Electronic Colloquium on Computational Complexity, Report No. 97 (2025)

On the Limits of Computationally Sound IPPs in the Isolated Model

Hadar Strauss
Weizmann Institute of Science

July 16, 2025

Abstract

Interactive proofs of proximity (IPPs) are a relaxation of interactive proofs, analogous to property testing, in which soundness is required to hold only for inputs that are *far* from the property being verified. In such proof systems, the verifier has oracle access to the input, and it engages in two types of activities before making its decision: querying the input oracle and communicating with the prover. The main objective is to achieve protocols where both the query and communication complexities are extremely low.

Of particular interest are IPPs in which the querying and the interacting activities are performed independently, with no information flow from one activity to the other. Such IPPs were systematically studied by Goldreich, Rothblum, and Skverer (ITCS 2023), who introduced two variants: the pre-coordinated model, where the querying and interacting activities may use a common source of randomness, and the isolated model, where the two activities are fully independent, each operating with a separate source of randomness.

We focus on what is possible under these models when soundness is relaxed to *computational soundness*. Our previous work (ECCC, TR24-131) showed that the pre-coordinated model becomes significantly more powerful under this relaxation. In this work, we consider the *isolated* model under the same relaxation and show a separation between the two models. We consider a property that, by our previous work, has a computationally sound IPP in the pre-coordinated model with poly-logarithmic complexities (assuming the existence of collision-resistant hashing functions), and show that any computationally sound IPP in the isolated model for this property must have either query complexity or communication complexity that is $n^{\Omega(1)}$, where n is the length of the input.

ISSN 1433-8092

Contents

1 Introduction

This work studies interactive proofs of proximity (IPPs) in which the queries to the input and the interaction with the prover are performed independently from one another. We study what is possible under such IPPs when the soundness condition is relaxed to *computational soundness*. Our focus is on showing a gap between two models of such IPPs: one in which the querying and the interacting components have a shared source of randomness, and one in which their randomness is independent. We begin with a wider background.

1.1 General background

Property testing. The field of property testing [5, 12] studies a relaxed notion of decision problems. Rather than deciding exact membership, a property tester is only required to distinguish (w.h.p.) between objects in the property and objects that are ϵ-*far* from the property, where $\epsilon > 0$ is a proximity parameter. An object $x \in \Sigma^n$ is considered ϵ-far from a property $\Pi_n \subseteq \Sigma^n$ if it differs from any object in the property on more than $\epsilon \cdot n$ locations.

Standard decision problems generally require reading the entire input, since flipping a single bit can change the decision. In contrast, relaxed decision opens the possibility of algorithms that (probabilistically) read only a sub-linear portion of the input. Thus, in property testing, the input is viewed as a huge object to which the tester gets only oracle access, and the goal is to obtain testers with extremely low query complexity (e.g., query complexity that is poly-logarithmic in the input size).

Interactive proofs of proximity. Interactive proofs of proximity (IPPs) [3, 11] extend the relaxation considered in property testing to the realm of proof systems, analogously to the extension of standard decision algorithms to standard interactive proofs (IPs). Specifically, an interactive proof of proximity for a property is a protocol between two parties, called a verifier and a prover. The verifier has oracle access to the input, and it interacts with an (untrusted) prover that tries to convince it to accept the input. The goal is for the verifier to be convinced to accept inputs that satisfy the property ("completeness"), and to not be fooled into accepting inputs that are far from the property ("soundness"). The prover is assumed to have explicit access to the input and is computationally unbounded.

The main complexity measures considered in IPPs are the verifier's query complexity and the communication complexity (i.e., the total number of bits exchanged during the interaction with the prover). Like the query complexity, the communication complexity should be sublinear in the input length. With linear communication complexity, the prover could simply send the entire input, and the verifier could verify that (a) the alleged input is indeed in the property (which requires no queries to the oracle); and (b) the alleged input is ϵ-close to the actual input, by checking for consistency with $O(1/\epsilon)$ random locations in the actual input. Another complexity measure of interest is the round complexity (the number of back-and-forth communication rounds). The goal is to obtain proof systems with significantly lower query complexity than a tester for the property can achieve, while also minimizing the communication and round complexities.

The isolated and pre-coordinated models. The verifier in an IPP performs two distinct activities: querying the input oracle and interacting with the prover. The general definition of IPPs allows the verifier to fully coordinate these two activities; that is, it can choose where to query the input based on its communication with the prover, and likewise, it can send challenges to the prover based on values seen in the input.

Goldreich, Rothblum, and Skverer [6] considered highly restricted models where the querying and interacting activities are assigned to separate modules such that no information can flow between them. The two modules feed their final views to a separate deciding module that decides whether to accept or reject based on the combined views.

They introduced two versions of this model. In the first model, called the isolated model, the querying and the interacting modules each get a separate and independent source of randomness, making them completely independent. The second model, called the pre-coordinated model, provides both modules with a shared source of randomness, which allows for some amount of coordination.[1]

Goldreich et al. showed that the isolated model is extremely weak; that is, it can only offer a very limited advantage over property testers. Specifically, they showed that IPPs in the isolated model that use q queries and c bits of communication can be emulated by property testers with query complexity $O(c \cdot q)$.

In contrast, they showed that the pre-coordinated model is much more powerful. They showed that there are pre-coordinated IPPs of extremely low complexity for properties that are extremely hard to test. They further showed that the pre-coordinated model can efficiently[2] emulate any public-coin IPP for any property of low-degree polynomials.

Still, they also showed that the pre-coordinated model is considerably limited compared to general IPPs. They showed that public-coin $O(1)$-round IPPs in the pre-coordinated model can be efficiently emulated by standard property testers.

1.2 This work – computationally sound IPPs.

In this work (as well as in our previous work [13]), we extend the study of the isolated and pre-coordinated models to the context of *computationally sound* interactive proofs of proximity (cs-IPPs). That is, we consider what is possible under these models when relaxing the soundness condition to hold only against computationally bounded provers.

In our previous work [13], we showed that relaxing the soundness condition to computational soundness significantly increases the power of the pre-coordinated model. Specifically, we showed that, assuming the existence of collision-resistant hashing functions (CRHF), any public-coin cs-IPP can be efficiently emulated by a cs-IPP in the pre-coordinated model.

The focus of this work is on cs-IPPs in the *isolated* model. Specifically, we show the following result:

Theorem 1.1 (separation between cs-IPPs in the isolated model and cs-IPPs in the pre-coordinated model)**.** *There exists a property* $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$ *such that:*

1. *Assuming the existence of strong CRHF,* Π *has a cs-IPP in the pre-coordinated model in which the query and communication complexities are poly-logarithmic in* n*.*
2. *Any cs-IPP for* Π *in the isolated model must have either query complexity or communication complexity that is* $n^{\Omega(1)}$*.*

The property that we consider is PERM, the set of all permutations over $[n]$. The first part of Theorem 1.1 follows immediately from our emulation of public-coin cs-IPPs by cs-IPPs in the pre-coordinated model. Specifically, PERM has the following simple public-coin IPP (which is also

[1] The isolated and pre-coordinated models were originally studied in [6] as restricted versions of IPPs with proof-oblivious queries. Proof-oblivious queries restrict only the information flow from the interacting module to the querying module.

[2] We usually consider an emulation of one model in a second model to be efficient if the communication and query complexity in the second model are polynomial in the complexities in the first model.

a cs-IPP), presented in [8, Sec. 4.1]: Given input $f : [n] \to [n]$, the verifier selects a random value $v \in [n]$ and sends it to the prover, who is required to reply with the pre-image of v under f. The verifier then queries f on this alleged pre-image, and checks that it indeed leads to the claimed value v. If f is ϵ-far from PERM, then more than $\epsilon \cdot n$ points in $[n]$ have no pre-image under f. This means that, with probability greater than ϵ, the random value v will have no pre-image under f; hence, no matter what the prover sends, the verifier will reject. To get constant soundness error we repeat this system $O(1/\epsilon)$ times. Note that this protocol is not pre-coordinated. However, using the emulation in [13], it can be transformed into a pre-coordinated protocol with computational soundness, while preserving poly-logarithmic query and communication complexities.[3]

Hence, the focus of this work is on showing the second part of Theorem 1.1, establishing a lower bound on cs-IPPs for PERM in the isolated model.

An open problem. A natural open question is whether all computationally sound isolated IPPs can be efficiently emulated by testers, as in the standard soundness case. The emulation shown by [6] for the standard soundness case does not extend to computational soundness. Their emulation relies on the fact that in standard soundness, the acceptance probability when interacting with the *optimal* prover strategy is at least 2/3 if the input is a YES-instance, and at most 1/3 if the input is a NO-instance. This enables the tester in their emulation to distinguish between YES and NO instances by estimating the optimal acceptance probability within an additive deviation of 1/6 (and accepting if the estimate is greater than 1/2). However, in computationally sound systems, only efficient provers must have low acceptance probability on NO-instances. The *optimal* prover strategy can succeed with arbitrary probability, breaking the emulation.

We stress that the method of [6] heavily relies on the ability to estimate the optimal strategy's acceptance probability, while making relatively few query access to the input. Specifically, their method relies on the fact that the optimal strategy for a given input f can be computed from the probabilities that the verifier accepts f given each possible interaction transcript.[4] However, the optimal strategy may not be implemented efficiently, and so (as stated above) its performance (on NO instances) is not relevant in the context of cs-IPPs.

1.3 Techniques

Our technique for showing the lower bound on isolated cs-IPPs for PERM is adopted from [6, Apdx. A.4], where a similar method was used in a related setting, albeit only for non-adaptive queries[5] (see Section 1.3.4 for further details on their setting). The main challenge in adapting their technique to our setting is in extending it to handle *adaptive* queries. When describing the technique, we first explain the overall approach while initially ignoring the issue of adaptive queries. This initial explanation, though presented in the context of cs-IPPs in the isolated model, is analogous to the approach of [6]. We then address the issue of adaptive queries and show how we handle it.

We also present an alternative technique for proving the lower bound in the special case of *non-adaptive* queries. This technique has a similar structure as the first and shares the same underlying intuition, but diverges in how it formally establishes this intuition. In our opinion, this approach is more straightforward and provides clearer insight into what underlies the lower bound. In addition, it improves the lower bound itself in the non-adaptive case (see Section 1.3.3 for a

[3] In fact, the query complexity will even be independent of n, specifically $O(1/\epsilon)$.

[4] They showed that it is possible to obtain an estimation of all these probabilities using only $O(q \cdot c)$ queries to f, where q is the query complexity and c is the communication complexity of the isolated IPP.

[5] A verifier uses non-adaptive queries if its queries do not depend on the answers it has received to previous queries.

detailed comparison of the bounds). However, this alternative technique does not extend readily to adaptive queries.

1.3.1 Main technique

Recall that the query complexity of *testing* PERM is $\Omega(\sqrt{n})$ (see [13, Apdx. A] following [8, Lem. 4.3]). In Theorem 5.1, we show that in any isolated cs-IPP for PERM the product of the query complexity and the communication complexity must be at least $n^{\Omega(1)}$. We show the lower bound by an indistinguishability argument.

In [13, Apdx. A], the lower bound for *testing* PERM is established by showing that a random permutation Π and a random function F (which is far from PERM w.h.p.) are indistinguishable using less than $\Omega(\sqrt{n})$ queries. We aim to show an analogous indistinguishability for isolated cs-IPPs when both the query and communication complexities are restricted. Specifically, we fix an arbitrary isolated cs-IPP with query complexity q and communication complexity c that we claim are impossible to achieve. We then construct a distribution over input functions G that is far from PERM (w.h.p.), and a corresponding distribution over (efficient) cheating provers P, such that the view of the verifier when it interacts with P on input G is indistinguishable from its view when interacting with the honest prover on a random permutation Π.

A natural first attempt is to take G to be a random function (as in the aforementioned lower bound for testers) and have the cheating prover emulate the honest prover on a random permutation. Since we are in the isolated model, where the verifier's messages are independent of the input, the interaction transcript will be distributed identically in both the honest and cheating interactions. The problem with this attempt is that there may be a correlation between the transcript and the query answers. For example, consider a simple system where the honest prover sends the first few locations of the input function, and the verifier queries these same locations and checks for consistency. If the cheating prover emulates the honest prover using a random permutation independent of G, then (w.h.p.) the values it sends will not be consistent with G, and the verifier will be able to distinguish the two cases.

Notice that we can fool the foregoing verifier (in the example) by simply having the cheating prover emulate the honest prover on a permutation that is consistent with the actual input G on the first few locations. For this to work, we need the first few locations in G to behave like a permutation.

Thus, we take the following approach: We identify a small set of "heavy" locations, which are the locations that the verifier queries with high probability. We take $G = G(\Pi, F)$ to be the function that on the heavy locations behaves like a random permutation Π, and on the remaining locations behaves like a random function F, and we have the cheating prover emulate the honest prover on the same permutation Π. Since there are only a small number of heavy locations, and on the remaining ("light") locations G behaves like a random function, G will be far from PERM with high probability.

We want to show that under this construction, the verifier's view when interacting with the cheating prover on G is indistinguishable from its view when interacting with the honest prover on Π. To do so, we will essentially need to show that in expectation over the randomness of the querying module, the interaction transcript together with the answers to heavy queries have little information on the answers to light queries. The basic intuition for this is that light queries are dispersed across many locations, and the short transcript can only contain significant information on a small number of them. To argue this formally, we will consider multiple independent executions of the querying module and show that we expect to see many different light locations across these executions. This results in a large total entropy, while the transcript can reduce the total entropy

by at most c bits.

Adaptive queries. Up to this point, we have described heavy locations as those that "the verifier queries with high probability", ignoring the issue of adaptive queries. When the queries are adaptive, the locations that are queried with high probability at the k^{th} query depend on the values observed in the first $k-1$ queries.

To illustrate, consider again the foregoing example in which the verifier queries the first few locations and checks for consistency with the prover's message. Consider a variant, where instead of querying the first few locations, the verifier adaptively queries a "chain" starting at location 1; that is, given oracle access to f, it begins by querying f at position 1, obtaining $v := f(1)$, then it queries position v, and continues in the same way for several steps, at each step querying the location given by the previous answer. The honest prover sends the same chain, and the verifier checks for consistency.

Following our general approach to fool this system, we would provide the verifier with an input function $G = G(\Pi, F)$ that behaves like a random permutation Π on heavy locations and like a random function F elsewhere, and let it interact with a cheating prover that emulates the honest prover on input Π. For the verifier to falsely accept G, the chain revealed by querying G must match the chain from Π sent by the cheating prover. This means that we want the heavy locations to be the locations of Π's chain. However, these locations depend on the values of Π at previous positions in the chain.

In light of this, we define the heavy locations recursively, query by query, branching according to the values assigned to the heavy locations of previous queries. Consequently, different permutations will correspond to different heavy locations, and $G(\pi, f)$ will equal π on the heavy locations that correspond to π, and equal f elsewhere.

We begin with a single set of heavy locations for the first query, denoted $\mathcal{H}_1$. Since the first query is independent of the input, these are simply the locations that the verifier queries with high probability on the first query. Next, we branch into different sets of heavy locations based on the possible values that can be observed in the first heavy locations. For each possible assignment of values $\bar{a}_1 \in [n]^{|\mathcal{H}_1|}$ to the locations in $\mathcal{H}_1$, we define $\mathcal{H}_2(\bar{a}_1)$ to be the locations that the verifier queries with high probability on its second query when the values it observed in the first query are given by $\bar{a}_1$ on heavy locations and by a random permutation (consistent with $\bar{a}_1$ on $\mathcal{H}_1$) on light locations. In other words, $\mathcal{H}_2(\bar{a}_1)$ consists of locations that the verifier queries with high probability on its second query when querying a random permutation Π, conditioned on Π agreeing with the assignment $\bar{a}_1$ on the locations in $\mathcal{H}_1$.

We continue recursively, such that in the k^{th} query we fix assignments of values $(\bar{a}_1, \ldots, \bar{a}_{k-1})$ to the heavy locations of all previous queries, where each $\bar{a}_i \in [n]^{|\mathcal{H}_i(\bar{a}_1,\ldots,\bar{a}_{i-1})|}$. For each such sequence of assignments, we define $\mathcal{H}_k(\bar{a}_1, \ldots, \bar{a}_{k-1})$ to be the locations that are queried with high probability in the k^{th} query when querying a random permutation Π, conditioned on Π agreeing with each of the assignments $\bar{a}_1, \ldots, \bar{a}_{k-1}$ on their corresponding heavy sets. The final heavy set of each permutation π is the union of the sets $\mathcal{H}_1, \mathcal{H}_2(\bar{a}_1), \ldots, \mathcal{H}_q(\bar{a}_1, \ldots, \bar{a}_{q-1})$, where each $\bar{a}_i$ is the values that π assigns to the corresponding heavy set $\mathcal{H}_i(\bar{a}_1, \ldots, \bar{a}_{i-1})$.

1.3.2 Alternative technique for non-adaptive queries

In Section 6, we present a different technique for proving the lower bound in the special case of non-adaptive queries. The general framework remains the same: We consider a distribution $G = G(\Pi, F)$ that behaves like a random permutation Π on "heavy" locations and like a random function F on the remaining locations. (We again show that the verifier's view when querying

$G = G(\Pi, F)$ and interacting with a cheating prover that emulates the honest prover on Π is indistinguishable from its view when querying Π and interacting with the honest prover.) However, we use a different definition of heavy locations than the one used in the main technique (actually, we are going to show two possible definitions, one that is the same as in the main technique, and one that is different). More importantly, the main difference in the new approach is in how we establish that the views are indistinguishable. Specifically, we use a different approach to argue that the interaction transcript, together with the answers to heavy queries, cannot have much information on the answers to light queries in expectation. While the previous approach analyzes the information on all the light queries together, in the new approach, we consider the information on each light query individually.

Consider again the example of the system in which the honest prover sends the first few locations of the input while the verifier queries the same locations and checks for consistency. Notice that there are two aspects in which the first few locations in this example are special. First, they are queried by the verifier with high probability, and second, these locations in Π have high mutual information with the interaction transcript $T(\Pi)$. In the general case, considering the mutual information with $T(\Pi)$ alone is insufficient; we need to consider the mutual information with $T(\Pi)$ together with answers to potential other queries made by the verifier.[6] Thus, we think of each location $i \in [n]$ as having an associated "information-weight", which represents the maximal information that can be gained on $\Pi(i)$ by $T(\Pi)$ and any other $q-1$ locations in Π. Specifically, this weight equals $\max_{S \subseteq [n] \setminus \{i\}, |S|<q} \left\{ \mathrm{I}\big(\Pi(i)\,;\,(T(\Pi), \Pi(S))\big) \right\}$.

Intuitively, since the interaction transcript is short and q is small, there cannot be many "information-heavy" locations. Indeed, in Lemma 6.2 we bound the total information-weight of all locations, which in turn bounds the number of information-heavy locations.

We show that we can effectively think of the verifier as aiming to maximize the expected total information-weight of the locations it queries. On the other hand, we can view designating certain locations as "heavy" (i.e., making them locations on which G behaves like a permutation), as preventing the verifier from gaining the information-weight of these locations. Our goal is to choose these locations such that the verifier cannot gain much information-weight (in expectation over its randomness).

This leads to two possible strategies: The first is to directly choose the heavy locations to be the information-heavy locations. For the second strategy, note that since there can only be a small number of information-heavy locations, the verifier will likely not hit them if it queries locations uniformly at random. Thus, in order to gain a high expected total information-weight, the verifier must query the information-heavy locations with high probability. Therefore, we can choose the heavy locations to be the locations that the verifier queries with high probability (this strategy corresponds to the definition of heavy locations used in the main technique). We show two proofs, following each of these strategies. The first proof, presented in Section 6.1, follows the second strategy, whereas the second proof, presented in Section 6.2, follows the first strategy.

