of $\langle \widetilde{\mathcal{P}}, \mathcal{V} \rangle$ on every input x corresponds to a positive-probability execution on the yes-instance $x^\star$, and is therefore accepting by perfect completeness. Hence $\widetilde{\mathcal{P}}$ makes the verifier accept every input with probability 1, contradicting soundness whenever an ε-far input exists. □

Repetitions. In the noisy setting, the standard repetition argument requires some care, since all repetitions are carried out with respect to the same fixed noisy copy of the input. A clean sufficient condition is that the verifier does not query its input during the interaction and that the verifier's query sets in the different repetitions be disjoint. The first condition ensures that the entire interaction is fixed before any oracle answers are seen, so the prover cannot learn information about the verifier's noisy local view during the interaction itself. Disjointness then guarantees that each repetition depends on fresh verifier randomness and on fresh coordinates of the fixed noise mask $\xi^{\mathcal{V}}$. In our positive results of Section 4, however, we prove the required bounds directly from the structure of the protocols.

3 Impossibility Results for Noisy IPPs

Throughout this section, we assume the existence of exponentially hard pseudorandom generators computable in NC^1, as in the lower bound of Kalai and Rothblum [KR15]. We prove two impossibility results for noisy IPPs, showing that even one of the two natural relaxations of the clean-prover or known-noise setting already make noisy IPPs impossible. First, in Section 3.2, we show that if the honest prover is noisy, then there exists a language in NC^1 for which no such protocol exists even when the exact noise rate is known. Second, in Section 3.3, we show that even with a clean honest prover, the same conclusion holds if the verifier knows only an upper bound on its noise rate. In fact, both lower bounds hold for the same language, defined in Section 3.1. The latter result shows that uncertainty of the verifier about its own noise channel is itself a genuine source of hardness: the lower bound continues to hold even if the clean prover is told the verifier's exact noise rate and only the verifier is not.[9]

Finally, in Section 3.4, we present a related but separate impossibility result for robust IPPs, showing that constant robustness cannot in general be achieved uniformly over NC^1 with both

[9] One might hope that an honest prover who knows the verifier's exact noise rate could communicate, or even interactively certify, this rate to the verifier. The lower bound rules this out in general. The protocol is required to work for every actual noise rate below the given upper bound; a cheating prover can use the honest prover strategy corresponding to a larger rate, so the verifier cannot soundly rely on the prover to identify its true noise rate.

sublinear query and communication complexity. Looking ahead, this will provide a useful point of comparison between noisy and robust IPPs, as the positive result of Section 4.1 uses a robust IPP for implementing noisy IPPs in a restricted setting.

3.1 A Common Hard Language

We first define the language used in both lower bounds. The construction is based on the following idea: if w is uniform and one observes one or two independent noisy copies of it at a positive constant noise rate, this view still leaves a constant amount of uncertainty about each bit of w, and hence linear uncertainty about w as a whole. Thus, a pairwise-independent hash (used as a randomness extractor) can extract a linear number of bits from w, such that these bits would be statistically close to uniform even given the noisy view(s) of w. Then, we use these extracted bits which are close-to-uniform as the seed of a pseudorandom generator. This gives a string which is computationally indistinguishable from a uniform one, and we define the language $\mathcal{L}$ such that whenever the verifier decides membership in $\mathcal{L}$, then it distinguished between this string and a truly uniformly random one. Since uniform strings will be far from the language, a low-communication noisy IPP would yield a distinguisher for the pseudorandom generator.

We refer the reader to Section A to the formal definition of pairwise-independent hash functions and a construction.

Fix constants $\varepsilon, \gamma \in (0, 1/2)$, where ε is a proximity parameter and γ is a noise rate. For every $k \in \mathbb{N}$, let $m = rk$, where $r = r(\gamma)$ is a sufficiently large constant integer to be fixed below. Let $\mathcal{H}$ be the pairwise independent hash family from Claim A.3, instantiated as functions $h : \{0,1\}^m \to \{0,1\}^k$, and write h_a for the function described by $a \in \{0,1\}^{m+2k-1}$.

Let $G_k : \{0,1\}^k \to \{0,1\}^{\lambda k}$ be an exponentially hard pseudorandom generator, where $\lambda > 1$ is a sufficiently large constant integer depending only on ε and γ. By a constant number of standard iterations, we may assume that the generator has any fixed constant stretch while preserving exponential hardness and NC^1 computability. Define

$$\mathcal{L} := \left\{ (a, G_k(h_a(w)), w) : a \in \{0,1\}^{m+2k-1},\ w \in \{0,1\}^m \right\}.$$

The input length is $n := (2r + \lambda + 2)k - 1 = \Theta(k)$.

The following lemma is information-theoretic; it works for any function $G_k : \{0,1\}^k \to \{0,1\}^{\lambda k}$ computable in NC^1, and does not use pseudorandomness of the PRG.

Lemma 3.1 (Properties of the hard language)**.** *For every fixed $\varepsilon, \gamma \in (0, 1/2)$, the constants r and λ above can be chosen such that $\mathcal{L} \in \mathsf{NC}^1$ and the following hold.*

1. *A uniformly random n-bit string is ε-far from $\mathcal{L}$ except with probability $2^{-\Omega(n)}$.*
2. *Let a, w, s be independent and uniform in $\{0,1\}^{m+2k-1}, \{0,1\}^m, \{0,1\}^k$, respectively. Let $\xi^{(1)}, \xi^{(2)} \leftarrow \mathsf{Bernoulli}(\gamma)^n$ be independent. For $t \in \{1, 2\}$, the joint distribution*

$$\left((a, G_k(h_a(w)), w) \oplus \xi^{(j)} \right)_{j \in [t]}$$

is within statistical distance $2^{-\Omega(k)}$ of

$$\left((a, G_k(s), w) \oplus \xi^{(j)} \right)_{j \in [t]}.$$

Proof. Membership in $\mathcal{L}$ is in NC^1: given (a, y, w), the circuit computes $h_a(w)$ (the hash functions from Definition A.2 are in logspace-uniform NC^1), then computes $G_k(h_a(w))$, and checks equality with y.

For Item 1, note that $|\mathcal{L}| \le 2^{2m+2k-1}$. A Hamming ball of relative radius ε in $\{0,1\}^n$ has size at most $2^{H_2(\varepsilon)n}$, where H_2 is the binary entropy function. Hence

$$\Pr_{z \leftarrow \mathsf{Unif}(\{0,1\}^n)}[\Delta(z, \mathcal{L}) \le \varepsilon] \le 2^{2m+2k+H_2(\varepsilon)n-n}.$$

Since $\varepsilon < 1/2$, we have $H_2(\varepsilon) < 1$, and choosing the constant stretch λ sufficiently large makes the right-hand side $2^{-\Omega(n)}$.

For Item 2, let $w \leftarrow \mathsf{Unif}(\{0,1\}^m)$. Given one γ-noisy observation of a coordinate w_i, its more likely value has conditional probability $1 - \gamma < 1$. Given two independent γ-noisy observations, if the two observed bits disagree then either value of w_i is equally likely. If they agree, say both observations equal b, then Bayes' rule gives

$$\Pr[w_i = b \mid \text{both observations equal } b] = \frac{\text{both observations are clean}}{\text{both observations are equal}} = \frac{(1-\gamma)^2}{(1-\gamma)^2 + \gamma^2} < 1.$$

Consequently, for every fixed realization z of either one or two noisy views of W, the conditional distribution $W \mid Z = z$ assigns probability at most p_γ^m to every $w \in \{0,1\}^m$, for some constant $p_\gamma < 1$ depending only on γ. Recall that the min-entropy of a random variable X is

$$H_\infty(X) := -\log_2 \left(\max_x \Pr[X = x] \right).$$

Thus, for every such z,

$$H_\infty(W \mid Z = z) \ge m \log_2(1/p_\gamma) = \Omega(m).$$

We choose r sufficiently large so that this quantity is at least $4k$. By the Leftover Hash Lemma for pairwise-independent hashing [Gol08, Sections D.2.3 and D.4.1.1],[10] for every fixed z,

$$(A, h_A(W) \mid Z = z)$$

is within statistical distance $2^{-\Omega(k)}$ of

$$(A, U_k),$$

where A is a uniform and independent hash seed and U_k is uniform over $\{0,1\}^k$ and independent of A. Averaging over z, it follows that $(A, h_A(W), Z)$ is within statistical distance $2^{-\Omega(k)}$ of (A, U_k, Z). This means that whenever a and s are uniform and independent, both

$$(a, h_a(w), w \oplus \xi) \quad \text{and} \quad \left(a, h_a(w), w \oplus \xi^{\mathcal{P}}, w \oplus \xi^{\mathcal{V}}\right)$$

are within statistical distance $2^{-\Omega(k)}$ of the corresponding distributions obtained by replacing $h_a(w)$ by s, where $\xi, \xi^{\mathcal{P}}, \xi^{\mathcal{V}} \leftarrow \mathsf{Bernoulli}(\gamma)^m$ are independent.

To complete the proof, we need to show that this property is preserved after applying the G_k and applying noise. Indeed, this step is simply applying the same randomized postprocessing to the

[10] In the terminology of [Gol08, Section D.4.1], a pairwise-independent hash family is viewed as a strong extractor: every random variable X with min-entropy at least t yields $(A, h_A(X))$ statistically close to (A, U_k), provided the output length k is sufficiently smaller than t. Indeed, we use the hash as a randomness extractor.

two statistically close distributions: apply G_k to the hash output and independently add γ-noise to the first and second blocks for each party. Statistical distance cannot increase under randomized postprocessing: any distinguisher for the postprocessed distributions immediately induces a distinguisher for the original distributions with the same advantage. This gives the claimed closeness of the complete noisy view(s), for both $t = 1$ and $t = 2$. □

We will use the same language for both lower bounds. More generally, for any fixed finite set of constant noise rates, the parameters r and λ can be chosen so that the lemma holds simultaneously for all of them: choose r large enough for all rates, and then choose λ sufficiently large.

We will also use an extension of a standard observation about the simulation of interactive proofs. We provide only a proof sketch, since this part is not the novelty of our lower bound. The underlying simulation goes back to Goldreich and Håstad [GH98], and variants of the same idea are used in lower bounds for IPPs, including [KR15, GR17]. We stress that this claim is the only place in our lower bounds where we restrict the verifier's running time to be polynomial. Since we think of the verifier in an interactive proof as an efficient procedure, this is a mild restriction. Nevertheless, to separate the different conceptual components of the lower bound, we note that this restriction can be removed at the price of a slightly weaker lower bound and a non-uniform hardness assumption.[11]

Claim 3.2 (Noisy IPPs simulation)**.** *Fix a noisy* IPP *with a polynomial-time verifier and total communication complexity c. For a fixed clean input x and a fixed actual verifier noise rate, let $p(x)$ be the maximum acceptance probability over all cheating prover strategies. Then, for any constant $\nu > 0$, there is a randomized algorithm that, given x, approximates $p(x)$ to additive error ν with probability at least $2/3$ in time $2^{O(c)}\mathrm{poly}(n)$, independently of the query complexity.*

Proof Sketch. Sample $O(2^{O(c)} + n)$ verifier random tapes, where the random tape also includes the verifier's persistent noise mask, and consider the following residual game tree. Its optimal acceptance probability can be computed by evaluating the game tree, averaging at verifier nodes and maximizing at prover nodes. Since a c-bit interaction has at most $2^{2^{O(c)}}$ deterministic prover strategies, a Chernoff bound and a union bound imply that, with high probability, all of their empirical acceptance probabilities approximate their true acceptance probabilities to additive error ν. The game tree can be evaluated in time $2^{O(c)}\mathrm{poly}(n)$ because the verifier runs in polynomial time. By a constant number of independent repetitions, the success probability of the approximation can be amplified to $1 - \tau$ for any constant $\tau > 0$. □

Notice that prover noise causes no difficulty here. In soundness the cheating prover has clean access to x, and hence is at least as powerful as an honest noisy prover: it may sample an independent persistent noise mask and emulate the honest prover on the resulting noisy view. Thus, whenever completeness gives acceptance probability at least $2/3$ to an honest noisy prover, the corresponding value $p(x)$ is at least $2/3$ as well.

[11] In particular, under the corresponding assumption that the PRG is secure against non-uniform circuits, one can obtain $q(n) + \mathsf{cc}(n) = \Omega(n/\log n)$. Here the dependence on q appears because, for an unrestricted verifier, describing its strategy must account not only for the communication transcript but also for the sequence of its oracle queries and answers; a verifier with query complexity q and communication complexity cc can be approximated by circuits of size $\exp(O((q + \mathsf{cc}) \log n))$. The same pseudorandomness argument then gives the claimed bound.

3.2 A Noisy Prover

We first consider the model in which both parties are noisy but the exact noise rate is known (see Definition 2.5). For our hard language, increasing the verifier's query complexity does not resolve the difficulty: even if $\mathcal{V}$ reads its entire noisy input, Lemma 3.1 shows that replacing the PRG seed $h_a(w)$ by an independent uniform seed changes the joint distribution of the two complete noisy views only negligibly. Thus the lower bound continues to hold even for $q = n$. Communication, on the other hand, is fundamentally different: when the noise rate is sufficiently smaller than the proximity parameter, linear communication allows an unbounded prover to exhaustively search for a yes-instance consistent with its noisy view and send it in full, after which the verifier checks its consistency with its own noisy view.

Theorem 3.3 (Impossibility with a noisy prover)**.** *Assume the existence of exponentially hard pseudorandom generators computable in* NC^1*. Fix constants* $\varepsilon, \eta \in (0, 1/2)$*. Then there exists a language* $\mathcal{L} \in \mathsf{NC}^1$ *for which there is no* (ε, η)*-noisy* IPP *with known noise rate, a polynomial-time verifier, and communication complexity* $o(n)$*. This holds regardless of the query complexity, number of rounds, or the running time of the honest prover.*

The same proof applies when the prover and verifier have different noise rates $\eta^{\mathcal{P}}, \eta^{\mathcal{V}} \in (0, 1/2)$ choosing the parameter r in Lemma 3.1 accordingly. Moreover, note that the theorem also rules out the general upper-bound model of Definition 2.6.

Proof. Let $\mathcal{L}$ be the language of Lemma 3.1 with $\gamma = \eta$, and suppose toward contradiction that it has such a protocol with communication $c = o(n)$. For a fixed clean input z, let $p(z)$ denote the maximum acceptance probability at actual noise rate η.

Sample a, w, s independently and uniformly, and set $z = (a, G_k(s), w)$. Consider the cheating prover that, given the clean z, samples an independent persistent η-noise mask and runs the honest prover on the resulting noisy view. By Item 2 of Lemma 3.1 (where $\xi^{(1)}, \xi^{(2)}$ correspond to the verifier's noise mask and the cheating prover's sampled noise mask), the joint noisy views of this prover and the verifier on the input z are statistically indistinguishable, up to $2^{-\Omega(k)}$, from their joint views in an honest execution on the random language member $x = (a, G_k(h_a(w)), w)$. Completeness with respect to x therefore gives

$$\mathop{\mathbb{E}}_{a,w,s \leftarrow \mathsf{Unif}(\{0,1\}^{2m+3k-1})} [p(a, G_k(s), w)] = \mathbb{E}[p(z)] \geq \frac{2}{3} - 2^{-\Omega(k)}.$$

On the other hand, by Item 1 of Lemma 3.1, a uniformly random $z' \in \{0,1\}^n$ is ε-far from $\mathcal{L}$ except with probability $2^{-\Omega(n)}$. Soundness therefore gives

$$\mathop{\mathbb{E}}_{z' \leftarrow \mathsf{Unif}(\{0,1\}^n)} [p(z')] \leq 1/3 + 2^{-\Omega(n)}.$$

Thus, approximating p distinguishes the distribution $(a, G_k(s), w)$ from uniform with constant advantage. By Claim 3.2, this can be done in time $2^{O(c)}\mathrm{poly}(n) = 2^{o(n)} = 2^{o(k)}$, contradicting pseudorandomness of G_k, since a and w are independent uniform strings. □

3.3 A Clean Prover and an Unknown Noise Rate

We now turn to the complementary setting where the honest prover is clean but the verifier knows only an upper bound on its noise rate. The previous lower bound exploited the residual uncertainty

present in the two noisy views; with a clean prover, that source of hardness disappears. Here, instead, the obstruction is the verifier's inability to distinguish corruption already present in the underlying input from corruption introduced by its own channel. This phenomenon persists even if the verifier reads its entire noisy input: the ambiguity concerns the source of the corruption, not the number of observed coordinates. Thus allowing $q = n$ does not help.

Theorem 3.4 (Impossibility with an unknown verifier noise rate)**.** *Assume the existence of exponentially hard pseudorandom generators computable in* NC^1*. Fix constants* $\varepsilon, \eta_b \in (0, 1/2)$*. Then there exists a language* $\mathcal{L} \in \mathsf{NC}^1$ *for which there is no* (ε, η_b)*-noisy* IPP *with unknown noise rate in the verifier-only-noise model, with a polynomial-time verifier and communication complexity* $o(n)$*. This holds regardless of the query complexity, number of rounds, or the running time of the honest prover.*

The lower bound also holds in the more general fully noisy upper-bound model. The proof below only uses the honest prover through completeness at the larger rate η_1; if the honest prover is noisy, the cheating prover in the soundness experiment can generate an independent η_1-noisy view of the hardwired language member and emulate that honest prover. In fact, only the verifier's ignorance of its exact rate is essential: the lower bound continues to hold even if the honest prover is additionally told the actual verifier noise rate. Together with Theorem 3.3, their extensions give two alternatives for proving a lower bound for noisy IPPs as per Definition 2.6.

Proof. Set $\eta_0 = \eta_b/4$, $\eta_1 = \eta_b/2$, and $\theta = \eta_b/(4 - 2\eta_b)$. Then, $\eta_0, \eta_1 \in (0, \eta_b)$ and

$$\eta_0 + \theta - 2\eta_0\theta = \eta_1,$$

or, equivalently, note that the XOR of independent bits distributed as $\mathsf{Bernoulli}(\eta_0)$ and $\mathsf{Bernoulli}(\theta)$ is distributed as $\mathsf{Bernoulli}(\eta_1)$. Let $\mathcal{L}$ be the language of Lemma 3.1 with $\gamma = \theta$, and suppose toward contradiction that the claimed protocol exists with communication $c = o(n)$. For a clean input z, let $p_{\eta_0}(z)$ denote the maximum acceptance probability when the actual verifier noise rate is η_0.

Sample a random member $x = (a, G_k(h_a(w)), w) \in \mathcal{L}$, independently sample $\xi \leftarrow \mathsf{Bernoulli}(\theta)^n$, and let $z = x \oplus \xi$. The parameter θ is used only to define this distribution over clean inputs z; it is not an actual noise rate of the protocol.

Consider the cheating prover strategy that ignores its clean input z and emulates the honest clean prover on x. This is a valid strategy under the standard soundness quantifier: for each fixed input z, soundness must hold against every computationally unbounded prover strategy, and one such strategy may have the fixed string x hardwired into its description. In particular, we do not claim that x can be recovered from z.

When this cheating prover interacts with the verifier on clean input z and actual noise rate η_0, the verifier sees

$$z \oplus \xi^{\mathcal{V}} = x \oplus \xi \oplus \xi^{\mathcal{V}},$$

where $\xi^{\mathcal{V}} \leftarrow \mathsf{Bernoulli}(\eta_0)^n$. By the choice of θ, the mask $\xi \oplus \xi^{\mathcal{V}}$ consists of independent $\mathsf{Bernoulli}(\eta_1)$ bits. Moreover, the verifier is given only η_b, and therefore runs exactly the same algorithm as it does on the clean input x when the actual noise rate is η_1. The interaction on input z at rate η_0 is thus distributed exactly as a completeness execution on x at rate η_1, and hence

$$\mathop{\mathbb{E}}_{x,\xi}[p_{\eta_0}(x \oplus \xi)] = \mathbb{E}[p_{\eta_0}(z)] \geq \frac{2}{3}.$$

By Item 2 of Lemma 3.1, applied with $t = 1$ and $\gamma = \theta$, the distribution of $x \oplus \xi = (a, G_k(h_a(w)), w) \oplus \xi$ is within statistical distance $2^{-\Omega(k)}$ of $(a, G_k(s), w) \oplus \xi$, where $s \leftarrow \mathsf{Unif}(\{0,1\}^k)$ is independent. By the pseudorandomness of G_k, $(a, G_k(s), w)$ is computationally indistinguishable from uniform against $2^{o(n)}$-time algorithms, and this remains true after independently applying $\mathsf{Bernoulli}(\theta)$-noise. Since a noisy uniform string is still uniform, it follows that $x \oplus \xi$ is computationally indistinguishable from $\mathsf{Unif}(\{0,1\}^n)$ against $2^{o(n)}$-time algorithms.

However, Item 1 of Lemma 3.1 and soundness at actual rate η_0 imply that

$$\mathop{\mathbb{E}}_{z' \leftarrow \mathsf{Unif}(\{0,1\}^n)} [p_{\eta_0}(z')] \leq 1/3 + 2^{-\Omega(n)}.$$

Approximating p_{η_0} therefore distinguishes $x \oplus \xi$ with constant advantage. Since this can be done in time $2^{O(c)}\mathrm{poly}(n) = 2^{o(n)}$ by Claim 3.2, this contradicts the PRG's security. □

3.4 A Related Impossibility Result for Robust IPPs

In this section we show that robust IPPs for NC^1 cannot achieve constant robustness, under the same cryptographic assumption used throughout this section. We note that the IPP of [RR20] for NC, which attains an essentially optimal query–communication tradeoff (under the same cryptographic assumption and up to polylogarithmic factors), and also achieves optimal robustness as observed by [HR24], has sublinear communication only when the proximity parameter is subconstant. By Remark 2.3, this already implies that its robustness cannot be constant if one wants sublinear communication, which aligns with our lower bound. However, looking ahead, this subconstant robustness will still suffice to construct noisy IPPs (in a restricted setting) with constant noise in the next section.

The proof has two steps. First, Lemma 3.5 shows that a sufficiently efficient robust IPP with constant robustness can be recursively composed with itself, reducing the verifier's query complexity to a constant while keeping the total communication sublinear. The resulting protocol is therefore an ordinary IPP whose query–communication product is $o(n)$. Combining this with the lower bound of Kalai and Rothblum (Theorem 3.6), we derive Corollary 3.7, the impossibility of robust IPPs with constant robustness.

Lemma 3.5 (Robust IPP composition)**.** *Fix a language $\mathcal{L} \subseteq \{0,1\}^n$ in NC^1, a proximity parameter $\varepsilon \in (0,1]$, and constants $\delta \in (0,1)$ and $\rho \in (0, \varepsilon]$.*

Assume that there is a uniform construction that, given an input length m, a succinct NC^1 description of a language $\mathcal{L}' \subseteq \{0,1\}^m$, and a proximity parameter $\varepsilon' \in [\rho, 1]$, outputs an (ε', ρ)-robust IPP for $\mathcal{L}'$ with a succinct decision predicate, query complexity at most $m^{1-\delta}$, communication complexity $\mathsf{cc}(m) = o(m)$, constant completeness and robust-soundness error at most $c/\log m$ for some absolute constant $c > 0$.

Then there exists an ε-IPP for $\mathcal{L}$ with query complexity q' and communication complexity cc' such that $q' \cdot \mathsf{cc}' = o(n)$.