1.3.3 Comparing the actual bounds

Our main technique, which handles adaptive queries, yields a lower bound of $q^5 \cdot c = \Omega(n/\log(n))$, where q is the query complexity and c is the communication complexity. To compare this with

[6] Think of the case where the honest prover sends the first few locations of Π each XORed with the last location of Π (i.e., $\Pi(n)$). Here, the mutual information between the transcript and each of the first few locations is very small. However, the verifier can still verify consistency by querying the first few locations as well as the last location and XORing the results. Indeed, when combined with the last location of Π, the transcript has significant mutual information with each of the first locations of Π.

the bound obtained by our alternative technique, which only handles non-adaptive queries, we first mention what the main technique achieves in the specialized case of non-adaptive queries. Recall that our main technique is an adaptation of the method from [6, Apdx. A.4], which was originally used for a related setting with *non-adaptive* queries and which we modify to handle adaptive queries. A direct adaptation of [6, Apdx. A.4] to our setting, without the modification, would yield a lower bound of $q^4 \cdot c^2 = \Omega(n/\log(n))$ for the non-adaptive case. Note that while our extension to the adaptive case worsens the dependence of the lower bound on q (from q^4 to q^5), due to a more refined analysis our dependence on c is improved (from c^2 to c). An analogous refinement can be applied to the non-adaptive case, which will improve its bound to $q^4 \cdot c = \Omega(n/\log(n))$.[7] Compared to this, our alternative technique for the non-adaptive case achieves a stronger lower bound of $q^3 \cdot c = \Omega(n)$.

1.3.4 The setting of [6, Apdx. A.4] and its connection to our setting

As mentioned at the beginning of Section 1.3, our main technique for establishing the lower bound on isolated cs-IPPs for `PERM` is adopted from [6, Apdx. A.4]. The contents of [6, Apdx. A.4] focuses on the property `PwI` (standing for Permutations with Inverse), consisting of pairs (π, π^{-1}) of a permutation π over $[n]$ and its inverse π^{-1}. For this property, they established a lower bound on MAPs, which are IPPs in which there is only a single message (a "proof") from the prover to the verifier. Specifically, they showed a lower bound on the tradeoff between the query complexity and the proof length of any MAP for `PwI` that uses non-adaptive queries.

The main challenge in adapting their technique to the setting of isolated cs-IPPs is in extending it to handle *adaptive* queries. We stress that this extension to adaptive queries does not apply to the setting of [6], since `PwI` has an efficient adaptive tester.

In Appendix D, we shed some light on why the technique from the setting of MAPs could be adapted to ours. We show that the lower bound for non-adaptive isolated IPPs for `PERM` can be extended to (non-adaptive) IPPs in a *hybrid* model that extends both isolated IPPs and MAPs. We further show a more general connection between the two settings (beyond just `PERM`), by showing that, similarly to the emulation of the isolated model by testers shown in [6, Thm. 1.2], the aforementioned hybrid model can be efficiently emulated by MAPs.

1.4 Organization

In Section 2 we give formal definitions of the computational models discussed in the introduction, as well as recall some basic definitions and claims from information theory. In Section 3 we present a simple isolated IPP for `PERM` that demonstrates a general tradeoff between the query and communication complexities.

The remaining sections are devoted to proving corresponding lower bounds. Section 4 contains preliminaries that are used in both subsequent sections; specifically, it includes notation and a claim showing that a partially random function is far from `PERM` with high probability. In Section 5, we establish our main result: a lower bound on the tradeoff between the query and communication complexities for any isolated cs-IPP for `PERM`. In Section 6 we present a tighter lower bound for the special case of non-adaptive queries.

Section 6 may be read before Section 5 and can serve as a warm-up to Section 5. While the proofs in these two sections follow different approaches, they share the same underlying framework and intuitions, with the non-adaptive case being technically simpler and potentially providing clearer insight into the main ideas.

[7] This improvement can be applied also to the case considered in [6].

2 Preliminaries and definitions

Property Testers, IPPs, MAPs, and cs-IPPs

A property is a collection of sets $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$ such that Π_n is a set of functions from $[n]$ to Σ.[8] The relative hamming distance between two functions $f, g : [n] \to \Sigma$ is the fraction of inputs on which they differ. We say f is ϵ-far from g if the relative hamming distance between f and g is greater than ϵ, and otherwise we say they are ϵ-close. A function $f : [n] \to \Sigma$ is ϵ-far from a property $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$ if it is ϵ-far from any $g \in \Pi_n$; otherwise, it is ϵ-close to Π. A tester for a property Π is a probabilistic algorithm that, on input parameters $n \in \mathbb{N}$, $\epsilon > 0$ and oracle access to a function $f : [n] \to \Sigma$, outputs 1 with probability at least $2/3$ if f is in Π, and outputs 0 with probability at least $2/3$ if f is ϵ-far from Π. The query complexity of the tester is $q : \mathbb{N} \times [0, 1] \to \mathbb{N}$ if, on input n, ϵ and oracle access to any $f : [n] \to \Sigma$, the tester makes at most $q(n, \epsilon)$ queries to f.

An interactive proof of proximity for a property Π is a two-party protocol for parties called verifier and prover. The verifier has oracle access to a function $f : [n] \to \Sigma$, and also gets explicit inputs n and $\epsilon > 0$. The prover gets f as explicit input, and its aim is to convince the verifier that f is in Π. We require that the prover can convince the verifier to accept any f in Π (w.h.p.), but cannot fool the verifier into accepting f that is ϵ-far from Π (except for with low probability). The prover is defined by its strategy, which is a (computationally unbounded) function that maps a party's input and all messages it has received so far, to the next message it will send.

Definition 2.1 (interactive proofs of proximity (IPPs))**.** *A randomized and interactive oracle machine, denoted V, constitutes a verifier for an interactive proof of proximity for a property $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$, if for every $\epsilon > 0$ the following two conditions hold.*

- **(*completeness*):** *There exists a prover P, called the honest prover, such that for any $n \in \mathbb{N}$, on input n, ϵ and oracle access to any $f \in \Pi_n$, after interacting with P that gets f as explicit input, V rejects with probability at most $1/3$.*
- **(*soundness*):** *For any prover P (referred to as a cheating prover), for any $n \in \mathbb{N}$, on input n, ϵ and oracle access to any $f : [n] \to \Sigma$ that is ϵ-far from Π_n, after interacting with P, the verifier V accepts with probability at most $1/3$.*

The system has perfect completeness if the verifier accepts each $f \in \Pi$ with probability 1. The functions $q, c, r : \mathbb{N} \times [0, 1] \to \mathbb{N}$ are the system's query complexity, communication complexity, and round complexity, respectively, if on input n, ϵ, on oracle access to any $f : [n] \to \Sigma$ and when interacting with any prover, the verifier makes at most $q(n, \epsilon)$ queries to f, the parties exchange at most $c(n, \epsilon)$ bits, and the interaction consists of at most $r(n, \epsilon)$ communication rounds (with two messages per round). The system is public-coin if each message sent by the verifier consists only of the outcomes of its coin tosses. The system uses non-adaptive queries if the verifier's queries to the input function are determined based on the verifier's randomness and the message it has received from the prover, but do not depend (directly) on the answers to prior queries.

More generally, IPPs with general soundness (resp., completeness) error $e : \mathbb{N} \times [0, 1] \to [0, 1]$ are defined by replacing the term $1/3$ in the soundness (resp., completeness) condition with $e(n, \epsilon)$. In that case, we may refer to $1 - e$ as the soundness (resp., completeness) of the IPP.

MA-proofs of proximity (MAPs) [9] are IPPs in which the communication is unidirectional, with the prover sending a single message. In this case, we call the prover's message a proof and instead of communication complexity we use the term proof length.

[8] Equivalently, properties are sometimes defined as sets of strings over alphabet Σ, rather than sets of functions from n to Σ.

The soundness condition in Definition 2.1 (which holds against computationally unbounded provers) is sometimes referred to as statistical soundness. In computationally-sound IPPs (cs-IPPs), the soundness condition is relaxed to hold only against computationally efficient provers. This leads to restricting also the computational power of the prover in the completeness condition. Actually, we even require the honest prover's strategy to be implementable in probabilistic polynomial-time, although our result holds also if it is allowed to be implemented by a non-uniform polynomial-size family of circuits.

Definition 2.2 (computationally sound interactive proofs of proximity (cs-IPPs))**.** *A randomized and interactive oracle machine, denoted V, constitutes a verifier for a computationally sound interactive proof of proximity for a property $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$, if for every $\epsilon > 0$ the following two conditions hold.*

- **(*completeness*):** *There exists a prover P, called the honest prover that can be implemented by a randomized, polynomial-time machine, such that for any $n \in \mathbb{N}$, on input n, ϵ and oracle access to any $f \in \Pi_n$, after interacting with P that gets f as explicit input, V rejects with probability at most $1/3$.*
- **(*computational soundness*):** *For any prover P (referred to as a cheating prover) that can be implemented by a (non-uniform) polynomial-size family of circuits, for all sufficiently large n's, on input n, ϵ and oracle access to any $f : [n] \to \Sigma$ that is ϵ-far from Π_n, after interacting with P, the verifier V accepts with probability at most $1/3$.*

Note that in the above definitions of IPPs and cs-IPPs, the verifier is not required to be computationally efficient. This follows the convention in the property testing literature, where the focus is on query complexity rather than computational complexity. This definition only strengthens our lower bound results, but weakens the emulations presented in Claim B.1 and Theorem C.1, in which the resulting verifier is computationally inefficient.[9]

The isolated and pre-coordinated models

To define the isolated and pre-coordinated models, we follow the framework of [6], where the verifier is decomposed into three modules: the querying module $\mathcal{Q}$, the interacting module $\mathcal{I}$, and the deciding module $\mathcal{D}$. The querying module is the only part that queries the input, and the interacting module is the only part that interacts with the prover. The final decision is made by the deciding module, which is fed with the outputs of the two other modules.

Both the isolated and pre-coordinated models are restricted forms of IPPs in which there is no information flow between the querying and the interacting modules. In the isolated model, the querying and the interacting modules each get a separate and independent source of randomness, whereas in the pre-coordinated model, the modules get a shared source of randomness.

Recall that in standard interactive proofs, one denotes the output of the verifier V when interacting with a prover P by $\langle P, V\rangle(x)$, where x is the common input. In extensions that allow private inputs, one uses the notation $\langle P(y), V(z)\rangle(x)$, where z and y are private inputs given to V and P, respectively. In the isolated model, we write the random variable representing the decision of the verifier as $\mathcal{D}(\mathcal{Q}^f(R_Q), \langle P(f), \mathcal{I}(R_I)\rangle)$, where R_Q and R_I are independent random variables representing the randomness of each module. In the pre-coordinated model, we write the random variable representing the decision as $\mathcal{D}(\mathcal{Q}^f(R), \langle P(f), \mathcal{I}(R)\rangle)$, where R is a random variable representing the shared randomness of both modules.

[9] We also note that this definition differs from the one we used in [13] (where we required the verifier to be computationally efficient). There, the efficiency of the verifier is necessary for the main result (i.e., [13, Theorem 1.4]).

Information theoretic functions

The entropy of a random variable X is defined as

$$\mathrm{H}(X) \stackrel{\text{def}}{=} \sum_x \Pr[X=x] \cdot \log_2\left(\frac{1}{\Pr[X=x]}\right)$$

The conditional entropy of a random variable X given another random variable Y is defined as

$$\mathrm{H}(X \mid Y) \stackrel{\text{def}}{=} \mathbb{E}_{y \sim Y}\left[\mathrm{H}(X \mid Y=y)\right]$$

where $\mathrm{H}(X \mid Y=y)$ is the entropy of $(X \mid Y=y)$; that is:

$$\mathrm{H}(X \mid Y=y) \stackrel{\text{def}}{=} \sum_x \Pr[X=x|Y=y] \cdot \log_2\left(\frac{1}{\Pr[X=x|Y=y]}\right)$$

The mutual information between random variables X and Y is defined as

$$\mathrm{I}(X;Y) \stackrel{\text{def}}{=} \mathrm{H}(X) - \mathrm{H}(X \mid Y) = \mathrm{H}(Y) - \mathrm{H}(Y \mid X)$$

The Kullback-Leibler (KL) divergence between random variables X and X' is defined as

$$\mathrm{D}(X \,||\, X') \stackrel{\text{def}}{=} \sum_x \Pr[X=x] \cdot \log_2\left(\frac{\Pr[X=x]}{\Pr[X'=x]}\right) = \mathbb{E}_{x \sim X}\left[\log_2\left(\frac{1}{\Pr[X'=x]}\right)\right] - \mathrm{H}(X)$$

Note that for the uniform distribution U over a support of size N, it holds that

$$\mathrm{D}(X \,||\, U) = \log_2(N) - \mathrm{H}(X)$$

The conditional KL divergence between random variables X and X' given another random variable Y is defined as

$$\begin{aligned}\mathrm{D}(X \mid Y \,||\, X' \mid Y) &\stackrel{\text{def}}{=} \mathbb{E}_{y \sim Y}\left[\mathrm{D}(X \mid Y=y \,||\, X' \mid Y=y)\right] \\ &= \mathbb{E}_{\substack{x \sim X \\ y \sim Y}}\left[\log_2\left(\frac{1}{\Pr[X'=x|Y=y]}\right)\right] - \mathrm{H}(X \mid Y)\end{aligned}$$

where $\mathrm{D}(X \mid Y=y \,||\, X' \mid Y=y)$ is the KL divergence between $(X \mid Y=y)$ and $(X' \mid Y=y)$; that is:

$$\mathrm{D}(X \mid Y=y \,||\, X' \mid Y=y) \stackrel{\text{def}}{=} \sum_x \Pr[X=x|Y=y] \cdot \log_2\left(\frac{\Pr[X=x|Y=y]}{\Pr[X'=x|Y=y]}\right)$$

We next present some simple claims regarding the foregoing quantities. The first claim states standard properties of entropy (see, e.g., the textbook [2]).

Claim 2.3. *For any random variables X and Y it holds that:*

- (conditioning reduces entropy) $\mathrm{H}(X \mid Y) \leq \mathrm{H}(X)$.
- (chain rule for entropy) $\mathrm{H}(X,Y) = \mathrm{H}(X) + \mathrm{H}(Y \mid X)$.
- (sub-additivity of entropy) $\mathrm{H}(X,Y) \leq \mathrm{H}(X) + \mathrm{H}(Y)$.
- $0 \leq \mathrm{H}(X) \leq \log_2(N)$ *where N is the support size of X.*

Claim 2.4. *For any random variables X, Y and Z, it holds that* $\mathrm{H}(X \mid Y, Z) \geq \mathrm{H}(X \mid Y) - \mathrm{H}(Z)$.

Proof: We have: $\mathrm{H}(X \mid Y, Z) = \mathrm{H}(X, Z \mid Y) - \mathrm{H}(Z \mid Y) \geq \mathrm{H}(X \mid Y) - \mathrm{H}(Z \mid Y) \geq \mathrm{H}(X \mid Y) - \mathrm{H}(Z)$. ■

Claim 2.5. *For any random variables X, Y and Y', it holds that*

$$\mathrm{D}(X, Y \,||\, X, Y') = \mathrm{D}(Y \mid X \,||\, Y' \mid X).^{10}$$

Proof: Noticing that $\frac{\Pr[X=x, Y=y]}{\Pr[X=x, Y'=y]} = \frac{\Pr[Y=y \mid X=x]}{\Pr[Y'=y \mid X=x]}$, the claim follows immediately from the definitions of KL divergence and conditional KL divergence. ■

Statistical distance and indistinguishability

The statistical distance between random variables X and X' is defined as

$$\Delta(X, X') \stackrel{\text{def}}{=} \frac{1}{2} \cdot \sum_x |\Pr[X = x] - \Pr[X' = x]| = \max_{f:\{0,1\}^* \to \{0,1\}} \{\Pr[f(X) = 1] - \Pr[f(X') = 1]\}$$

We say that a (deterministic) algorithm A distinguishes between a pair of sequences of random variables $\{X_n\}_{n \in \mathbb{N}}$ and $\{X'_n\}_{n \in \mathbb{N}}$ if

$$|\Pr[A(X_n) = 1] - \Pr[A(X'_n) = 1]| = \Omega(1).$$

Note that it implies that $\Delta(X_n, X'_n) = \Omega(1)$. We say that a pair of sequences of random variables $\{X_n\}_{n \in \mathbb{N}}$ and $\{X'_n\}_{n \in \mathbb{N}}$ are indistinguishable if $\Delta(X_n, X'_n) \neq \Omega(1)$ (i.e., for every constant $\alpha > 0$, for infinitely many n's it holds that $\Delta(X_n, X'_n) < \alpha$). For simplicity of presentation, we will often consider random variables X and X' that implicitly depend on $n \in \mathbb{N}$. In such cases, we say that an algorithm A distinguishes X and X' if it distinguishes the corresponding sequences $\{X_n\}_{n \in \mathbb{N}}$ and $\{X'_n\}_{n \in \mathbb{N}}$. Likewise, we say that X and X' are indistinguishable if the corresponding sequences $\{X_n\}_{n \in \mathbb{N}}$ and $\{X'_n\}_{n \in \mathbb{N}}$ are indistinguishable.

3 An upper bound on the isolated model for PERM

In this section we present an isolated IPP for PERM that demonstrates a general tradeoff between the query and communication complexities. We stress that this IPP is *statistically* sound.

Theorem 3.1. *For every $q > 0$, there exists an isolated IPP for* PERM *with query complexity q, communication complexity $O\left(\frac{n}{q} \cdot \log(n)\right)$, soundness $\Omega(\epsilon)$, and perfect completeness.*

Proof: We begin with an outline of the proof. Recall that if $f : [n] \to [n]$ is ϵ-far from PERM, then more than $\epsilon \cdot n$ points in $[n]$ have no pre-image under f. The isolated verifier that we construct will ask the prover to provide the pre-image of $m = \Theta(n/q)$ randomly selected points in $[n]$, and separately query the input on q random locations. The choice of m such that $m \cdot q = \Omega(n)$ ensures that w.h.p., there would be a collision between one of the queries and one of the alleged pre-images provided by the prover, in which case the verifier can check that the query answer matches the prover's claim. If the input is ϵ-far from PERM, then with probability greater than ϵ, the value (sent by the verifier) for which the prover has claimed to provide the pre-image that collided with a query, does not have a pre-image, meaning the check will fail.

We proceed with the formal description of the protocol. Given input $f : [n] \to [n]$, the isolated protocol proceeds as follows.

[10] This claim is a special case of the chain rule for KL divergence, that asserts that for any random variables X, X', Y and Y', it holds that $\mathrm{D}(X, Y \,||\, X', Y') = \mathrm{D}(X \,||\, X') + \mathbb{E}_{x \sim X}\left[\mathrm{D}(Y \mid X = x \,||\, Y' \mid X' = x)\right]$.

- The interacting module selects $m = \Theta(n/q)$ distinct random points $v_1, \ldots, v_m \in [n]$ and sends them to the prover.
- The prover sends what it claims to be the pre-images of $v_1, \ldots, v_m$ under f, denoted $\bar{a} = (a_1, \ldots, a_m)$ respectively. If $\bar{a}$ contains duplicates, the verifier immediately rejects.
- The querying module queries f on q random points $i_1, \ldots, i_q \in [n]$, obtaining their values $f(i_1), \ldots, f(i_q)$.
- The deciding module accepts if and only if for every $l \in [m]$ such that a_l was queried (i.e., $a_l = i_k$ for some $k \in [q]$), the claimed image v_l equals the actual image (i.e., $f(i_k)$).

Clearly, the protocol has perfect completeness. For soundness, consider an arbitrary function f that is ϵ-far from PERM. Let a collision be the event that there exists $l \in [m]$ such that a_l equals to one of the verifier's queries i_k. When a collision occurs, let L be the random variable representing the index such that a_L equals to one of the verifier's queries (if there is more than one collision, then we take L to be the first such index).

We will show that with probability at least $\Omega(\epsilon)$, both a collision occurs and the corresponding value v_L has no pre-image under f, which together guarantee the verifier rejects. We do so in two steps: we first show that a collision occurs with constant probability, and then show that conditioned on a collision occurring, v_L has no pre-image under f with probability greater than ϵ.

To argue that a collision occurs with constant probability, we first fix $v_1, \ldots, v_m$ to arbitrary values, which in turn fixes the (optimal) prover's answers $a_1, \ldots, a_m$. We can assume that $a_1, \ldots, a_m$ are all distinct, since otherwise the verifier immediately rejects. Now, notice that the queries $i_1, \ldots, i_q$ are independent, and each will collide with one of the (distinct) values $a_1, \ldots, a_m$ with probability m/n. Thus, the expected number of collisions is $q \cdot m/n = \Omega(1)$, and by the Chernoff Bound, there will be a collision with constant probability.

Next, we condition on a collision occurring and show that v_L has no pre-image under f with probability greater than ϵ. First, since f is ϵ-far from PERM, more than $\epsilon \cdot n$ points in $[n]$ have no pre-image under f. Now, observe that conditioned on a collision occurring, v_L is uniformly distributed in $[n]$. This is because each v_i is uniformly distributed in $[n]$ and the values of $v_1, \ldots, v_m$ do not affect the probability of collisions (since $i_1, \ldots, i_q$ are uniform over $[n]$ and independent of $v_1, \ldots, v_m$). Hence, condition on a collision occurring, the uniform v_L has no pre-image under f with probability greater than ϵ.

All together, the probability that both a collision occurs and v_L has no pre-image under f is at least $\Omega(\epsilon)$, concluding the soundness claim. ■

4 Preliminaries for the lower bounds

This section establishes notation and a claim that are used in both the lower bound proofs in Sections 5 and 6.

4.1 Notation

For a function $f : [n] \to [n]$ and a subset $S \subseteq [n]$, we reserve the notation $f(S)$ to denote the restriction of f to locations S; that is, if $S = \{i_1, \ldots, i_k\}$ where $i_1 < \cdots < i_k$, then $f(S) \stackrel{\text{def}}{=} (f(i_1), \ldots, f(i_k))$. Note that this notation differs from its common use to represent the image set $\{f(i) : i \in S\}$. Consequently, we denote the image of S under f by $\text{Img}(f, S) \stackrel{\text{def}}{=} \{f(i) : i \in S\}$. For

a sequence $\overline{S} = (i_1, \ldots, i_k)$ of elements in $[n]$, we denote by $f(\overline{S})$ the sequence $(f(i_1), \ldots, f(i_k))$. For any $l \in [k]$, we denote by $\overline{S}_{[l]}$ the prefix $(i_1, ..., i_l)$.

4.2 A partially random function is far from PERM w.h.p.

In [8, Claim 4.5] it was shown that a random function is ϵ-far from PERM with high probability (for sufficiently small $\epsilon > 0$). Here, we generalize this claim to show that also a function that agrees with a random function on half of its inputs is far from PERM with high probability.

Claim 4.1. *Let F be a random function from $[n]$ to $[n]$, and let $S \subseteq [n]$ be a set of size at least $\frac{n}{2}$. Let G be a function from $[n]$ to $[n]$ that agrees with F on S and is fixed to arbitrary values on the remaining locations $[n] \setminus S$. Then, for all sufficiently small $\epsilon > 0$, the probability that G is ϵ-close to* PERM *is at most* $\exp(-\Omega(n))$.

Proof: We follow closely the proof of [8, Claim 4.5]. Notice that if G is ϵ-close to PERM then it must hold that $|\operatorname{Img}(F, S)| \geq |S| - \epsilon \cdot n$. Thus, we will show that for all sufficiently small $\epsilon > 0$, it holds that

$$\Pr\left[\left|\operatorname{Img}(F, S)\right| \geq |S| - \epsilon \cdot n\right] \leq \exp(-\Omega(n))$$

Consider an arbitrary subset $T \subseteq S$ of size $\frac{n}{4} + \epsilon \cdot n$, and note that $|S \setminus T| \geq \frac{n}{2} - (\frac{n}{4} + \epsilon \cdot n) \geq \frac{n}{8}$ for sufficiently small ϵ. Note that the values of F at distinct locations are independent, and in particular, $\operatorname{Img}(F, T)$ and $\operatorname{Img}(F, S \setminus T)$ are independent. Observe that if $|\operatorname{Img}(F, S)| \geq |S| - \epsilon \cdot n$, then both the following must hold:

1. $|\operatorname{Img}(F, T)| \geq n/4$, since $|\operatorname{Img}(F, S)| \leq |\operatorname{Img}(F, T)| + (|S| - |T|)$.
2. $|\operatorname{Img}(F, S \setminus T) \cap \operatorname{Img}(F, T)| \leq \epsilon \cdot n$, since there cannot be more than $\epsilon \cdot n$ collisions between the values of F on T and on $S \setminus T$.

To bound the probability that both conditions are satisfied, we first fix $\operatorname{Img}(F, T)$ such that it satisfies item (1). Now, the expected number of collisions in $\operatorname{Img}(F, S \setminus T) \cap \operatorname{Img}(F, T)$ is $\frac{|\operatorname{Img}(F,T)|}{n} \cdot |S \setminus T| \geq \frac{1}{4} \cdot \frac{n}{8} = \frac{n}{32}$, since each element in $S \setminus T$ is mapped by F into $\operatorname{Img}(F, T)$ with probability $\frac{|\operatorname{Img}(F,T)|}{n} \geq \frac{1}{4}$. Therefore, by Chernoff Bound, for every $\epsilon < \frac{1}{32}$ the probability of having at most $\epsilon \cdot n$ collisions (as required by item (2)) is at most $\exp(-\Omega(n))$. ■

5 A lower bound on the isolated model for PERM

In this section we establish the second part of Theorem 1.1, by showing the following lower bound on isolated cs-IPPs for PERM.

Theorem 5.1. *If* PERM *can be verified by a computationally-sound IPP in the isolated model that has query complexity $q > 0$ and communication complexity $c > 0$, then $q^5 \cdot c = \Omega(n/\log(n))$.*

5.1 Proof of Theorem 5.1

Consider an arbitrary computationally sound isolated IPP that has communication complexity $c > 0$ and uses $q > 0$ (possibly adaptive) queries. Without loss of generality, we assume that the querying module does not make the same query more than once.

Let R_Q and R_I denote the randomness of the querying and interacting modules, respectively. For every r, for every $k \in [q]$ and every $\bar{a} \in [n]^{k-1}$, let $Q_k(\bar{a}, r)$ denote the k^{th} query made by the

querying module with randomness r after obtaining $\bar{a}$ as the answers to the first $k-1$ queries. Let $\overline{Q}_{[k]}(\bar{a}, r) \stackrel{\text{def}}{=} (Q_1(r), Q_2(a_1, r), \ldots, Q_k(\bar{a}_{[k-1]}, r))$ be the sequence of the first k queries given answers $\bar{a}$ and randomness r. For any function f, we define $Q_k^f(r) \stackrel{\text{def}}{=} Q_k(\bar{a}, r)$ where $\bar{a} = (a_1, \ldots, a_{k-1})$ such that $a_i = f(Q_i(\bar{a}_{[i-1]}, r))$ for each $i \in [k-1]$. Accordingly, we define $\overline{Q}^f_{[k]}(r) \stackrel{\text{def}}{=} (Q_1^f(r), \ldots, Q_k^f(r))$, and denote the complete sequence by $\overline{Q}^f(r) \stackrel{\text{def}}{=} \overline{Q}^f_{[q]}(r)$. We remove the overline and write $Q_{[k]}(\bar{a}, r)$ (resp., $Q^\pi_{[k]}(r)$) when we refer to the *set* of the queries in $\overline{Q}_{[k]}(\bar{a}, r)$ (resp., $\overline{Q}^\pi_{[k]}(r)$) rather than the sequence.

Note that $\overline{Q}^f_{[k]}(r)$ does not depend on all of f, but rather only on the value of f on the prefix $\overline{Q}^f_{[k-1]}(r)$. In particular, we will use the fact that if two functions f and g satisfy that $f(\overline{Q}^f_{[k]}(r)) = g(\overline{Q}^g_{[k]}(r))$, it implies that $\overline{Q}^f_{[k]}(r) = \overline{Q}^g_{[k]}(r)$ (in addition to $Q^f_{k+1}(r) = Q^g_{k+1}(r)$).

Next, we turn to define heavy locations as locations that are queried with high probability by the querying module. Since queries are adaptive, these locations depend on the answers to previous queries. Hence, we define the heavy locations recursively, query by query, each time fixing values for the *previous heavy locations only*. Let $\alpha > 0$ be a threshold parameter to be determined later (specifically, we will set $\alpha = O(q^2/n)$). Since the first query is independent of the input, the heavy locations of the first query are simply those that are queried with high probability in the first query:

$$\mathcal{H}_1 \stackrel{\text{def}}{=} \left\{ i \in [n] : \Pr_{R_Q}[Q_1(R_Q) = i] \geq \alpha \right\}$$

Next, we consider all possible assignments of values to the first heavy set. For each assignment $\bar{a}_1 \in [n]^{|\mathcal{H}_1|}$, we define $\mathcal{H}_2(\bar{a}_1)$ to be the locations that are queried with high probability in the second query when querying a random permutation Π, conditioned on Π agreeing with the assignment $\bar{a}_1$ on $\mathcal{H}_1$:

$$\mathcal{H}_2(\bar{a}_1) \stackrel{\text{def}}{=} \left\{ i \in [n] : \Pr_{\Pi, R_Q}\left[Q_2^\Pi(R_Q) = i \;\middle|\; \Pi(\overline{\mathcal{H}}_1) = \bar{a}_1\right] \geq \alpha \right\}$$

We continue recursively. For the k^{th} query, we consider all assignments $\bar{a}_{[k-1]} = (\bar{a}_1, \ldots, \bar{a}_{k-1})$ to the heavy locations of the first $k-1$ queries, such that $\bar{a}_i \in [n]^{|\mathcal{H}_i(\bar{a}_{[i-1]})|}$ for each $i \in [k-1]$.[11] We denote the sequence of heavy locations for the first $k-1$ queries corresponding to $\bar{a}_{[k-2]}$ by $\mathcal{H}_{[k-1]}(\bar{a}_{[k-2]}) \stackrel{\text{def}}{=} (\mathcal{H}_1, \mathcal{H}_2(\bar{a}_1) \ldots, \mathcal{H}_{k-1}(\bar{a}_{[k-2]}))$. We define $\mathcal{H}_k(\bar{a}_{[k-1]})$ to be the locations that are queried with high probability in the k^{th} query when querying a random permutation Π, conditioned on Π agreeing with all assignment in $\bar{a}_{[k-1]}$ on the corresponding sets in $\mathcal{H}_{[k-1]}(\bar{a}_{[k-2]})$:

$$\mathcal{H}_k(\bar{a}_{[k-1]}) \stackrel{\text{def}}{=} \left\{ i \in [n] : \Pr_{\Pi, R_Q}\left[Q_k^\Pi(R_Q) = i \;\middle|\; \Pi(\overline{\mathcal{H}}_{[k-1]}(\bar{a}_{[k-2]})) = \bar{a}_{[k-1]}\right] \geq \alpha \right\}$$

where $\Pi(\overline{\mathcal{H}}_{[k-1]}(\bar{a}_{[k-2]}))$ denotes the application of Π to each element in $\overline{\mathcal{H}}_{[k-1]}(\bar{a}_{[k-2]})$ in order; that is, $\Pi(\overline{\mathcal{H}}_{[k-1]}(\bar{a}_{[k-2]})) = (\Pi(\mathcal{H}_1), \Pi(\mathcal{H}_2(\bar{a}_1)), \ldots, \Pi(\mathcal{H}_{k-1}(\bar{a}_{[k-2]})))$.

For every permutation π, we define $\mathcal{H}^\pi_k \stackrel{\text{def}}{=} \mathcal{H}_k(\bar{a}_{[k-1]})$ where $\bar{a}_{[k-1]} = (\bar{a}_1, \ldots, \bar{a}_{k-1})$ such that $\bar{a}_i = \pi(\mathcal{H}_i(\bar{a}_{[i-1]}))$ for each $i \in [k-1]$. Accordingly, we define $\overline{\mathcal{H}}^\pi_{[k]} \stackrel{\text{def}}{=} (\mathcal{H}^\pi_1, \ldots, \mathcal{H}^\pi_k)$, and denote the

[11] We will actually only care about assignments that are consistent with some permutation; that is, assignments $\bar{a}_{[k-1]} = (\bar{a}_1, \ldots, \bar{a}_{k-1})$ for which there exists some permutation π such that, for all $i \in [k-1]$, it holds that $\bar{a}_i = \pi(\overline{\mathcal{H}}_i(\bar{a}_{[i-1]}))$.

complete sequence by $\overline{\mathcal{H}}^\pi \stackrel{\text{def}}{=} \overline{\mathcal{H}}^\pi_{[q]}$. We remove the overline and write $\mathcal{H}_{[k]}(\bar{a}_{[k-1]})$ (resp., $\mathcal{H}^\pi_{[k]}$) to denote the union of $\overline{\mathcal{H}}_{[k]}(\bar{a}_{[k-1]})$ (resp., $\overline{\mathcal{H}}^\pi_{[k]}$); that is, $\mathcal{H}_{[k]}(\bar{a}_{[k-1]}) \stackrel{\text{def}}{=} \bigcup_{i\in[k]} \mathcal{H}_i(\bar{a}_{[i-1]})$, and similarly $\mathcal{H}^\pi_{[k]} \stackrel{\text{def}}{=} \bigcup_{i\in[k]} \mathcal{H}^\pi_i$. We also denote the complementary sets of light locations: $\mathcal{L}_k(\bar{a}_{[k-1]}) \stackrel{\text{def}}{=} [n] \setminus \mathcal{H}_k(\bar{a}_{[k-1]})$ and $\mathcal{L}_{[k]}(\bar{a}_{[k-1]}) \stackrel{\text{def}}{=} [n] \setminus \mathcal{H}_{[k]}(\bar{a}_{[k-1]})$, and similarly $\mathcal{L}^\pi_k \stackrel{\text{def}}{=} [n] \setminus \mathcal{H}^\pi_k$, $\mathcal{L}^\pi_{[k]} \stackrel{\text{def}}{=} [n] \setminus \mathcal{H}^\pi_{[k]}$ and $\mathcal{L}^\pi \stackrel{\text{def}}{=} \mathcal{L}^\pi_{[q]}$.

Note that, similarly to $\overline{Q}^\pi_{[k]}(r)$, also $\overline{\mathcal{H}}^\pi_{[k]}$ does not depend on all of π, but rather only on its values on the prefix $\overline{\mathcal{H}}^\pi_{[k-1]}$. This property will be important for us later.

Note that each $\mathcal{H}^\pi_k$ satisfies that $|\mathcal{H}^\pi_k| \leq 1/\alpha$, thus $|\mathcal{H}^\pi| \leq q/\alpha$. Denote $h \stackrel{\text{def}}{=} q/\alpha$, the bound on the size of the heavy locations. We will later set α such that $h \leq n/2$, which ensures that the function G we will define next is far from PERM with high probability.

For each permutation π over $[n]$ and each function f from $[n]$ to $[n]$, let $G(\pi, f) : [n] \to [n]$ be the function that agrees with π on $\mathcal{H}^\pi$, and agrees with f on $\mathcal{L}^\pi$ (i.e., denoting $g = G(\pi, f)$, it holds that $g(i) = \pi(i)$ for every $i \in \mathcal{H}^\pi$, and $g(i) = f(i)$ for every $i \in \mathcal{L}^\pi$). Let Π be a random permutation over $[n]$ and let F be a random function from $[n]$ to $[n]$ such that Π and F are independent. Let $G = G(\Pi, F)$, and let $\epsilon > 0$ be a sufficiently small proximity parameter. For each fixed permutation π, since $|\mathcal{L}^\pi| \geq n/2$, Claim 4.1 implies that the probability that $G(\pi, F)$ is ϵ-close to PERM is at most $\exp(-\Omega(n))$. Hence, the probability that G is ϵ-close to PERM is at most $\exp(-\Omega(n))$.

For each permutation π and each randomness r of the interacting module, let $T(\pi, r)$ denote the transcript of the interaction with the honest prover on input π, when the interacting module uses randomness r. Consider the verifier's view when interacting with the honest prover on input Π:

$$X \stackrel{\text{def}}{=} (T(\Pi, R_I), R_I, \Pi(\overline{Q}^\Pi(R_Q)), R_Q)$$

For each fixed permutation π, define the cheating prover P_π that has π hard-wired and emulates the honest prover with input π. Consider the verifier's view when interacting with P_Π on input $G = G(\Pi, F)$. Note that the same permutation Π is used both in G and in the prover; in other words, we sample $\pi \sim \Pi$ and $f \sim F$, and consider the verifier's view when interacting with P_π on $G(\pi, f)$. The view is:

$$X' \stackrel{\text{def}}{=} (T(\Pi, R_I), R_I, G(\overline{Q}^G(R_Q)), R_Q)$$

We claim that the verifier must distinguish between the views X and X'.

Claim 5.2. *Let $\mathcal{D}$ denote the deciding module of the verifier. Then,*

$$\Pr[\mathcal{D}(X) = 1] - \Pr[\mathcal{D}(X') = 1] = \Omega(1)$$

Proof: We show that for all sufficiently large n's it must hold that

$$\Pr[\mathcal{D}(X) = 1] - \Pr[\mathcal{D}(X') = 1] \geq \frac{1}{3} - \Pr[G \text{ is } \epsilon\text{-close to PERM}] \tag{1}$$

Since $\Pr[G \text{ is } \epsilon\text{-close to PERM}] = o(1)$, this will establish the claim.

For each fixed permutation π and function f, let X_π denote the view X when Π is fixed to π, and let $X'_{\pi,f}$ denote the view X' when Π is fixed to π and F is fixed to f. That is, X_π is the view of the verifier when interacting with the honest prover on input π, and $X'_{\pi,f}$ is the view of the verifier when interacting with P_π on input $G(\pi, f)$. The completeness condition implies that for each fixed permutation π it holds that $\Pr[\mathcal{D}(X_\pi) = 1] \geq 2/3$. Therefore,

$$\Pr[\mathcal{D}(X) = 1] \geq \frac{2}{3} \tag{2}$$

For starters, suppose that the verifier is statistically sound (whereas we are only guaranteed that it is computationally sound). In this case, for each fixed permutation π and function f such that $G(\pi, f)$ is ϵ-far from $\texttt{PERM}$, it holds that $\Pr[\mathcal{D}(X'_{\pi,f}) = 1] \leq 1/3$. Therefore,

$$\begin{aligned}\Pr[\mathcal{D}(X') = 1] &= \sum_{\pi,f} \Pr[\Pi = \pi, F = f] \cdot \Pr[\mathcal{D}(X'_{\pi,f}) = 1] \\ &\leq \frac{1}{3} + \Pr_{\Pi,F}[G(\Pi, F) \text{ is } \epsilon\text{-close to } \texttt{PERM}]\end{aligned} \tag{3}$$

Combining Eq. (3) with Eq. (2) it follows that Eq. (1) holds for every n.

We turn to the (real) case where the verifier is only computationally sound. We show that for all sufficiently large n's, for every fixed permutation π over $[n]$ and function f from $[n]$ to $[n]$ such that $G(\pi, f)$ is ϵ-far from $\texttt{PERM}$, it holds that $\Pr[\mathcal{D}(X'_{\pi,f}) = 1] \leq 1/3$. This implies Eq. (3) holds for all sufficiently large n's, which together with Eq. (2) proves the claim.