In fact, the composed protocol constructed in the proof is itself an (ε, ρ)-robust IPP; we state only the weaker conclusion of a standard IPP, which is all that is needed for the application to Theorem 3.6. Moreover, note that the restriction $\varepsilon' \geq \rho$ is necessary: by Remark 2.3, the robustness parameter cannot exceed the proximity parameter.

Since the proof is somewhat technical, and is not the emphasis of this section, we defer it to Section B.

Kalai and Rothblum [KR15] showed that (under a reasonable cryptographic assumption) there exists a language in NC^1 for which the product of the query and communication complexities cannot be sublinear.

Theorem 3.6 ([KR15])**.** *Assume the existence of exponentially hard pseudorandom generators computable in* NC^1 *and an error-correcting code with* NC^1 *encoding. Then there exists a constant* $\varepsilon > 0$ *such that for every* q*, there exists a language* $\mathcal{L} \in \mathsf{NC}^1$ *such that every* ε*-*IPP *for* $\mathcal{L}$ *(with any number of rounds) with a polynomial-time verifier, query complexity* q*, and communication complexity* cc*, satisfies* $q \cdot \mathsf{cc} = \Omega(n)$*.*

Kalai and Rothblum instantiate their impossibility result using Spielman's error-correcting codes [Spi95], and the constant ε is determined by the code's guaranteed adversarial error-correction radius. More generally, one may replace Spielman's code by any code family satisfying the properties required for the [KR15] lower bound, namely constant rate, an NC^1-computable encoding, and resilience to a constant fraction of adversarial errors.

Combining Lemma 3.5 and Theorem 3.6 yields the following corollary: there is no general theorem guaranteeing, for every language in NC^1, a robust IPP with polynomially shrinking query complexity, sublinear communication, and constant robustness. When we use the term "uniform construction" we mean exactly this, namely, an algorithm that given a succinct NC^1 description of $\mathcal{L}$ outputs a robust IPP with the stated complexity and error bounds. This negative result does not contradict [RVW13]'s positive result, who provide a uniform construction for NC, since the IPP in their construction does not provide constant robustness.

We stress that our negative result is specific to NC^1; we do not know whether sublinear IPPs with constant robustness may still exist for smaller complexity classes.

Corollary 3.7 (Formal version of Theorem 1.4)**.** *Assume the hypotheses of Theorem 3.6. Then there exists a constant* $\tau > 0$ *such that, for every fixed* $\delta > 0$ *and every constant robustness parameter* $\rho \in (0, \tau]$*, there is no uniform construction that, for every language* $\mathcal{L} \in \mathsf{NC}^1$ *and every proximity parameter* $\varepsilon \in [\rho, 1]$*, provides an* (ε, ρ)*-robust* IPP *with a succinct decision predicate that simultaneously satisfies query complexity* $q(n, \varepsilon) \leq n^{1-\delta}$*, communication complexity* $\mathsf{cc}(n) = o(n)$*, polynomial-time verifier, constant completeness and robust-soundness error* $O(1/\log n)$*, regardless of the number of rounds or the running time of the prover.*

4 Positive Results for Noisy IPPs

4.1 Robust IPPs Imply Noisy IPPs with a Clean Prover and Known Noise Rate

In the previous section we saw that, under the cryptographic assumption of [KR15], one cannot obtain sublinear-query, sublinear-communication robust IPPs with constant robustness for all of NC^1 uniformly. In this section we ask to what extent robust IPPs can nevertheless be used to construct noisy IPPs with constant noise. We consider the verifier-only-noise model, where the honest prover has clean access to the input and the verifier knows its exact noise rate, which is the easiest variant of a noisy IPP. We refer the reader to Definitions 2.2 and 2.5 for the definitions of robust and noisy IPPs.

The main observation is that knowing the exact noise rate allows the verifier to calibrate its consistency test to the expected amount of random noise. Consequently, the robustness parameter

need not be comparable to the noise rate: even subconstant robustness is enough, provided the verifier makes sufficiently many queries to detect a small shift in the disagreement rate.

Lemma 4.1 (Robust IPPs imply noisy IPPs with a known verifier noise rate)**.** *Fix a language $\mathcal{L} \subseteq \{0,1\}^n$, a proximity parameter $\varepsilon \in (0,1]$, a robustness parameter $\rho \in (0,1]$, and a fixed constant verifier noise rate $\eta^{\mathcal{V}} \in [0, 1/2)$. Suppose there exists an (ε, ρ)-robust IPP for $\mathcal{L}$ with q queries, completeness error α, and robust-soundness error β.*

Then there exists an $(\varepsilon, \eta^{\mathcal{V}})$-noisy IPP with known noise rate for $\mathcal{L}$ in the verifier-only-noise model. Its completeness and soundness errors are $\alpha + \exp(-\Omega(\rho^2 q))$ and $\beta + \exp(-\Omega(\rho^2 q))$, the query complexity remains q, the round complexity is increased by one, and the communication complexity and the running times of the verifier and the prover are increased by $O(q \log n)$.

In particular, the robustness parameter ρ may be subconstant even when the verifier's noise rate is a fixed constant, so long as $\rho^2 q = \Omega(1)$. For example, $\rho = n^{-1/3}$ together with $q = n^{2/3}$, or $\rho = 1/\log n$ together with $q = \log^2 n$, both lie in this regime.

Proof. Let $\langle \mathcal{P}^R, \mathcal{V}^R \rangle$ be the given robust IPP. We define the noisy IPP $\langle \mathcal{P}, \mathcal{V} \rangle$ as follows:

1. $\mathcal{P}$ and $\mathcal{V}$ execute the interaction phase of $\langle \mathcal{P}^R, \mathcal{V}^R \rangle$. Let τ denote the resulting transcript, $Q \subseteq [n]$ denote $\mathcal{V}^R$'s query set (of size q), and $\phi : \{0,1\}^q \to \{0,1\}$ denote its decision predicate.
2. $\mathcal{V}$ sends Q to $\mathcal{P}$.
3. $\mathcal{P}$ replies with the clean restriction $z := x|_Q$.
4. $\mathcal{V}$ queries its noisy oracle on all coordinates in Q, obtaining $y := x|_Q \oplus \xi^{\mathcal{V}}|_Q$. It accepts if and only if $\phi(z) = 1$ and
$$\Delta(y, z) \leq \eta^{\mathcal{V}} + \frac{1 - 2\eta^{\mathcal{V}}}{2}\rho.$$

The verifier's test is calibrated to the following calculation. Fix the transcript and the string z sent by the prover, and write $d = \Delta(x|_Q, z)$. On coordinates where z agrees with $x|_Q$, the noisy view disagrees with z with probability $\eta^{\mathcal{V}}$; on coordinates where they disagree, it disagrees with z with probability $1 - \eta^{\mathcal{V}}$. Since Q and z are fixed before the verifier reads its noisy oracle, these events are independent across queried coordinates. Therefore for any z at distance d it holds that

$$\mathbb{E}[\Delta(y, z) \mid Q, z] = \eta^{\mathcal{V}} + (1 - 2\eta^{\mathcal{V}})d. \tag{2}$$

Completeness. If $x \in \mathcal{L}$, the honest prover sends $z = x|_Q$, and hence the expectation in (2) is exactly $\eta^{\mathcal{V}}$. Thus the threshold in the verifier's test (in Step 4) exceeds the expectation by $(1 - 2\eta^{\mathcal{V}})\rho/2$. A Chernoff bound implies that the consistency test fails with probability $\exp(-\Omega(\rho^2 q))$. Together with the completeness error α of the robust protocol, this gives the claimed bound.

Soundness. Assume that x is ε-far from $\mathcal{L}$. Except with probability β, robust soundness guarantees that $\Delta(x|_Q, A) > \rho$, where $A := \{u \in \{0,1\}^q : \phi(u) = 1\}$. Condition on such a transcript. If a cheating prover sends any $z \in A$, then $d = \Delta(x|_Q, z) > \rho$, and hence by (2) the expected disagreement is larger than $\eta^{\mathcal{V}} + (1 - 2\eta^{\mathcal{V}})\rho$. The acceptance threshold is therefore below the expectation by at least $(1 - 2\eta^{\mathcal{V}})\rho/2$. Another Chernoff bound shows that the consistency test passes with probability at most $\exp(-\Omega(\rho^2 q))$, and adding the robust-soundness error β concludes the lemma. □

We now instantiate the transformation using the IPP of Rothblum and Rothblum [RR20], whose robustness was made explicit by Herman and Rothblum [HR24, Theorem 4.7]. This gives the following corollary. We state it for every fixed $\eta^{\mathcal{V}} < 1/2$.

Corollary 4.2 (Noisy IPPs for NC with a known verifier noise rate)**.** *Fix a constant proximity parameter $\varepsilon > 0$ and a constant verifier noise rate $\eta^{\mathcal{V}} < 1/2$. Then every language $\mathcal{L} \in \mathsf{NC}$ has an $(\varepsilon, \eta^{\mathcal{V}})$-noisy IPP with known noise rate in the verifier-only-noise model, with query complexity and communication complexity $\widetilde{O}(n^{2/3})$. The protocol has* polylog(n) *rounds, the verifier runs in time $\widetilde{O}(n^{2/3})$, and the honest prover runs in time* poly(n).

Proof. We run the protocol of [RR20] with proximity parameter $\delta = n^{-1/3}$. For sufficiently large n, we have $\delta < \varepsilon$, and therefore every input that is ε-far from $\mathcal{L}$ is also δ-far from $\mathcal{L}$. Thus, although the resulting robust IPP is defined for the smaller proximity parameter δ, its soundness guarantee applies in particular to all inputs on which soundness of the desired ε-noisy IPP is required.

By [HR24, Theorem 4.7], the resulting protocol has perfect completeness, robustness $\rho = \Theta(\delta)$, and communication complexity $\widetilde{O}(\delta n)$. Moreover, repeating only its query phase r times gives query complexity $\widetilde{O}(r/\delta)$ and robust-soundness error $\exp(-\Omega(r))$, without increasing the communication of the interaction phase.

To apply Lemma 4.1 with constant error, we need $\rho^2 q = \Omega(1)$. Since $\rho = \Theta(\delta)$ and $q = \widetilde{\Theta}(r/\delta)$, it suffices to take $r = \Theta(1/\delta)$, up to the polylogarithmic factors hidden in the notation. This gives

$$q = \widetilde{O}\left(\frac{1}{\delta^2}\right) = \widetilde{O}(n^{2/3}),$$

while the robust-soundness error is already $o(1)$. The constant in the choice of r can be taken sufficiently large, depending only on the fixed $\eta^{\mathcal{V}}$, so that the additional $\exp(-\Omega(\rho^2 q))$ error from Lemma 4.1 is also smaller than any desired constant.

It remains to account for communication. The communication of the robust IPP is $\widetilde{O}(\delta n) = \widetilde{O}(n^{2/3})$. The transformation of Lemma 4.1 additionally has the clean prover send the verifier its restriction to the q queried coordinates (and the verifier specify these queries), which costs $\widetilde{O}(q) = \widetilde{O}(n^{2/3})$ bits. Hence the total communication remains $\widetilde{O}(n^{2/3})$.[12] The round complexity and running-time bounds follow directly from [RR20] and Lemma 4.1. □

The corollary highlights a qualitative difference between robustness and random noise. For a known constant noise rate, the verifier can tolerate noise using only subconstant robustness: it suffices to increase the number of queries until the small shift in the expected disagreement rate becomes detectable. In contrast, Corollary 3.7 shows, under the cryptographic assumption of [KR15], that constant robustness cannot in general be achieved with both sublinear query and communication complexity even for all languages in NC^1. Thus, in the clean-prover setting with a known noise rate, noisy IPPs with constant noise rate are strictly easier to obtain than IPPs with constant robustness.

4.2 Noisy IPP for Hamming Weight

In this section we construct noisy IPPs for the Hamming Weight problem. We refer the reader to Section 1.2.2 for an overview. As stated there, the result extends verbatim to the more general

[12] The choice $\delta = n^{-1/3}$ balances the $\widetilde{O}(1/\delta^2)$ query complexity with the $\widetilde{O}(\delta n)$ communication complexity. More generally, taking $\delta = n^{-\alpha}$ gives $q = \widetilde{O}(n^{2\alpha})$ and $\mathsf{cc} = \widetilde{O}(n^{1-\alpha})$; e.g., $\alpha = 1/4$ gives $q = \widetilde{O}(\sqrt{n})$ and $\mathsf{cc} = \widetilde{O}(n^{3/4})$.

setting in which the prover and verifier have different noise rates, with corresponding upper bounds. For simplicity, throughout this section we state the result under a common upper bound η_b.

We begin by formally defining the search variant of the Hamming Weight problem. Although one could also consider its decision analogue, we find the search version to be the more natural and more general task.

Definition 4.3 (The Hamming Weight Problem (HW)). *For an input $x \in \{0,1\}^n$ and an accuracy parameter $\varepsilon \in (0,1]$, the search version of the Hamming Weight problem (denoted HW) asks to output a value $\sigma \in \{0,\ldots,n\}$ satisfying*

$$|\mathsf{wt}(x) - \sigma| \leq \varepsilon n.$$

Theorem 4.4 (Noisy IPP for HW). *For every function $r : \mathbb{N} \to \mathbb{N}$, every proximity parameter $\varepsilon \in (0,1]$, every error parameter $\delta \in (0,1/2]$, and every upper bound $\eta_b \leq c\varepsilon$ for some $c = c(\delta) < 1$, there exists an $r(n)$-round (ε, η_b)-noisy IPP with unknown noise rate for HW such that, for every input $x \in \{0,1\}^n$ and every actual noise rate $\eta \in [0, \eta_b]$, the verifier either rejects or outputs a value $\sigma \in \{0,\ldots,n\}$ such that the following hold:*

- Completeness: *After interacting with the honest prover $\mathcal{P}$, the verifier $\mathcal{V}$ outputs a value σ satisfying $|\mathsf{wt}(x) - \sigma| \leq \varepsilon n$ with probability at least $1 - \delta$.*
- Soundness: *For every unbounded cheating prover $\widetilde{\mathcal{P}}$ (that knows the clean input x and the actual noise rate η), after interacting with $\mathcal{V}$, the probability that $\mathcal{V}$ outputs a value σ satisfying $|\mathsf{wt}(x) - \sigma| > \varepsilon n$ is at most δ.*

The verifier makes $O\left(\frac{\log(1/\delta)}{\varepsilon}\right)$ queries, the honest prover runs in time $O(n)$, and the communication and verification time are $O\left(\frac{rn^{1/r}\log n \cdot \log(1/\delta)}{\varepsilon}\right)$.

In particular, setting $r(n) = \log n$ and $\delta = 1/3$ yields a protocol with communication and verification time $O\left(\log^2 n/\varepsilon\right)$ whenever $\eta_b \leq \varepsilon/(3 + 18\ln 6)$.

Proof. Let $r := r(n)$ and $k := n^{1/r}$, while assuming for simplicity that k is an integer; the general case only changes floors and ceilings. Let $T := \left\lceil \frac{2\ln(2/\delta)}{\varepsilon} \right\rceil$. The noisy IPP $\langle \mathcal{P}, \mathcal{V} \rangle$ for HW works as follows:

1. $\mathcal{P}$ sends a claimed weight $\sigma \in \{0,\ldots,n\}$.
2. $\mathcal{P}$ and $\mathcal{V}$ run T copies of the following r-round subprotocol in parallel. In each copy, the first block is $[n]$ and the first claim is σ. For each of the r rounds of a single copy:
   (a) Both parties partition the current block into k equal parts.
   (b) $\mathcal{P}$ sends the claimed weights of the k parts.
   (c) $\mathcal{V}$ checks that each claimed weight is an integer between 0 and the size of the corresponding part, and that the k claimed weights sum to the current claim. If not, it rejects.
   (d) $\mathcal{V}$ chooses one of the k parts uniformly at random (and independently of its choices in the other copies), sends its index to $\mathcal{P}$, and both parties continue to the next round with that block and its claimed weight.

3. At the end of each copy, $\mathcal{V}$ has a singleton block $\{i\}$ and a claimed weight $\widetilde{\sigma}_i \in \{0,1\}$. It queries the bit $x_i^{\mathcal{V}}$ and rejects if $\widetilde{\sigma}_i \neq x_i^{\mathcal{V}}$.

4. $\mathcal{V}$ outputs σ.

In completeness, we will roughly need $T \leq \frac{\delta}{2\eta_b(1-\eta_b)}$ while in soundness we will roughly need $T \geq \frac{\ln(2/\delta)}{\varepsilon}$. For this range to be non-empty, we let $c = c(\delta) := \frac{\delta}{1+6\ln(2/\delta)}$ and recall the assumption that $\eta_b \leq c\varepsilon$. The T defined above satisfies this inequality.

Completeness. The honest prover $\mathcal{P}$ uses its noisy copy $x^{\mathcal{P}} = x \oplus e^{\mathcal{P}}$, sends $\sigma = \mathsf{wt}(x^{\mathcal{P}})$, and in every round sends the exact weights of the k sub-blocks in $x^{\mathcal{P}}$.

First, since

$$|\sigma - \mathsf{wt}(x)| \leq \mathsf{wt}(e^{\mathcal{P}}),$$

Markov's inequality implies that

$$\Pr\left[|\sigma - \mathsf{wt}(x)| > \varepsilon n\right] \leq \Pr\left[\mathsf{wt}(e^{\mathcal{P}}) > \varepsilon n\right] \leq \frac{\eta}{\varepsilon} \leq \frac{\eta_b}{\varepsilon}.$$

Hence, it remains to show that $\mathcal{V}$ outputs σ with probability at least $1-\delta$.

In a single repetition, under the honest strategy $\mathcal{V}$ rejects if the final sampled coordinate i satisfies $x_i^{\mathcal{P}} \neq x_i^{\mathcal{V}}$. For every coordinate, this happens with probability $2\eta(1-\eta) \leq 2\eta_b$. By a union bound over the T repetitions, the completeness error is at most $\eta_b/\varepsilon + 2\eta_b T$. Since $\delta \leq 1/2$ and $\varepsilon \leq 1$, we have $\ln(2/\delta) \geq 1$ and therefore $T \leq \frac{3\ln(2/\delta)}{\varepsilon}$. Thus

$$\frac{\eta_b}{\varepsilon} + 2\eta_b T \leq \frac{(1+6\ln(2/\delta))\eta_b}{\varepsilon} \leq \delta$$

by the assumed bound on η_b. Therefore, with probability at least $1-\delta$, the verifier outputs a value σ satisfying $|\mathsf{wt}(x) - \sigma| \leq \varepsilon n$.

Soundness. Fix any prover strategy, and suppose that the cheating prover $\widetilde{\mathcal{P}}$'s top-level claim satisfies

$$|\sigma - \mathsf{wt}(x)| > \varepsilon n.$$

We show that in this case, $\mathcal{V}$ rejects with probability at least $1-\delta$.

Fix one repetition, and let $a_0 = \sigma$ and $b_0 = \mathsf{wt}(x)$. For every round $\ell \in \{0,\ldots,r-1\}$, let a_ℓ denote the prover's current claimed weight and let b_ℓ denote the true Hamming weight of the current block in the clean input x. Suppose that at round ℓ the current block is partitioned into k children, with prover claims $a_{\ell+1}^{(1)},\ldots,a_{\ell+1}^{(k)}$ and true clean weights $b_{\ell+1}^{(1)},\ldots,b_{\ell+1}^{(k)}$. If the consistency check fails then $\mathcal{V}$ rejects immediately, so we may assume it passes. Therefore

$$\sum_{j=1}^{k} a_{\ell+1}^{(j)} = a_\ell,$$

and trivially $\sum_{j=1}^{k} b_{\ell+1}^{(j)} = b_\ell$. Hence,

$$\frac{1}{k}\sum_{j=1}^{k}\left|a_{\ell+1}^{(j)} - b_{\ell+1}^{(j)}\right| \geq \frac{1}{k}\left|\sum_{j=1}^{k}\left(a_{\ell+1}^{(j)} - b_{\ell+1}^{(j)}\right)\right| = \frac{|a_\ell - b_\ell|}{k}.$$

Since $\mathcal{V}$ chooses the next child uniformly at random, the expected discrepancy after one more round is at least the current discrepancy divided by k. Iterating this over the r rounds, and recalling that $k^r = n$, shows that if $Y \in \{0,1\}$ denotes the indicator of the event that the prover's final claimed bit differs from the clean bit at the sampled coordinate, then

$$\mathbb{E}[Y] \geq \frac{|\sigma - \mathsf{wt}(x)|}{k^r} > \varepsilon.$$

In particular,

$$\Pr[Y = 1] > \varepsilon.$$

Conditioned on the event $Y = 1$, the verifier rejects unless its queried bit was corrupted, and hence rejects with probability $1 - \eta$. Conditioned on the complementary event $Y = 0$, the verifier rejects only if its queried bit was corrupted, which happens with probability η. Therefore, the rejection probability in a single repetition is at least

$$\varepsilon(1 - \eta) + (1 - \varepsilon)\eta = \eta + \varepsilon(1 - 2\eta).$$

Equivalently, the acceptance probability in a single repetition is at most $1 - \eta - \varepsilon(1 - 2\eta)$. Because the noise is persistent, the acceptance events in the T repetitions are not necessarily independent, so we cannot directly amplify the preceding one-repetition bound. Instead, first consider the execution with respect to the clean input. For repetition $j \in [T]$ and round ℓ, let

$$D_{\ell,j} := \frac{|a_{\ell,j} - b_{\ell,j}|}{n/k^\ell}$$

denote the normalized discrepancy in that repetition. Conditioned on the history before the verifier's choices at round ℓ, the choices in the different repetitions are independent, and the argument above gives $\mathbb{E}[D_{\ell+1,j} \mid \text{history}] \geq D_{\ell,j}$ for every j. Hence

$$\mathbb{E}\left[\prod_{j=1}^{T}(1 - D_{\ell+1,j}) \;\middle|\; \text{history}\right] = \prod_{j=1}^{T}\mathbb{E}[1 - D_{\ell+1,j} \mid \text{history}] \leq \prod_{j=1}^{T}(1 - D_{\ell,j}).$$

Iterating over the r rounds, and using that at the final singleton $D_{r,j} = Y_j \in \{0,1\}$, we obtain

$$\Pr[Y_1 = \cdots = Y_T = 0] \leq \left(1 - \frac{|\sigma - \mathsf{wt}(x)|}{n}\right)^T < (1 - \varepsilon)^T \leq \exp(-\varepsilon T) \leq \left(\frac{\delta}{2}\right)^2 \leq \frac{\delta}{2}.$$

It remains to account for the verifier's noise. If some $Y_j = 1$ but the verifier nevertheless accepts that repetition, then the verifier's noise must flip the final queried coordinate in repetition j. Since the verifier makes no oracle queries before these coordinates are fixed, each such coordinate is independent of the verifier's noise mask and is corrupted with probability η. Therefore, by a union bound over the T repetitions, the probability that any clean inconsistency is hidden by noise is at most

$$T\eta \leq T\eta_b \leq \frac{3\ln(2/\delta)}{\varepsilon}\eta_b \leq \frac{3\delta\ln(2/\delta)}{1 + 6\ln(2/\delta)} \leq \frac{\delta}{2}.$$

Consequently, the probability that $\mathcal{V}$ accepts an incorrect top-level claim is at most δ.

Complexities. At the end of each repetition, $\mathcal{V}$ queries exactly one bit of its noisy oracle, and therefore the total query complexity is $T = O\left(\frac{\log(1/\delta)}{\varepsilon}\right)$.