Assume toward contradiction that for infinitely many n's there exist corresponding π_n and f_n such that $G(\pi_n, f_n)$ is ϵ-far from $\texttt{PERM}$ and yet $\Pr[\mathcal{D}(X'_{\pi_n,f_n}) = 1] > 1/3$. Consider a cheating prover that for each such n implements P_{π_n}. A polynomial-size family of circuits can implement this cheating prover, because it may have π_n hard-wired and emulate the polynomial-time honest prover. However, this cheating prover makes the verifier falsely accept $G(\pi_n, f_n)$ with probability greater than $1/3$ for infinity many n's, contradicting the computational soundness of the verifier. □

In contrast to Claim 5.2 we will show that the views X and X' are indistinguishable unless $q^5 \cdot c = \Omega(n/\log(n))$. We will show the indistinguishability claim even for each fixed value of R_I. Furthermore, we will show that the views are indistinguishable even when they are extended to include the value of Π on *all* the heavy locations.[12]

Lemma 5.3. *Let r_I be an arbitrary randomness string of the interacting module, and let $T(\Pi) \stackrel{def}{=} T(\Pi, r_I)$. Let*

$$\begin{aligned}Y &\stackrel{def}{=} \left(T(\Pi), \Pi(\overline{\mathcal{H}}^{\Pi}), \Pi\big(\overline{Q}^{\Pi}(R_Q) \cap \mathcal{L}^{\Pi}\big), R_Q\right) \\ Y' &\stackrel{def}{=} \left(T(\Pi), \Pi(\overline{\mathcal{H}}^{\Pi}), F\big(\overline{Q}^{G}(R_Q) \cap \mathcal{L}^{\Pi}\big), R_Q\right)\end{aligned}$$

Let $\gamma > 0$ be a sufficiently small constant ($\gamma \leq 1/64$ will suffice). If $q^5 \cdot c \leq \gamma \cdot \frac{n}{\log(n)}$, then $\Delta(Y, Y') = O\big(\big(\frac{q^5 \cdot c}{n/\log(n)}\big)^{1/6}\big)$.

Note that we can indeed reconstruct the complete sequence of answers $\Pi(\overline{Q}^{\Pi}(R_Q))$ from the pair $(\Pi(\overline{\mathcal{H}}^{\Pi}), \Pi(\overline{Q}^{\Pi}(R_Q) \cap \mathcal{L}^{\Pi}))$ (which is necessary for the indistinguishability of Y and Y' to imply that of X and X'). Specifically, we rely on the fact that from $\Pi(\overline{\mathcal{H}}^{\Pi})$ we can recursively recover $\overline{\mathcal{H}}^{\Pi}$, which allows us to determine for each query whether it belongs to $\mathcal{H}^{\Pi}$ or $\mathcal{L}^{\Pi}$, and if it is heavy, its position within $\overline{\mathcal{H}}^{\Pi}$.

Lemma 5.3 implies that Y and Y' (and hence X and X') are indistinguishable unless $q^5 \cdot c = \Omega(n/\log(n))$. To see this: Let $\gamma > 0$ be the constant in Lemma 5.3. Assume $q^5 \cdot c \neq \Omega(n/\log(n))$, meaning that for any $\delta > 0$, for infinitely many n's, it holds that $q^5 \cdot c \leq \delta \cdot n/\log(n)$. This implies that, for any $\delta < \gamma$, for infinitely many n's the lemma's condition that $q^5 \cdot c \leq \gamma \cdot n/\log(n)$ holds. Hence, by the lemma, for all those n's it holds that $\Delta(Y, Y') = O\big(\big(\frac{q^5 \cdot c}{n/\log(n)}\big)^{1/6}\big) = O(\delta^{1/6})$, meaning that Y and Y' are indistinguishable.

[12] Recall that for any random variables X and X', and any function f, it holds that $\Delta(f(X), f(X')) \leq \Delta(X, X')$.

Before proceeding to the proof of Lemma 5.3, we state a central lemma that will be needed for the proof. This lemma bounds the probability that the same light location is queried in two independent executions of the querying module, when querying a random permutation.

Lemma 5.4. *Let R_1, R_2 be two independent random variables, each distributed identically to R_Q. If $q = O(\sqrt{n})$, then:*

$$\Pr_{R_1,R_2,\Pi}\left[Q^{\Pi}(R_1) \cap Q^{\Pi}(R_2) \cap \mathcal{L}^{\Pi} \neq \emptyset\right] = O(q^2 \cdot \alpha).$$

Intuitively, this lemma holds due to the bound on the probability of querying light locations. However, the two executions query the same random permutation Π, introducing a dependency that necessitates a more delicate argument. We first prove Lemma 5.3 assuming Lemma 5.4, and then prove Lemma 5.4 in Section 5.1.1.

Proof of Lemma 5.3. By Pinsker's inequality it holds that $\Delta(Y, Y') \leq \sqrt{\frac{1}{2}\,\mathrm{D}(Y \,||\, Y')}$, where D is the KL divergence. Therefore, we focus on upper bounding $\mathrm{D}(Y \,||\, Y')$. Let Y_r and Y'_r denote the views Y and Y' when fixing R_Q to r, and note that $\mathrm{D}(Y \,||\, Y') = \mathbb{E}_{r\sim R_Q}\left[\mathrm{D}(Y_r \,||\, Y'_r)\right]$. As for $\mathrm{D}(Y_r \,||\, Y'_r)$, we have

$$\begin{aligned}
\mathrm{D}(Y_r \,||\, Y'_r) &= \mathrm{D}\big(T(\Pi), \Pi(\overline{\mathcal{H}}^{\Pi}), \Pi(\overline{Q}^{\Pi}(r) \cap \mathcal{L}^{\Pi}) \,\big|\big|\, T(\Pi), \Pi(\overline{\mathcal{H}}^{\Pi}), F(\overline{Q}^{G}(r) \cap \mathcal{L}^{\Pi})\big) \\
&= \mathrm{D}\big(\Pi(\overline{Q}^{\Pi}(r) \cap \mathcal{L}^{\Pi}) \,\big|\, \Pi(\overline{\mathcal{H}}^{\Pi}), T(\Pi) \,\big|\big|\, F(\overline{Q}^{G}(r) \cap \mathcal{L}^{\Pi}) \,\big|\, \Pi(\overline{\mathcal{H}}^{\Pi}), T(\Pi)\big) \\
&= \mathbb{E}_{\substack{\bar{b}\sim\Pi(\overline{Q}^{\Pi}(r)\cap\mathcal{L}^{\Pi}) \\ \bar{a}\sim\Pi(\overline{\mathcal{H}}^{\Pi}) \\ \tau\sim T(\Pi)}}\left[\log\left(\frac{1}{\Pr_{\Pi,F}\left[F(\overline{Q}^{G}(r)\cap\mathcal{L}^{\Pi}) = \bar{b} \,\middle|\, \Pi(\overline{\mathcal{H}}^{\Pi}) = \bar{a}, T(\Pi) = \tau\right)\right]}\right)\right] \qquad (4) \\
&\qquad - \mathrm{H}\big(\Pi(\overline{Q}^{\Pi}(r) \cap \mathcal{L}^{\Pi}) \,\big|\, \Pi(\overline{\mathcal{H}}^{\Pi}), T(\Pi)\big)
\end{aligned}$$

where the second equality uses the KL divergence chain rule (see also Claim 2.5), and the third equality is by the definition of conditional KL divergence. We can express the expectation in Eq. (4) as expectation over Π itself:

$$\mathbb{E}_{\pi\sim\Pi}\left[\log\left(\frac{1}{\Pr_{\Pi,F}\left[F(\overline{Q}^{G}(r)\cap\mathcal{L}^{\Pi}) = \pi(\overline{Q}^{\pi}(r)\cap\mathcal{L}^{\pi}) \,\middle|\, \Pi(\overline{\mathcal{H}}^{\Pi}) = \pi(\overline{\mathcal{H}}^{\pi}), T(\Pi) = T(\pi)\right]}\right)\right]$$

Consider the probability term in the denominator. Recall that the heavy locations of each permutation are determined only by the value of the permutation on previous heavy locations. This implies that the condition $\Pi(\overline{\mathcal{H}}^{\Pi}) = \pi(\overline{\mathcal{H}}^{\pi})$ is equivalent to $\Pi(\overline{\mathcal{H}}^{\pi}) = \pi(\overline{\mathcal{H}}^{\pi})$. Note that this condition also means that $\mathcal{L}^{\Pi} = \mathcal{L}^{\pi}$. Similarly, the queries to each function are determined only by the function's value on previous queries. The fact that Π agrees with π on the heavy locations and F agrees with π on the light queries means that G agrees with π on all queries, which in turn means that we can replace $\overline{Q}^{G}(r)$ with $\overline{Q}^{\pi}(r)$. Thus, the probability term becomes:

$$\begin{aligned}
&\Pr_{\Pi,F}\left[F(\overline{Q}^{G}(r)\cap\mathcal{L}^{\pi}) = \pi(\overline{Q}^{\pi}(r)\cap\mathcal{L}^{\pi}) \,\middle|\, \Pi(\mathcal{H}^{\pi}) = \pi(\mathcal{H}^{\pi}), T(\Pi) = T(\pi)\right] \\
&= \Pr_{\Pi,F}\left[F(Q^{\pi}(r)\cap\mathcal{L}^{\pi}) = \pi(Q^{\pi}(r)\cap\mathcal{L}^{\pi}) \,\middle|\, \Pi(\mathcal{H}^{\pi}) = \pi(\mathcal{H}^{\pi}), T(\Pi) = T(\pi)\right] \\
&= \Pr_{F}[F(Q^{\pi}(r)\cap\mathcal{L}^{\pi}) = \pi(Q^{\pi}(r)\cap\mathcal{L}^{\pi})] \\
&= \frac{1}{n^{|Q^{\pi}(r)\cap\mathcal{L}^{\pi}|}}
\end{aligned}$$

where the second equality is since F and Π are independent. Denoting $\overline{Q}_L^\pi(r) \stackrel{\text{def}}{=} \overline{Q}^\pi(r) \cap \mathcal{L}^\pi$, and accordingly the set $Q_L^\pi(r) \stackrel{\text{def}}{=} Q^\pi(r) \cap \mathcal{L}^\pi$, and combining all the above, we get

$$\mathrm{D}(Y_r \,||\, Y'_r) = \mathbb{E}_{\pi\sim\Pi}\left[|Q_L^\pi(r)|\right] \cdot \log(n) - \mathrm{H}\big(\Pi(\overline{Q}_L^\Pi(r)) \,\big|\, \Pi(\overline{\mathcal{H}}^\Pi), T(\Pi)\big)$$

Taking expectation over $r \sim R_Q$, we have

$$\mathrm{D}(Y \,||\, Y') = \mathop{\mathbb{E}_{r\sim R_Q}}_{\pi\sim\Pi}\left[|Q_L^\pi(r)|\right] \cdot \log(n) - \mathbb{E}_{r\sim R_Q}\left[\mathrm{H}\big(\Pi(\overline{Q}_L^\Pi(r)) \,\big|\, \Pi(\overline{\mathcal{H}}^\Pi), T(\Pi)\big)\right] \tag{5}$$

We next focus on lower bounding the second term of Eq. (5). Recall that we expect this term to be large since the light queries are dispersed across many locations, while the short transcript can only have significant information on a small number of them. To argue this formally, we consider multiple independent executions of the querying module. We show that we expect to see many different light locations across the executions, making their total entropy large, while the transcript can reduce this total entropy by at most c bits. Let $R_1, \ldots, R_t$ be t independent random variables, each distributed identically to R_Q, where $t \in \mathbb{N}$ is a parameter to be set later. Denote $\overline{R} = (R_1, \ldots, R_t)$ and $\bar{r} = (r_1, \ldots, r_t)$. Then,

$$\begin{aligned}
t \cdot \mathbb{E}_{r\sim R_Q}\left[\mathrm{H}\big(\Pi(\overline{Q}_L^\Pi(r)) \,\big|\, \Pi(\overline{\mathcal{H}}^\Pi), T(\Pi)\big)\right] &= \sum_{i\in[t]} \mathbb{E}_{r_i\sim R_i}\left[\mathrm{H}\big(\Pi(\overline{Q}_L^\Pi(r_i)) \,\big|\, \Pi(\overline{\mathcal{H}}^\Pi), T(\Pi)\big)\right] \\
&= \mathbb{E}_{\bar{r}\sim\overline{R}}\left[\sum_{i\in[t]} \mathrm{H}\big(\Pi(\overline{Q}_L^\Pi(r_i)) \,\big|\, \Pi(\overline{\mathcal{H}}^\Pi), T(\Pi)\big)\right] \\
&\geq \mathbb{E}_{\bar{r}\sim\overline{R}}\left[\mathrm{H}\left(\left(\Pi(\overline{Q}_L^\Pi(r_i))\right)_{i\in[t]} \,\middle|\, \Pi(\overline{\mathcal{H}}^\Pi), T(\Pi)\right)\right] \\
&\geq \mathbb{E}_{\bar{r}\sim\overline{R}}\left[\mathrm{H}\left(\left(\Pi(\overline{Q}_L^\Pi(r_i))\right)_{i\in[t]} \,\middle|\, \Pi(\overline{\mathcal{H}}^\Pi)\right)\right] - c
\end{aligned} \tag{6}$$

where the last inequality uses the fact that, for any random variables X, Y, Z, it holds that $\mathrm{H}(X \,|\, Y, Z) \geq \mathrm{H}(X \,|\, Y) - \mathrm{H}(Z)$ (see Claim 2.4), and that $\mathrm{H}\big(T(\Pi)\big) \leq c$ since $|T(\Pi)| \leq c$. Now, for each $\bar{r}$ it holds that

$$\begin{aligned}
&\mathrm{H}\left(\left(\Pi(\overline{Q}_L^\Pi(r_i))\right)_{i\in[t]} \,\middle|\, \Pi(\overline{\mathcal{H}}^\Pi)\right) \\
&= \mathbb{E}_{\pi\sim\Pi}\left[\log\left(\frac{1}{\mathrm{Pr}_\Pi\left[\Pi\left(\bigcup_{i\in[t]} Q_L^\pi(r_i)\right) = \pi\left(\bigcup_{i\in[t]} Q_L^\pi(r_i)\right) \,\middle|\, \Pi(\mathcal{H}^\pi) = \pi(\mathcal{H}^\pi)\right]}\right)\right]
\end{aligned} \tag{7}$$

We next lower bound the logarithm term in Eq. (7). Recall that $h = q/\alpha$ bounds the size of $\mathcal{H}^\pi$ for every π. The logarithm term is lower bounded by

$$\begin{aligned}
&\log\left(\prod_{i\in\left[\left|\bigcup_{i\in[t]} Q_L^\pi(r_i)\right|\right]} (n - h + 1 - i)\right) \\
&\geq \log\left((n - h - t\cdot q)^{\left|\bigcup_{i\in[t]} Q_L^\pi(r_i)\right|}\right) \\
&\geq \left|\bigcup_{i\in[t]} Q_L^\pi(r_i)\right| \cdot \log\left(n - 2h\right) \\
&\geq \left|\bigcup_{i\in[t]} Q_L^\pi(r_i)\right| \cdot \left(\log(n) - O\left(\frac{h}{n}\right)\right)
\end{aligned} \tag{8}$$

where the first inequality uses that $|Q_L^\pi(r_i)| \le q$, the second inequality assumes that $h \ge t \cdot q$ (which will be verified once we set t and h), and the last inequality uses $\log(n-2h) = \log(n) + \log\left(1-\frac{2h}{n}\right)$ and $-\log\left(1-\frac{2h}{n}\right) = O\left(\frac{h}{n}\right)$ which assumes that $\frac{2h}{n} < \frac{1}{2}$ (also to be verified later). Combining Eq. (8) with Eq. (7), we get

$$\mathrm{H}\left(\left(\Pi(\overline{Q}_L^\Pi(r_i))\right)_{i\in[t]} \,\middle|\, \Pi(\overline{\mathcal{H}}^\Pi)\right) \ge \mathbb{E}_{\pi\sim\Pi}\left[\left|\bigcup_{i\in[t]} Q_L^\pi(r_i)\right|\right] \cdot \left(\log(n) - O\left(\frac{h}{n}\right)\right) \tag{9}$$

Using $\left|\bigcup_{i\in[t]} Q_L^\pi(r_i)\right| \ge \sum_{i\in[t]} |Q_L^\pi(r_i)| - \sum_{i,j\in\binom{t}{2}} |Q_L^\pi(r_i)\cap Q_L^\pi(r_j)|$, we have

$$\mathbb{E}_{\substack{\bar{r}\sim\overline{R}\\ \pi\sim\Pi}}\left[\left|\bigcup_{i\in[t]} Q_L^\pi(r_i)\right|\right] \ge t\cdot \mathbb{E}_{\substack{r\sim R_Q\\ \pi\sim\Pi}}\left[|Q_L^\pi(r)|\right] - t^2\cdot \mathbb{E}_{\substack{r_1\sim R_1\\ r_2\sim R_2\\ \pi\sim\Pi}}\left[|Q_L^\pi(r_1)\cap Q_L^\pi(r_2)|\right] \tag{10}$$

Combining Eq. (6), (9) and (10) and dividing by t, we have

$$\begin{aligned}
&\mathbb{E}_{r\sim R_Q}\left[\mathrm{H}\big(\Pi(\overline{Q}_L^\Pi(r)) \,\big|\, \Pi(\overline{\mathcal{H}}^\Pi), T(\Pi)\big)\right]\\
&\ge \left(\mathbb{E}_{\substack{r\sim R_Q\\ \pi\sim\Pi}}\left[|Q_L^\pi(r)|\right] - t\cdot \mathbb{E}_{\substack{r_1\sim R_1\\ r_2\sim R_2\\ \pi\sim\Pi}}\left[|Q_L^\pi(r_1)\cap Q_L^\pi(r_2)|\right]\right)\cdot\left(\log(n) - O\left(\frac{h}{n}\right)\right) - \frac{c}{t}\\
&\ge \mathbb{E}_{\substack{r\sim R_Q\\ \pi\sim\Pi}}\left[|Q_L^\pi(r)|\right]\cdot\log(n) - t\cdot \mathbb{E}_{\substack{r_1\sim R_1\\ r_2\sim R_2\\ \pi\sim\Pi}}\left[|Q_L^\pi(r_1)\cap Q_L^\pi(r_2)|\right]\cdot\log(n) - O\left(\frac{q\cdot h}{n}\right) - \frac{c}{t}
\end{aligned}$$

where in the last inequality we have used $\mathbb{E}\left[|Q_L^\pi(r)|\right] \le q$. Combining this with Eq. (5), we have

$$\mathrm{D}(Y \,||\, Y') \le t\cdot \mathbb{E}_{\substack{r_1\sim R_1\\ r_2\sim R_2\\ \pi\sim\Pi}}\left[|Q_L^\pi(r_1)\cap Q_L^\pi(r_2)|\right]\cdot\log(n) + O\left(\frac{q\cdot h}{n}\right) + \frac{c}{t} \tag{11}$$

Note that

$$\mathbb{E}_{\substack{r_1\sim R_1\\ r_2\sim R_2\\ \pi\sim\Pi}}\left[|Q_L^\pi(r_1)\cap Q_L^\pi(r_2)|\right] \le \Pr_{R_1,R_2,\Pi}\left[Q_L^\Pi(R_1)\cap Q_L^\Pi(R_2)\neq\emptyset\right]\cdot q \tag{12}$$

By Lemma 5.4, it holds that

$$\Pr_{R_1,R_2,\Pi}\left[Q_L^\Pi(R_1)\cap Q_L^\Pi(R_2)\neq\emptyset\right] = O(q^2\cdot\alpha) \tag{13}$$

where we rely on our hypothesis that $q^5\cdot c \le \gamma\cdot\frac{n}{\log(n)}$, which ensures the hypothesis of Lemma 5.4 (i.e., $q = O(\sqrt{n})$). Combining Eq. (11), (12), and (13), we get

$$\mathrm{D}(Y \,||\, Y') = O\left(t\cdot q^3\cdot\alpha\cdot\log(n) + \frac{q\cdot h}{n} + \frac{c}{t}\right)$$

Recall that $h = q/\alpha$. Setting $h = \frac{c\cdot n}{t\cdot q}$ and $t = \left(\frac{c^2\cdot n}{q^5\log(n)}\right)^{1/3}$, we get

$$\mathrm{D}(Y \,||\, Y') = O\left(\frac{t\cdot q^4}{h}\cdot\log(n) + \frac{q\cdot h}{n} + \frac{c}{t}\right) = O\Big(\Big(\frac{q^5\cdot c}{n/\log(n)}\Big)^{1/3}\Big)$$

Since $\Delta(Y,Y') \leq \sqrt{\frac{1}{2}\,\mathrm{D}(Y \,||\, Y')}$, we have that $\Delta(Y,Y') = O\Big(\Big(\frac{q^5 \cdot c}{n/\log(n)}\Big)^{1/6}\Big)$, as claimed.

We are left to verify the assumptions we had in Eq. (8), that $h \leq n/4$ and $h \geq t \cdot q$. (We also had the assumption that $h \leq n/2$ to ensure that G is ϵ-far from PERM w.h.p., but this is already implied by $h \leq n/4$.) Recall our hypothesis that $q^5 \cdot c \leq \gamma \cdot n/\log(n)$ for a sufficiently small $\gamma > 0$. This hypothesis implies that $h = (c \cdot n^2 \cdot q^2 \cdot \log(n))^{1/3} \leq \gamma^{1/3} \cdot n/q$, which is less than $n/4$ for sufficiently small γ. Turning to verify the assumption $h \geq t \cdot q$, we have $h/tq = (q^4 \cdot n \cdot \log^2(n)/c)^{1/3} \geq \gamma^{-1/3} q^3 \log(n) \geq 1$, as desired. □

Completing the proof of Theorem 5.1. Lemma 5.3 implies that unless $q^5 \cdot c = \Omega(n/\log(n))$ it holds that X and X' are indistinguishable (even for any fixed value of R_I, and even when X and X' are extended to include the value of Π on all the heavy locations). On the other hand, by Claim 5.2, the verifier must distinguish between X and X'. Thus, we conclude that $q^5 \cdot c = \Omega(n/\log(n))$ must hold, completing the proof.

5.1.1 Proof of Lemma 5.4.

Recall that in Lemma 5.4 we want to bound the probability of a collision between the light queries of two independent executions querying a random permutation. Specifically, we want to show that:

$$p \stackrel{\text{def}}{=} \Pr\left[Q^{\Pi}_{[q]}(R_1) \cap Q^{\Pi}_{[q]}(R_2) \cap \mathcal{L}^{\Pi}_{[q]} \neq \emptyset\right] = O(q^2 \cdot \alpha)$$

For intuition, consider first the simplified case where all locations are light across all queries; that is, for each query $k \in [q]$ and location $i \in [n]$, we have $\Pr_{\Pi,R_Q}[Q^{\Pi}_k(R_Q) = i] < \alpha$. We aim to show that for any $k, l \in [q]$, the probability that $Q^{\Pi}_k(R_1) = Q^{\Pi}_l(R_2)$ is $O(\alpha)$. What prevents us from deducing this immediately from the condition on light locations is the dependence between $Q^{\Pi}_k(R_1)$ and $Q^{\Pi}_l(R_2)$ through Π. However, $Q^{\Pi}_k(R_1)$ and $Q^{\Pi}_l(R_2)$ depend on Π only through its value on the previous queries, $\Pi(\overline{Q}^{\Pi}_{[k-1]}(R_1))$ and $\Pi(\overline{Q}^{\Pi}_{[l-1]}(R_2))$, respectively. If we assume inductively that there are no collisions between $Q^{\Pi}_{[k-1]}(R_1)$ and $Q^{\Pi}_{[l-1]}(R_2)$, then $\Pi(\overline{Q}^{\Pi}_{[k-1]}(R_1))$ and $\Pi(\overline{Q}^{\Pi}_{[l-1]}(R_2))$ are "almost independent", since a permutation restricted to a small set of locations is close to a random function. We will show that this near-independence is sufficient.

Proceeding to the proof, we begin by decomposing the probability of a collision in the first q light queries based on whether there is a collision in the previous $q-1$ light queries, and continue recursively:

$$\begin{aligned}
p &= \Pr\left[Q^{\Pi}_{[q]}(R_1) \cap Q^{\Pi}_{[q]}(R_2) \cap \mathcal{L}^{\Pi}_{[q]} \neq \emptyset\right] \\
&\leq \Pr\left[\begin{array}{cc} & Q^{\Pi}_{[q]}(R_1) \cap Q^{\Pi}_{[q]}(R_2) \cap \mathcal{L}^{\Pi}_{[q]} \neq \emptyset \\ \wedge & Q^{\Pi}_{[q-1]}(R_1) \cap Q^{\Pi}_{[q-1]}(R_2) \cap \mathcal{L}^{\Pi}_{[q-1]} = \emptyset \end{array}\right] + \Pr\left[Q^{\Pi}_{[q-1]}(R_1) \cap Q^{\Pi}_{[q-1]}(R_2) \cap \mathcal{L}^{\Pi}_{[q-1]} \neq \emptyset\right] \\
&\vdots \\
&\leq \sum_{k \in [q]} \Pr\left[\begin{array}{cc} & Q^{\Pi}_{[k]}(R_1) \cap Q^{\Pi}_{[k]}(R_2) \cap \mathcal{L}^{\Pi}_{[k]} \neq \emptyset \\ \wedge & Q^{\Pi}_{[k-1]}(R_1) \cap Q^{\Pi}_{[k-1]}(R_2) \cap \mathcal{L}^{\Pi}_{[k-1]} = \emptyset \end{array}\right]
\end{aligned}$$

We next take a union bound over the possible collisions. If there are no collisions between the first $k-1$ light queries, then for a collision to exist between the first k light queries, it must involve $Q^{\Pi}_k(R_1)$ or $Q^{\Pi}_k(R_2)$. Specifically, there must exist some $l \in [k]$ such that $Q^{\Pi}_k(R_1) = Q^{\Pi}_l(R_2)$ and $Q^{\Pi}_l(R_2) \in \mathcal{L}^{\Pi}_{[k]}$, or the same with R_1 and R_2 reversed (which give us a factor 2). Furthermore, we

can replace the condition $Q_l^\Pi(R_2) \in \mathcal{L}_{[k]}^\Pi$ with the weaker condition $Q_l^\Pi(R_2) \in \mathcal{L}_k^\Pi$ (since $\mathcal{L}_{[k]}^\Pi$, the set of locations that are light in all the first k queries, is a subset of $\mathcal{L}_k^\Pi$, the set of locations that are light in the k^{th} query). Similarly, we can replace the condition $Q_{[k-1]}^\Pi(R_1) \cap Q_{[k-1]}^\Pi(R_2) \cap \mathcal{L}_{[k-1]}^\Pi = \emptyset$ with the weaker condition $Q_{[k-1]}^\Pi(R_1) \cap Q_{[l-1]}^\Pi(R_2) \cap \mathcal{L}_{[k-1]}^\Pi = \emptyset$ (since $Q_{[l-1]}^\Pi(R_2)$ is a subset of $Q_{[k-1]}^\Pi(R_2)$). We get:

$$p \leq 2 \cdot \sum_{k \in [q]} \sum_{l \in [k]} \Pr \left[\begin{array}{c} Q_k^\Pi(R_1) = Q_l^\Pi(R_2) \ \ \wedge \ \ Q_l^\Pi(R_2) \in \mathcal{L}_k^\Pi \\ \wedge \ \ Q_{[k-1]}^\Pi(R_1) \cap Q_{[l-1]}^\Pi(R_2) \cap \mathcal{L}_{[k-1]}^\Pi = \emptyset \end{array} \right] \tag{14}$$

We next show that each term in the summation is bounded by $O(\alpha)$. Consider any $k \in [q]$ and $l \in [k]$. We will show the bound by showing it holds even when conditioning on any value of Π on $\overline{\mathcal{H}}_{[k-1]}^\Pi$ and $\overline{Q}_{[l-1]}^\Pi(R_2) \cap \mathcal{L}_{[k-1]}^\Pi$, and on any value of R_2. We begin by conditioning on the value of Π on $\overline{\mathcal{H}}_{[k-1]}^\Pi$. Specifically, we condition on $\Pi(\overline{\mathcal{H}}_{[k-1]}(\bar{a}_{[k-2]})) = \bar{a}_{[k-1]}$, where each $\bar{a}_i$ is an arbitrary assignment to the elements in $\mathcal{H}_i(\bar{a}_{[i-1]})$. Note that this conditioning fixes $\mathcal{L}_{[k-1]}^\Pi$ to $\mathcal{L}_{[k-1]}(\bar{a}_{[k-2]})$ and $\mathcal{L}_k^\Pi$ to $\mathcal{L}_k(\bar{a}_{[k-1]})$. We get:

$$\Pr \left[\begin{array}{c} Q_k^\Pi(R_1) = Q_l^\Pi(R_2) \ \ \wedge \ \ Q_l^\Pi(R_2) \in \mathcal{L}_k(\bar{a}_{[k-1]}) \\ \wedge \ \ Q_{[k-1]}^\Pi(R_1) \cap Q_{[l-1]}^\Pi(R_2) \cap \mathcal{L}_{[k-1]}(\bar{a}_{[k-2]}) = \emptyset \end{array} \ \middle| \ \Pi(\overline{\mathcal{H}}_{[k-1]}(\bar{a}_{[k-2]})) = \bar{a}_{[k-1]} \right]$$

Notice that if $Q_k^\Pi(R_1)$ was independent of $Q_l^\Pi(R_2)$ conditioned on $\Pi(\overline{\mathcal{H}}_{[k-1]}(\bar{a}_{[k-2]})) = \bar{a}_{[k-1]}$, then we would be done, since by definition of $\mathcal{H}_k(\bar{a}_{[k-1]})$, the probability of $Q_k^\Pi(R_1)$ equaling a location in $\mathcal{L}_k(\bar{a}_{[k-1]})$ given that $\Pi(\overline{\mathcal{H}}_{[k-1]}(\bar{a}_{[k-2]})) = \bar{a}_{[k-1]}$ is less than α. However, $Q_k^\Pi(R_1)$ and $Q_l^\Pi(R_2)$ are dependent through Π. Recall that $Q_k^\Pi(R_1)$ and $Q_l^\Pi(R_2)$ depend on Π only through its values on the previous queries $\overline{Q}_{[k-1]}^\Pi(R_1)$ and $\overline{Q}_{[l-1]}^\Pi(R_2)$ respectively. Furthermore, we have that the light locations within $\overline{Q}_{[k-1]}^\Pi(R_1)$ and $\overline{Q}_{[l-1]}^\Pi(R_2)$ are disjoint, and the heavy locations are fixed. Per the discussion at the beginning of the proof, we use the fact that the light queries are disjoint, and their number is small (at most $q = O(\sqrt{n})$), to argue that the value of Π on them are "almost independent" (even conditioned on $\Pi(\overline{\mathcal{H}}_{[k-1]}(\bar{a}_{[k-2]})) = \bar{a}_{[k-1]}$). Specifically, we condition on an arbitrary value of Π on the light queries from the second execution (i.e., $\Pi(\overline{Q}_{[l-1]}^\Pi(R_2) \cap \mathcal{L}_{[k-1]}(\bar{a}_{[k-2]}))$), and show that the conditional probability is at most an $O(1)$ factor larger than the same probability without this condition.

First, we fix R_2 to an arbitrary value r_2. Now, let $\bar{b}_{[l-1]}$ be an arbitrary sequence, such that for each $i \in [l-1]$, if $Q_i(\bar{b}_{[i-1]}, r_2)$ is heavy (i.e., $Q_i(\bar{b}_{[i-1]}, r_2) \in \mathcal{H}_{[k-1]}(\bar{a}_{[k-2]})$), then b_i equals to $\bar{a}_{[k-1]}$ at the location corresponding to $Q_i(\bar{b}_{[i-1]}, r_2)$. We also require that $\bar{b}_{[l-1]}$ leads to a light l^{th} query; that is, $Q_l(\bar{b}_{[l-1]}, r_2) \in \mathcal{L}_k(\bar{a}_{[k-1]})$. Let $\bar{b}_L$ denote the subsequence of $\bar{b}_{[l-1]}$ at indices i for which $Q_i(\bar{b}_{[i-1]}, r_2) \in \mathcal{L}_{[k-1]}(\bar{a}_{[k-2]})$. We condition on $\Pi(\overline{Q}_{[l-1]}(\bar{b}_{[l-2]}, r_2) \cap \mathcal{L}_{[k-1]}(\bar{a}_{[k-2]})) = \bar{b}_L$. To simplify the expressions that follow, we write $\overline{\mathcal{H}}_{[k-1]} = \overline{\mathcal{H}}_{[k-1]}(\bar{a}_{[k-2]})$, and $\mathcal{L}_{[k-1]} = \mathcal{L}_{[k-1]}(\bar{a}_{[k-2]})$. We have:

$$\Pr \left[\begin{array}{c} Q_k^\Pi(R_1) = Q_l(\bar{b}_{[l-1]}, r_2) \\ \wedge \ Q_{[k-1]}^\Pi(R_1) \cap Q_{[l-1]}(\bar{b}_{[l-2]}, r_2) \cap \mathcal{L}_{[k-1]} = \emptyset \end{array} \ \middle| \ \begin{array}{c} \Pi(\overline{\mathcal{H}}_{[k-1]}) = \bar{a}_{[k-1]}, \\ \Pi(\overline{Q}_{[l-1]}(\bar{b}_{[l-2]}, r_2) \cap \mathcal{L}_{[k-1]}) = \bar{b}_L \end{array} \right] \tag{15}$$

In order to upper bound Eq. (15), we first establish a general claim:

Claim: Let Π be a uniform permutation over $[n]$. Let $\overline{S}_1, \overline{S}_2, \overline{S}_3$ be three disjoint sequences of distinct elements in $[n]$. Let $\bar{a}, \bar{b}, \bar{c}$ be disjoint sequences of distinct elements in $[n]$ such that $|\bar{a}| = |\overline{S}_1|$,

$|\bar{b}| = |\overline{S}_2|$ and $|\bar{c}| = |\overline{S}_3|$. Then:

$$\Pr[\Pi(\overline{S}_3) = \bar{c} \mid \Pi(\overline{S}_1) = \bar{a}, \Pi(\overline{S}_2) = \bar{b}] \leq \left(\frac{n - |\bar{a}|}{n - |\bar{a}| - |\bar{b}| - |\bar{c}|}\right)^{|\bar{c}|} \cdot \Pr[\Pi(\overline{S}_3) = \bar{c} \mid \Pi(\overline{S}_1) = \bar{a}]$$

Proof: We have:

$$\frac{\Pr[\Pi(\overline{S}_3) = \bar{c} \mid \Pi(\overline{S}_1) = \bar{a}, \Pi(\overline{S}_2) = \bar{b}]}{\Pr[\Pi(\overline{S}_3) = \bar{c} \mid \Pi(\overline{S}_1) = \bar{a}]} = \frac{\prod_{i \in [|\bar{c}|]}(n - |\bar{a}| + 1 - i)}{\prod_{i \in [|\bar{c}|]}(n - |\bar{a}| - |\bar{b}| + 1 - i)} \leq \frac{(n - |\bar{a}|)^{|\bar{c}|}}{(n - |\bar{a}| - |\bar{b}| - |\bar{c}|)^{|\bar{c}|}}$$

□

Using the claim, we show that Eq. (15) is upper bounded by:

$$\left(\frac{n-h}{n-h-l-k}\right)^k \cdot \Pr\left[\begin{array}{c} Q_k^{\Pi}(R_1) = Q_l(\bar{b}_{[l-1]}, r_2) \\ \wedge\ Q_{[k-1]}^{\Pi}(R_1) \cap Q_{[l-1]}(\bar{b}_{[l-2]}, r_2) \cap \mathcal{L}_{[k-1]} = \emptyset \end{array} \;\middle|\; \Pi(\overline{\mathcal{H}}_{[k-1]}) = \bar{a}_{[k-1]} \right] \tag{16}$$

To do so we decompose Eq. (15) in terms of all possible values of $R_1 = r_1$ and all possible values $\bar{c}_{[k-1]}$ for fixing $\Pi(\overline{Q}_{[k-1]}^{\Pi}(r_1) \cap \mathcal{L}_{[k-1]})$, defined analogously to the values $\bar{b}_{[l-1]}$ we used for fixing $\Pi(\overline{Q}_{[l-1]}^{\Pi}(r_2) \cap \mathcal{L}_{[k-1]})$. We sum only over those $\bar{c}_{[k-1]}$ that satisfy the condition $Q_k(\bar{c}_{[k-1]}, r_1) = Q_l(\bar{b}_{[l-1]}, r_2)$ and the disjointness condition $Q_{[k-1]}(\bar{c}_{[k-2]}, r_1) \cap Q_{[l-1]}(\bar{b}_{[l-2]}, r_2) \cap \mathcal{L}_{[k-1]} = \emptyset$. Each term in the decomposition has the form

$$\Pr\left[\; \Pi(\overline{Q}_{[k-1]}(\bar{c}_{[k-2]}, r_1) \cap \mathcal{L}_{[k-1]}) = \bar{c}_L \;\middle|\; \begin{array}{c} \Pi(\overline{\mathcal{H}}_{[k-1]}) = \bar{a}_{[k-1]}, \\ \Pi(\overline{Q}_{[l-1]}(\bar{b}_{[l-2]}, r_2) \cap \mathcal{L}_{[k-1]}) = \bar{b}_L \end{array} \right]$$

We can apply the claim to each such term since the three sequences involved are disjoint (by the disjointness condition above, and the fact that the heavy and light locations are disjoint). Applying the claim to each term, while using the size bounds $|\bar{c}_L| \leq |\bar{c}_{[k-1]}| < k$, $|\bar{b}_L| \leq |\bar{b}_{[l-1]}| < l$, and $|\bar{a}_{[k-1]}| \leq h$, gives us Eq. (16).

We turn to upper bound the term $\left(\frac{n-h}{n-h-l-k}\right)^k$ in Eq. (16). For this, we will use the Lemma's hypothesis that $q = O(\sqrt{n})$. We have that

$$\left(\frac{n-h}{n-h-l-k}\right)^k \leq \left(\frac{n-h}{n-h-2q}\right)^q = \left(1 + \frac{2q}{n-h-2q}\right)^q = \left(1 + O\left(\frac{q}{n}\right)\right)^q \leq e^{O(\frac{q^2}{n})} = O(1)$$

where the first inequality uses $k, l \leq q$, the second equality uses $h \leq n/2$ and $q = O(\sqrt{n})$, and the last equality also uses $q = O(\sqrt{n})$. As for the probability term in Eq. (16), it is upper bounded by

$$\Pr\left[Q_k^{\Pi}(R_1) = Q_l(\bar{b}_{[l-1]}, r_2) \;\middle|\; \Pi(\overline{\mathcal{H}}_{[k-1]}) = \bar{a}_{[k-1]}\right] < \alpha \tag{17}$$

where the inequality is by definition of $\mathcal{H}_k(\bar{a}_{[k-1]})$, since $Q_l(\bar{b}_{[l-1]}, r_2) \in \mathcal{L}_k(\bar{a}_{[k-1]})$. Thus, Eq. (16), and hence Eq. (15), is bounded by $O(\alpha)$. Since this is true for all $\bar{a}_{[k-1]}$, $\bar{b}_{[l-1]}$ and r_2, we have that each term in the summation in Eq. (14) is bounded by $O(\alpha)$. Hence, we get that $p = O(q^2 \cdot \alpha)$, as claimed. □

An inessential comment about the proofs of Lemma 5.3 and Lemma 5.4. Note that in the analysis of Lemma 5.3 and Lemma 5.4 we have mainly considered the distribution of Π when its value on the heavy locations is fixed arbitrarily. We could have conditioned a priori on an arbitrary fixed value for $\Pi(\mathcal{H}^{\Pi})$, rather than making the argument for a random value of $\Pi(\mathcal{H}^{\Pi})$ (and considering the expectation over all possible fixed values for it). This would have complicated the analysis of Lemma 5.4, since there we conditioned only on the value of Π on a prefix of the heavy locations $\mathcal{H}^{\Pi}_{[k-1]}$ for each $k \in [q]$, rather than on all heavy locations (which is needed in order to invoke the definition of the k^{th} heavy set in Eq. (17)). Nevertheless, we could have fixed the value of Π on all heavy locations a priori and removed the extra conditioning on the heavy locations of the last $q-k-1$ queries, similar to how we removed the conditioning on the light queries in Eq. (16). To ensure that this does not have a significant effect on the probabilities, we would have needed to ensure that the number of heavy locations is sufficiently small (specifically, we would have needed to ensure that $h = O(n/q)$, which holds under our current parameter setting).