In each round of each repetition, $\mathcal{P}$ sends $k = n^{1/r}$ integers in $\{0, \ldots, n\}$, each using $O(\log n)$ bits. In addition, at the beginning of the protocol $\mathcal{P}$ sends the top-level claim σ, which also uses $O(\log n)$ bits. Therefore, the total communication is

$$O(\log n) + O\left(Trn^{1/r} \log n\right) = O\left(\frac{rn^{1/r} \log n \cdot \log(1/\delta)}{\varepsilon}\right).$$

The verifier's running time is dominated by checking the consistency constraints and sampling the random path in each repetition, and is therefore $O\left(\frac{rn^{1/r} \log n \cdot \log(1/\delta)}{\varepsilon}\right)$. The honest prover can compute the Hamming weights of all blocks in time $O(n)$. □

Remark 4.5 (Extending to the doubly-sublinear regime.)**.** *The protocol of Theorem 4.4 can also be made doubly-sublinear, in the sense of [AGR25]. The only place where the honest prover currently uses linear access to its noisy input is in computing the exact Hamming weights of the blocks encountered in the recursion. Following [AGR25], these exact weights can instead be replaced by estimates obtained from random samples of the corresponding blocks. This preserves the verifier's query complexity, while the honest prover makes $O\left(\frac{r^3 n^{1/r} \log n \cdot \log(1/\delta)}{\varepsilon^3}\right)$ queries to its noisy input. As in the standard noisy setting of Theorem 4.4, for constant δ and $r = \log n$, both the verifier and the honest prover have polylogarithmic query complexity.*

4.3 Noisy IPP for Permutations

In this section we construct noisy IPPs for the Permutation problem. We refer the reader to Section 1.2.2 for an overview. As stated there and in the previous section, this result extends to the more general setting in which the prover and verifier may have different noise rates, but for simplicity we state it under a common upper bound η_b.

For $n \in \mathbb{N}$, let PERM denote the set of all permutations $f : [n] \to [n]$.

Theorem 4.6 (Noisy IPP for PERM)**.** *For every proximity parameter $\varepsilon \in (0, 1]$ and every upper bound $\eta_b \leq \varepsilon/25$, there exists a one-round (ε, η_b)-noisy IPP for PERM in the following noisy function-oracle model.*

Let $f : [n] \to [n]$ be the input function, and let $\eta \in [0, \eta_b]$ be the actual noise rate. For each party $A \in \{\mathcal{V}, \mathcal{P}\}$, there is a random set $B^A \subseteq [n]$ obtained by including each point independently with probability η, where $B^{\mathcal{V}}$ and $B^{\mathcal{P}}$ are independent. The party A receives oracle access to a function $\widetilde{f}^A : [n] \to [n]$ such that

$$\widetilde{f}^A(i) = f(i) \qquad \text{for every } i \notin B^A,$$

while on the corrupted set B^A the values $\widetilde{f}^A(i)$ may be arbitrary, but are fixed for the entire protocol. Both parties are given only the upper bound η_b. The protocol satisfies the following:

- Completeness: *If $f \in$ PERM, then after interacting with the honest prover $\mathcal{P}$, the verifier $\mathcal{V}$ accepts with probability at least $2/3$.*
- Soundness: *If f is ε-far from PERM, then for every unbounded cheating prover $\widetilde{\mathcal{P}}$ (who knows the clean function f and the actual noise rate η), after interacting with $\mathcal{V}$, the verifier accepts with probability at most $1/3$.*

The verifier makes $O(1/(\varepsilon - \eta_b))$ queries, the total communication is $O(\log n/(\varepsilon - \eta_b))$, the verifier runs in time $O(1/(\varepsilon - \eta_b))$ and the honest prover runs in time $O(n + 1/(\varepsilon - \eta_b))$.

Proof. We use the natural noisy analogue of the one-round protocol of [GLR21], repeated in parallel. Fix a repetition parameter T to be determined below.

1. $\mathcal{V}$ samples $r_1, \ldots, r_T \sim [n]$ independently and uniformly at random and sends them to $\mathcal{P}$.
2. $\mathcal{P}$ sends back claimed preimages $(x_1, \ldots, x_T) \in [n]^T$.
3. $\mathcal{V}$ accepts if and only if for every $j \in [T]$, $\widetilde{f}^{\mathcal{V}}(x_j) = r_j$.

For the analysis, let

$$\beta_{\mathcal{V}} := \frac{|B^{\mathcal{V}}|}{n}, \qquad \beta_{\mathcal{P}} := \frac{|B^{\mathcal{P}}|}{n}.$$

Let $\beta' := \eta_b + \varepsilon/1000$. Since $|B^{\mathcal{V}}|, |B^{\mathcal{P}}| \sim \mathsf{Bin}(n, \eta)$ independently and $\eta \leq \eta_b$, a Chernoff bound implies that, with probability at least $1 - \exp(-\Omega(\varepsilon n))$, both $\beta_{\mathcal{V}} \leq \beta'$ and $\beta_{\mathcal{P}} \leq \beta'$. We condition on this event and at the end absorb the failure probability into the constant error.

Completeness. The honest prover preprocesses its noisy oracle $\widetilde{f}^{\mathcal{P}}$ by scanning all points in $[n]$ and, for every $y \in [n]$, storing one arbitrary preimage of y under $\widetilde{f}^{\mathcal{P}}$ if such a preimage exists. Upon receiving $(r_1, \ldots, r_T)$, it returns for each j the stored preimage of r_j if one exists, and an arbitrary point otherwise.

Assume that $f \in \mathsf{PERM}$, and fix one trial $j \in [T]$. Let $x_j^* := f^{-1}(r_j)$, which is well defined because f is a permutation. We claim that the verifier accepts the j^{th} trial whenever all of the following hold:

1. $x_j^* \notin B^{\mathcal{V}}$;
2. $x_j^* \notin B^{\mathcal{P}}$;
3. $r_j \notin \widetilde{f}^{\mathcal{P}}(B^{\mathcal{P}})$.

Indeed, under the second event, we have

$$\widetilde{f}^{\mathcal{P}}(x_j^*) = f(x_j^*) = r_j.$$

Also, since f is injective, every uncorrupted point $z \neq x_j^*$ satisfies

$$\widetilde{f}^{\mathcal{P}}(z) = f(z) \neq r_j.$$

The third condition rules out corrupted points whose value under $\widetilde{f}^{\mathcal{P}}$ equals r_j. Hence x_j^* is the unique preimage of r_j under $\widetilde{f}^{\mathcal{P}}$, so the honest prover returns x_j^*. By the first condition,

$$\widetilde{f}^{\mathcal{V}}(x_j^*) = f(x_j^*) = r_j,$$

and so the verifier accepts this trial. These three conditions hold with probability at least $1 - \beta_{\mathcal{V}} - \beta_{\mathcal{P}} - \frac{|\widetilde{f}^{\mathcal{P}}(B^{\mathcal{P}})|}{n}$, and since $|\widetilde{f}^{\mathcal{P}}(B^{\mathcal{P}})| \leq |B^{\mathcal{P}}| = \beta_{\mathcal{P}} n$, we get that $\mathcal{V}$ accepts in the j^{th} repetition with probability at least $1 - \beta_{\mathcal{V}} - 2\beta_{\mathcal{P}} \geq 1 - 3\beta'$ by our initial conditioning. Hence, the overall acceptance probability is at least $(1 - 3\beta')^T$.

Soundness. Assume that f is ε-far from PERM. As in the standard (noise-free) analysis, this implies $|\mathsf{Im}(f)| \le (1-\varepsilon)n$.

Fix one trial $j \in [T]$, the verifier's noisy oracle $\widetilde{f}^{\mathcal{V}}$, and consider an arbitrary cheating prover $\widetilde{\mathcal{P}}$. To make the verifier accept, $\widetilde{\mathcal{P}}$ must return some x_j such that $\widetilde{f}^{\mathcal{V}}(x_j) = r_j$. If $r_j \notin \mathsf{Im}(f)$, this can happen only if $x_j \in B^{\mathcal{V}}$ and the corrupted value at x_j equals r_j. Hence every target $r_j \notin \mathsf{Im}(f)$ that can possibly be accepted must lie in the set $\widetilde{f}^{\mathcal{V}}(B^{\mathcal{V}})$, whose size is at most $|B^{\mathcal{V}}| = \beta_{\mathcal{V}} n$. It follows that the set of all targets that can possibly be accepted has size at most

$$|\mathsf{Im}(f)| + |\widetilde{f}^{\mathcal{V}}(B^{\mathcal{V}})| \le (1-\varepsilon)n + \beta_{\mathcal{V}} n = (1-\varepsilon+\beta_{\mathcal{V}})n.$$

Therefore $\mathcal{V}$ accepts in the j^{th} repetition with probability at most $1-\varepsilon+\beta_{\mathcal{V}} \le 1-\varepsilon+\beta'$, and the overall acceptance probability is at most $(1-\varepsilon+\beta')^T$.

Complexities. This is a one-round protocol. The verifier makes T oracle queries and runs in time $O(T)$, the communication is $O(T \log n)$, and the honest prover runs in time $O(n+T)$. Choose $T := \lceil \frac{3}{2\varepsilon} \rceil$. Since $\beta' \le (1/25 + 1/1000)\varepsilon = 41\varepsilon/1000$, conditioned on the good-noise-size event,

$$(1-3\beta')^T \ge 1 - 3\beta' T \ge 1 - \frac{123}{1000}\left(\frac{3}{2} + \varepsilon\right) > \frac{2}{3},$$

whereas

$$(1-\varepsilon+\beta')^T \le \exp\bigl(-(\varepsilon-\beta')T\bigr) \le \exp\left(-\frac{959}{1000}\cdot\frac{3}{2}\right) < \frac{1}{3}.$$

The exponentially small probability of the conditioning failure is absorbed by the constant slack. Since $\eta_b \le \varepsilon/25$, we have $T = \Theta(1/\varepsilon) = \Theta(1/(\varepsilon - \eta_b))$, giving the claimed complexity bounds. □

5 Noisy PAC Verification

In this section we prove our main positive result for interactive proofs in the presence of noise. In Section 5.1, we define the Noisy PAC Verification model. In Section 5.2, we present two noisy variants of the query-to-samples transformation: one in which only the verifier receives noisy samples, and one in which both parties have noise. Next, in Section 5.3, we construct a PAC verification protocol for sparse Fourier functions, assuming the noise rate is sufficiently small relative to the target error parameter. Finally, in Section 5.4, we justify this assumption on the noise rate.

Parameter regime. Throughout this section, asymptotic $o(1)$ and negligible-error terms are with respect to $n \to \infty$ in the standard efficient regime, where t, $1/\varepsilon$, $1/\delta$, and $1/(1-2\eta)$ are polynomially bounded in n (and analogously with η_b in the upper-bound model). We only invoke the generic query-to-samples reductions when their resulting sample complexity is polynomial in n. In particular, repeated sampled points occur with negligible probability, and the size conditions required by the Fourier subroutines below hold for all sufficiently large n.

5.1 Definition

We start by recalling the notion of PAC verification from [GRSY21, GJK+24]. In [GRSY21] it is formulated in a general setting where there is an unknown distribution G over labeled samples

(x, y) that we wish to approximate, and a special case is when there exists a target function f such that $\Pr_{(x,y)\leftarrow G}[y = f(x)] = 1$. Some of the positive results in [GRSY21, GJK+24] apply in this fully general setting (the Thresholds result of [GRSY21]), while the others are for the more specific setting where the goal is to learn an underlying function f (such as the Junta and Fourier-sparse results). It is in this functional setting that we will introduce and analyze the effect of noise.

We now formally define a noisy variant of PAC verification. The only change is in the access model: the prover and verifier each see (possibly different) noisy views of the target function f, either through sample access or membership queries. We place no computational restrictions on the cheating prover; in particular, we allow it full access to the underlying (clean/noiseless) function f.

Definition 5.1 ($(\varepsilon, \delta, \eta)$-Noisy PAC verification)**.** *For parameters $\delta, \varepsilon > 0$ and $\eta \in [0, 1/2)$, a concept class $\mathcal{C}$ is* $(\varepsilon, \delta, \eta)$-Noisy PAC Verifiable *with hypothesis class $\mathcal{H}$ over a distribution $\mathcal{D}$ on $\{0,1\}^n$, if there exists an interactive protocol between a verifier $\mathcal{V}$ and a prover $\mathcal{P}$, taking explicit inputs $1^n, \varepsilon, \delta, \eta$ and noisy oracle access to a target function $f : \{0,1\}^n \to \{0,1\}$, such that*

- *the honest prover gets full access*[13] *to a noisy membership queries oracle f_η, where for every input x, $f_\eta(x) = f(x) \oplus b^{\mathcal{P}}(x)$ and $b^{\mathcal{P}}(x) \leftarrow \mathsf{Bernoulli}(\eta)$ is independent of x and independent across inputs x;*
- *the verifier gets noisy random samples $(x, y^{\mathcal{V}})$ with $x \leftarrow \mathcal{D}$ and $y^{\mathcal{V}} = f(x) \oplus b^{\mathcal{V}}(x)$, where $b^{\mathcal{V}}(x) \leftarrow \mathsf{Bernoulli}(\eta)$ and is independent of x and across inputs x, and also independent of all $\{b^{\mathcal{P}}(x)\}_x$.*

In both cases, the noise is persistent*: each input x is always paired with the same (possibly flipped) label. At the end of the protocol the verifier outputs either a hypothesis $h \in \mathcal{H}$ or $\perp$ (i.e.,* reject*), and the following hold for every f:*

1. ***Completeness.*** *There exists an honest prover $\mathcal{P}$ such that the output of the interaction satisfies*
$$\Pr\left[(h \neq \perp) \wedge (\Delta(f, h) \leq \Delta(f, \mathcal{C}) + \varepsilon)\right] \geq 1 - \delta.$$
2. ***Soundness.*** *For every unbounded cheating prover $\widetilde{\mathcal{P}}$ (that has access to the clean f), the output of the interaction satisfies*
$$\Pr\left[(h \neq \perp) \wedge (\Delta(f, h) > \Delta(f, \mathcal{C}) + \varepsilon)\right] \leq \delta.$$

The probability is over the randomness of the samples, the internal randomness of $\mathcal{P}$ and $\mathcal{V}$ and the randomness of the noise masks $\{b^{\mathcal{P}}\}$ and $\{b^{\mathcal{V}}\}$.

We say the protocol has sample complexity $q = q(n, 1/\varepsilon, 1/\delta, \eta)$ if $\mathcal{V}$ uses at most q noisy samples. The protocol has communication complexity $\mathsf{cc} = \mathsf{cc}(n, 1/\varepsilon, 1/\delta, \eta)$ if the total number of bits exchanged between $\mathcal{V}$ and $\mathcal{P}$ is at most cc. Its running times are $\mathcal{V}_{\mathsf{time}} = \mathcal{V}_{\mathsf{time}}(n, 1/\varepsilon, 1/\delta, \eta)$ for the verifier and $\mathcal{P}_{\mathsf{time}} = \mathcal{P}_{\mathsf{time}}(n, 1/\varepsilon, 1/\delta, \eta)$ for the honest prover.

All distances in this section are under $\mathcal{D}$, so we omit it in our notation of $\Delta(\cdot)$.

[13] By "full access" we mean that we do not account for its query complexity.

Remark 5.2. *Definition 5.1 is stated in the most general (improper) setting, with a* concept class $\mathcal{C}$ *and a* hypothesis class $\mathcal{H}$*. The output of the protocol, h, is required only to lie in $\mathcal{H}$, while its quality is measured with respect to $\mathcal{C}$, namely via the excess loss $\Delta(f,h) - \Delta(f,\mathcal{C})$.*

In our PAC verification protocol, we work with a single class of functions (Fourier-sparse functions) and say that this class is noisy PAC verifiable. We implicitly use it as the concept *class, and we do not restrict the prover's hypothesis beyond having small loss (compared to the best function in this class). Equivalently, one may take the hypothesis class to be the set of all Boolean functions, or all efficiently computable Boolean functions.*

To stay consistent with the notation in [GJK+24], we denote this concept class by $\mathcal{H}$, even though, conceptually, it plays the role of the concept class $\mathcal{C}$ in Definition 5.1. We stress that omitting an explicit hypothesis class does not *mean we are in the proper setting $\mathcal{H} = \mathcal{C}$; rather, it reflects that we allow improper hypotheses and do not restrict $\mathcal{H}$, while the performance benchmark is always the fixed concept class.*

A few additional remarks are in order. First, the original definitions in [GRSY21, GJK+24] consider a more general setting in which we allow a multiplicative slack parameter $\alpha \geq 1$ on top of the additive error ε, namely the distance of h from f is compared against $\alpha \cdot \Delta(f,\mathcal{C}) + \varepsilon$. Since in all our applications we take $\alpha = 1$, we omit this parameter from the definition and simply compare to $\Delta(f,\mathcal{C}) + \varepsilon$.

Second, as noted in [GRSY21, GJK+24], this model trivializes in the realizable case: the verifier can estimate the error of the hypothesis sent by the prover with respect to f using a small number of random samples, and accept if this estimate is at most ε. This holds even in the noisy setting, and will require the verifier to use $O(1/(1-2\eta)^2\varepsilon^2)$ samples instead of $O(1/\varepsilon^2)$. Hence, throughout this work, we restrict attention to the agnostic setting.

Finally, we only spell out the definition in the setting where the prover has query access and the verifier has sample access (in this regime the improvement is qualitative, and may also be quantitative), as this is the setting that will be useful for us. Variants in which both parties have samples or both have queries are also natural, and in such symmetric cases an interactive protocol is only interesting if it achieves a quantitative improvement in the verifier's query (or sample) complexity.

5.2 Query-to-Samples Reductions

The previous works on PAC verification ([GRSY21, GJK+24]) show how, under some conditions on the query distribution, a PAC-verification protocol that uses membership queries for the verifier can be converted into one where the verifier only receives random samples. Intuitively, the samples-only verifier embeds one random labeled sample inside the membership queries that the queries-verifier needs, and asks the prover to label all queries. It then checks that the embedded query is labeled correctly and rejects otherwise. Repeating this enough times allows the verifier to detect cheating with high probability, since the prover doesn't know the index of the embedded query. This was first fully formalized in [GJK+24, Section 3.2]. We recall their theorem.

Throughout this subsection, we consider proof-only protocols in the standard form in which the prover's proof π specifies a candidate output h_π, and the verifier either outputs h_π or rejects. In particular, all non-rejecting executions with the same proof π have the same output. This is the form used in our application to Fourier verification below.

Theorem 5.3 (Query-to-samples reduction [GJK+24, Theorem 3.2]). *Let $\mathcal{H} = \mathcal{H}_n$ be a hypothesis class and let $\mathcal{D}$ be a distribution over $\{0,1\}^n$. Suppose that $\mathcal{H}$ is (ε, δ)-PAC-verifiable*[14] *with respect to $\mathcal{D}$ by a proof-only (1-message) protocol $(\mathcal{P}^Q, \mathcal{V}^Q)$, in which the verifier uses $q = q(\varepsilon, \delta)$ membership queries, the prover has full access to the input, the communication complexity is $\mathsf{cc} = \mathsf{cc}(\varepsilon, \delta)$, and the running time of the verifier and the prover are $\mathcal{V}_{\mathsf{time}}$ and $\mathcal{P}_{\mathsf{time}}$, respectively.*

Assume further that (1) Each membership query made by $\mathcal{V}^Q$ has marginal distribution $\mathcal{D}$. (2) For every $x \in \{0,1\}^n$, it is possible to efficiently construct a membership-query set Q *for $\mathcal{V}^Q$ conditioned on the event that a uniformly random query index equals x.*[15]

Then, $\mathcal{H}$ is (ε, δ)-PAC-verifiable with respect to $\mathcal{D}$ by a 3-message protocol $(\mathcal{P}^S, \mathcal{V}^S)$ in which the verifier uses $\Theta\left(q \cdot \log(1/\delta)\right)$ random samples, the prover has full access to the input, the communication complexity is $\Theta\left(\mathsf{cc} + n \cdot q^2 \cdot \log(1/\delta)\right)$, and the running time of the verifier and the prover are $(\mathcal{V}_{\mathsf{time}} + T_{\mathrm{query}}) \cdot \Theta\left(q \cdot \log(1/\delta)\right)$ and $\mathcal{P}_{\mathsf{time}} + \Theta\left(n \cdot q^2 \cdot \log(1/\delta)\right)$, respectively, where T_{query} is the time needed to construct a membership-query set.

Proof sketch. The transformed protocol $\left(\mathcal{P}^S, \mathcal{V}^S\right)$ works as follows. The prover first sends a proof π (the same proof as in $(\mathcal{P}^Q, \mathcal{V}^Q)$). The verifier then draws

$$t = \Theta\left(q \cdot \log(1/\delta)\right)$$

independent random samples $(x_i, f(x_i))$. For each x_i it uses the membership-set constructor to build a query set Q_i distributed exactly as the membership queries of $\mathcal{V}^Q$, conditioned on embedding x_i at a uniformly random position. It sends all unlabeled sets $\{Q_i\}_{i \in [t]}$ to the prover, who returns labelings $\tilde{f}(Q_i)$. For each i, the verifier checks that $\tilde{f}(x_i) = f(x_i)$; if the check fails it rejects immediately. Otherwise, it runs $\mathcal{V}^Q$ on the proof π with the $\tilde{f}(Q_i)$ as the query answers (i.e., uses $\tilde{f}$ as the membership query oracle), obtaining an output h_i. Finally, it rejects if more than half of the t executions reject; otherwise it outputs h_i from an arbitrary accepting execution. The analysis shows completeness error 2δ which can be reduced to δ (by taking slightly larger constants), and soundness error δ as stated. □

Next, we consider a more general setting in which the verifier's random samples are corrupted by noise. The verifier no longer has access to clean labeled samples; instead it only sees labels flipped independently with some known rate $\eta < 1/2$. As in Theorem 5.3, we allow the prover (both the honest and the cheating one) to have full noiseless membership query access to the input f. Our goal is to adapt the query-to-samples reduction so that it remains complete and sound even when the verifier's sample oracle is noisy.

In the noisy setting, the verifier can only compare the prover's label $\tilde{f}(x_i)$ to a *possibly flipped* label, and therefore a single mismatch is no longer a definitive certificate of cheating. Instead, we interpret the plant-and-check step as a sequence of biased coin flips, where the bias depends on the frequency with which the prover mislabels the planted queries. To compensate for this loss of a "single-mismatch" certificate, our reduction repeats the check $\Theta(q^2)$ times, incurring a quadratic overhead in the query complexity compared to the noiseless protocol. See Remark 5.6 for discussion of why we believe this quadratic blow-up is inherent.

From this point on, we restrict attention to the uniform distribution $\mathcal{D} = \mathsf{Unif}(\{0,1\}^n)$. The reason is that the concentration argument for the above biased-coin process requires the mismatch

[14] Formally, this means that $\mathcal{H}$ is $(\varepsilon, \delta, \eta)$-noisy PAC verifiable in the sense of Definition 5.1 with $\eta = 0$, which coincides with the standard noiseless PAC verification setting of [GRSY21, GJK+24].

[15] See [GJK+24, Appendix A] for a formalization of this notion.

indicators to be (essentially) independent. Under *persistent* noise, however, this independence can fail if the verifier samples the same point x more than once, since repeated occurrences of x are paired with the same (possibly flipped) label. When $\mathcal{D} = \mathsf{Unif}(\{0,1\}^n)$, a standard collision bound implies that, for our parameter settings, all sampled points are distinct with overwhelming probability, and hence the mismatch indicators behave as independent coin flips. Since we only apply the resulting theorems in the uniform case, we adopt this assumption to simplify the presentation.

Theorem 5.4 (Query-to-samples reduction with verifier noise)**.** *Let $\mathcal{H}$, $\mathcal{D}$, $(\mathcal{P}^Q, \mathcal{V}^Q)$ and T_{query} be the same as in Theorem 5.3, and assume they satisfy the same assumptions. Fix a noise rate $\eta \in [0, 1/2)$.*

Then, $\mathcal{H}$ is $(\varepsilon, \delta, \eta)$-noisy PAC verifiable with verifier noise only[16] *with respect to $\mathcal{D}$ by a 3-message protocol $(\mathcal{P}^S, \mathcal{V}^{S,\eta})$, in which the verifier uses $\Theta\left(q^2 \cdot (1-2\eta)^{-2} \cdot \log(1/\delta)\right)$ random noisy samples, the communication complexity is $\Theta\left(\mathsf{cc} + n \cdot q^3 \cdot (1-2\eta)^{-2} \cdot \log(1/\delta)\right)$, and the running time of the verifier and the prover are $(\mathcal{V}_{\mathsf{time}} + T_{\text{query}}) \cdot \Theta\left(q^2 \cdot (1-2\eta)^{-2} \cdot \log(1/\delta)\right)$ and $\mathcal{P}_{\mathsf{time}} + \Theta\left(n \cdot q^3 \cdot (1-2\eta)^{-2} \cdot \log(1/\delta)\right)$, respectively.*

Remark 5.5. *All of the query-to-samples reductions (both the noisy and the noiseless cases) are stated for boolean functions $f : \{0,1\}^n \to \{0,1\}$, but the distinction between range $\{0,1\}$ and $\{-1,1\}$ is immaterial for the analysis and complexities. Indeed, the reduction only ever uses equality tests between labels and the induced bias in these events. All expectations that drive the Chernoff bounds are the same, and the dependence on the parameters q, δ, η (and in particular the $(1-2\eta)$ factors) is unchanged. Thus we may state it in either convention and apply it to the Fourier-sparse protocol (where $f : \{0,1\}^n \to \{-1,1\}$) without affecting the complexities.*

Proof. The protocol $(\mathcal{P}^S, \mathcal{V}^{S,\eta})$ is defined as follows.

Input. A function $f : \{0,1\}^n \to \{0,1\}$, parameters $\varepsilon, \delta > 0$, noise rate $\eta \in [0, 1/2)$ and a proof-only (1-message) protocol $(\mathcal{P}^Q, \mathcal{V}^Q)$ with membership queries.

1. $\mathcal{P}^S$ sends a proof string π (the same proof used in $(\mathcal{P}^Q, \mathcal{V}^Q)$).
2. $\mathcal{V}^{S,\eta}$ takes $t = \Theta\left(q^2 \cdot (1-2\eta)^{-2} \cdot \log(1/\delta)\right)$ random noisy samples $(x_i, y_i)_{i \in [t]}$. For each x_i, it constructs a membership query set Q_i for $\mathcal{V}^Q$ (distributed as in Theorem 5.3, conditioned on embedding x_i at a uniformly random position).
3. $\mathcal{V}^{S,\eta}$ sends all unlabeled query sets $\{Q_i\}_{i \in [t]}$ to the prover.
4. $\mathcal{P}^S$ responds with labelings $\tilde{f}(Q_i) = \{(x, \tilde{f}(x)) : x \in Q_i\}$ for all $i \in [t]$.
5. For each $i \in [t]$:
   (a) $\mathcal{V}^{S,\eta}$ defines a mismatch indicator $Z_i \in \{0,1\}$ by $Z_i := \mathbf{1}\left[\tilde{f}(x_i) \neq y_i\right]$,
   (b) $\mathcal{V}^{S,\eta}$ simulates $\mathcal{V}^Q$ with proof π and queries given by $\tilde{f}(Q_i)$, obtaining an output $h_i \in \mathcal{H} \cup \{\perp\}$. Note that π remains fixed across all iterations.
6. Let $Z := \sum_{i=1}^{t} Z_i$ denote the total number of mismatches.

[16] Formally, Definition 5.1 is stated for the setting in which both parties observe noisy labels. In this theorem we consider the intermediate model where only the verifier sees noisy labels while the prover has access to the clean input; equivalently, in the notation of Definition 5.1 we replace f_η by f on the prover side.

(a) If Z exceeds a threshold $T_Z := (\eta + \gamma) \cdot t$ for $\gamma = (1 - 2\eta)/20q$, then $\mathcal{V}^{S,\eta}$ rejects.

(b) Otherwise, if $\mathcal{V}^Q$ rejects in more than $t/2$ of the iterations, $\mathcal{V}^{S,\eta}$ rejects.

Otherwise, it outputs h_i of an arbitrary non-rejecting iteration.

Proof overview. Intuitively, under an honest prover we have $\tilde{f} = f$, so the mismatch indicators Z_i behave like i.i.d. $\mathsf{Bernoulli}(\eta)$ bits and so the first test (Step 6a) passes, and then, the total number of rejections in the second test (Step 6b) is dominated by the completeness error δ of $(\mathcal{P}^Q, \mathcal{V}^Q)$. Under a cheating prover, either many of the sets Q_i are mislabeled, which forces the mismatch rate above η by a noticeable amount (and the first test will fail), or most Q_i's are faithful and then the original soundness analysis for $(\mathcal{P}^Q, \mathcal{V}^Q)$ applies, forcing at least half of the iterations to reject.

We now analyze completeness and soundness. Throughout we fix f and consider all probabilities over the randomness of $\mathcal{V}^{S,\eta}$, $\mathcal{P}^S$, and the noisy oracle (i.e., the flips that $\mathcal{V}^{S,\eta}$ sees in its noisy samples).

For each $i \in [t]$, let

$$f(Q_i) := \{(x, f(x)) : x \in Q_i\}$$

denote the correct labeling of the i^{th} query set, and define the set of *bad* indices

$$B := \{i \in [t] : \tilde{f}(Q_i) \neq f(Q_i)\}.$$

Let $t_{\text{bad}} := |B|$.

Completeness. Assume f satisfies the completeness condition of $(\mathcal{P}^Q, \mathcal{V}^Q)$, and that P^S acts honestly in the protocol, i.e., it uses the honest proof π and responds with $\tilde{f} = f$ on all queries.

In this case, every query set is faithful, so $t_{\text{bad}} = 0$, and for each i we have

$$Z_i = \mathbf{1}[f(x_i) \neq y_i] = \mathbf{1}[b_i = 1],$$

where $b_i \leftarrow \mathsf{Bernoulli}(\eta)$. Thus $\{Z_i\}$ are i.i.d. $\mathsf{Bernoulli}(\eta)$ random variables, independent of the internal randomness of $\mathcal{V}^Q$, and (conditioned on the event that all planted points are distinct) mutually independent. Since this distinctness event holds with all but negligible probability under our parameter settings, we omit its failure probability in the theorem statement. Their sum is distributed $Z \leftarrow \mathsf{Bin}(t, \eta)$, therefore its expected value is $\mathbb{E}[Z] = \eta t$ and its variance is $\mathsf{Var}(Z) = t\eta(1-\eta) \leq t/4$. Recall that the first test of $\mathcal{V}^{S,\eta}$ is whether Z exceeds its expectation by more than $\gamma = \frac{1-2\eta}{20q}$. By a Chernoff bound,

$$\Pr[Z > T_Z] = \Pr[Z > (\eta + \gamma)t] \leq \exp\left(-\Omega(\gamma^2 t)\right) = \exp\left(-\Omega\left(\frac{t \cdot (1-2\eta)^2}{(20q)^2}\right)\right) \leq \frac{\delta}{2},$$

where the last inequality follows by our choice of $t = \Theta\left(q^2 \cdot (1-2\eta)^{-2} \cdot \log(1/\delta)\right)$ with appropriate constants.

The rest of the completeness analysis follows the same lines of the noiseless case (as presented in [GJK+24]). Conditioned on the event $Z \leq T_Z$, every iteration uses exactly the same query distribution as $\mathcal{V}^Q$ (because the membership-set constructor preserves the distribution of queries), and the prover supplies correct labels. Thus each individual simulation of $(\mathcal{P}^Q, \mathcal{V}^Q)$ rejects with

probability at most δ. Let A be the number of accepting iterations, then $\mathbb{E}[A] \geq (1-\delta)t$. For $\delta < 1/3$ and t large enough (any $t > 18\log(2/\delta)$) we have

$$\Pr\left[A < t/2\right] \leq \delta/2,$$

by another application of Chernoff (we can take t large enough to handle both bounds simultaneously; note that this only affects constants).

Putting the two failure events together and using a union bound we obtain that the probability that $\mathcal{V}^{S,\eta}$ rejects or outputs a hypothesis which is ε-far from optimal is at most δ.

Soundness. Fix a cheating prover strategy. As in the original analysis, we consider two cases depending on t_{bad}.

Case 1: Many dishonest query sets, $t_{\text{bad}} \geq t/10$.

For each bad index $i \in B$, the prover mislabels at least one point in Q_i. Let $\ell_i \geq 1$ be the number of mislabeled points in Q_i. Recall that in the membership-set construction the position of the planted sample x_i within Q_i is a uniformly random index $j_i \in [q]$ independent of any other randomness. Conditioned on Q_i and the prover's labeling $\tilde{f}(Q_i)$, the probability (over the noise bit b_i and the random choice of j_i) that the mismatch indicator Z_i equals 1 is

$$\begin{aligned}
\Pr[Z_i = 1] &= \Pr[\tilde{f}(x_i) \neq y_i] \\
&= \Pr[\tilde{f}(x_i) \neq f(x_i) \oplus b_i] \\
&= \Pr[\tilde{f}(x_i) \neq f(x_i)] \cdot (1-\eta) + \Pr[\tilde{f}(x_i) = f(x_i)] \cdot \eta \\
&= \frac{\ell_i}{q} \cdot (1-\eta) + \left(1 - \frac{\ell_i}{q}\right) \cdot \eta \\
&= \eta + \frac{\ell_i}{q} \cdot (1-2\eta) \\
&\geq \eta + \frac{1-2\eta}{q}.
\end{aligned}$$

Thus for every $i \in B$,

$$\mathbb{E}[Z_i] \geq \eta + \frac{1-2\eta}{q}.$$

On the other hand, for a faithful query set ($i \notin B$) we have $\mathbb{E}[Z_i] = \eta$. Hence, the expected total number of mismatches $Z = \sum_i Z_i$ is

$$\mathbb{E}[Z] \geq t_{\text{bad}} \cdot \left(\eta + \frac{1-2\eta}{q}\right) + (t - t_{\text{bad}}) \cdot \eta = \eta t + t_{\text{bad}} \cdot \frac{1-2\eta}{q} \geq \eta t + \frac{1-2\eta}{q} \cdot \frac{t}{10},$$

using the assumption that $t_{\text{bad}} \geq t/10$ in the last inequality. Recall the threshold $T_Z = (\eta+\gamma) \cdot t = (\eta + \frac{1-2\eta}{20q}) \cdot t$. Conditioned on the query sets and the prover's labelings, the variables Z_i are independent and bounded in $[0,1]$, and hence a Chernoff bound gives

$$\Pr[Z \leq T_Z] \leq \exp\left(-\left(\frac{1-2\eta}{20q}\right)^2 t\right) \leq \delta/2,$$

by the choice of t.

Thus, in Case 1, the verifier rejects in the first test with probability at least $1 - \delta/2$.

Case 2: Few dishonest query sets, $t_{\text{bad}} < t/10$. This is the same argument as in the noiseless case. Consider the t executions in which every Q_i is labeled correctly according to f. These are independent executions of $(\mathcal{P}^Q, \mathcal{V}^Q)$ with the fixed proof π, so by soundness and a Chernoff bound, with probability at least $1 - \delta/2$, at least $3t/5$ of them reject. Since $t_{\text{bad}} < t/10$, the actual executions differ from these executions on fewer than $t/10$ indices. Thus, with probability at least $1 - \delta/2$, more than $3t/5 - t/10 = t/2$ actual executions reject. In particular, with probability at least $1 - \delta/2$, the verifier rejects in the second test.

Finally, combining the two cases, we obtain that the verifier rejects with probability at least $1 - \delta$. □

Remark 5.6 (The quadratic blow-up in q)**.** *In the noiseless reduction of Theorem 5.3, a single planted sample behaves as a* one-sided *test for cheating: whenever the prover mislabels any point in Q_i, the chance that the planted query is the mislabeled one is at least $1/q$, and this event certifies cheating with certainty. Thus, if there are $t_{\text{bad}} \geq q \log(1/\delta)$ dishonest query sets, the probability that none of them triggers the test is at most $(1 - 1/q)^{t_{\text{bad}}} \leq \exp(-t_{\text{bad}}/q) \leq \delta$. This leads to a sample complexity $t = \Theta(q \log(1/\delta))$ as stated in Theorem 5.3.*

With noisy samples, the same check becomes a biased coin *rather than a one-sided test. For a faithful query set, we have $\Pr[Z_i = 1] = \eta$. For a query set where the prover mislabels exactly one point, the probability that the mismatch test applies increases only to $\Pr[Z_i = 1] = \eta + \frac{1-2\eta}{q}$, an additive gap of $\Delta = (1-2\eta)/q$. Thus, the planted sample only distinguishes faithful from mislabeled query sets with a bias Δ. Distinguishing a $\mathsf{Bernoulli}(\eta)$ coin from a $\mathsf{Bernoulli}(\eta+\Delta)$ coin with error probability at most δ requires $\Omega(\Delta^{-2} \log(1/\delta)) = \Omega\left(\frac{q^2}{(1-2\eta)^2} \log(1/\delta)\right)$ independent samples. This is why we find the quadratic blow-up in q as inherent for this plant-and-check approach.*

Next, we move to the most general setting of this reduction, in which both the verifier and the prover experience noise (see Definition 5.1 for the formal definition). In this setting, the verifier cannot hope to reconstruct query access to the *clean* function f from noisy samples on its side and noisy membership queries on the prover's side. At best, it can implement query access to the prover's noisy oracle f_η. Accordingly, the reduction is from *noisy* queries to noisy samples: given noisy samples from f and interaction with a prover that has noisy membership query access to f_η, the verifier gains simulated noisy membership query access where the noise is distributed exactly as in the prover's oracle.

In particular, this means that the reduction is only useful for protocols $(\mathcal{P}^Q, \mathcal{V}^Q)$ which remain complete even when the verifier's query access is only to a *noisy* oracle rather than to the clean function (where the noise is random), and remain sound even when the verifier's query access is to an *adversarially-corrupted* oracle, since we recall that, in soundness, the cheating prover has access to the clean function f and may choose to adaptively flip labels wherever it chooses. We formalize this notion in the next definition.

Definition 5.7 (Adversarially corrupted query oracle)**.** *Let $f : \{0,1\}^n \to \{-1,1\}$ be a Boolean function and let $\eta \in [0, 1/2)$. An* η-adversarially corrupted query oracle *for f is an interactive oracle, denoted by $\tilde{f}$, that answers a (finite) sequence of queries $x_1, x_2, \ldots \in \{0,1\}^n$ with labels $\tilde{f}(x_1), \tilde{f}(x_2), \ldots \in \{-1,1\}$, possibly* adaptively *as a function of the queried points and their order. For any randomized query algorithm $\mathcal{A}$ that makes q queries, let*

$$Q := (x_1, \ldots, x_q)$$

denote the (multi)set of query points asked by $\mathcal{A}$, and define the number of corrupted answers by

$$\mathrm{err}(Q, \tilde{f}, f) := \left|\left\{i \in [q] : \tilde{f}(x_i) \neq f(x_i)\right\}\right|.$$

We say that $\tilde{f}$ is η-adversarially corrupted with respect to $\mathcal{A}$ *if for every execution,*

$$\mathrm{err}(Q, \tilde{f}, f) \leq 4\eta q + 1.$$

The constant 4 is chosen for the application below; for smaller soundness error, this factor can be reduced, approaching 2 in the limit.

In particular, the adversarial noise rate is measured only relative to the query set produced by $\mathcal{A}$ (and not relative to $|\{0,1\}^n|$), and $\tilde{f}$ need not be defined on points that are never queried.

For the non-adaptive query algorithms considered below, we allow $\widetilde{f}$ to choose its answers after seeing the entire realized query sequence Q. Moreover, note that the same definition holds in case the range of f is $\{0,1\}$. We now present the reduction in the case where both parties have noise.

Theorem 5.8 (Query-to-samples reduction with verifier and prover noise)**.** *Let $\mathcal{H}$, $\mathcal{D}$, $(\mathcal{P}^Q, \mathcal{V}^Q)$ and T_{query} be the same as in Theorem 5.3, and assume they satisfy the same assumptions. Fix a noise rate $\eta \in [0, 1/2)$, and assume further that*

- *the completeness error of $\mathcal{V}^Q$ is at most $\delta/8 + o(1)$ even when $\mathcal{V}^Q$ only has access to $\mathcal{P}^Q$'s noisy membership queries oracle f_η as per Definition 5.1;*
- *the soundness error of $\mathcal{V}^Q$ is at most $\delta/8$ even when $\mathcal{V}^Q$ only has access to an η-adversarially-corrupted queries oracle $\tilde{f}$ as per Definition 5.7.*

Then, $\mathcal{H}$ is $(\varepsilon, \delta, \eta)$-Noisy PAC verifiable with respect to $\mathcal{D}$ by a 3-message protocol $(\mathcal{P}^{S,\eta}, \mathcal{V}^{S,\eta})$ in which the verifier uses $\Theta\left(q^2 \cdot (1-2\eta)^{-2} \cdot \log(1/\delta)\right)$ random noisy samples, the communication complexity is $\Theta\left(\mathsf{cc} + n \cdot q^3 \cdot (1-2\eta)^{-2} \cdot \log(1/\delta)\right)$, and the running time of the verifier and the prover are $(\mathcal{V}_{\mathsf{time}} + T_{\mathrm{query}}) \cdot \Theta\left(q^2 \cdot (1-2\eta)^{-2} \cdot \log(1/\delta)\right)$ and $\mathcal{P}_{\mathsf{time}} + \Theta\left(n \cdot q^3 \cdot (1-2\eta)^{-2} \cdot \log(1/\delta)\right)$, respectively.

$\mathcal{V}^Q$ may also use noisy samples, and these will be added as is to its overall sample complexity.

Note that the last sentence is not redundant. In the completeness case, the verifier can emulate noisy sample access by querying the noisy oracle f_η on uniformly random inputs. In the soundness case, however, the verifier only has access to an adversarially corrupted oracle $\tilde{f}$, which need not correspond to noisy sample access. Luckily, providing samples is trivial: $\mathcal{V}^{S,\eta}$ draws the required noisy samples directly and feeds them to $\mathcal{V}^Q$.

Proof. The protocol $(\mathcal{P}^{S,\eta}, \mathcal{V}^{S,\eta})$ is identical to the one in the verifier-only-noise variant of Theorem 5.4, except that now the honest prover labels queries according to its noisy oracle f_η. The only changes are in the soundness analysis and in the choice of the threshold parameter T_Z, which we take to be

$$T_Z := (2\eta(1-\eta) + \gamma) \cdot t,$$

where $\gamma = (1-2\eta)/20q$ (the same as in Theorem 5.4). For a fixed iteration $i \in [t]$, let Z_i be the mismatch indicator as defined in the protocol, namely

$$Z_i := \mathbf{1}\left[\tilde{f}(x_i) \neq y_i\right],$$

where (x_i, y_i) is the i^{th} noisy sample seen by the verifier and $\tilde{f}(x_i)$ is the label returned by the prover on the planted point. Recall that $y_i = f(x_i) \oplus b^{\mathcal{V}}(x_i)$ where $b^{\mathcal{V}}(x_i) \sim \mathsf{Bernoulli}(\eta)$, and that the prover's noisy oracle is $f_\eta(x) = f(x) \oplus b^{\mathcal{P}}(x)$ with $b^{\mathcal{P}}(x) \sim \mathsf{Bernoulli}(\eta)$, where the two noise masks are independent as per Definition 5.1.

Completeness. If $\mathcal{P}^{S,\eta}$ is honest, then $\tilde{f} = f_\eta$, and for each iteration i we have

$$\Pr[Z_i = 1] = \Pr[f_\eta(x_i) \neq y_i] = \Pr[b^{\mathcal{P}}(x_i) \neq b^{\mathcal{V}}(x_i)] = 2\eta(1-\eta).$$

Moreover, with probability $1 - o(1)$ (over the verifier's samples), the planted points $\{x_i\}_{i\in[t]}$ are all distinct, in which case the corresponding noise bits $\{b^{\mathcal{V}}(x_i), b^{\mathcal{P}}(x_i)\}_{i\in[t]}$ are independent. Conditioning on this event, $\{Z_i\}_{i\in[t]}$ are i.i.d. $\mathsf{Bernoulli}(2\eta(1-\eta))$ random variables in $[0,1]$, and hence

$$Z := \sum_{i=1}^{t} Z_i \sim \mathsf{Bin}\left(t, 2\eta(1-\eta)\right), \qquad \mathbb{E}[Z] = 2\eta(1-\eta)\, t.$$

By a Chernoff bound,

$$\Pr\left[Z > T_Z\right] \leq \exp\left(-\Omega(\gamma^2 t)\right) \leq \delta/2,$$

by our choice of t and γ (with appropriate constants). Letting R be the number of iterations in which $\mathcal{V}^Q$ rejects, by the completeness assumption, $\mathbb{E}[R] \leq (\delta/8 + o(1))t$, and hence Markov gives $\Pr[R > t/2] \leq \delta/4 + o(1)$. Together with the $\delta/2$ error probability of the first test, the overall completeness error is at most δ.

Soundness. Fix an arbitrary (computationally unbounded) cheating prover strategy. Recall that the cheating prover may have access to the clean function f, and may choose its answers adaptively as a function of the entire collection of query sets $\{Q_i\}_{i\in[t]}$.

As in the noiseless reduction, the verifier performs two tests. The first test (Step 6a) checks that the total number of mismatches

$$Z := \sum_{i=1}^{t} Z_i \qquad \text{where} \qquad Z_i := \mathbf{1}\left[\tilde{f}(x_i) \neq y_i\right]$$

does not exceed the threshold $T_Z = (2\eta(1-\eta) + \gamma)t$. The second test (Step 6b) checks that the simulated verifier $\mathcal{V}^Q$ rejects in at least $t/2$ iterations.

For each iteration i, let

$$\ell_i := \left|\left\{x \in Q_i : \tilde{f}(x) \neq f(x)\right\}\right|$$

be the number of disagreements between the prover's labeling and the clean function f on Q_i, and let

$$L := \sum_{i=1}^{t} \ell_i$$

be the total number of such disagreements across all t query sets. Note that L is defined only for the analysis (the verifier does not know f), and it may be chosen adversarially and adaptively by the cheating prover.

We split into two cases depending on the average disagreement rate $L/(tq)$.