6 A lower bound on the isolated model with non-adaptive queries for PERM

In this section we present a lower bound on isolated cs-IPPs for PERM in the special case where the verifier uses non-adaptive queries.

Theorem 6.1. *If* PERM *can be verified by a computationally-sound isolated IPP that uses non-adaptive queries and has query complexity $q > 0$ and communication complexity $c > 0$, then $q^3 \cdot c = \Omega(n)$.*

Before proceeding to the proof, we establish a lemma that bounds the total information that any c-bit transcript $T(\Pi)$ can have on individual locations in a random permutation Π, when it is combined with the value of Π at any $q-1$ other locations.

Lemma 6.2. *Let $n \in \mathbb{N}$, and let $q, c \in \mathbb{N}\setminus\{0\}$ such that $q \cdot c = O(n)$. Let Π be a random permutation over $[n]$. Let $T(\Pi)$ be any random variable over $\{0,1\}^c$ that may depend on Π. For each $i \in [n]$, let S_i be any subset of $[n] \setminus \{i\}$ such that $|S_i| < q$. Then,*

$$\sum_{i\in[n]} \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi)\big) = O\left(\sqrt{q \cdot c \cdot n}\right).$$

Proof: First, it holds that

$$\sum_{i\in[n]} \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi)\big) = \sum_{i\in[n]} \mathrm{H}\big(\Pi(i)\big) - \sum_{i\in[n]} \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i), T(\Pi)\big) \tag{18}$$

For the first term of Eq. (18), since each $\Pi(i)$ is uniform over $[n]$ we have that

$$\sum_{i\in[n]} \mathrm{H}\big(\Pi(i)\big) = n \cdot \log(n) \tag{19}$$

To lower bound the second term of Eq. (18), we first establish the following:

Claim: There exists a partition P of $[n]$ into $O(q)$ parts such that for each part $B \in P$ and each $i \in B$ it holds that S_i does not intersect B.

Proof: Consider a directed graph with vertex set $[n]$, where each vertex i has outgoing edges to all vertices in S_i. Notice that the desired partition is equivalent to a $O(q)$-coloring of this graph

(i.e., a coloring of the vertices with $O(q)$ colors, such that every two nodes that are connected by a directed edge are assigned different colors). Such a coloring exists since the out-degree of each node in the graph is upper bounded by q (see Appendix A). $\square$

Let $k \in \mathbb{N} \setminus \{0\}$ be a parameter we will set later, satisfying $k \leq n/2q$. Consider a partition P of $[n]$ obtained by taking the $O(q)$-way partition from the previous claim and splitting each part into subparts of size k (except possibly one smaller subpart per original part). The resulting partition has at most $n/k + O(q) = O(n/k)$ parts. For each part $B \in P$, define $S_B = \bigcup_{i \in B} S_i$. Note that $|S_B| \leq |B| \cdot \max_{i \in B} \{|S_i|\} \leq k \cdot (q-1)$, and that B and S_B do not intersect. Then,

$$\begin{aligned}\sum_{i \in [n]} \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i), T(\Pi)\big) &= \sum_{B \in P} \sum_{i \in B} \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i), T(\Pi)\big) \\ &\geq \sum_{B \in P} \sum_{i \in B} \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_B), T(\Pi)\big) \\ &\geq \sum_{B \in P} \mathrm{H}\big(\Pi(B) \,\big|\, \Pi(S_B), T(\Pi)\big)\end{aligned} \tag{20}$$

Recall that for any random variables X, Y, Z, it holds that $\mathrm{H}(X \,|\, Y, Z) \geq \mathrm{H}(X \,|\, Y) - \mathrm{H}(Z)$ (see Claim 2.4). Therefore, for each $B \in P$ it holds that

$$\begin{aligned}\mathrm{H}\big(\Pi(B) \,\big|\, \Pi(S_B), T(\Pi)\big) &\geq \mathrm{H}\big(\Pi(B) \,\big|\, \Pi(S_B)\big) - \mathrm{H}\big(T(\Pi)\big) \\ &\geq \mathrm{H}\big(\Pi(B) \,\big|\, \Pi(S_B)\big) - c\end{aligned} \tag{21}$$

since $|T(\Pi)| \leq c$. To lower bound $\mathrm{H}\big(\Pi(B) \,\big|\, \Pi(S_B)\big)$, we use the fact that B and S_B do not intersect:

$$\begin{aligned}\mathrm{H}\big(\Pi(B) \,\big|\, \Pi(S_B)\big) &= \log\left(\prod_{i \in [|B|]} (n - |S_B| + 1 - i)\right) \\ &> \log\left((n - |S_B| - |B|)^{|B|}\right) \\ &\geq |B| \cdot \log\left(n - k \cdot q\right) \\ &= |B| \cdot \left(\log(n) - O\left(\frac{k \cdot q}{n}\right)\right)\end{aligned} \tag{22}$$

where the last inequality uses $\log(n - k \cdot q) = \log(n) + \log\left(1 - \frac{k \cdot q}{n}\right)$, and $-\log\left(1 - \frac{k \cdot q}{n}\right) = O\left(\frac{k \cdot q}{n}\right)$ which relies on $\frac{k \cdot q}{n} < \frac{1}{2}$. Combining Eq. (20), (21) and (22), we get

$$\begin{aligned}\sum_{i \in [n]} \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i), T(\Pi)\big) &\geq \sum_{B \in P} |B| \cdot \left(\log(n) - O\left(\frac{k \cdot q}{n}\right)\right) - |P| \cdot c \\ &= n \cdot \left(\log(n) - O\left(\frac{k \cdot q}{n}\right)\right) - O\left(\frac{n}{k}\right) \cdot c\end{aligned}$$

since $|P| = O(n/k)$. Combining this with Eq. (19) and Eq. (18), we get that

$$\sum_{i \in [n]} \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi)\big) = O\left(k \cdot q + \frac{n \cdot c}{k}\right)$$

Setting $k = \Theta\left(\sqrt{\frac{c \cdot n}{q}}\right)$, we get that

$$\sum_{i \in [n]} \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi)\big) = O\left(\sqrt{q \cdot c \cdot n}\right)$$

as desired. Due to the lemma's hypothesis that $q \cdot c = O(n)$, by setting $k = \gamma \cdot \sqrt{\frac{c \cdot n}{q}}$ with a sufficiently small constant $\gamma > 0$, we can ensure that k satisfies our requirement that $k \leq n/2q$. ■

6.1 Proof of Theorem 6.1

Consider an arbitrary computationally sound isolated IPP that has communication complexity $c > 0$ and uses $q > 0$ non-adaptive queries. Without loss of generality, we assume that the querying module does not make the same query more than once.

Let R_Q and R_I denote the randomness of the querying and interacting modules, respectively. For each randomness string r of the querying module, let $Q(r)$ be the set of q (non-adaptive) queries made by the querying module with randomness r. Consider the following set of heavy locations, denoted

$$\mathcal{H} \stackrel{\text{def}}{=} \left\{ i \in [n] : \Pr_{R_Q}[i \in Q(R_Q)] \geq 2q/n \right\}$$

Additionally, denote the complementary set of light locations by $\mathcal{L} \stackrel{\text{def}}{=} [n] \setminus \mathcal{H}$. Note that $|\mathcal{H}| \leq n/2$ since $\sum_{i \in [n]} \Pr_{R_Q}[i \in Q(R_Q)] = q$.

Let Π be a random permutation over $[n]$ and let F be a random function from $[n]$ to $[n]$ such that Π and F are independent. Let $G = G(\Pi, F)$ be the function that agrees with Π on $\mathcal{H}$, and agrees with F on $\mathcal{L}$; that is, $G(i) = \Pi(i)$ for every $i \in \mathcal{H}$, and $G(i) = F(i)$ for every $i \in \mathcal{L}$. Since $|\mathcal{L}| \geq \frac{n}{2}$, Claim 4.1 implies that for all sufficiently small $\epsilon > 0$ it holds that G is ϵ-far from PERM with high probability.

Let $Q_H(r) \stackrel{\text{def}}{=} Q(r) \cap \mathcal{H}$ and $Q_L(r) \stackrel{\text{def}}{=} Q(r) \cap \mathcal{L}$ denote the heavy and light queries, respectively, when the querying module uses randomness r. For each permutation π and each randomness string r of the interacting module, let $T(\pi, r)$ denote the transcript of the interaction between the honest prover and the interacting module with randomness r on input π. Consider the verifier's view when interacting with the honest prover on input Π:

$$\begin{aligned} X &\stackrel{\text{def}}{=} (T(\Pi, R_I), R_I, \Pi(Q(R_Q)), R_Q) \\ &\equiv (T(\Pi, R_I), R_I, \Pi(Q_H(R_Q)), \Pi(Q_L(R_Q)), R_Q) \end{aligned} \tag{23}$$

For each fixed permutation π, define the cheating prover P_π that has π hard-wired and emulates the honest prover with input π. Consider the verifier's view when interacting with P_Π on input $G = G(\Pi, F)$. Note that the same permutation Π is used both in G and in the prover; in other words, we sample $\pi \sim \Pi$ and $f \sim F$, and consider the verifier's view when interacting with P_π on $G(\pi, f)$. The view is:

$$\begin{aligned} X' &\stackrel{\text{def}}{=} (T(\Pi, R_I), R_I, G(Q(R_Q)), R_Q) \\ &\equiv (T(\Pi, R_I), R_I, \Pi(Q_H(R_Q)), F(Q_L(R_Q)), R_Q) \end{aligned} \tag{24}$$

Since G is ϵ-far from PERM with high probability, the completeness and computational soundness conditions imply that the verifier must distinguish between X and X'; see Claim 5.2 for an identical argument. In contrast, we will show that the views X and X' are indistinguishable unless $q^3 \cdot c = \Omega(n)$. We will establish the indistinguishability claim even for any fixed value of R_I.

Claim 6.3. *Let r_I be an arbitrary randomness string of the interacting module, and let $T(\Pi) \stackrel{def}{=}$*

$T(\Pi, r_I)$.[13] *Let Y and Y' denote the views X and X' when fixing R_I to r_I:*

$$Y \stackrel{def}{=} (T(\Pi), \Pi(Q_H(R_Q)), \Pi(Q_L(R_Q)), R_Q)$$
$$Y' \stackrel{def}{=} (T(\Pi), \Pi(Q_H(R_Q)), F(Q_L(R_Q)), R_Q)$$

If $q \cdot c = O(n)$, then $\Delta(Y, Y') = O\big(\big(\frac{q^3 \cdot c}{n}\big)^{1/4}\big)$.

This claim implies that Y and Y' (and hence X and X') are indistinguishable unless $q^3 \cdot c = \Omega(n)$. To see this: Fix an arbitrary constant $\gamma > 0$. The claim implies that there exists some constant $\gamma' > 0$ such that for any sufficiently large n, if $q \cdot c \leq \gamma \cdot n$, then $\Delta(Y, Y') \leq \gamma' \cdot \big(\frac{q^3 \cdot c}{n}\big)^{1/4}$. Thus, for any (arbitrarily small) constant $\delta > 0$, if for infinitely many n's it holds that $q^3 \cdot c \leq \delta \cdot n$, then (since $q \cdot c \leq q^3 \cdot c \leq \delta \cdot n \leq \gamma \cdot n$ for sufficiently small δ) for infinitely many n's it holds that $\Delta(Y, Y') \leq \gamma' \cdot \big(\frac{q^3 \cdot c}{n}\big)^{1/4} \leq \gamma' \cdot \delta$, meaning that Y and Y' are indistinguishable.

Proof: By Pinsker's inequality it holds that $\Delta(Y, Y') \leq \sqrt{\frac{1}{2} \mathrm{D}(Y \,||\, Y')}$, where D is the KL divergence. Therefore, we focus on upper bounding $\mathrm{D}(Y \,||\, Y')$. Let Y_r and Y'_r denote the views Y and Y' when fixing R_Q to r; that is, $Y_r = (T(\Pi), \Pi(Q_H(r)), \Pi(Q_L(r)))$ and similarly for Y'_r. Note that

$$\mathrm{D}(Y \,||\, Y') = \mathbb{E}_{r \sim R_Q}\big[\mathrm{D}(Y_r \,||\, Y'_r)\big].$$

It holds that

$$\begin{aligned}
\mathrm{D}(Y_r \,||\, Y'_r) &= \mathrm{D}\big(T(\Pi), \Pi(Q_H(r)), \Pi(Q_L(r)) \,\big|\big|\, T(\Pi), \Pi(Q_H(r)), F(Q_L(r))\big) \\
&= \mathrm{D}\big(\Pi(Q_L(r)) \,\big|\, \Pi(Q_H(r)), T(\Pi) \,\big|\big|\, F(Q_L(r)) \,\big|\, \Pi(Q_H(r)), T(\Pi)\big) \\
&= |Q_L(r)| \cdot \log(n) - \mathrm{H}\big(\Pi(Q_L(r)) \,\big|\, \Pi(Q_H(r)), T(\Pi)\big) \qquad (25)
\end{aligned}$$

where the second equality uses the KL divergence chain rule (see also Claim 2.5), and the third equality follows since for the uniform distribution U over a set of size N and any random variable X it holds that $\mathrm{D}(X \,||\, U) = \log(N) - \mathrm{H}(X)$, and $F(Q_L(r))$ is uniform over $[n]^{|Q_L(r)|}$ and independent of $(\Pi(Q_H(r)), T(\Pi))$. To lower bound the second term in Eq. (25), observe that from the chain rule and the fact that conditioning reduces entropy, it holds that

$$\begin{aligned}
\mathrm{H}\big(\Pi(Q_L(r)) \,\big|\, \Pi(Q_H(r)), T(\Pi)\big) &= \sum_{i \in Q_L(r)} \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(Q_H(r) \cup Q_L^{<i}(r)), T(\Pi)\big) \\
&\geq \sum_{i \in Q_L(r)} \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(Q(r) \setminus \{i\}), T(\Pi)\big)
\end{aligned}$$

where $Q_L^{<i}(r)$ denotes the set of elements in $Q_L(r)$ that are smaller than i. Turning to the first term of Eq. (25), notice that $|Q_L(r)| \cdot \log(n) = \sum_{i \in Q_L(r)} \mathrm{H}\big(\Pi(i)\big)$ since for every $i \in [n]$ the marginal distribution of $\Pi(i)$ is uniform over $[n]$. Thus,

$$\begin{aligned}
\mathrm{D}(Y_r \,||\, Y'_r) &\leq \sum_{i \in Q_L(r)} \Big(\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(Q(r) \setminus \{i\}), T(\Pi)\big)\Big) \\
&= \sum_{i \in Q_L(r)} \mathrm{I}\big(\Pi(i); \Pi(Q(r) \setminus \{i\}), T(\Pi)\big)
\end{aligned} \qquad (26)$$

[13] Note that here $T(\Pi)$ is a deterministic function of Π, whereas in Lemma 6.2 the notation $T(\Pi)$ refers to any random variable that may depend on Π.

For each $i \in \mathcal{L}$, define

$$S_i \stackrel{\text{def}}{=} \arg \max_{S \subseteq [n] \setminus \{i\}, |S| < q} \left\{ \mathrm{I}\big(\Pi(i); \Pi(S), T(\Pi)\big) \right\} \tag{27}$$

Then,

$$\mathrm{D}(Y_r \,||\, Y'_r) \leq \sum_{i \in Q_L(r)} \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi)\big) \tag{28}$$

Note that each term in the summation in Eq. (28) is independent of r. Taking expectation over r, we get

$$\begin{aligned} \mathrm{D}(Y \,||\, Y') &= \mathbb{E}_{r \sim R_Q} \left[\mathrm{D}(Y_r \,||\, Y'_r) \right] \\ &\leq \mathbb{E}_{r \sim R_Q} \left[\sum_{i \in Q_L(r)} \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi)\big) \right] \\ &= \sum_{i \in \mathcal{L}} \Pr_{R_Q}[i \in Q(R_Q)] \cdot \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi)\big) \end{aligned} \tag{29}$$

By definition of the light locations, for every $i \in \mathcal{L}$ we have that $\Pr_{R_Q}[i \in Q(R_Q)] < \frac{2q}{n}$. Therefore,

$$\mathrm{D}(Y \,||\, Y') < \frac{2q}{n} \cdot \sum_{i \in \mathcal{L}} \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi)\big) \tag{30}$$

Note that the sets $\{S_i\}_{i \in \mathcal{L}}$ satisfy the hypothesis of Lemma 6.2. Therefore, assuming $q \cdot c = O(n)$, we can apply Lemma 6.2 to obtain

$$\sum_{i \in \mathcal{L}} \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi)\big) = O(\sqrt{q \cdot c \cdot n}).$$

Combining this with Eq. (30), we get that $\mathrm{D}(Y \,||\, Y') = O\Big(\sqrt{\frac{q^3 \cdot c}{n}}\Big)$. Recalling that by Pinsker's inequality $\Delta(Y, Y') \leq \sqrt{\frac{1}{2} \mathrm{D}(Y \,||\, Y')}$, it follows that $\Delta(Y, Y') = O\big(\big(\frac{q^3 \cdot c}{n}\big)^{1/4}\big)$, as claimed. □

Claim 6.3 implies that unless $q^3 \cdot c = \Omega(n)$, it holds that X and X' are indistinguishable (even for any fixed value of R_I). Since the verifier must distinguish between X and X', we conclude that $q^3 \cdot c = \Omega(n)$, completing the proof.

6.2 Alternative proof, achieving a slightly weaker lower bound

This proof follows a similar structure to the previous one, with the main difference being an alternative definition of the heavy locations. Additionally, in this approach, we will not fix the randomness of the interacting module R_I but rather fix the randomness of the querying module R_Q.

We make the assumption that the interacting module sends its entire randomness at the end of the interaction, thereby R_I is included in the interaction transcript. The length of the randomness R_I can be assumed to be at most $O(c + \log(n))$ (see Appendix B), hence modifying a general system to satisfy this assumption increases its communication complexity (from c) to at most $O(c + \log(n))$. We will prove that under this assumption $q^3 \cdot c = \Omega(n)$, which will establish a slightly weaker bound of $q^3 \cdot (c + \log(n)) = \Omega(n)$. We note that in the case of *statistical* soundness (which we cannot

assume here), one may assume without loss of generality that the isolated IPP is public-coin, since any (statistically sound) isolated IPP can be emulated by a public-coin isolated IPP with the same asymptotic query, communication and round complexities (see Appendix C). This means the entire randomness of the interaction is automatically included in the transcript, and thus the additive $\log(n)$ term can be avoided.

Similarly to Eq. (27), for each $i \in [n]$, denote $S_i = \arg\max_{S \subseteq [n] \setminus \{i\}, |S| < q} \{\mathrm{I}\big(\Pi(i); \Pi(S), T(\Pi, R_I)\big)\}$. That is, S_i is the set of $q-1$ locations of Π (excluding i) that together with $T(\Pi, R_I)$, provide maximal information about $\Pi(i)$. We change the definition of heavy locations to:

$$\mathcal{H} \stackrel{\text{def}}{=} \big\{ i \in [n] : \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi, R_I)\big) \geq \alpha \big\}$$

where α is chosen so as to ensure that there are at most $n/2$ heavy locations; that is

$$\alpha \stackrel{\text{def}}{=} \frac{2}{n} \cdot \sum_{i \in [n]} \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi, R_I)\big)$$

Note that by Lemma 6.2, if $q \cdot c = O(n)$, then we have that $\sum_{i \in [n]} \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi, R_I)\big) = O(\sqrt{q \cdot c \cdot n})$. Therefore, if $q \cdot c = O(n)$, then $\alpha = O\left(\sqrt{\frac{q \cdot c}{n}}\right)$.