Case 1: Disagreement rate is too large, $\frac{L}{tq} \geq \eta + \frac{1}{10q}$. Fix an iteration $i \in [t]$ and condition on Q_i and on the prover's labeling $\tilde{f}(Q_i)$. By embeddability, the planted sample x_i is embedded at a uniformly random position in Q_i, independent of all other randomness. Therefore,

$$\Pr[\tilde{f}(x_i) \neq f(x_i)] = \frac{\ell_i}{q}.$$

Recalling that $y_i = f(x_i) \oplus b^{\mathcal{V}}(x_i)$ with $b^{\mathcal{V}}(x_i) \sim \mathsf{Bernoulli}(\eta)$ independent of everything else, we have

$$\begin{aligned}\Pr[Z_i = 1] &= \Pr[\tilde{f}(x_i) \neq f(x_i)] \cdot (1-\eta) + \Pr[\tilde{f}(x_i) = f(x_i)] \cdot \eta \\ &= \eta + (1-2\eta) \cdot \frac{\ell_i}{q}.\end{aligned}$$

Summing over $i \in [t]$ and using linearity of expectation, we obtain

$$\mathbb{E}[Z] = \sum_{i=1}^{t} \mathbb{E}[Z_i] = \eta t + (1-2\eta) \cdot \frac{1}{q} \sum_{i=1}^{t} \ell_i = \eta t + (1-2\eta) \cdot \frac{L}{q}. \tag{3}$$

Under the assumption of this case, $L/q \geq \eta t + t/(10q)$, and therefore

$$\begin{aligned}\mathbb{E}[Z] &\geq \eta t + (1-2\eta)\left(\eta t + \frac{t}{10q}\right) \\ &= \left(\eta + \eta(1-2\eta)\right) t + \frac{1-2\eta}{10q}\, t \\ &= 2\eta(1-\eta)\, t + 2\gamma t,\end{aligned}$$

where in the last equality we used $\gamma = (1-2\eta)/(20q)$. Recalling that the first test rejects when $Z > T_Z = (2\eta(1-\eta) + \gamma)t$, we have $\mathbb{E}[Z] \geq T_Z + \gamma t$. By a Chernoff bound and our choice of t,

$$\Pr[Z \leq T_Z] \leq \exp\left(-\Omega(\gamma^2 t)\right) \leq \delta/2.$$

Thus, in Case 1 the verifier rejects in the first test with probability at least $1 - \delta/2$.

Case 2: Disagreement rate is within the allowed budget, $\frac{L}{tq} < \eta + \frac{1}{10q}$. Define the set of *heavily corrupted* indices by

$$B := \{i \in [t] : \ell_i > 4\eta q + 1\}, \qquad t_{\text{bad}} := |B|.$$

Since each $i \in B$ contributes more than $4\eta q + 1$ disagreements, we have

$$L = \sum_{i=1}^{t} \ell_i \geq t_{\text{bad}} \cdot (4\eta q + 1).$$

On the other hand, the assumption of this case implies

$$L < \left(\eta + \frac{1}{10q}\right) tq = \eta t q + \frac{t}{10}.$$

Combining the two bounds and dividing by $4\eta q + 1$, we obtain

$$t_{\text{bad}} < t \cdot \frac{\eta q + 1/10}{4\eta q + 1} \leq t \cdot \frac{1}{4},$$

where the last inequality holds since $4(\eta q + 1/10) \leq 4\eta q + 1$. Thus, letting $G := [t] \setminus B$ be the set of *good* indices, we have $|G| \geq 3t/4$, and for every $i \in G$,

$$\left|\left\{x \in Q_i : \tilde{f}(x) \neq f(x)\right\}\right| = \ell_i \leq 4\eta q + 1.$$

For every $i \in [t]$, define a modified labeling $\widetilde{f}_i'$ as follows: if $\ell_i \leq 4\eta q + 1$, let $\widetilde{f}_i' = \widetilde{f}$ on Q_i; otherwise, let $\widetilde{f}_i' = f$ on Q_i. Thus every modified execution uses an η-adversarially corrupted query oracle. Let A be the number of modified executions in which $\mathcal{V}^Q$ does not reject. By the soundness assumption and linearity of expectation, $\mathbb{E}[A] \leq (\delta/8)t$, and hence Markov's inequality gives $\Pr[A \geq t/4] \leq \delta/2$. The actual and modified executions differ only on indices in B, and $t_{\text{bad}} < t/4$. Therefore, whenever $A < t/4$, fewer than $t/2$ actual executions do not reject, so $\mathcal{V}^{S,\eta}$ rejects in the second test. Thus, with probability at least $1 - \delta/2$, $\mathcal{V}^{S,\eta}$ rejects in the second test. □

5.3 Noisy PAC Verification of Sparse Fourier Functions

For Fourier analysis, it is convenient to switch from the $\{0,1\}$-valued convention of Definition 5.1 to functions with range $\{-1,1\}$; as noted in Remark 5.5, this does not affect the model or its complexities. We first recall a few facts about Boolean functions. Throughout, we write $\{0,1\}^n$ for the domain, even when it is more natural to view it as the vector space $\mathbb{F}_2^n$. The set of all functions $\{f : \{0,1\}^n \to \mathbb{R}\}$ forms a real inner-product space with inner product

$$\langle f, g \rangle := \mathop{\mathbb{E}}_{x \leftarrow \mathsf{Unif}(\{0,1\}^n)} [f(x)g(x)].$$

For each $\gamma \in \{0,1\}^n$ let $\chi_\gamma(x) := (-1)^{\langle \gamma, x \rangle}$ (with the inner product over $\mathbb{F}_2$) denote the corresponding *Fourier character*. The family $\{\chi_\gamma : \gamma \in \{0,1\}^n\}$ forms an orthonormal basis for this space. Consequently, every function $f : \{0,1\}^n \to \mathbb{R}$ can be uniquely written as

$$f(x) = \sum_{\gamma \in \{0,1\}^n} \widehat{f}(\gamma)\chi_\gamma(x),$$

called the Fourier expansion of f, where the coefficients are given by

$$\widehat{f}(\gamma) = \langle f, \chi_\gamma \rangle$$

and are referred to as the *Fourier coefficients* of f. When $f : \{0,1\}^n \to \{-1,1\}$ we have $\widehat{f}(\gamma) \in [-1,1]$ for all γ. When convenient, we sometimes refer to a vector γ with $\widehat{f}(\gamma) \neq 0$ as a "Fourier coefficient"; the intended meaning will always be clear from the context. Any $\gamma \in \{0,1\}^n$ such that the absolute value of its Fourier coefficient $|\widehat{f}(\gamma)|$ is "large" is called a *heavy vector*.

Next, we recall two useful tools from [GJK+24] and refer the reader there for the proofs. We begin with a lemma that provides a way to estimate the sum of fourth powers of the Fourier coefficients corresponding to the vectors in an affine subspace.

Lemma 5.9 ([GJK+24, Lemma 4.4]). *Let $V + h$ be an affine subspace of $\{0,1\}^n$. For any $f : \{0,1\}^n \to \{-1,1\}$ the following holds:*

$$\sum_{\gamma \in V+h} \widehat{f}(\gamma)^4 = \mathop{\mathbb{E}}_{\substack{x,y,z \leftarrow \mathsf{Unif}(\{0,1\}^n) \\ w \leftarrow \mathsf{Unif}(V^\perp)}} \left[\chi_h(w) \cdot f(x) \cdot f(y) \cdot f(z) \cdot f(x+y+z+w)\right].$$

The following claim shows that if a set $V \subseteq \{0,1\}^n$ contains at most one heavy vector, then the fourth root of the sum of fourth powers of the coefficients over V is close to the maximum corresponding coefficient in V.

Claim 5.10 ([GJK+24, Claim 4.5]). *Let $\varepsilon' \in (0,1)$, and let $V \subseteq \{0,1\}^n$ be a set with at most one vector whose corresponding Fourier coefficient has absolute value at least ε'. Then*

$$\left| \sqrt[4]{\sum_{\gamma \in V} \widehat{f}(\gamma)^4} - \left|\widehat{f}(\gamma^\star)\right| \right| \le \sqrt{\varepsilon'},$$

where $\gamma^\star = \arg\max_{\gamma \in V} \left|\widehat{f}(\gamma)\right|$.

Next, we move on to proving a noisy variant of Claim 4.6 in [GJK+24]. The claim will show how to estimate the maximum coefficient in a subspace given access to adversarially corrupted queries.

Claim 5.11 (Noisy version of [GJK+24, Claim 4.6]). *Let $s \in [n]$ and let $(V_a)_{a \in \{0,1\}^s}$ for $V_a = V_0 + h_a$ be 2^s affine subspaces of $\{0,1\}^n$ that have at most one vector whose corresponding Fourier coefficient has absolute value at least ε'. Let*

$$\gamma_a^\star \in \arg\max_{\gamma \in V_a} |\widehat{f}(\gamma)|.$$

Then, for every $\varepsilon', \delta \in (0,1)$ and $\eta \le (\varepsilon')^2/100$, there is a randomized algorithm that, given $\varepsilon', \delta, \eta$ and query access to an η-adversarially corrupted query oracle $\tilde{f}$ as per Definition 5.7, makes at most $q = O\left(\frac{s+\log(1/\delta)}{(\varepsilon')^4}\right)$ queries to $\tilde{f}$, runs in time $O\left(\frac{2^s(s+\log(1/\delta))}{(\varepsilon')^4}\right)$ and outputs values $(\kappa_a)_a$ such that

$$\left|\kappa_a - \left|\widehat{f}(\gamma_a^\star)\right|\right| \le 2\sqrt{\varepsilon'}$$

with probability at least $1 - \delta - q^2 2^{-n}$ simultaneously for all $a \in \{0,1\}^s$.

Proof. The proof follows the structure of the noiseless setting, with an extra analysis step accounting for the adversarial corruptions and for the negligible probability of collisions among the queried points.

Let $V_a = V_0 + h_a$ be as in the statement, and let

$$\omega_a := \sum_{\gamma \in V_a} \widehat{f}(\gamma)^4.$$

By Claim 5.10 (applied to V_a and f), we know that

$$\left|\omega_a^{1/4} - \left|\widehat{f}(\gamma_a^\star)\right|\right| \le \sqrt{\varepsilon'}.$$

Thus, it suffices to estimate ω_a up to additive error at most $(\varepsilon')^2$. Indeed, if an estimation $\tilde{\omega_a} \in [0,1]$ satisfies

$$|\tilde{\omega_a} - \omega_a| \le (\varepsilon')^2,$$

then by concavity of the fourth-root function on $[0,1]$ (which implies that $|\alpha^{1/4} - \beta^{1/4}| \le |\alpha - \beta|^{1/4}$ for any $\alpha, \beta \in [0,1]$), we get

$$|\tilde{\omega_a}^{1/4} - \omega_a^{1/4}| \le \sqrt{\varepsilon'},$$

and so

$$\left|\tilde{\omega_a}^{1/4} - \left|\widehat{f}(\gamma_a^\star)\right|\right| \le \left|\tilde{\omega_a}^{1/4} - \omega_a^{1/4}\right| + \left|\omega_a^{1/4} - \left|\widehat{f}(\gamma_a^\star)\right|\right| \le 2\sqrt{\varepsilon'}.$$

The algorithm. Recall the identity from Lemma 5.9: if $x, y, z \leftarrow \mathsf{Unif}(\{0,1\}^n)$ and $w \leftarrow \mathsf{Unif}(V_0^\perp)$, then

$$\omega_a = \mathop{\mathbb{E}}_{x,y,z,w}\left[\chi_{h_a}(w) \cdot f(x) \cdot f(y) \cdot f(z) \cdot f(x+y+z+w)\right], \tag{4}$$

where χ_{h_a} is the Fourier character associated with h_a.

Fix an integer m (to be chosen later) and repeat the following experiment independently for $i = 1, \ldots, m$: sample $x_i, y_i, z_i \leftarrow \mathsf{Unif}(\{0,1\}^n)$ and $w_i \leftarrow \mathsf{Unif}(V_0^\perp)$, and set

$$t_i := x_i + y_i + z_i + w_i.$$

Query $\tilde{f}$ on the four points x_i, y_i, z_i, t_i and define the base product

$$B_i := \tilde{f}(x_i) \cdot \tilde{f}(y_i) \cdot \tilde{f}(z_i) \cdot \tilde{f}(t_i) \in \{-1, 1\}.$$

For each $a \in \{0,1\}^s$ define

$$\tilde{Z}_{i,a} := \chi_{h_a}(w_i) \cdot B_i \in \{-1,1\}, \qquad \tilde{\omega}_a := \frac{1}{m}\sum_{i=1}^{m} \tilde{Z}_{i,a}.$$

Finally, define

$$\kappa_a := \left(\max\{0, \tilde{\omega}_a\}\right)^{1/4}.$$

Note that the same queried values (the products B_i) are reused for all a, and the time to compute all $\tilde{\omega}_a$ is $O(2^s \cdot m)$.

Collision probability. Let Q denote the multiset of all queried points

$$Q := (x_1, y_1, z_1, t_1, \ldots, x_m, y_m, z_m, t_m), \qquad q := |Q| = 4m,$$

where $t_i := x_i + y_i + z_i + w_i$ and $w_i \leftarrow \mathsf{Unif}(V_0^\perp)$. Let $\mathcal{E}$ be the event that all the q queried points are distinct.

Fix any two distinct query occurrences $Q_u \neq Q_v$ in the list Q. We claim that $\Pr[Q_u = Q_v] \le 2^{-n}$. If the two occurrences belong to different repetitions, this follows because the repetitions are independent and every queried point is uniform over $\{0,1\}^n$. Within a single repetition, collisions among x_i, y_i, z_i have probability 2^{-n}. Moreover, for example,

$$\Pr[t_i = x_i] = \Pr[y_i + z_i + w_i = 0] = 2^{-n},$$

since, conditioned on w_i, the sum $y_i + z_i$ is uniform over $\{0,1\}^n$; the cases $t_i = y_i$ and $t_i = z_i$ are identical. Hence, by a union bound,

$$\Pr[\neg\mathcal{E}] \le \binom{q}{2} 2^{-n} \le q^2 2^{-n}.$$

Sampling error in the noiseless experiment. Define the corresponding noiseless random variables

$$Z_{i,a} := \chi_{h_a}(w_i) \cdot f(x_i) \cdot f(y_i) \cdot f(z_i) \cdot f(t_i) \in \{-1, 1\}, \qquad \bar{\omega}_a := \frac{1}{m} \sum_{i=1}^{m} Z_{i,a}.$$

By Equation (4), for every fixed a we have $\mathbb{E}[Z_{i,a}] = \omega_a$, and the variables $\{Z_{i,a}\}_{i=1}^{m}$ are independent. By Chernoff, for every fixed a,

$$\Pr\left[|\bar{\omega}_a - \omega_a| > \frac{(\varepsilon')^2}{2} \right] \leq 2 \exp\left(-\frac{m(\varepsilon')^4}{2} \right),$$

and a union bound over all $a \in \{0,1\}^s$ gives

$$\Pr\left[\exists a \in \{0,1\}^s : \ |\bar{\omega}_a - \omega_a| > \frac{(\varepsilon')^2}{2} \right] \leq 2^s \cdot 2 \exp\left(-\frac{m(\varepsilon')^4}{2} \right).$$

Thus, setting

$$m = \Theta\left(\frac{s + \log(1/\delta)}{(\varepsilon')^4} \right)$$

ensures that with probability at least $1 - \delta$,

$$\forall a \in \{0,1\}^s, \qquad |\bar{\omega}_a - \omega_a| \leq \frac{(\varepsilon')^2}{2}. \tag{5}$$

Robustness to adversarial corruptions. Assume now that the oracle answers satisfy the promised corruption bound

$$\left| \{ x \in Q : \ \tilde{f}(x) \neq f(x) \} \right| \leq 4\eta q + 1.$$

Condition on the event $\mathcal{E}$. Then each queried point appears in Q exactly once, and each corrupted answer affects exactly one base product B_i (namely, the unique i for which the corrupted point is one of x_i, y_i, z_i, t_i). Let

$$T := \left\{ i \in [m] : \exists u \in \{x_i, y_i, z_i, t_i\} \text{ such that } \tilde{f}(u) \neq f(u) \right\}$$

be the set of *corrupted repetitions*. Since each $i \in T$ accounts for at least one corrupted query, we have $|T| \leq 4\eta q + 1 = 16\eta m + 1$. Hence, $\tilde{Z}_{i,a} = Z_{i,a}$ for all $i \notin T$, and always $\tilde{Z}_{i,a}, Z_{i,a} \in \{-1, 1\}$. Therefore, for every a,

$$|\tilde{\omega}_a - \bar{\omega}_a| = \left| \frac{1}{m} \sum_{i=1}^{m} (\tilde{Z}_{i,a} - Z_{i,a}) \right| \leq \frac{1}{m} \sum_{i \in T} |\tilde{Z}_{i,a} - Z_{i,a}| \leq \frac{1}{m} \cdot |T| \cdot 2 \leq 32\eta + \frac{2}{m} \leq (\varepsilon')^2/2,$$

where the last inequality crucially uses the assumption on η, and also the choice of m. Combining with (5), we obtain that on the event $\mathcal{E}$ and the sampling-good event (5), $|\tilde{\omega}_a - \omega_a| \leq (\varepsilon')^2$ for every a. By a union bound, with probability at least $1 - \delta - q^2 2^{-n}$ both events hold. Since $\tilde{\omega}$ is an average of $\{-1, 1\}$-valued variables, $\max\{0, \tilde{\omega}_a\} \in [0, 1]$ for all a as well, which concludes the proof.

Finally, since each repetition uses 4 queries to $\tilde{f}$, the total number of queries is $q = 4m = O\left(\frac{s + \log(1/\delta)}{(\varepsilon')^4} \right)$, and the running time is $O(2^s \cdot m) = O\left(\frac{2^s \cdot (s + \log(1/\delta))}{(\varepsilon')^4} \right)$, as claimed. □

Next, we present the noisy variant of [GJK+24, Theorem 4.2] where the noise is *adversarial* (and not random). This theorem gives a randomized algorithm that, given query access to an adversarially corrupted query oracle, efficiently estimates the values of the t heaviest Fourier coefficients. Note that the algorithm only approximates the *values* $\widehat{f}(\gamma) \in [-1,1]$, and does not identify the corresponding vectors $\gamma \in \{0,1\}^n$ themselves.

Theorem 5.12 ($\mathsf{GL}^\star$ with adversarial noise, noisy version of [GJK+24, Theorem 4.2])**.** *For parameters $\varepsilon, \delta > 0$, an integer $t \le 2^n$ and $\eta = O(\varepsilon^4)$, there exists a randomized algorithm that, given query access to an η-adversarially corrupted query oracle $\tilde{f}$ as per Definition 5.7, makes $\mathrm{poly}(\log t, 1/\varepsilon, \log(1/\delta))$ queries to $\tilde{f}$ and outputs $\sigma_1 \ge \cdots \ge \sigma_t$ in $[0,1]$, such that*

$$\Pr\left[\left|\left|\widehat{f}(\gamma_i)\right| - \sigma_i\right| \le \varepsilon \text{ for all } i \in [t]\right] \ge 1 - \delta - o(1),$$

where $\gamma_1, \ldots, \gamma_t$ are the vectors in $\{0,1\}^n$ corresponding to the t Fourier coefficients of f with largest absolute value, listed in non-increasing order. Its running time is $\mathrm{poly}\left(n, t, \frac{1}{\varepsilon}, \frac{1}{\delta}\right)$.

Proof sketch. The argument follows the proof of [GJK+24, Theorem 4.2] almost verbatim, with the only change being that Claim 4.6 is replaced by its noisy analogue, Claim 5.11. Concretely, we run the same algorithm $\mathsf{GL}^\star$ (Algorithm 3 in [GJK+24]) with the same choice of parameters t, ε, δ and the same random partitioning of $\{0,1\}^n$ into 2^s affine subspaces, where

$$s := 2\log\left(\frac{16}{\varepsilon^4} + t\right) + \log\left(\frac{2}{\delta}\right).$$

The only place where the presence of noise changes the analysis is in the estimation of the quantities ω_a associated with each affine subspace V_a. In the noiseless setting, Claim 4.6 guarantees that one can estimate ω_a to within additive error ε by applying it with error parameter $\varepsilon' = (\varepsilon/2)^2$ noiseless queries. In the noisy setting, Claim 5.11 provides the same additive accuracy guarantee for ω_a when queries are answered by $\tilde{f}$, at the cost of an additional additive error of $q^2 2^{-n}$, which is negligible in n by the choice of s. Note that, as commented before Claim 5.11, the noiseless version works for any subspace and then invoked separately for each of them, whereas our version works simultaneously for all subspaces together (to avoid the situation where all the corruption is concentrated on a single subspace).

Taking the error parameter when invoking Claim 5.11 to be $\delta/2$, the same case analysis as in the noiseless setting implies that, with probability at least $1 - \delta/2 - o(1) \ge 1 - \delta$, the output values $\sigma_1 \ge \cdots \ge \sigma_t$ satisfy for all $i \in [t]$

$$\left|\left|\widehat{f}(\gamma_i)\right| - \sigma_i\right| \le \varepsilon.$$

The total number of queries to $\tilde{f}$ is $O\left(\left(\log(t + 1/\varepsilon) + \log(1/\delta)\right)/\varepsilon^8\right)$, and the running time of the algorithm is $\mathrm{poly}\left(n, t, 1/\varepsilon, 1/\delta\right)$. □

Learning a t-sparse function means learning its "heavy vectors": the vectors of the top t Fourier coefficients of a given function. The following definition formalizes what we mean by "top" coefficients.

Definition 5.13 (ε-top set, [GJK+24, Definition 4.1])**.** *A set $\Gamma \subseteq \{0,1\}^n$ is called an ε-top set with respect to a Boolean function $f : \{0,1\}^n \to \{-1,1\}$ if for every $\alpha \in \{0,1\}^n \setminus \Gamma$ and every $\beta \in \Gamma$ we have $|\widehat{f}(\alpha)| \le |\widehat{f}(\beta)| + \varepsilon$.*

We move on to the interactive setting. Recall that our original goal was a noisy protocol for sparse Fourier functions, and that we planned to achieve one by first constructing such a protocol in which the verifier has query access, and then applying a query-to-samples reduction on it to achieve a protocol in which the verifier only uses samples. We recall the definition of noisy sample access and noisy membership query access from the definition of noisy PAC verification (Definition 5.1), with adaptation to the setting in which $f : \{0,1\}^n \to \{-1,1\}$ instead of $f : \{0,1\}^n \to \{0,1\}$:

- *Noisy membership queries:* Access to a noisy membership queries oracle f_η, where for every input x, $f_\eta(x) = f(x) \cdot b(x)$, where $b(x) \in \{-1,1\}$ satisfies

$$\Pr[b(x) = -1] = \eta \text{ and } \Pr[b(x) = 1] = 1 - \eta,$$

  and is independent of x and across inputs x;

- *Noisy samples:* Access to random samples (x, y) with $x \leftarrow \mathcal{D}$ and $y = f(x) \cdot b'(x)$, where $b'(x) \in \{-1,1\}$ satisfies

$$\Pr[b'(x) = -1] = \eta \text{ and } \Pr[b'(x) = 1] = 1 - \eta,$$

  and is independent of x and across inputs x.

In any settings in which both oracles appear, they are always independent of each other.

The following theorem gives an interactive proof for learning the top t vectors associated with the largest t Fourier coefficients of a Boolean function, even when the verifier and prover have access to the same noisy membership query oracle. The verifier also has access to noisy samples, which is compatible with the query-to-samples reduction (the samples will not be delegated to the prover for labeling, only the queries). The protocol outputs an ε-top set Γ of t vectors and it is sound even against cheating provers that have access to the clean function f.

This task is harder than merely approximating the largest t Fourier coefficients' values, which is achieved by Theorem 5.12: that algorithm approximates the values $\widehat{f}(\gamma)$ but does not recover the vectors γ themselves. In particular, it only estimates, for a randomly chosen affine subset, the maximum coefficient inside the subset, without revealing which vector in the subset attains this value. Following [GJK+24], we address this by letting the prover learn the vectors of the top coefficients using its noisy membership query access, send these vectors to the verifier, and then have the verifier check them using the algorithm of Theorem 5.12.

Theorem 5.14 (Noisy version of [GJK+24, Theorem 4.3])**.** *There exists a single-message protocol with verifier $\mathcal{V}$ and prover $\mathcal{P}$, such that, given $\varepsilon, \delta > 0$, $\eta = O(\varepsilon^4)$ and $2^{-n/3} \leq (1 - 2\eta)\varepsilon/96$, and $t \in \mathbb{N}$, the verifier makes $\mathrm{poly}(\log t, 1/\varepsilon, \log(1/\delta))$ noisy membership queries and $O\left(t \cdot \log(t/\delta)/\varepsilon^2(1-2\eta)^2\right)$ noisy samples, the communication complexity is $t \cdot n$, the verifier's running time is $\mathrm{poly}(n, t, 1/\varepsilon, 1/(1-2\eta), 1/\delta)$, and the prover's running time is $\mathrm{poly}(n, t, 1/\varepsilon, 1/(1-2\eta), 1/\delta)$. The verifier outputs a set Λ, which is either a set of size t or $\perp$, such that:*

- Completeness. *There exists an honest prover $\mathcal{P}$ such that when $\mathcal{V}$ and $\mathcal{P}$ have query access to the same noisy membership queries oracle f_η and the verifier also has noisy sample access, the output of the interaction Λ satisfies $\Pr[\Lambda$ is $\varepsilon/3$-top$] \geq 1 - \delta - o(1)$.*

- Soundness. *For any (unbounded) prover $\mathcal{P}$, even when $\mathcal{V}$ only has access to an η-adversarially-corrupted queries oracle $\tilde{f}$ and noisy samples, and $\mathcal{P}$ has clean access to f (and to f_η), the output of the interaction Λ satisfies $\Pr[(\Lambda \neq \perp) \wedge (\Lambda$ is not ε-top$)] \leq \delta + o(1)$.*

Proof sketch. The proof follows the same high-level structure as [GJK+24, Theorem 4.3], with the only change that all oracle accesses are replaced by accesses to f_η, and the uses of GL and $\mathsf{GL}^\star$ are replaced by their noisy analogues from Theorems C.1 and 5.12.

First, in place of the noiseless learning step in [GJK+24], the honest prover runs Theorem C.1 with parameters $(t, \varepsilon/3, \delta/3)$ using query access to f_η, and sends the resulting set $\Lambda' \subseteq \{0,1\}^n$ of size t.

The verifier then performs exactly the same verification procedure as in the noiseless version, except that wherever it estimates Fourier coefficients by sampling, here it implements the same step using its noisy sample access to f: it draws independent points $x \leftarrow \mathsf{Unif}(\{0,1\}^n)$ together with noisy labels $y(x) := f(x) \cdot b^{\mathcal{V}}(x)$, where $\Pr[b^V(x) = -1] = \eta$ and $\Pr[b^V(x) = 1] = 1 - \eta$, averages the values $y(x)\chi_\gamma(x)$ to estimate

$$\widehat{f \cdot b^{\mathcal{V}}}(\gamma) = \mathop{\mathbb{E}}_x [y(x)\chi_\gamma(x)],$$

and rescales by the factor $1 - 2\eta$ to obtain an estimate for $|\widehat{f}(\gamma)|$. This rescaling is valid since

$$\mathbb{E}\left[\widehat{f \cdot b^{\mathcal{V}}}(\gamma)\right] = (1 - 2\eta)\widehat{f}(\gamma),$$

where the expectation is over the noise mask (see Theorem C.1 for full details). Thus, the same concentration argument gives the required additive accuracy (with success probability at least $1 - \delta/3$ over the noise mask), using $O\left(\frac{t \cdot \log(t/\delta)}{\varepsilon^2(1-2\eta)^2}\right)$ noisy samples. Note that the query-to-samples reduction that we apply in the next theorem explicitly allows the verifier to obtain noisy samples in addition to simulating query access (see the last sentence of Theorem 5.8 and the discussion following it).

By a Chernoff bound, after increasing the adversarial-noise from η to $\eta + O(\log(1/\delta)/q) = O(\varepsilon^4)$, where q is the number of queries in the following invocation of Theorem 5.12, the random noisy oracle satisfies the required adversarial-corruption bound with probability at least $1 - \delta/3$. In addition, in place of $\mathsf{GL}^\star$ on noiseless access to f, the verifier runs the noisy algorithm from Theorem 5.12 with parameters $(t, \varepsilon/6, \delta/3)$ using its query access (which is f_η in completeness and $\tilde{f}$ in soundness), obtaining the same type of estimates used in the noiseless analysis with $\mathrm{poly}(\log t, 1/\varepsilon, \log(1/\delta))$ queries.

The verifier applies the same acceptance condition as in the noiseless version, and the completeness argument is identical. Soundness in the model where the verifier can only access an η-adversarially corrupted oracle follows from the soundness condition of Theorem 5.12.

The stated query complexity and running times follow directly from Theorems C.1 and 5.12 together with the verifier's coefficient-estimation step. Finally, although $\mathcal{P}$ and $\mathcal{V}$ query the same (persistent) noisy oracle f_η, this causes no issue since the analysis is carried out conditioned on the fixed function f_η, and never uses any independence between the prover's message and the verifier's tests. □

Next, we present the main result of this section: an interactive protocol for verifying sparse Fourier functions in the presence of noise. Rather than outputting a single hypothesis as in standard PAC verification, the verifier outputs a set that approximates the t largest Fourier coefficients, for any prescribed t. We adopt this formulation because the verifier needs access to the coefficients themselves in order to check the prover. Since this guarantee is slightly different from the usual PAC-verification formulation, we avoid using the term "PAC verification" in the theorem statement,

consistent with the presentation in [GJK+24, Theorem 4.1] and [GRSY21, Lemma 2.3] (see also [GRSY21, Remark 2.4]). As shown in [GRSY21, Appendix G], estimating the heavy coefficients implies learning Fourier-sparse functions.

Theorem 5.15 (Verifying Fourier coefficients with noise (Theorem 1.8, formal))**.** *Fix parameters $\varepsilon, \delta \in (0,1)$, $t \in \mathbb{N}$ and a noise rate $\eta = O(\varepsilon^4)$. There exists a 3-message protocol in which the prover $\mathcal{P}$ is given noisy membership query access and the verifier $\mathcal{V}$ is given noisy sample access as defined in Definition 5.1, such that, given ε, δ, t and η, the verifier uses* $\mathrm{poly}\left(t, \frac{1}{\varepsilon}, \frac{1}{1-2\eta}, \log\frac{1}{\delta}\right)$ *noisy samples, the communication complexity is* $n \cdot \mathrm{poly}\left(t, \frac{1}{\varepsilon}, \frac{1}{1-2\eta}, \log\frac{1}{\delta}\right)$*, the verifier's running time is* $\mathrm{poly}\left(n, t, \frac{1}{\varepsilon}, \frac{1}{1-2\eta}, \frac{1}{\delta}\right)$*, and the prover's running time is* $\mathrm{poly}\left(n, t, \frac{1}{\varepsilon}, \frac{1}{1-2\eta}, \frac{1}{\delta}\right)$*. The verifier outputs Λ, which is either a set of size t or $\perp$, such that:*

- Completeness. *There exists an honest prover $\mathcal{P}$ such that when $\mathcal{P}$ has noisy membership query access and $\mathcal{V}$ has noisy sample access, the output of the interaction Λ satisfies* $\Pr[\Lambda \text{ is } \varepsilon/3\text{-top}] \geq 1 - \delta - o(1)$.
- Soundness. *For any (possibly unbounded) prover $\mathcal{P}$, even when $\mathcal{V}$ only has noisy sample access and $\mathcal{P}$ has clean access to f (and to f_η), the output of the interaction Λ satisfies* $\Pr[(\Lambda \neq \perp) \wedge (\Lambda \text{ is not } \varepsilon\text{-top})] \leq \delta + o(1)$.

Proof sketch. The proof follows the same flow as the noiseless version of [GJK+24]: we apply the query-to-samples reduction of Theorem 5.8 to the query-based protocol of Theorem 5.14. Three conditions are needed to invoke this reduction:

1. *Constructing membership-query sets.* For every $x \in \{0,1\}^n$, it should be possible to efficiently construct a membership-query set Q for the verifier conditioned on the event that a uniformly random query position equals x. In [GJK+24, Theorem 4.1 and Proposition A.2], it is shown that the verifier's query pattern in the query-based protocol is linear (see [GJK+24, Definition 6.1]), and by Lemma 6.2 there, it is possible to construct the embedded query sets required by the query-to-samples transformation. Our noisy analogue of $\mathsf{GL}^\star$ does not change the distribution or structure of the verifier's query locations, so the same construction applies here.

2. *Uniform marginals for each query.* Each query made by the verifier should have marginal distribution Unif, since this is the distribution with respect to which the protocol's guarantee is defined. In the noiseless protocol, the marginal distribution of every query is uniform, and the same remains true in our noisy variant because only the oracle answers are affected by noise, while the verifier's query locations are generated exactly as before.

3. *Robustness to noisy membership queries.* By construction of the protocol described in Theorem 5.14, the completeness guarantee of the verifier holds when its queries are answered by the prover's noisy oracle f_η, and its soundness guarantee holds when its queries are answered by an η-adversarially corrupted oracle. This is exactly the access assumed by Theorem 5.8.

Hence we may apply Theorem 5.8 (with error parameter $\delta/8$) to Theorem 5.14, obtaining a 3-message protocol in which the verifier replaces its query access by random noisy samples from f. Completeness and soundness are inherited in a black-box way from the query-based protocol, and

the stated sample complexity, communication complexity and running times follow by plugging the parameters of Theorem 5.14 into Theorem 5.8. □

Remark 5.16. *Besides the fully noisy model of Theorem 5.15, it is also natural to consider the intermediate setting in which only the verifier experiences noise, while the prover has clean membership query access to f. In this case, we can apply the one-sided noisy query-to-sample reduction (Theorem 5.4) to the same query-based Fourier verifier from Theorem 5.14 (with $\mathcal{P}$ answering the verifier's membership queries using clean access to f). This yields a protocol in which the verifier uses only noisy samples and the prover uses clean queries, whereas the analysis is simplified significantly: the verifier never has to deal with adversarially-corrupted queries, as the cheating prover has no η-fraction "cheating budget" (this "budget" stems from the difference between the noise rate of the* honest prover *and the clean labels). The sample, communication, and time complexities are of the same order (and slightly better in their dependence on η) as in the fully noisy case, so for simplicity we state and analyze only the stronger double-noisy version in the main theorem.*

5.4 Justifying the Assumption on the Noise Rate

Theorem 5.15 assumes a noise rate $\eta = O(\varepsilon^4)$. This dependence arises from the robustness requirements of the query-based Fourier verifier (Theorem 5.14) and, ultimately, from the fact that the verifier must distinguish genuine Fourier structure from noise using only corrupted access. In this section we give evidence that some non-trivial relationship between η and the target accuracy parameter ε is unavoidable even for general protocols in our noisy access model. Concretely, we show that if the noise rate is as large as $\Omega(\varepsilon)$, then a natural class of "query-to-sample-style" protocols for identifying heavy Fourier coefficients would contradict the LPN assumption.

We first recall the *decisional* version of the Learning Parity with Noise problem (LPN). This formulation is polynomially equivalent (in terms of hardness) to the more common *search* version, which has long served as a well-studied and widely used computational hardness assumption in cryptography and complexity theory.

Definition 5.17. *Fix $n, M \geq 1$ and a noise rate $p' \in (0, 1/2)$. An* LPN sample *is a pair (A, y) where $A \in \{0,1\}^{M \times n}$ and $y \in \{0,1\}^M$. The* LPN *problem is to distinguish between the following two distributions, both efficiently samplable in time* $\mathrm{poly}(n, M)$ *given (n, M, p'):*

- YES*: sample $A \leftarrow \{0,1\}^{M \times n}$ and $s \leftarrow \{0,1\}^n$ uniformly, sample $e \sim \mathsf{Bernoulli}(p')^M$, and output $(A, y) := (A, As \oplus e)$, where As is computed over $\mathbb{F}_2$.*
- NO*: sample $A \leftarrow \{0,1\}^{M \times n}$ and $u \leftarrow \{0,1\}^M$ uniformly, and output $(A, y) := (A, u)$.*

An algorithm D is a distinguisher for LPN *with advantage ε if*

$$|\Pr[D(A, y) = 1 \mid (A, y) \sim \text{YES}] - \Pr[D(A, y) = 1 \mid (A, y) \sim \text{NO}]| \geq \varepsilon.$$

We say that LPN *is hard (for parameters n, M, p') if every efficient D has negligible advantage.*

Theorem 5.18 (A barrier when the noise exceeds the accuracy)**.** *Assume that* LPN *is hard for $M = O(n/\varepsilon^2)$ samples and noise rate $1/4$, as per Definition 5.17. Fix parameters $\delta \in [0, 1/3)$, $\varepsilon \in (0, 1/2)$, and a noise rate $\eta \in [\varepsilon, 1/2)$.*

Then there is no protocol in the noisy access model of Definition 5.1 for the task of outputting an explicit nonempty ε-top set which, with $M = O(n/\varepsilon^2)$ noisy verifier samples and $\mathrm{poly}(n, 1/\varepsilon, 1/\eta)$ verifier running time, satisfies:

- ***Completeness:*** *there exists an efficient prover strategy (with access to the same noise model as in Definition 5.1) such that, for every f, causes the verifier to accept with probability at least $1-\delta$ and output an ε-top set for f; and*
- ***Soundness:*** *for every f and every (possibly unbounded) cheating prover (with access to the clean f), the verifier either rejects or outputs an ε-top set for f with probability at least $1-\delta$.*

Proof. We assume towards contradiction that such a protocol $\langle \mathcal{P}, \mathcal{V} \rangle$ exists. We will define a distribution over inputs f for which soundness fails.

Let $\gamma^\star \leftarrow \{0,1\}^n$ be a uniformly random vector. Define the noise rate $p := \frac{1-2\varepsilon}{2} \geq 0$, and let $\alpha : \{0,1\}^n \to \{0,1\}$ be a random noise mask with i.i.d. coordinates sampled from $\mathsf{Bernoulli}(p)$, that is,

$$\forall x,\ \Pr[\alpha(x) = 1] = p.$$

Define

$$f(x) := \chi_{\gamma^\star}(x) \cdot (-1)^{\alpha(x)},$$

where we recall that $\chi_\gamma(x) := (-1)^{\langle \gamma, x \rangle}$ in $\{-1,1\}$-encoding. Intuitively, f is a parity $\chi_{\gamma^\star}$ multiplied by independent noise of bias 2ε.

Claim 5.19. *With probability $1-o(1)$ over the choice of $\gamma^\star$ and α, every nonempty ε-top set for f must contain $\gamma^\star$.*

Proof. First,

$$\widehat{f}(\gamma^\star) = \mathop{\mathbb{E}}_x \left[(-1)^{\alpha(x)} \right]$$

is an average of 2^n independent $\{-1,1\}$ random variables with expectation 2ε. Hence, by a Chernoff bound,

$$\Pr\left[\left| \hat{f}(\gamma^\star) - 2\varepsilon \right| > \varepsilon/5 \right] \leq 2\exp(-\Omega(2^n \varepsilon^2)).$$

Thus, with probability $1-o(1)$, $|\hat{f}(\gamma^\star)| \geq 9\varepsilon/5$. Fix any $\gamma \neq \gamma^\star$. Then

$$\hat{f}(\gamma) = \mathop{\mathbb{E}}_x \left[\chi_{\gamma^\star \oplus \gamma}(x) \cdot (-1)^{\alpha(x)} \right],$$

which is an average of 2^n independent $\{-1,1\}$ random variables. Moreover, its expectation over α is 0: indeed, for each fixed x we have

$$\mathop{\mathbb{E}}_{\alpha(x)} \left[(-1)^{\alpha(x)} \right] = 1 - 2p = 2\varepsilon,$$

and hence

$$\mathop{\mathbb{E}}_\alpha \left[\hat{f}(\gamma) \right] = \mathop{\mathbb{E}}_x \left[\chi_{\gamma^\star \oplus \gamma}(x) \cdot \mathop{\mathbb{E}}_{\alpha(x)} \left[(-1)^{\alpha(x)} \right] \right] = 2\varepsilon \cdot \mathop{\mathbb{E}}_x [\chi_{\gamma^\star \oplus \gamma}(x)] = 0,$$

since $\gamma \neq \gamma^\star$. A Chernoff bound gives

$$\Pr\left[|\hat{f}(\gamma)| > \varepsilon/5 \right] \leq 2\exp\big(-\Omega(2^n \varepsilon^2) \big).$$

By a union bound over all $\gamma \neq \gamma^\star$, with probability $1-o(1)$ we have $|\hat{f}(\gamma)| \leq \varepsilon/5$ for all $\gamma \neq \gamma^\star$. Now if a set Γ does not contain $\gamma^\star$, pick any $\beta \in \Gamma$. Then the ε-top condition would require

$$|\hat{f}(\gamma^\star)| \leq |\hat{f}(\beta)| + \varepsilon \leq \varepsilon/5 + \varepsilon,$$

which is a contradiction. Hence $\gamma^\star \in \Gamma$. □

We next show that when $\eta \geq \varepsilon$, there exists an η-rate flipping strategy that (i) is indistinguishable from honest noise in the mismatch test, and yet (ii) completely erases the correlation with $\chi_{\gamma^\star} \cdot$.

Claim 5.20. *Assume $\eta \geq \varepsilon$. There exists a distribution over functions $r : \{0,1\}^n \to \{0,1\}$ such that:*

1. *For every $x \in \{0,1\}^n$, $\Pr[r(x) = 1] = \eta$.*
2. *For every $x \in \{0,1\}^n$, $\alpha(x) \oplus r(x)$ is uniform in $\{0,1\}$.*
3. *Letting $\tilde{f}(x) := f(x) \cdot (-1)^{r(x)} = \chi_{\gamma^\star}(x) \cdot (-1)^{\alpha(x) \oplus r(x)}$, the function $\tilde{f}$ is distributed as a uniformly random Boolean function (and in particular is independent of $\gamma^\star$).*

Proof. We define r by sampling its values independently across points, conditioned on α. Let

$$a := \frac{\eta - \varepsilon}{1 - 2\varepsilon}, \quad b := \frac{\eta + \varepsilon}{1 + 2\varepsilon}.$$

Note that $a \geq 0$ exactly when $\eta \geq \varepsilon$, and trivially $a, b \leq 1$ since $\eta < 1/2$. For each x:

- if $\alpha(x) = 1$, sample $r(x) \sim \mathsf{Bernoulli}(a)$;
- if $\alpha(x) = 0$, sample $r(x) \sim \mathsf{Bernoulli}(b)$.

Then for every fixed x, using $\Pr[\alpha(x) = 1] = (1 - 2\varepsilon)/2$ and $\Pr[\alpha(x) = 0] = (1 + 2\varepsilon)/2$,

$$\Pr[r(x) = 1] = \frac{1 + 2\varepsilon}{2} \cdot b + \frac{1 - 2\varepsilon}{2} \cdot a = \eta,$$

proving Item 1. Moreover,

$$\Pr[\alpha(x) \oplus r(x) = 1] = \Pr[\alpha(x) = 1, r(x) = 0] + \Pr[\alpha(x) = 0, r(x) = 1] = \frac{1 - 2\varepsilon}{2}(1 - a) + \frac{1 + 2\varepsilon}{2} b = \frac{1}{2},$$

proving Item 2.

Finally, by Item 2 and the fact that the values $\{\alpha(x) \oplus r(x)\}_x$ are independent across x, the mask $(-1)^{\alpha(x) \oplus r(x)}$ is i.i.d. uniform in $\{-1, 1\}$ across x. Multiplying a uniformly random truth table by the fixed parity $\chi_{\gamma^\star}$ is a bijection, so $\tilde{f}$ is itself uniformly random and independent of $\gamma^\star$. □

We now use the assumed protocol to solve LPN with noise rate $1/4$. Let $(z_i, \ell_i)_{i \in [M]}$ be the given LPN samples. Sample a uniformly random function $\tilde{f} : \{0,1\}^n \to \{-1, 1\}$ (lazily, as needed), and run the honest prover $\mathcal{P}$ using $\tilde{f}$ as its noisy membership-query oracle.

Set

$$d := 2\eta(1 - \eta), \ \kappa := \varepsilon(1 - 2\eta), \ \text{and } \theta_0 := d - 2\kappa, \ \theta_1 := d + 2\kappa.$$

Since $\eta \geq \varepsilon$, we have $\theta_0, \theta_1 \in [0, 1]$. For each i, write $\tilde{f}(z_i) = (-1)^{\tilde{b}_i}$, let $t_i := \ell_i \oplus \tilde{b}_i$, sample $c_i \leftarrow \mathsf{Bernoulli}(\theta_{t_i})$, and give $\mathcal{V}$ the noisy sample $(z_i, \tilde{f}(z_i)(-1)^{c_i})$. We condition on the event that the M sample points are distinct, whose failure probability is $o(1)$.

If the LPN instance is a NO instance, then each t_i is uniform, and hence

$$\Pr[c_i = 1] = \frac{\theta_0 + \theta_1}{2} = d = 2\eta(1 - \eta).$$

Therefore the joint distribution of $\tilde{f}$ and the verifier's samples is exactly that obtained by choosing a uniformly random underlying function and giving $\mathcal{P}$ and $\mathcal{V}$ two independent η-noisy views of it. By completeness, $\Pr[\mathcal{V} \text{ accepts} \mid \text{NO}] \geq 1 - \delta - o(1)$.

Now suppose the LPN instance is a YES instance with secret $\gamma^\star$, so that $\ell_i = \langle \gamma^\star, z_i \rangle \oplus e_i$ with $e_i \leftarrow \mathsf{Bernoulli}(1/4)$. Write $\tilde{f}(z_i) = \chi_{\gamma^\star}(z_i)(-1)^{u_i}$. Then $t_i = u_i \oplus e_i$, and therefore

$$\Pr[c_i = 1 \mid u_i = 0] = d - \kappa, \qquad \Pr[c_i = 1 \mid u_i = 1] = d + \kappa.$$

On the other hand, from Claim 5.20, where $u = \alpha \oplus r$, Bayes' rule gives

$$\Pr[r = 1 \mid u = 0] = \eta - \varepsilon \quad \text{and} \quad \Pr[r = 1 \mid u = 1] = \eta + \varepsilon.$$

After adding an independent verifier-noise bit of rate η, the corresponding disagreement probabilities with $\tilde{f}$ are $d - \kappa$ and $d + \kappa$, respectively.

Hence the simulated YES execution is exactly distributed as an execution on the hard function $f(x) = \chi_{\gamma^\star}(x)(-1)^{\alpha(x)}$ with the cheating prover that samples r as in Claim 5.20, defines $\tilde{f} = f \cdot (-1)^r$, and runs the honest prover $\mathcal{P}$ using $\tilde{f}$ as its oracle. This is a valid cheating strategy, since the prover is unbounded and, on the high-probability event established in the proof of Claim 5.19, can recover $\gamma^\star$ from f.

If the verifier's acceptance probabilities in the YES and NO LPN cases differ by a non-negligible amount, this already distinguishes LPN. Otherwise, $\Pr[\mathcal{V} \text{ accepts} \mid \text{YES}] \geq 1 - \delta - o(1)$. By soundness and Claim 5.19, $\Pr[\gamma^\star \in \Gamma \mid \text{YES}] \geq 1 - 2\delta - o(1)$, which is a positive constant since $\delta < 1/3$.