We define $G = G(\Pi, F)$ as in the previous proof, only now with the new definition of the heavy locations. Note that, since we still have at most $n/2$ heavy locations, G remains far from PERM with high probability. Consider again the honest and cheating views from Eq. (23) and (24), but now under the new definition of the heavy locations. Note that since R_I is part of the interaction transcript, we can simplify the pair $(T(\Pi, R_I), R_I)$ to $T(\Pi, R_I)$. Hence, we have

$$X \stackrel{\text{def}}{=} (T(\Pi, R_I), \Pi(Q(R_Q)), R_Q) \equiv (T(\Pi, R_I), \Pi(Q_H(R_Q)), \Pi(Q_L(R_Q)), R_Q)$$
$$X' \stackrel{\text{def}}{=} (T(\Pi, R_I), G(Q(R_Q)), R_Q) \equiv (T(\Pi, R_I), \Pi(Q_H(R_Q)), F(Q_L(R_Q)), R_Q)$$

Since G remains far from PERM with high probability, the verifier still must distinguish between the views X and X'. On the other hand, we will show that the views X and X' are indistinguishable unless $q^3 \cdot c = \Omega(n)$. We will establish the indistinguishability claim even for any fixed value of R_Q. The following claim is analogous to Claim 6.3, with the only difference being the underlying definition of the heavy locations and the fact that we consider the views when fixing R_Q rather than when fixing R_I.

Claim 6.4. *Let r be an arbitrary randomness string of the querying module. Let Z and Z' denote the views X and X' when fixing R_Q to r:*

$$Z \stackrel{def}{=} (T(\Pi, R_I), \Pi(Q_H(r)), \Pi(Q_L(r)))$$
$$Z' \stackrel{def}{=} (T(\Pi, R_I), \Pi(Q_H(r)), F(Q_L(r)))$$

If $q \cdot c = O(n)$, then $\Delta(Z, Z') = O\big(\big(\frac{q^3 \cdot c}{n}\big)^{1/4}\big)$.

Proof: We can bound $\mathrm{D}(Z \,||\, Z')$ using the same analysis used to bound $\mathrm{D}(Y_r \,||\, Y'_r)$ in Claim 6.3 from Eq. (25) to Eq. (28), only replacing $T(\Pi)$ with $T(\Pi, R_I)$. Analogously to Eq. (28), we obtain:

$$\mathrm{D}(Z \,||\, Z') \leq \sum_{i \in Q_L(r)} \mathrm{I}\big(\Pi(i); \Pi(S_i), T(\Pi, R_I)\big) \tag{31}$$

By our new definition of the heavy locations, each term in the summation in Eq. (31) is smaller than α, since we sum over light locations only. If $q \cdot c = O(n)$ we have that $\alpha = O\left(\sqrt{\frac{q \cdot c}{n}}\right)$. Thus, we have

$$\mathrm{D}(Z \,||\, Z') < q \cdot \alpha = O\left(\sqrt{\frac{q^3 \cdot c}{n}}\right)$$

Since $\Delta(Z, Z') \leq \sqrt{\frac{1}{2}\,\mathrm{D}(Z \,||\, Z')}$, the claim follows. □

Claim 6.4 implies that X and X' are indistinguishable unless $q^3 \cdot c = \Omega(n)$, completing the proof.

Acknowledgments

I am deeply grateful to my advisor, Oded Goldreich. First, for his guidance throughout this entire research project, and, in particular, for his invaluable feedback during the writing stages. Second, I am grateful to Oded for key insights that significantly contributed to the actual results in this paper.

References

[1] Mihir Bellare, Oded Goldreich, and Shafi Goldwasser. "Randomness in interactive proofs". In: *computational complexity* 3 (1993), pp. 319–354.

[2] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[3] Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. "Fast approximate probabilistically checkable proofs". In: *Information and Computation* 189.2 (2004), pp. 135–159.

[4] Oded Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*. Vol. 17. Algorithms and Combinatorics. Springer, 1998.

[5] Oded Goldreich, Shafi Goldwasser, and Dana Ron. "Property testing and its connection to learning and approximation". In: *J. ACM* 45.4 (1998), pp. 653–750.

[6] 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)*. Vol. 251. 2023, 59:1–59:16.

[7] Oded Goldreich and Or Sheffet. "On the randomness complexity of property testing". In: *Computational Complexity* 19 (2010), pp. 99–133.

[8] Tom Gur, Yang P. Liu, and Ron D. Rothblum. "An Exponential Separation Between MA and AM Proofs of Proximity". In: *Comp. Complexity* 30.2 (2021), p. 12.

[9] Tom Gur and Ron Rothblum. "Non-interactive proofs of proximity". In: *Computational Complexity* 27.1 (2018), pp. 99–207. Preliminary version in ECCC, TR13-078, 2013.

[10] Donald E. Knuth and Andrew C. Yao. "The Complexity of Nonuniform Random Number Generation". In: *Algorithms and Complexity: New Directions and Recent Results*. Academic Press, 1976, pp. 357–428.

[11] 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. Association for Computing Machinery, 2013, pp. 793–802.

[12] Ronitt Rubinfeld and Madhu Sudan. "Robust Characterizations of Polynomials with Applications to Program Testing". In: *SIAM Journal on Computing* 25.2 (1996), pp. 252–271.

[13] Hadar Strauss. "Emulating Computationally Sound Public-Coin IPPs in the Pre-Coordinated Model". In: *Electronic Colloquium on Computational Complexity (ECCC)* TR24-131 (2024).

Appendices

A Coloring a directed graph with bounded out-degree

Claim A.1. *Let $k \in \mathbb{N}$. If G is a directed graph where each node has out-degree at most k, then G is $(2k+1)$-colorable.*

Proof: Let n be the number of nodes in G. Since each node has at most k outgoing edges, the total number of edges in G is at most $n \cdot k$. Therefore, there must be some node v that has at most k incoming edges. Hence, v shares an edge with at most $2k$ nodes. By induction, the graph obtained by removing v from G is $(2k+1)$-colorable. The nodes that share an edge with v use at most $2k$ colors, so there is a free color available for v among the $2k+1$ colors. ■

Note that the $2k+1$ bound in Claim A.1 is tight: Consider a clique of $2k+1$ nodes, where the nodes are ordered in a cycle such that each node has outgoing edges to the k nodes following it clockwise, and incoming edges from the k nodes preceding it anticlockwise.

B Reducing the amount of randomness in the isolated model

Claim B.1 (on the randomness complexity of cs-IPPs in the isolated model)**.** *Let $\Pi = \cup_{n\in\mathbb{N}}\Pi_n$ be a property such that Π_n is a set of functions from $[n]$ to Σ, where $|\Sigma| = \text{poly}(n)$. If Π has a computationally-sound IPP in the isolated model with communication complexity $c(n)$, then it has a computationally-sound IPP in the isolated model with the same order of communication, query, and round complexities, in which the randomness complexity of the interacting module is $O(c(n) + \log(n))$. Furthermore, if the original IPP uses non-adaptive queries, then so does the resulting IPP. Additionally, perfect completeness is preserved.*

Proof: The proof follows the standard argument for randomness reduction (see, for example, [7, Thm. 3], [8, Apdx. A], and [11, Lem. 4.8]). We detail only the modifications needed for the current setting (i.e., computationally-sound isolated IPPs).

Consider a matrix in which each column corresponds to a possible pair of prover strategy and input function, and each row corresponds to a possible randomness string of the interacting module. Each entry contains the probability (over the randomness of the querying module) that the verifier accepts, given the corresponding prover strategy, input function, and interacting module randomness. There are at most $2^{2^{c(n)}}$ prover strategies, and $\text{poly}(n)^n$ input functions, therefore the matrix has at most $2^{2^{c(n)}} \cdot \text{poly}(n)^n$ columns. Proceeding in the standard argument of randomness reduction, there exists a multi-set of size $\Theta(\log(2^{2^{c(n)}} \cdot \text{poly}(n)^n))$ of the matrix's rows that preserves the average of all columns up to an additive deviation of $1/7$. This implies an isolated IPP for which the randomness complexity of the interacting module is $O(\log(\log(2^{2^{c(n)}} \cdot \text{poly}(n)^n))) = O(c(n) + \log(n))$, and the computational-soundness and completeness errors are $1/3 + 1/7$. We reduce the error via $O(1)$ sequential repetitions of the entire system, which increases the communication, query, and round complexities by a constant factor. ■

Note that the resulting interacting module explicitly stores a multi-set of size $\Omega(2^{c(n)})$ for each input size n. Therefore, the resulting interacting module is non-uniform, and if $c(n)$ is super-logarithmic it requires a super-polynomial circuit size.

C Emulation of the isolated model with public coins

In this section we show that any IPP in the isolated model can be emulated by a public-coin IPP in the isolated model with the same query, communication, and round complexities (up to constant factors). We stress that this only applies to *statistically sound* IPPs, since the emulation does not preserve the computational complexity of the honest prover.

Theorem C.1. *Suppose that a property* Π *has an* r*-round IPP in the isolated model with communication complexity* c *and query complexity* q*. Then,* Π *has an* r*-round* public-coin *IPP in the isolated model with communication complexity* $O(c)$ *and query complexity* $O(q)$*. Furthermore, if the original IPP uses non-adaptive queries, then so does the resulting IPP. In addition, perfect completeness is preserved.*

The basic idea of the emulation. In each interaction round, we consider the distribution of the next message of the original verifier given the interaction transcript up to that round. Instead of directly sending a message according to this distribution, we send coins that are used to sample from this distribution (according to a predetermined sampling process).

Note that this emulation requires both the verifier and the honest prover of the resulting system to know the distributions of the original verifier's messages, and hence, in general, both will be inefficient. Furthermore, in a general IPP, these distributions may depend on the full input, since the verifier's messages can depend on the query answers. Hence, we rely on the fact that the original IPP is in the isolated model, where the verifier messages are oblivious of the input, to ensure that the new verifier can know these distributions without explicit access to the input. A similar approach can be used to transform any IPP in which the communication with the prover is oblivious of the input to a public-coin IPP, see Remark C.3. We note that this emulation strategy is folklore in the case of IPs, where there is free access to the input.

C.1 Preliminaries

Sampling from an arbitrary distribution. We will need the following well-known fact, which bounds the number of coins needed in order to sample from an arbitrary distribution.

Lemma C.2 (sampling from an arbitrary distribution using only fair coins [10])**.** *For any discrete random variable* X*, there exist an (unbounded time) algorithm that samples from* X *(i.e., for every* x *in the range of* X *the algorithm outputs* x *with probability* $\Pr[X = x]$*), such that the expected number of fair coin tosses required by the algorithm is at most* $\mathrm{H}(X) + 2$*.*

We note that the reason that we are using the sampler of Lemma C.2, rather than a more straightforward *approximate* sampler that has a worst-case guarantee on the number of coins used, is that it is needed in order to preserve the communication complexity up to constant factors.[14]

Game tree and value. The proof of Theorem C.1 uses the formulation of game trees for interactive proof systems (cf. [1, Sec. 4] and [4, Apdx. C.1]). For a fixed input f, the game tree captures all possible executions of the interactive protocol. Each path from the root to a leaf represents a possible interaction transcript. The root corresponds to the empty transcript, and each subsequent

[14]Specifically, the issue with using an approximate sampler is that in order to maintain an overall constant error, we will need the sampling error at each interaction round to be inversely proportional to the number of rounds r. On the other hand, the straightforward approximator of a distribution over N values, uses $\log_2(N/\epsilon)$ coins in order to get a distribution that is ϵ-close to the original one.

level alternates between prover and verifier messages. Each internal node represents a partial transcript and branches according to the possible next messages: If the transcript ends with a prover message, then the node's children represent all the possibilities for the verifier's next message, and if it ends with a verifier message, the node's children represent all the possibilities for the prover's next message.

We associate each node in the game tree with a value. The value of an internal node corresponding to partial transcript τ' is the maximum probability over all prover strategies that the verifier accepts f, conditioned on the corresponding sequence of messages sent during the execution equaling τ'. The value can be computed recursively: If the node is associated with the verifier, then its value is the expected value of its children, where the expectation is according to the distribution of the next verifier message, conditioned on the previous sequence of messages equaling τ'. If the node is associated with the prover, then its value is the maximal value among its children. Note that the value of the root of the game tree (corresponding to the empty transcript λ) represents the maximum probability over all prover strategies that the verifier accepts f. This value should be at least $2/3$ if f is a YES-instance, and at most $1/3$ otherwise.

Value in the isolated model. We next present in more detail the recursive computation of the value, focusing on the isolated model. Fix an arbitrary isolated IPP and an arbitrary input f. Let $\mathcal{Q}$, $\mathcal{I}$ and $\mathcal{D}$ denote the querying, interacting, and deciding modules of the IPP, respectively, and let R_I and R_Q denote the randomness of the interacting and querying modules, respectively. For any partial transcript τ', let $\mathcal{I}(r, \tau')$ denote the next message sent by the interacting module on partial transcript τ' when using randomness r. Let $R_I^{\tau'}$ be a random variable that is uniform over all randomness strings of the interacting module that are consistent with the partial transcript τ'.[15] As was observed by [6], in the isolated model the value of a leaf corresponding to a (full) transcript τ equals:

$$p_\tau^f \stackrel{\text{def}}{=} \Pr\left[\mathcal{D}(\tau, R_I^\tau, \mathcal{Q}^f(R_Q)) = 1\right]$$

The value of a node that corresponds to partial transcript τ' ending with a prover message equals:

$$p_{\tau'}^f \stackrel{\text{def}}{=} \mathbb{E}_{\alpha \sim \mathcal{I}(R_I^{\tau'}, \tau')}\left[p_{\tau', \alpha}^f\right]$$

The value of a node that corresponds to partial transcript τ' ending with a verifier message equals:

$$p_{\tau'}^f \stackrel{\text{def}}{=} \max_\beta \left\{p_{\tau', \beta}^f\right\}$$

Communication value (in the isolated model). For the purpose of our proof, we will define for each node in the game tree an additional value, which we will call the communication value. The communication value of a node corresponding to partial transcript τ' is the expected communication complexity of the remaining interaction from that point, assuming the prover uses the strategy that maximizes expected communication complexity. The expectation is over the distribution of the subsequent messages conditioned on the previous sequence of messages equaling τ'. For the root of the game tree, this represents the overall expected communication complexity when interacting with the worst-case prover strategy. Note that, since we are in the isolated model, where the verifier's messages are independent of the input f, the communication value is the same for all f.

[15] More explicitly, r is consistent with a partial transcript τ' if for every prefix of τ' of the form (τ'', α), where α represents a verifier message, it holds that $\mathcal{I}(r, \tau'') = \alpha$.

The communication value can be computed recursively: The communication value of a leaf is 0. For a partial transcript τ' that ends with a prover message, the communication value equals:

$$C_{\tau'} \stackrel{\text{def}}{=} \mathbb{E}_{\alpha \sim \mathcal{I}(R_I^{\tau'}, \tau')} \left[|\alpha| + C_{\tau', \alpha} \right]$$

For a partial transcript τ' that ends with a verifier message, the communication value equals:

$$C_{\tau'} \stackrel{\text{def}}{=} \max_{\beta} \left\{ |\beta| + C_{\tau', \beta} \right\}$$

where the maximum is taken over all the possible prover messages $\beta \in \{0,1\}^l$ such that l is the maximum number of bits that the verifier will read from the message that follows the partial transcript τ'.

C.2 Proof of Theorem C.1

The emulation. Given a general isolated IPP for a property Π, our aim is to construct a public-coin isolated IPP for Π that preserves the complexities of the original IPP up to constant factors. The IPP that we will construct will not preserve the worst-case communication complexity c of the original IPP, but its communication value (i.e., its *expected* communication when interacting with the worst-case prover) will be bounded by $O(c)$. This IPP can easily be modified to have $O(c)$ worst-case communication complexity. Specifically, we can modify the IPP to halt and accept if the communication exceeds 10 times its expectation, which by Markov's inequality occurs with probability at most 0.1. This will increase the soundness error by at most 0.1, which can be reduced back down via $O(1)$ parallel repetitions.

We begin by establishing notation for the original IPP (which will be identical to the notation used in the preliminaries). Let $\mathcal{Q}$, $\mathcal{I}$ and $\mathcal{D}$ denote the querying, interacting, and deciding modules of the original IPP, and let R_I and R_Q denote the randomness of the interacting and querying modules, respectively. Let $\mathcal{I}(r, \tau')$ denote the next message sent by the interacting module on partial transcript τ' when using randomness r, and let $R_I^{\tau'}$ denote the random variable that is uniform over all randomness strings of the interacting module that are consistent with the partial transcript τ'.

Additionally, for any partial transcript τ', let $S_{\tau'}$ be the procedure guaranteed by Lemma C.2 that samples from the distribution of $\mathcal{I}(R_I^{\tau'}, \tau')$. Let $W_{\tau'}$ be the random variable representing the coins used by $S_{\tau'}$. Note that by Lemma C.2 the expected length of $W_{\tau'}$ is at most $\mathrm{H}(\mathcal{I}(R_I^{\tau'}, \tau')) + 2$.

We construct new interacting and deciding modules, denoted $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{D}}$, respectively, that together with $\mathcal{Q}$ will constitute the claimed public-coin IPP. At a high level, the new interacting module $\tilde{\mathcal{I}}$ simulates the original interaction, such that at each round, if the original verifier would have sent a message α distributed according to $\mathcal{I}(R_I^{\tau'}, \tau')$, the new interacting module will send coins w distributed according to $W_{\tau'}$, such that the message w will correspond to the original message $\alpha = S_{\tau'}(w)$. Note that the corresponding message α has the original distribution since $S_{\tau'}(W_{\tau'}) \sim \mathcal{I}(R_I^{\tau'}, \tau')$ by definition of $S_{\tau'}$.

More concretely, for any partial transcript of the new system $\tilde{\tau}' = (w_1, \beta_1, \ldots, w_i, \beta_i)$, we define the corresponding original transcript $\tau' = (\alpha_1, \beta_1, \ldots, \alpha_i, \beta_i)$, where, for each $j \in [i]$, it holds that $\alpha_j = S_{\alpha_1, \beta_1, \ldots, \alpha_{j-1}, \beta_{j-1}}(w_j)$. At each interaction round, given partial transcript $\tilde{\tau}'$, the new interacting module $\tilde{\mathcal{I}}$ samples $w \sim W_{\tau'}$, and sends w to the prover. The new deciding module $\tilde{\mathcal{D}}$ receives the full interaction transcript $\tilde{\tau}$ from the interacting module, and translates it to the corresponding view of the original interacting module: (τ, r), where it samples $r \sim R_I^{\tau}$. It then emulates the original deciding module $\mathcal{D}$ on the view (τ, r) along with the view from the querying module $\mathcal{Q}$.

Correctness. To show that the resulting system is complete and sound, we show that for every input f, the value of the game tree of the new system equals to that of the original system.