As $\mathcal{V}$ runs in polynomial time, $|\Gamma| = \mathrm{poly}(n, 1/\varepsilon)$. Using $O(\log |\Gamma|)$ fresh LPN samples, we can test all candidates in Γ and identify $\gamma^\star$ with constant probability, contradicting the standard search–decision equivalence for LPN. □

5.5 Extensions

So far, in Definition 5.1 and throughout Sections 5.2 and 5.3, the verifier and the honest prover are given the exact noise rate η as an explicit input. In some settings, however, it is more natural to assume that the parties only know an upper bound $\eta_b \geq \eta$ on the actual noise rate. In this case, the cheating prover (that we do not want to restrict) may know the true η and exploit the gap $\eta_b - \eta$ as a "cheating budget", by introducing additional corruptions while still remaining statistically consistent with the verifier's noisy view. In this subsection we show that all of our protocols extend to this setting with only minor changes: roughly, we tune all thresholds and repetition parameters using η_b (the worst case), and require robustness against $2\eta_b$-adversarial query corruptions.

Definition 5.21 ($(\varepsilon, \delta, \eta_b)$-Noisy PAC verification with unknown noise rate)**.** *Parameters $\delta, \varepsilon > 0$ and an* upper bound $\eta_b \in [0, 1/2)$*, a concept class $\mathcal{C}$ is* $(\varepsilon, \delta, \eta_b)$-Noisy PAC Verifiable with unknown noise rate *with hypothesis class $\mathcal{H}$ over a distribution $\mathcal{D}$ on $\{0,1\}^n$, if there exists an interactive protocol between a verifier $\mathcal{V}$ and a prover $\mathcal{P}$, taking explicit inputs $1^n, \varepsilon, \delta, \eta_b$ and noisy oracle access to a target function $f : \{0,1\}^n \to \{0,1\}$, such that the following holds.*

Fix an actual *noise rate $\eta \in [0, \eta_b]$ (which is not given to $\mathcal{V}$ or $\mathcal{P}$) for which the honest prover and verifier receive oracle access as in Definition 5.1 with rate η, namely:*

- *the honest prover gets full access to a noisy membership queries oracle f_η, where for every input x, $f_\eta(x) = f(x) \oplus b^{\mathcal{P}}(x)$ and $b^{\mathcal{P}}(x) \leftarrow \mathsf{Bernoulli}(\eta)$ is independent of x and independent across inputs x;*

- *the verifier gets noisy random samples* $(x, y^{\mathcal{V}})$ *with* $x \leftarrow \mathcal{D}$ *and* $y^{\mathcal{V}} = f(x) \oplus b^{\mathcal{V}}(x)$*, where* $b^{\mathcal{V}}(x) \leftarrow \mathsf{Bernoulli}(\eta)$ *and is independent of* x *and across inputs* x*, and also independent of all* $\{b^{\mathcal{P}}(x)\}_x$*.*

In both cases, the noise is persistent*: each input* x *is always paired with the same (possibly flipped) label. At the end of the protocol the verifier outputs either a hypothesis* $h \in \mathcal{H}$ *or* $\perp$*, and the following hold for every* f *and every actual noise rate* $\eta \in [0, \eta_b]$*:*

1. ***Completeness.*** *There exists an honest prover* $\mathcal{P}$ *(that only knows* η_b*) such that the output of the interaction satisfies*

$$\Pr\left[(h \neq \perp) \wedge (\Delta(f, h) \leq \Delta(f, \mathcal{C}) + \varepsilon)\right] \geq 1 - \delta.$$

2. ***Soundness.*** *For every unbounded cheating prover* $\widetilde{\mathcal{P}}$ *(that has access to the clean* f *and* η*), the output of the interaction satisfies*

$$\Pr\left[(h \neq \perp) \wedge (\Delta(f, h) > \Delta(f, \mathcal{C}) + \varepsilon)\right] \leq \delta.$$

The probability is over the randomness of the samples, the internal randomness of $\mathcal{P}$ *and* $\mathcal{V}$*, and the randomness of the noise masks* $\{b^{\mathcal{P}}\}$ *and* $\{b^{\mathcal{V}}\}$*.*

Note that when $\eta_b = \eta$ is exact, Definition 5.21 coincides with Definition 5.1.

We briefly describe the required changes to Theorems 5.4 and 5.8. The major change is that the verifier-only-noise reduction requires a much stronger soundness assumption for it to apply. In both reductions, the verifier's first test compares the empirical mismatch rate (i.e., based on the prover's labeling) to its expected baseline, and is tuned to detect an additive deviation of order $(1 - 2\eta)/q$. When only η_b is known, we tune these parameters to the worst case $\eta = \eta_b$.

- **Thresholds and repetition.** Replace every occurrence of η in the definition of γ, the threshold T_Z, and the repetition parameter t by η_b. Concretely, in the verifier-only-noise reduction one sets

$$\gamma := \frac{1 - 2\eta_b}{20q}, \qquad T_Z := (\eta_b + \gamma) \cdot t, \qquad t := \Theta\left(q^2 \cdot (1 - 2\eta_b)^{-2} \cdot \log(1/\delta)\right),$$

  and in the two-noise reduction one sets

$$T_Z := (2\eta_b(1 - \eta_b) + \gamma) \cdot t$$

  with the same γ and t.

- **Soundness assumptions.** In Theorem 5.8, define $\eta_b^\star := 2\eta_b(1 - \eta_b)$. When only η_b is known, we require that the soundness error of $\mathcal{V}^Q$ be at most δ even when $\mathcal{V}^Q$ only has access to an $\eta_b^\star$-adversarially corrupted query oracle. Accordingly, throughout Case 2 of the proof we replace the "heavily corrupted" threshold $4\eta q + 1$ by $4\eta_b^\star q + 1$. Indeed, if $L/(tq) \geq \eta_b^\star + 1/(10q)$, then for every $\eta \leq \eta_b$, Equation (3) gives

$$\frac{\mathbb{E}[Z]}{t} \geq \eta + (1 - 2\eta)\eta_b^\star + \frac{1 - 2\eta}{10q} \geq \eta_b^\star + 2\gamma,$$

so the Case 1 concentration argument remains valid. Otherwise, $L/(tq) < \eta_b^\star + 1/(10q)$, and the Case 2 argument gives that fewer than $t/4$ iterations have more than $4\eta_b^\star q + 1$ corruptions.

Importantly, the same issue arises already in the verifier-only-noise reduction of Theorem 5.4. While Theorem 5.4 in the known-noise setting does not assume any robustness of $\mathcal{V}^Q$ beyond standard (clean-query) soundness, in the upper-bound setting a cheating prover can use the slack $\eta_b - \eta$ as a "cheating budget" to introduce additional label corruptions while still passing the mismatch test calibrated to η_b. As a result, extending Theorem 5.4 to the upper-bound setting also requires the same soundness assumption of the double-noisy version: namely, that $\mathcal{V}^Q$ remains sound even with access to an η_b-*adversarially corrupted* query oracle.

With these changes, the completeness analyses become only easier when the true η is smaller than η_b (since the mismatch baseline decreases), while soundness is preserved because we explicitly treat the slack $\eta_b - \eta$ as part of the adversarial corruption budget.

Next, we describe the changes needed for extending the Fourier verification protocols (Theorems 5.14 and 5.15) to the setting where only an upper bound $\eta_b \geq \eta$ is known. In the query-based protocol of Theorem 5.14, the only places where the exact noise rate is used are (i) in the robustness parameter for adversarial query corruptions, and (ii) in the verifier's coefficient-estimation step, where noisy samples are averaged and then rescaled by $1 - 2\eta$ to estimate $|\widehat{f}(\gamma)|$. When only η_b is known, we replace the adversarial corruption parameter η everywhere on the soundness side by $\eta_b^\star$. For the verifier's coefficient-estimation step, define

$$\kappa := \frac{1 - 2\eta}{1 - 2\eta_b} \geq 1.$$

Dividing the noisy sample average by $1 - 2\eta_b$ yields an estimate of $\kappa\widehat{f}(\gamma)$ rather than of $\widehat{f}(\gamma)$. However,

$$0 \leq \kappa - 1 = \frac{2(\eta_b - \eta)}{1 - 2\eta_b} \leq \frac{2\eta_b}{1 - 2\eta_b} = O(\eta_b).$$

Since $|\widehat{f}(\gamma)| \leq 1$ and $\eta_b = O(\varepsilon^4)$, this introduces only an additional $O(\varepsilon^4)$ additive error.

To absorb this error, in the upper-bound variant we run both the direct coefficient-estimation step and the $\mathsf{GL}^\star$ estimation step with accuracy $\varepsilon/12$ instead of $\varepsilon/6$; this changes the complexities only by constant factors. Taking the implicit constant in $\eta_b = O(\varepsilon^4)$ sufficiently small, we may assume that

$$\frac{2\eta_b}{1 - 2\eta_b} \leq \varepsilon/6.$$

The original acceptance threshold $2\varepsilon/3$ can then be kept unchanged. Indeed, in completeness, if the claimed set is $\varepsilon/3$-top, the gap between an unclaimed coefficient and the smallest claimed coefficient is at most $\varepsilon/3$, and the two $\varepsilon/12$ estimation errors leave a gap of at most $\varepsilon/2 < 2\varepsilon/3$. In soundness, if the claimed set is not ε-top, some unclaimed coefficient exceeds some claimed coefficient by more than ε; after the two $\varepsilon/12$ estimation errors and the additional rescaling error of at most $\varepsilon/6$, the detected gap remains larger than $2\varepsilon/3$. Thus the same acceptance test remains complete and sound.

The condition $\eta = O(\varepsilon^4)$ in Theorem 5.14 is strengthened to $\eta_b = O(\varepsilon^4)$. The proof of Theorem C.1 is invoked with $\tau_b := (1 - 2\eta_b)\varepsilon/16 - 2^{-n/3}$ as the GL threshold, with estimating $\widehat{f}_\eta(\gamma)$ to accuracy $(1-2\eta_b)\varepsilon/16$, and ranking the candidates by $|\widetilde{f}_\eta(\gamma)|$. Since $1-2\eta \geq 1-2\eta_b$ and the factor

$1-2\eta$ is common to all Fourier coefficients, the same ε-top-set guarantee follows. Finally, Theorem 5.15 follows verbatim by applying the upper-bound version of the query-to-samples reduction (Theorem 5.8 with parameter η_b) to the modified query-based verifier.

Throughout in Sections 5.2 and 5.3, all sample/query, communication, and running-time bounds remain the same up to constants, except that every occurrence of η (and of $(1-2\eta)^{-1}$) is replaced by the upper bound η_b.

We conclude with an upper-bound version of Theorem 5.15.

Theorem 5.22 (Verifying Fourier coefficients with an upper bound on the noise rate)**.** *Fix parameters $\varepsilon, \delta \in (0,1)$, $t \in \mathbb{N}$ and an upper bound $\eta_b = O(\varepsilon^4)$. There exists a 3-message protocol in the model of Definition 5.21 such that, for every actual noise rate $\eta \in [0, \eta_b]$, the verifier uses $\mathrm{poly}\left(t, \frac{1}{\varepsilon}, \frac{1}{1-2\eta_b}, \log\frac{1}{\delta}\right)$ noisy samples, the communication complexity is $n \cdot \mathrm{poly}\left(t, \frac{1}{\varepsilon}, \frac{1}{1-2\eta_b}, \log\frac{1}{\delta}\right)$, and the running times of the verifier and the prover are $\mathrm{poly}\left(n, t, \frac{1}{\varepsilon}, \frac{1}{1-2\eta_b}, \frac{1}{\delta}\right)$. The verifier outputs Λ, which is either a set of size t or $\perp$, such that:*

- Completeness. *There exists an honest prover $\mathcal{P}$ (that only knows η_b) such that for every $\eta \le \eta_b$, when $\mathcal{P}$ has noisy membership query access and $\mathcal{V}$ has noisy sample access, the output of the interaction Λ satisfies $\Pr[\Lambda$ is $\varepsilon/3$-top$] \ge 1 - \delta - o(1)$.*
- Soundness. *For any (possibly unbounded) prover $\mathcal{P}$ and every $\eta \le \eta_b$, even when $\mathcal{V}$ only has noisy sample access and $\mathcal{P}$ has clean access to f and η, the output of the interaction Λ satisfies $\Pr[(\Lambda \neq \perp) \wedge (\Lambda$ is not ε-top$)] \le \delta + o(1)$.*

Different noise rates for the prover and the verifier. All of the results in this section extend with only notational changes to the setting where the prover and verifier experience different noise rates, denoted $\eta^{\mathcal{P}}, \eta^{\mathcal{V}} \in [0, 1/2)$. We note that it is most natural to consider the regime $\eta^{\mathcal{P}} \le \eta^{\mathcal{V}}$, in which the prover's labels are at least as reliable as the verifier's, as suggested in [GRSY21] and discussed above.

Indeed, the analysis does not rely on these rates being equal. We briefly sketch the required changes. First, in the two-rate model, the baseline disagreement probability on a planted point is

$$\bar{\eta} := \Pr[b^{\mathcal{P}}(x) \neq b^{\mathcal{V}}(x)] = \eta^{\mathcal{P}}(1-\eta^{\mathcal{V}}) + \eta^{\mathcal{V}}(1-\eta^{\mathcal{P}}) = \eta^{\mathcal{P}} + \eta^{\mathcal{V}} - 2\eta^{\mathcal{P}}\eta^{\mathcal{V}},$$

so every place where we previously used $2\eta(1-\eta)$ is replaced by $\bar{\eta}$. Second, when the verifier estimates Fourier coefficients from its own noisy samples, the appropriate rescaling factor is $(1-2\eta^{\mathcal{V}})$. Finally, the soundness loss in the query-to-samples step is governed by the prover-side noise budget: since $\mathcal{V}$ must tolerate mismatch rates consistent with an honest prover that answers according to noise rate $\eta^{\mathcal{P}}$, a cheating prover can hide adversarial mislabelings within this tolerance, yielding an $\eta^{\mathcal{P}}$-adversarial corruption soundness requirement for the underlying query-based verifier in Theorem 5.14. Consequently, the condition becomes $\eta^{\mathcal{P}} = O(\varepsilon^4)$.

The same extension applies when the parties only know upper bounds $\eta^{\mathcal{P}} \le \eta_b^{\mathcal{P}}$ and $\eta^{\mathcal{V}} \le \eta_b^{\mathcal{V}}$: the verifier tunes all thresholds to the worst case induced by $(\eta_b^{\mathcal{P}}, \eta_b^{\mathcal{V}})$ (equivalently, to an upper bound on $\bar{\eta}$), uses $(1-2\eta_b^{\mathcal{V}})$ as a conservative rescaling factor, and soundness uses the corresponding worst-case adversarial-corruption budget.

Acknowledgments

We thank Oded Goldreich for pointing out the importance of considering the model in which the noise rate is only upper bounded, for helpful discussions comparing different variants of interactive proofs of proximity, and for directing us to relevant related work.

References

[AGR25] Noga Amir, Oded Goldreich, and Guy N. Rothblum. Doubly Sub-Linear Interactive Proofs of Proximity. In Raghu Meka, editor, *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, volume 325 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:25, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[AL88] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine learning*, 2(4):343–370, 1988.

[AR23] Noga Amit and Guy N. Rothblum. Constant-round arguments from one-way functions. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 1537–1544, New York, NY, USA, 2023. Association for Computing Machinery.

[BDFPJ26] Mark Bun, Rathin Desai, and Renato Ferreira Pinto Jr. Testing distributions against bounded distinguishers. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, pages 1847–1856, 2026.

[BEFLR20] Omri Ben-Eliezer, Eldar Fischer, Amit Levi, and Ron D. Rothblum. Hard Properties with (Very) Short PCPPs and Their Applications. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:27, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[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*, STOC '04, page 1–10, New York, NY, USA, 2004. Association for Computing Machinery.

[CEH+24] Matthias C Caro, Jens Eisert, Marcel Hinsche, Marios Ioannou, Alexander Nietner, and Ryan Sweke. Interactive proofs for verifying (quantum) learning and testing. *arXiv preprint arXiv:2410.23969*, 2024.

[CHI+24] Matthias C. Caro, Marcel Hinsche, Marios Ioannou, Alexander Nietner, and Ryan Sweke. Classical Verification of Quantum Learning. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:23, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[DRTV16] Kashyap Dixit, Sofya Raskhodnikova, Abhradeep Thakurta, and Nithin Varma. Erasure-Resilient Property Testing. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 91:1–91:15, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

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

[FV13] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[GGR18] Oded Goldreich, Tom Gur, and Ron Rothblum. Proofs of proximity for context-free languages and read-once branching programs. *Information and Computation*, 261, 02 2018.

[GH98] Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Information Processing Letters*, 67(4):205–214, 1998.

[GJK+24] Tom Gur, Mohammad Mahdi Jahanara, Mohammad Mahdi Khodabandeh, Ninad Rajgopal, Bahar Salamatian, and Igor Shinkar. On the power of interactive proofs for learning. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1063–1070, New York, NY, USA, 2024. Association for Computing Machinery.

[GKS93] Sally A Goldman, Michael J Kearns, and Robert E Schapire. Exact identification of read-once formulas using fixed points of amplification functions. *SIAM Journal on Computing*, 22(4):705–726, 1993.

[GLR21] Tom Gur, Yang P Liu, and Ron D Rothblum. An exponential separation between ma and am proofs of proximity. *computational complexity*, 30(2):12, 2021.

[Gol08] Oded Goldreich. *Computational complexity - a conceptual perspective*. Cambridge University Press, 2008.

[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)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 39:1–39:43, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[GRS23] Oded Goldreich, Guy N Rothblum, and Tal Skverer. On interactive proofs of proximity with proof-oblivious queries. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, pages 59–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2023.

[GRSY21] Shafi Goldwasser, Guy N. Rothblum, Jonathan Shafer, and Amir Yehudayoff. Interactive Proofs for Verifying Machine Learning. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41:1–41:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[GS16] Venkatesan Guruswami and Adam Smith. Optimal rate code constructions for computationally simple channels. *J. ACM*, 63(4), September 2016.

[GSS+25] Cornelia Gruber, Patrick Oliver Schenk, Malte Schierholz, Frauke Kreuter, and Göran Kauermann. Sources of uncertainty in supervised machine learning–a statisticians' view. *arXiv preprint ArXiv:2305.16703*, 2025.

[HR24] Tal Herman and Guy Rothblum. Interactive proofs for general distribution properties. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 528–538, 2024.

[JSS99] Jeffrey Jackson, Eli Shamir, and Clara Shwartzman. Learning with queries corrupted by classification noise. *Discrete Applied Mathematics*, 92(2-3):157–175, 1999.

[Kea98] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

[KM93] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

[KR15] Yael Tauman Kalai and Ron D. Rothblum. Arguments of proximity - [extended abstract]. In *CRYPTO*, pages 422–442. Springer, 2015.

[KRV22] Iden Kalemaj, Sofya Raskhodnikova, and Nithin Varma. Sublinear-Time Computation in the Presence of Online Erasures. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 90:1–90:25, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[MS23] Saachi Mutreja and Jonathan Shafer. Pac verification of statistical algorithms. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 5021–5043. PMLR, 12–15 Jul 2023.

[MSD24] Yinghao Ma, Jiaxi Su, and Dong-Ling Deng. Classical verification of quantum learning advantages with noises. *arXiv preprint arXiv:2411.09210*, 2024.

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

[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*, page 108–138, Berlin, Heidelberg, 2020. Springer-Verlag.

[RRR16] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 49–62, New York, NY, USA, 2016. Association for Computing Machinery.

[RRR18] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Efficient batch verification for up. In *Proceedings of the 33rd Computational Complexity Conference*, CCC '18, Dagstuhl, DEU, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[RVW13] Guy N. Rothblum, Salil Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 793–802, New York, NY, USA, 2013. Association for Computing Machinery.

[RZR25] Noga Ron-Zewi and Ron Rothblum. Proving as fast as computing: Succinct arguments with constant prover overhead. *J. ACM*, 72(2), March 2025.

[Spi95] Daniel A Spielman. Linear-time encodable and decodable error-correcting codes. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 388–397, 1995.

[Tel20] Roei Tell. A note on tolerant testing with one-sided error. In *Computational Complexity and Property Testing: On the Interplay Between Randomness and Computation*, pages 173–177. Springer, 2020.

A Pairwise-Independent Hashing

Definition A.1 (Pairwise independence)**.** *A family $\mathcal{H}$ of functions $h : \{0,1\}^q \to \{0,1\}^m$ is pairwise independent if for every distinct $x, y \in \{0,1\}^q$ and every $u, v \in \{0,1\}^m$,*

$$\Pr_{h \leftarrow \mathcal{H}}[h(x) = u \wedge h(y) = v] = 2^{-2m}.$$

Definition A.2 (Toeplitz hash family [Gol08, Construction D.3])**.** *Fix integers $m, q \geq 1$. For a Toeplitz matrix $T \in \{0,1\}^{m \times q}$ and a vector $b \in \{0,1\}^m$, define the map $h_{T,b} : \{0,1\}^q \to \{0,1\}^m$ by*

$$h_{T,b}(z) = Tz \oplus b.$$

We call $\{h_{T,b}\}$ the Toeplitz hash family, and its description length is $q + 2m - 1$ bits.

Claim A.3 (Toeplitz is pairwise independent, [Gol08, Proposition 8.25])**.** *Let $\mathcal{H} = \{h_{T,b}\}$ be as in Definition A.2, with T uniform over all $m \times q$ Toeplitz matrices and b uniform in $\{0,1\}^m$, independently. Then $\mathcal{H}$ is pairwise independent in the sense of Definition A.1.*

B Proof of the Robust-IPP Lower Bound

In this section we prove Lemma 3.5, deferred from Section 3.4.

Proof. The idea is to iterate the robust protocol t times, thereby driving the verifier's query complexity down to a constant while keeping the communication sublinear. In the first stage, we use proximity parameter ε and robustness parameter ρ. In every subsequent stage, we use both parameters equal to ρ. This is valid because robust soundness implies that with high probability, after the first stage the restricted input is ρ-far from the current accepting set, and the same argument then propagates this ρ-distance throughout the remaining iterations. Thus, while the input length shrinks at each stage, the relevant distance parameter remains fixed.

Fix

$$n_0 := n \qquad \text{and} \qquad \varepsilon_0 := \varepsilon,$$

and for each $i \geq 0$ define

$$n_{i+1} := q(n_i, \varepsilon_i)^{17} \qquad \text{and} \qquad \varepsilon_{i+1} := \rho.$$

Since $q(m, \varepsilon') \leq m^{1-\delta}$ for every input length m and proximity parameter ε', we have

$$n_{i+1} \leq n_i^{1-\delta} \qquad \text{for every } i \geq 0.$$

Since the completeness of the underlying robust protocols is bounded by an absolute positive constant, there is an absolute constant $a > 0$ such that, if we run K independent copies of such a protocol on a yes-instance, the probability that all K copies reject is at most $\exp(-aK)$. Fix a sufficiently large absolute constant K_0 such that $\exp(-aK_0) \leq 1/12$, and a sufficiently large absolute constant C such that $\exp(-aC \log\log m) \leq 1/\log^2 m$ for all sufficiently large m.

Fix a sufficiently large constant $N_0 = N_0(c, \delta)$, to be specified below, and let t be the minimal index such that $n_t \leq N_0$. Since the input length shrinks by a polynomial factor at each round, it follows that $t = O(\log\log n)$. The final query complexity is $q' = n_t \leq N_0 = O(1)$.

Selective amplification at each stage. At stage 0, we run K_0 independent copies of the communication phase of the corresponding robust protocol. At every subsequent stage $i \geq 1$, we run $K_i := \lceil C \log\log n_i \rceil$ independent copies of its communication phase. For each copy $j \in [K_i]$, the verifier obtains a query set $Q_{i,j}$ and a decision predicate $\phi_{i,j}$, and sends $Q_{i,j}$ together with the succinct description $\langle \phi_{i,j} \rangle$ to the prover. The prover then sends an index $j_i \in [K_i]$. We retain only this selected copy and write

$$Q_i := Q_{i,j_i}, \qquad \phi_i := \phi_{i,j_i}, \qquad A_i := \{z \in \{0,1\}^{|Q_i|} : \phi_i(z) = 1\}.$$

The verifier discards the other copies and continues the recursive composition only with the selected query set and predicate. In completeness, the honest prover chooses any copy j_i for which $\phi_{i,j_i}(x_i|_{Q_{i,j_i}}) = 1$, if such a copy exists, and chooses an arbitrary index otherwise.

The composed IPP is now defined recursively using these selectively amplified stages. In the first stage, both parties execute the K_0 communication phases of the robust IPP for the original language $\mathcal{L}$, with proximity parameter $\varepsilon_0 = \varepsilon$ and robustness parameter ρ; by definition, the verifier does not access the input at this point. After the prover selects j_0, let $Q_0 \subseteq [n]$ be the selected query set, of size $q_0 = n_1$, and let $\phi_0 : \{0,1\}^{q_0} \to \{0,1\}$ be the selected predicate. Let

$$A_0 := \{z \in \{0,1\}^{q_0} : \phi_0(z) = 1\}.$$

[17] Formally, at each stage one may take $n_{i+1} := |Q_i|$, the size of the selected query set. We use $q(n_i, \varepsilon_i)$ only as an upper bound on this realized length. All subsequent inequalities use only $n_{i+1} \leq n_i^{1-\delta}$, so the argument applies verbatim to these execution-dependent lengths.

The input to the next stage is the restricted string $x_1 := x|_{Q_0} \in \{0,1\}^{q_0}$, and the next stage uses proximity parameter $\varepsilon_1 = \rho$ and robustness parameter ρ.

Note that $A_0 \in \mathsf{NC}^1$ by the assumption that the decision predicate has a succinct description (see Definition 2.4). Hence we may apply the robust protocol again, now to the language $A_0 \subseteq \{0,1\}^{q_0}$ and the input x_1. Again, both parties execute only the communication phases, now in K_1 independent copies. After the prover selects one of them, the verifier retains a query set $Q_1 \subseteq [q_0]$ of size $q_1 = n_2$, and a predicate $\phi_1 : \{0,1\}^{q_1} \to \{0,1\}$, with accepting set

$$A_1 := \{z \in \{0,1\}^{q_1} : \phi_1(z) = 1\}.$$

The input to the next stage becomes $x_2 := x_1|_{Q_1}$.

The parties continue in this manner for $i = 2, \ldots, t-1$. At stage i, they run K_i independent copies of the communication phase of the robust protocol for the language A_{i-1} on input x_i, with proximity parameter ε_i and robustness parameter ρ. After the prover selects one copy, the verifier retains a query set $Q_i \subseteq [n_i]$ of size $q_i = n_{i+1}$ and a predicate $\phi_i : \{0,1\}^{q_i} \to \{0,1\}$, with accepting set

$$A_i := \{z \in \{0,1\}^{q_i} : \phi_i(z) = 1\}.$$

As above, each A_i is in NC^1. For every $i < t-1$, the next-stage input is defined by $x_{i+1} := x_i|_{Q_i}$.

At the end, the verifier makes a single batch of oracle queries to the original input x, corresponding to the coordinates of the final restricted string $x_t = x|_{Q_0, Q_1, \ldots, Q_{t-1}}$, and accepts if and only if $\phi_{t-1}(x_t) = 1$, or equivalently, if and only if $x_t \in A_{t-1}$.

Queries and communication. Let q' and cc' denote the query and communication complexities of the composed protocol. As we already saw, $q' = O(1)$: although we run several copies of the communication phase at each stage, only the query set of the selected copy is followed, and the verifier makes oracle queries only once, at the end of the entire composition.

For the communication complexity, stage i contains K_i copies of the communication of the underlying robust protocol on inputs of length n_i. In addition, the verifier sends the K_i query sets and succinct decision predicates to the prover, and the prover sends the selected index j_i. Thus,

$$\mathsf{cc}' \le \sum_{i=0}^{t-1} K_i \mathsf{cc}(n_i) + \sum_{i=0}^{t-1} K_i n_{i+1} \log n_i + \sum_{i=0}^{t-1} K_i \widetilde{O}(n_{i+1}) + \sum_{i=0}^{t-1} O(\log K_i).$$

At stage 0, $K_0 = O(1)$, and therefore $K_0 \mathsf{cc}(n) = o(n)$ and $K_0 n_1 \log n = O(n^{1-\delta} \log n) = o(n)$, and likewise $K_0 \widetilde{O}(n_1) = o(n)$. For every $i \ge 1$, we have $n_i \le n_1 \le n^{1-\delta}$ and $K_i = O(\log \log n_i) = O(\log n)$. Since $\mathsf{cc}(m) = o(m)$, in particular $\mathsf{cc}(m) = O(m)$, and therefore

$$\sum_{i=1}^{t-1} K_i \mathsf{cc}(n_i) \le O(\log \log n) \cdot O(\log n) \cdot O(n^{1-\delta}) = o(n).$$

Similarly,

$$\sum_{i=1}^{t-1} K_i n_{i+1} \log n_i \le O(\log \log n) \cdot O(n^{1-\delta} \log^2 n) = o(n),$$

and

$$\sum_{i=1}^{t-1} K_i \widetilde{O}(n_{i+1}) \le n^{1-\delta} \mathrm{polylog}(n) = o(n).$$

The communication used to specify the selected indices is only polylog(n). Hence $\mathsf{cc}' = o(n)$, and since $q' = O(1)$, we conclude that $q' \cdot \mathsf{cc}' = o(n)$.

Completeness. Assume that $x \in \mathcal{L}$. Conditioned on all previous stages having succeeded, the current restricted input belongs to the current language.

At stage 0, the honest prover has K_0 independent copies from which to choose. By our choice of K_0, the probability that all of them reject is at most $\exp(-aK_0) \le \frac{1}{12}$.

At every stage $i \ge 1$, the probability that all K_i copies reject is at most

$$\exp(-aK_i) \le \exp(-aC \log\log n_i) \le \frac{1}{\log^2 n_i}.$$

Therefore, by a union bound, the total completeness error is at most

$$\frac{1}{12} + \sum_{i=1}^{t-1} \frac{1}{\log^2 n_i}.$$

We claim that the latter sum can be made arbitrarily small by choosing N_0 sufficiently large. Since

$$\log n_{i+1} \le (1-\delta) \log n_i,$$

we have

$$\frac{1}{\log^2 n_i} \le (1-\delta)^2 \frac{1}{\log^2 n_{i+1}}.$$

Hence

$$\sum_{i=1}^{t-1} \frac{1}{\log^2 n_i} \le \frac{1}{\log^2 n_{t-1}} \sum_{j=0}^{\infty} (1-\delta)^{2j} \le \frac{1}{(1-(1-\delta)^2)\log^2 N_0},$$

where the last inequality uses $n_{t-1} > N_0$. Choosing N_0 sufficiently large so that this quantity is at most $1/12$, the total completeness error is at most $1/6$, and in particular at most $1/3$.

Soundness. This is where robustness is used at every phase. Let x be ε-far from $\mathcal{L}$. We first record the effect of selective amplification on robust soundness. At stage i, suppose that the current input x_i is ε_i-far from the current language. For each of the K_i copies, robust soundness implies that the probability that its restricted input is less than ρ-far from its accepting set is at most $c/\log n_i$. Therefore, by a union bound, the probability that *any* of the K_i copies is robustly bad is at most $K_i c/\log n_i$. Since the prover can select a robustly bad copy only if at least one such copy exists, the same quantity upper-bounds the robust-soundness error of the selectively amplified stage.

For each $i = 0, \ldots, t-1$, let

$$\mathsf{Good}_i := \{\Delta(x_{i+1}, A_i) > \varepsilon_{i+1}\}.$$

At stage 0, since $x_0 = x$ is $\varepsilon_0 = \varepsilon$-far from $\mathcal{L}$, we have

$$\Pr[\mathsf{Good}_0] \ge 1 - \frac{K_0 c}{\log n_0}.$$

Moreover, for every $i = 1, \ldots, t-1$,

$$\Pr\left[\mathsf{Good}_i \mid \mathsf{Good}_0 \wedge \cdots \wedge \mathsf{Good}_{i-1}\right] \geq 1 - \frac{K_i c}{\log n_i}.$$

Indeed, conditioned on $\mathsf{Good}_0, \ldots, \mathsf{Good}_{i-1}$, the current input x_i is ε_i-far from A_{i-1}, and so the amplified robust-soundness bound applies at stage i.

For $i \geq 1$, since $K_i = \lceil C \log\log n_i \rceil$, we have

$$\frac{K_i c}{\log n_i} \leq c\,\frac{C \log\log n_i + 1}{\log n_i}.$$

We claim that these errors are summable along the recursion. Let us denote $L_i := \log n_i$. Since $L_{i+1} \leq (1-\delta) L_i$, for every $i \leq t-1$,

$$L_i \geq \frac{L_{t-1}}{(1-\delta)^{t-1-i}}.$$

For sufficiently large N_0, the function $u \longmapsto \frac{C \log u + 1}{u}$ is decreasing for all $u \geq \log N_0$, therefore, writing $j = t - 1 - i$,

$$\frac{C \log L_i + 1}{L_i} \leq \frac{(1-\delta)^j}{L_{t-1}} \left(C \log L_{t-1} + Cj \log \frac{1}{1-\delta} + 1 \right).$$

Summing over $i = 1, \ldots, t-1$, and extending the corresponding sum over j to infinity, gives

$$\sum_{i=1}^{t-1} \frac{K_i c}{\log n_i} \leq \frac{c}{L_{t-1}} \left(\frac{C \log L_{t-1} + 1}{\delta} + \frac{C(1-\delta)}{\delta^2} \log \frac{1}{1-\delta} \right).$$

Since $L_{t-1} = \log n_{t-1} > \log N_0$, the right-hand side is $O\left(\frac{\log\log N_0}{\log N_0}\right)$, which tends to zero as $N_0 \to \infty$. Moreover, since $n > N_0$ whenever the composition is nontrivial, $\frac{K_0 c}{\log n} \leq \frac{K_0 c}{\log N_0}$, and thus by choosing N_0 sufficiently large, we can ensure that

$$\frac{K_0 c}{\log N_0} + \sum_{i=1}^{t-1} \frac{K_i c}{\log n_i} \leq \frac{1}{3}.$$

Consequently, the probability that there exists a good event that does not happen is upper-bound by $1/3$. If all the events $\mathsf{Good}_0, \ldots, \mathsf{Good}_{t-1}$ occur, then in particular $x_t \notin A_{t-1}$, and therefore the verifier rejects. Thus the acceptance probability on every ε-far input is at most $1/3$.

To conclude, we note that the assumption in the lemma's statement is stronger than needed for the proof: it would suffice to assume that for every input length m and every language $\mathcal{L}' \subseteq \{0,1\}^m$ in NC^1, there exists a (ρ, ρ)-robust IPP for $\mathcal{L}'$ with the same additional properties. □

Remark B.1 (Why robustness is necessary in Lemma 3.5)**.** *Composition replaces the top instance x by a smaller instance $x|_Q$, determined by a query set Q and a predicate ϕ with accepting set A. For even a single step of the recursion to be sound on no-instances, we need an invariant that $x|_Q$ already has nontrivial distance from A. A plain IPP does not provide such a distance guarantee; it may only certify that $x|_Q \notin A$ without ruling out that $x|_Q$ is within one bit of A. Robustness supplies the needed gap, making the recursive step sound.*

C Noisy Goldreich-Levin

In this section, we prove a noisy analogue of the Goldreich-Levin learning algorithm. The proof of Theorem C.1 consists of two conceptually separate parts. First, we show that the heavy Fourier coefficients of the noisy function f_η are essentially the same as those of the original function f: for every $\gamma \in \{0,1\}^n$ we have $\widehat{f_\eta}(\gamma) = (1-2\eta)\widehat{f}(\gamma) \pm 2^{-\Omega(n)}$, so the noise merely rescales all coefficients by the same factor (up to a negligible error) and does not change which γ-s are heavy. Second, we apply the standard Goldreich-Levin algorithm to f_η and wrap its threshold-based guarantee into an ε-top-set guarantee (see Definition 5.13). This wrapper is not special to the noisy setting; it is implicit already in the noiseless use of GL in [GJK+24], and we spell it out explicitly here for completeness.

Theorem C.1 (Noisy Goldreich-Levin Learning Algorithm)**.** *Let $f : \{0,1\}^n \to \{-1,1\}$ and let $\eta \in [0, 1/2)$. Let f_η be the noisy membership queries oracle defined in Section 5.3. Fix parameters $1 \le t \le 2^n$ and $\varepsilon, \delta \in (0,1)$. There is a randomized algorithm that, given query access to f_η, and to explicit parameters $n, t, \varepsilon, \delta, \eta$, runs in time*

$$\mathrm{poly}\left(t, n, \frac{1}{\varepsilon}, \frac{1}{1-2\eta}, \log\frac{1}{\delta}\right)$$

and outputs a set $\Gamma \subseteq \{0,1\}^n$ of size t, such that Γ is an ε-top set (with respect to f) with probability at least $1 - \delta - o(1)$, provided that $2^{-n/3} \le ((1-2\eta)\varepsilon)/32$.

Proof. We start with relating $\widehat{f_\eta}$ and $\widehat{f}$. Fix $\gamma \in \{0,1\}^n$, and let $\{b(x)\}_{x\in\{0,1\}^n}$ be the noise mask, where the bits are i.i.d. and satisfy $\Pr[b(x) = -1] = \eta$ for every $x \in \{0,1\}^n$. Then,

$$\widehat{f_\eta}(\gamma) = \mathop{\mathbb{E}}_{x}\left[f_\eta(x)\cdot\chi_\gamma(x)\right] = 2^{-n}\sum_{x\in\{0,1\}^n} f(x)\cdot b(x)\cdot\chi_\gamma(x).$$

Taking expectation over the random noise mask, we obtain

$$\mathop{\mathbb{E}}_{\{b(x)\}_x}\left[\widehat{f_\eta}(\gamma)\right] = 2^{-n}\sum_{x\in\{0,1\}^n} f(x)\chi_\gamma(x)\cdot\mathbb{E}[b(x)] = (1-2\eta)\cdot\widehat{f}(\gamma).$$

Define $Y_x := f(x)b(x)\chi_\gamma(x) \in \{-1,1\}$. Then $\{Y_x\}_{x\in\{0,1\}^n}$ are independent and

$$\widehat{f_\eta}(\gamma) - (1-2\eta)\widehat{f}(\gamma) = 2^{-n}\sum_{x\in\{0,1\}^n}\left(Y_x - \mathbb{E}[Y_x]\right).$$

By a Chernoff bound, for an average of 2^n independent random variables in $[-1,1]$, for any $\rho > 0$,

$$\Pr\left[\left|\widehat{f_\eta}(\gamma) - (1-2\eta)\widehat{f}(\gamma)\right| > \rho\right] \le 2\exp\left(-2^{n-1}\rho^2\right).$$

Choosing $\rho = 2^{-n/3}$ gives

$$\Pr_b\left[\left|\widehat{f_\eta}(\gamma) - (1-2\eta)\widehat{f}(\gamma)\right| > 2^{-n/3}\right] \le 2\exp\left(-2^{n/3-1}\right).$$

By a union bound over all $\gamma \in \{0,1\}^n$, with probability at least

$$1 - 2^n\cdot 2\exp\left(-2^{n/3-1}\right) \ge 1 - 2^{-\Omega(n)},$$

the inequality

$$\left|\widehat{f_\eta}(\gamma) - (1-2\eta)\widehat{f}(\gamma)\right| \leq 2^{-n/3}$$

holds simultaneously for every $\gamma \in \{0,1\}^n$. In the remainder of the proof we condition on this global event and incur an additional failure probability of at most $2^{-\Omega(n)}$.

We now describe the algorithm. Let $\tau := \varepsilon/16$ and define

$$\tau_\eta := (1-2\eta)\tau - 2^{-n/3}.$$

Note that by the assumption that $2^{-n/3} \leq ((1-2\eta)\varepsilon)/32$, it holds that $\tau_\eta > 0$.

We invoke the standard Goldreich-Levin algorithm on f_η with threshold τ_η and failure probability $\delta/3$, and obtain a list $L \subseteq \{0,1\}^n$ satisfying, with probability at least $1-\delta/3$, the following properties:

1. Completeness: For every $\gamma \in \{0,1\}^n$, if $|\widehat{f_\eta}(\gamma)| \geq \tau_\eta$ then $\gamma \in L$.
2. Soundness: For every $\gamma \in L$, we have $|\widehat{f_\eta}(\gamma)| \geq \tau_\eta/2$.
3. Size and running time:[18] $|L| \leq 4/\tau_\eta^2$, and the running time is poly $\left(n, \frac{1}{\tau_\eta}, \log\frac{1}{\delta}\right)$.

Next, for each $\gamma \in L$ we estimate $\widehat{f_\eta}(\gamma)$ by sampling. Draw m uniform and independent points $x_1, \ldots, x_m \leftarrow \mathsf{Unif}(\{0,1\}^n)$ and compute

$$\widetilde{f_\eta}(\gamma) := \frac{1}{m}\sum_{i=1}^{m} f_\eta(x_i)\chi_\gamma(x_i).$$

By a Chernoff bound, for each fixed γ,

$$\Pr\left[\left|\widetilde{f_\eta}(\gamma) - \widehat{f_\eta}(\gamma)\right| > (1-2\eta)\frac{\varepsilon}{16}\right] \leq 2\exp\left(-2m\left((1-2\eta)\frac{\varepsilon}{16}\right)^2\right).$$

Choosing

$$m = \Theta\left(\frac{1}{(1-2\eta)^2\varepsilon^2}\log\frac{|L|}{\delta}\right)$$

and taking a union bound over all $\gamma \in L$ implies that, with probability at least $1-\delta/3$,

$$\left|\widetilde{f_\eta}(\gamma) - \widehat{f_\eta}(\gamma)\right| \leq (1-2\eta)\frac{\varepsilon}{16}$$

holds simultaneously for all $\gamma \in L$.

Define the estimated values of the Fourier coefficients of f by

$$\widetilde{\omega}(\gamma) := \frac{|\widetilde{f_\eta}(\gamma)|}{1-2\eta}.$$

[18] This third property is either explicitly included in the usual GL theorem statement, or follows immediately from Parseval.

On the global event we conditioned on earlier, and the sampling-accuracy event above, we have for every $\gamma \in L$,

$$\begin{aligned}\left||\widetilde{f}_\eta(\gamma)| - (1-2\eta)|\widehat{f}(\gamma)|\right| &= \left||\widetilde{f}_\eta(\gamma)| - |(1-2\eta)\widehat{f}(\gamma)|\right| \\ &\le \left|\widetilde{f}_\eta(\gamma) - (1-2\eta)\widehat{f}(\gamma)\right| \\ &\le \left|\widetilde{f}_\eta(\gamma) - \widehat{f}_\eta(\gamma)\right| + \left|\widehat{f}_\eta(\gamma) - (1-2\eta)\widehat{f}(\gamma)\right| \\ &\le (1-2\eta)\frac{\varepsilon}{16} + 2^{-n/3}.\end{aligned}$$

Dividing by $1-2\eta$ gives

$$\left|\widetilde{\omega}(\gamma) - |\widehat{f}(\gamma)|\right| \le \frac{\varepsilon}{16} + \frac{2^{-n/3}}{1-2\eta} \le \frac{\varepsilon}{8}, \tag{6}$$

where the last inequality uses $\tau_\eta > 0$, which implies $2^{-n/3} \le (1-2\eta)\varepsilon/16$. Overall, we got that the values $\widetilde{\omega}(\gamma)$ estimate $|\widehat{f}(\gamma)|$ up to an additive error of at most $\varepsilon/8$ for every $\gamma \in L$.

Finally, we output Γ as follows. If $|L| \ge t$, let Γ be the set of the t elements of L with the largest values of $\widetilde{\omega}(\gamma)$. If $|L| < t$, define $\Gamma := L$ and add arbitrary distinct elements from $\{0,1\}^n \setminus L$ until $|\Gamma| = t$.

We prove that Γ is an ε-top set for f. Let $\beta^\star = \operatorname{argmin}_{\beta\in\Gamma}\left\{\left|\widehat{f}(\beta)\right|\right\}$. It suffices to show that for every $\alpha \notin \Gamma$,

$$|\widehat{f}(\alpha)| \le |\widehat{f}(\beta^\star)| + \varepsilon,$$

because then for every $\beta \in \Gamma$ we also have $|\widehat{f}(\beta^\star)| \le |\widehat{f}(\beta)|$, and hence $|\widehat{f}(\alpha)| \le |\widehat{f}(\beta)| + \varepsilon$ as required by Definition 5.13. Next, we fix $\alpha \notin \Gamma$ and split into two cases:

- **Case 1:** $\alpha \notin L$**.** By the GL guarantee, $|\widehat{f}_\eta(\alpha)| < \tau_\eta$. By the global event relating $\widehat{f}_\eta$ and $\widehat{f}$,

  $$(1-2\eta)|\widehat{f}(\alpha)| \le |\widehat{f}_\eta(\alpha)| + 2^{-n/3} < \tau_\eta + 2^{-n/3} = (1-2\eta)\tau,$$

  so $|\widehat{f}(\alpha)| < \tau = \varepsilon/16$. Since $|\widehat{f}(\beta^\star)| \ge 0$, this implies $|\widehat{f}(\alpha)| \le |\widehat{f}(\beta^\star)| + \varepsilon$.

- **Case 2:** $\alpha \in L \setminus \Gamma$**.** By construction of Γ (taking the top t elements in L by $\widetilde{\omega}$), we have $\widetilde{\omega}(\alpha) \le \widetilde{\omega}(\beta^\star)$. Using Equation (6) on α and on $\beta^\star$,

  $$|\widehat{f}(\alpha)| \le \widetilde{\omega}(\alpha) + \frac{\varepsilon}{8} \le \widetilde{\omega}(\beta^\star) + \frac{\varepsilon}{8} \le |\widehat{f}(\beta^\star)| + \frac{\varepsilon}{4}.$$

In both cases we have shown $|\widehat{f}(\alpha)| \le |\widehat{f}(\beta^\star)| + \varepsilon$, and therefore Γ is an ε-top set with respect to f.

It remains to bound the failure probability and running time. The failure events are: failure of the global Fourier-relation event (probability $2^{-\Omega(n)}$), failure of the GL algorithm (probability at most $\delta/3$), and failure of the coefficient-estimation step (probability at most $\delta/3$ as well). Thus the overall success probability is at least $1 - \delta - 2^{-\Omega(n)} = 1 - \delta - o(1)$. The running time is the GL running time plus the coefficient-estimation step, which is poly $\left(t, n, \frac{1}{\varepsilon}, \frac{1}{1-2\eta}, \log\frac{1}{\delta}\right)$. □

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il