Let f be an arbitrary input. Let $p^f_{\tau'}$ denote the value of the game tree of the original system at the node corresponding to partial transcript τ'. Similarly, let $\tilde{p}^f_{\tilde{\tau}'}$ denote the value of the game tree of the new system at the node corresponding to partial transcript $\tilde{\tau}'$. We show by reverse induction on the round number that for any partial transcript $\tilde{\tau}'$ it holds that $\tilde{p}^f_{\tilde{\tau}'} = p^f_{\tau'}$. The base case, where we have a full transcript $\tilde{\tau}$, follows immediately from the construction of the new deciding module:

$$\tilde{p}^f_{\tilde{\tau}} = \Pr\left[\tilde{\mathcal{D}}(\tilde{\tau}, \mathcal{Q}^f(R_Q)) = 1\right] = \Pr\left[\mathcal{D}(\tau, R^\tau_I, \mathcal{Q}^f(R_Q)) = 1\right] = p^f_\tau$$

For the inductive step, we first consider a partial transcript $\tilde{\tau}'$ that ends with a verifier message. We have:

$$\tilde{p}^f_{\tilde{\tau}'} = \max_\beta \left\{\tilde{p}^f_{\tilde{\tau}',\beta}\right\} = \max_\beta \left\{p^f_{\tau',\beta}\right\} = p^f_{\tau'}$$

where the second equality is by the induction hypothesis. Now, for an interaction transcript $\tilde{\tau}'$ that ends with a prover message, we have:

$$\tilde{p}^f_{\tilde{\tau}'} = \mathbb{E}_{w\sim W_{\tau'}}\left[\tilde{p}^f_{\tilde{\tau}',w}\right] = \mathbb{E}_{w\sim W_{\tau'}}\left[p^f_{\tau',S_{\tau'}(w)}\right] = \mathbb{E}_{\alpha\sim\mathcal{I}(R^{\tau'}_I,\tau')}\left[p^f_{\tau',\alpha}\right] = p^f_{\tau'}$$

where the second equality follows from the preceding inductive case (for transcripts ending with a verifier message), and the third equality follows because $S_{\tau'}(W_{\tau'}) \sim \mathcal{I}(R^{\tau'}_I, \tau')$ by definition of $S_{\tau'}$. This completes the inductive claim. We conclude that $\tilde{p}^f_\lambda = p^f_\lambda$, establishing the completeness and soundness of the resulting system.

Complexities. Clearly, the emulation preserves the round and query complexities. We are left to verify the communication complexity. Let c denote the communication complexity of the original system. Recall that we only need to show that the *communication value* of the resulting system is bounded by $O(c)$. Let $\tilde{C}_{\tilde{\tau}'}$ denote the communication value of the game tree of the resulting system at the node corresponding to partial transcript $\tilde{\tau}'$, and similarly, let $C_{\tau'}$ denote the communication value of the game tree of the original system at the node corresponding to partial transcript τ'. Let m be the number of messages the verifier sends in the original protocol (and hence also in the resulting protocol). We will show that:

$$\tilde{C}_\lambda \leq C_\lambda + 2m \tag{32}$$

Note that $C_\lambda \leq c$ (because c is the worst case communication complexity, whereas C_λ is the expected communication complexity (when interacting with the worst-case prover)) and that $m \leq c$ (because in each message at least 1 bit is exchanged). Hence, this will establish that $\tilde{C}_\lambda = O(c)$, as claimed. We establish Eq. (32) by showing that for any partial transcript $\tilde{\tau}'$ it holds that

$$\tilde{C}_{\tilde{\tau}'} \leq C_{\tau'} + 2\cdot(m-i)$$

where i is the number of verifier messages in $\tilde{\tau}'$. We show this by reverse induction on the round number. For the base case, where we have a full transcript $\tilde{\tau}$, we simply have $\tilde{C}_{\tilde{\tau}} = 0 = C_\tau$. Now, for a partial transcript $\tilde{\tau}'$ that ends with a verifier message, we have

$$\tilde{C}_{\tilde{\tau}'} = \max_\beta \left\{|\beta| + \tilde{C}_{\tilde{\tau}',\beta}\right\} \leq \max_\beta \left\{|\beta| + C_{\tau',\beta}\right\} + 2\cdot(m-i) = C_{\tau'} + 2\cdot(m-i)$$

where the inequality is since, by the induction hypothesis, $\tilde{C}_{\tilde{\tau}',\beta} \leq C_{\tau',\beta} + 2 \cdot (m-i)$. We turn to the case where the partial transcript $\tilde{\tau}'$ ends with a prover message. By the definition of $\tilde{C}_{\tilde{\tau}'}$, we have that

$$\tilde{C}_{\tilde{\tau}'} = \mathbb{E}_{w \sim W_{\tau'}} \left[|w|\right] + \mathbb{E}_{w \sim W_{\tau'}} \left[\tilde{C}_{\tilde{\tau}',w}\right] \tag{33}$$

For the first term of Eq. (33), we have:

$$\mathbb{E}_{w \sim W_{\tau'}} \left[|w|\right] \leq \mathrm{H}(\mathcal{I}(R_I^{\tau'}, \tau')) + 2 \leq \mathbb{E}_{\alpha \sim \mathcal{I}(R_I^{\tau'}, \tau')} \left[|\alpha|\right] + 2 \tag{34}$$

where the first inequality is by the guarantee of Lemma C.2, and second inequality is since for any random variable X over $\{0,1\}^*$ it holds that $\mathrm{H}(X) \leq \mathbb{E}[|X|]$. We turn to the second term of Eq. (33). By the preceding inductive case (for transcripts ending with a verifier message), we have that

$$\tilde{C}_{\tilde{\tau}',w} \leq C_{\tau',S_{\tau'}(w)} + 2 \cdot (m - (i+1))$$

Hence,

$$\begin{aligned} \mathbb{E}_{w \sim W_{\tau'}} \left[\tilde{C}_{\tilde{\tau}',w}\right] &\leq \mathbb{E}_{w \sim W_{\tau'}} \left[C_{\tau',S_{\tau'}(w)}\right] + 2 \cdot (m - (i+1)) \\ &= \mathbb{E}_{\alpha \sim \mathcal{I}(R_I^{\tau'}, \tau')} \left[C_{\tau',\alpha}\right] + 2 \cdot (m - (i+1)) \end{aligned} \tag{35}$$

where the equality is since $S_{\tau'}(W_{\tau'}) \sim \mathcal{I}(R_I^{\tau'}, \tau')$ by definition of $S_{\tau'}$. Combining Eq. (34) and (35) with Eq. (33), we get that

$$\begin{aligned} \tilde{C}_{\tilde{\tau}'} &\leq \mathbb{E}_{\alpha \sim \mathcal{I}(R_I^{\tau'}, \tau')} \left[|\alpha|\right] + 2 + \mathbb{E}_{\alpha \sim \mathcal{I}(R_I^{\tau'}, \tau')} \left[C_{\tau',\alpha}\right] + 2 \cdot (m - (i+1)) \\ &= C_{\tau'} + 2 \cdot (m - i) \end{aligned}$$

where the equality follows since $C_{\tau'} = \mathbb{E}_{\alpha \sim \mathcal{I}(R_I^{\tau'}, \tau')} \left[|\alpha|\right] + \mathbb{E}_{\alpha \sim \mathcal{I}(R_I^{\tau'}, \tau')} \left[C_{\tau',\alpha}\right]$ by definition. This completes the inductive claim, establishing that $\tilde{C}_\lambda \leq C_\lambda + 2m$, as desired. ■

Remark C.3 (public-coin emulation of IPPs in which the interaction is oblivious of the input)**.** *Consider IPPs where there is no information flow from the querying module to the interacting module, i.e., where the decision can be written as* $\mathcal{D}(\mathcal{Q}^f(\langle P(f), \mathcal{I}(R_I)\rangle))$*. Any such IPP can be transformed to a* public-coin *IPP by a similar emulation to that of Theorem C.1. The only modification needed is that, at the end of the interaction, the new interacting module* $\tilde{\mathcal{I}}$ *passes the interaction transcript* $\tilde{\tau}$ *to a new querying module* $\tilde{Q}$*, which emulates the original querying module* Q *on* (τ, r) *where* $r \sim R_I^\tau$*. The deciding module remains identical to the original system. The analysis is the same as that of Theorem C.1, with the only difference being that the value of a leaf is replaced with:* $p_\tau^f = \Pr\left[\mathcal{D}(\mathcal{Q}^f(\tau, R_I^\tau)) = 1\right]$*.*

D MAPs and the hybrid model

The purpose of this appendix is to shed some light on why the lower bound technique of [6, Apdx. A.4], which was used in the setting of MAPs, could be adapted to our isolated model setting. We start by showing that our technique for proving the non-adaptive lower bound for isolated IPPs for `PERM` (Theorem 6.1) can be adapted to obtain an analogous lower bound on MAPs for `PERM`. We complement this by showing a corresponding upper bound on MAPs for `PERM`, exemplifying a general tradeoff between the query complexity and the proof length, analogous to the tradeoff

between the query and communication complexities we showed for isolated IPPs in Theorem 3.1. Then, in Section D.3 we show that the lower bounds on non-adaptive MAPs and isolated IPPs for PERM can be extended to a *hybrid* model that extends both MAPs and isolated IPPs. Finally, in Section D.4 we establish a more general connection between the two settings (beyond just PERM), by showing that, similarly to the emulation of the isolated model by testers shown in [6, Thm. 1.2], our hybrid model can be efficiently emulated by MAPs.

We begin with the upper bound for MAPs.

D.1 An upper bound on MAPs for PERM

In this section we present a MAP for PERM that demonstrates a general tradeoff between the query complexity and proof length, analogous to the tradeoff between the query and communication complexities we showed for isolated IPPs in Theorem 3.1.

Theorem D.1. *For every $q > 0$, there exists a MAP for* PERM *with query complexity q, proof length $O\left(\frac{n}{q} \cdot \log(n)\right)$, soundness $\Omega(\epsilon)$, and perfect completeness.*

Proof idea. Note that a permutation can be viewed as a collection of directed cycles. Hence, if we aim to find a pre-image of some point $i \in [n]$ in a permutation π, one approach is to start by querying the permutation at position i, obtaining its value $v := \pi(i)$, then querying position v, and so on, each step querying the answer from the last step, until completing the cycle by reaching the value i. This approach is only effective if the cycle that i resides on is small. Thus, we use the MAP-proof to specify $O(n/q)$ cutting points across the cycles that break each cycle into segments of length at most q. With these cutting points, we can avoid traversing the entire cycle and instead traverse only the segment containing i: when we reach the cutting point at the end of the segment, we jump to the previous cutting point (i.e., the cutting point at the start of the segment).

Proof: The honest prover partitions each cycle of the permutation into segments of length at most q by selecting cutting points among the nodes such that in total there are at most $2n/q$ cutting points. The MAP-proof consists of a list of cutting points for each cycle, ordered as they appear when traversing the cycle.

The verifier samples a random point $i \in [n]$ and makes at most q steps, starting at position i. At each step, it queries the input function at the position equal to its previous answer (or i for the first step), unless that position is a cutting point that appears in the proof - in which case it jumps to (i.e., queries) the previous cutting point in that cycle's list, and continues the process from that new position. (Note that if the sampled point is a cutting point, the verifier will immediately jump to the previous cutting point.) The verifier continues this process until either it finds a pre-image of i (i.e., one of its queries returns the value i) and accepts, or it reaches q queries without finding such a pre-image and rejects.

Completeness follows from the fact that the verifier traverses a segment of length at most q that contains i. For soundness, if f is ϵ-far from PERM, then at least $\epsilon \cdot n$ points in $[n]$ have no pre-image under f. The verifier will sample such a point with probability at least ϵ and will certainly reject. ■

D.2 A lower bound on MAPs with non-adaptive queries for PERM

In this section we show that our technique for proving the non-adaptive lower bound for isolated IPPs for PERM (Theorem 6.1) can be adapted to obtain an analogous lower bound on MAPs for PERM.

Unlike the lower bound we showed for the isolated model, we now consider statistical soundness rather than computational soundness. The reason is that in the context of MAPs, there is no distinction between statistical and computational soundness when considering non-uniform cheating provers. This is since the optimal strategy in a MAP is simply to produce the best possible proof string, which a non-uniform polynomial-size cheating prover can always explicitly store.

Theorem D.2. *If* PERM *can be verified by a MAP that uses non-adaptive queries and has query complexity q and proof length c, then $q^3 \cdot c = \Omega(n)$.*

We note that in [8, Lem. 4.3] it was shown that any (possibly adaptive) MAP for PERM with query complexity $q > 0$ and proof length $c > 0$ must satisfy $q \cdot c = \Omega(\sqrt{n})$. Theorem D.2 provides a tighter bound on c when $q = o(n^{1/4})$, when restricting to non-adaptive queries.

Notation. For random variables X, X', Y and Z, and an element z in the range of Z, we denote:

$$\mathrm{H}(X \,|\, Y, Z = z) \stackrel{\text{def}}{=} \mathbb{E}_{y \sim (Y|Z=z)} \left[\mathrm{H}(X \,|\, Y = y, Z = z)\right]$$

$$\mathrm{D}(X \,|\, Y, Z = z \,||\, X' \,|\, Y, Z = z) \stackrel{\text{def}}{=} \mathbb{E}_{y \sim (Y|Z=z)} \left[\mathrm{D}(X \,|\, Y = y, Z = z \,||\, X' \,|\, Y = y, Z = z)\right]$$

Proof overview. We follow an approach similar to the isolated model lower bound. We show indistinguishability between the verifier's view when querying a random permutation (with an honest proof) and its view when querying a random function that behaves like a permutation on a small set of "heavy" locations while receiving an honest proof of the same permutation. More specifically, we consider a random permutation Π and a random function F. We sample $\pi \sim \Pi$ and $f \sim F$, and construct the function $G = G(\pi, f)$ that equals π on selected "heavy" locations and equals f on the remaining locations. We then consider the verifier's view when querying $G = G(\pi, f)$ while receiving an honest proof for π, denoted $\mathcal{W}(\pi)$. The key difference from the isolated model is that here the verifier can choose its queries based on the proof it receives. However, we can also be adaptive in where we place π, choosing different heavy locations according to its proof $\mathcal{W}(\pi)$. This is possible because the proof depends only on π, unlike a general interaction transcript which also depends on the verifier's randomness. Thus, rather than having a single set of heavy locations, we will define for each possible proof string ω a corresponding set of ω-heavy locations.

For each possible proof ω, we think of each location $i \in [n]$ as having an associated "ω-information-weight". Recall that in the isolated model, the information-weight of i was:

$$\max_{S \subseteq [n] \setminus \{i\}, |S| < q} \left\{\mathrm{I}\big(\Pi(i); \Pi(S), T(\Pi)\big)\right\}$$

or equivalently,

$$\max_{S \subseteq [n] \setminus \{i\}, |S| < q} \left\{\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S), T(\Pi)\big)\right\}$$

In the current case, the ω-information-weight of i is:

$$\max_{S \subseteq [n] \setminus \{i\}, |S| < q} \left\{\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S), \mathcal{W}(\Pi) = \omega\big)\right\}.$$

Unlike the expression for information-wight we had in the isolated model, the expression $\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S), \mathcal{W}(\Pi) = \omega\big)$ does not directly correspond to the standard definition of mutual information. However, when taking the expectation of this expression over $\omega \sim \mathcal{W}(\Pi)$ we get the mutual information between $\Pi(i)$ and $(\Pi(S), \mathcal{W}(\Pi))$; that is,

$$\mathbb{E}_{\omega \sim \mathcal{W}(\Pi)} \Big[\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S), \mathcal{W}(\Pi) = \omega\big)\Big] = \mathrm{I}\big(\Pi(i); \Pi(S), \mathcal{W}(\Pi)\big).$$

We show that in expectation over $\omega \sim \mathcal{W}(\Pi)$, we can bound the total ω-information weight of all locations (see Lemma D.3). This in turn bounds the number of information-heavy locations of a *typical* proof $\omega \sim \mathcal{W}(\Pi)$.

Similarly to the isolated model case, we can effectively think of the verifier as getting a proof ω and aiming to maximize the expected total ω-information-weight of the locations it queries. On the other hand, we can view designating locations as ω-heavy as preventing the verifier from gaining the weight of these locations. Our aim is to choose these locations such that in expectation over the proof $\omega \sim \mathcal{W}(\Pi)$, the verifier cannot gain much ω-information-weight (in expectation over its randomness).

Since a typical proof has only a small number of information-heavy locations, we can use our strategies from the isolated model per each proof. In the first approach, we define the ω-heavy locations to be the locations that the verifier queries with high probability when receiving the proof ω, and in the second approach, we define the ω-heavy locations to be the information-heavy locations of ω. Note that in the second approach, the number of ω-heavy locations is bounded only in expectation (over $\omega \sim \mathcal{W}(\Pi)$). We show that this is sufficient to ensure that the constructed input is far from PERM with high probability. A proof of Theorem D.2 following the first approach is presented in Section D.2.2, and an alternative proof following the second approach is presented in Section D.2.3.

D.2.1 Preliminaries

Before proceeding to the proof of Theorem D.2, we state a generalization of Lemma 6.2 that allows the sets S_i (representing the verifier's possible queries) to depend on the given proof.

Lemma D.3. *Let $n \in \mathbb{N}$, and let $q, c \in \mathbb{N} \setminus \{0\}$ such that $q \cdot c = O(n)$. Let Π be a random permutation over $[n]$. Let $\mathcal{W}(\Pi)$ be any random variable over $\{0,1\}^c$ that may depend on Π. For each $\omega \in \{0,1\}^c$, for each $i \in [n]$, let S_i^ω be any subset of $[n] \setminus \{i\}$ such that $|S_i^\omega| < q$. Then,*

$$\sum_{i \in [n]} \mathrm{I}\big(\Pi(i); \Pi(S_i^{\mathcal{W}(\Pi)}), \mathcal{W}(\Pi)\big) = O\left(\sqrt{q \cdot c \cdot n}\right)$$

This lemma differs from Lemma 6.2 in two ways. First, $\mathcal{W}(\Pi)$ appears instead of $T(\Pi)$, but this difference is immaterial as both are treated as general random variables. Second, and more importantly, the sets S_i now depend on $\mathcal{W}(\Pi)$. This dependency makes the concrete steps of the proof more complex, but the overall proof follows similar lines to that of Lemma 6.2. The proof is therefore deferred to Appendix E.

Note that while in the current setting $\mathcal{W}(\Pi)$ is a deterministic function of Π (representing Π's proof), Lemma D.3 allows $\mathcal{W}(\Pi)$ to be any random variable. We will use the more general statement later in the proof of Theorem D.8.

D.2.2 Proof of Theorem D.2

The proof closely resembles that of Theorem 6.1, presented in Section 6.1. The main difference is that here the queries depend on the proof string, which requires more careful handling.

Consider an arbitrary MAP that has proof length $c > 0$ and uses $q > 0$ non-adaptive queries. Without loss of generality, we assume the verifier does not make the same query more than once.

Let R denote the randomness of the verifier. For each proof string $\omega \in \{0,1\}^c$ and each randomness string r, let $Q(\omega, r)$ be the set of q (non-adaptive) queries made by the verifier given

proof ω and randomness r. For each proof string $\omega \in \{0,1\}^c$, consider the following set of ω-heavy locations, denoted

$$\mathcal{H}_\omega \stackrel{\text{def}}{=} \Big\{ i \in [n] : \Pr_R[i \in Q(\omega, R)] \geq 2q/n \Big\}$$

Additionally, denote the complementary set of ω-light locations by $\mathcal{L}_\omega \stackrel{\text{def}}{=} [n] \setminus \mathcal{H}_\omega$. Note that for every ω we have that $|\mathcal{H}_\omega| \leq n/2$ since $\sum_{i \in [n]} \Pr_R[i \in Q(\omega, R)] = q$.

For each permutation π, let $\mathcal{W}(\pi)$ be an honest proof for π. Let Π be a random permutation over $[n]$ and let F be a random function from $[n]$ to $[n]$ such that Π and F are independent. Let $G = G(\Pi, F)$ be the function that agrees with Π on $\mathcal{H}_{\mathcal{W}(\Pi)}$, and agrees with F on $\mathcal{L}_{\mathcal{W}(\Pi)}$; that is, $G(i) = \Pi(i)$ for every $i \in \mathcal{H}_{\mathcal{W}(\Pi)}$, and $G(i) = F(i)$ for every $i \in \mathcal{L}_{\mathcal{W}(\Pi)}$. Let $\epsilon > 0$ be a sufficiently small proximity parameter. For each fixed permutation π, since $|\mathcal{L}_{\mathcal{W}(\pi)}| \geq n/2$, Claim 4.1 implies that $G(\pi, F)$ is ϵ-far from PERM with high probability. Hence, G is ϵ-far from PERM with high probability.

Let $Q_H(\omega, r) \stackrel{\text{def}}{=} Q(\omega, r) \cap \mathcal{H}_\omega$ and $Q_L(\omega, r) \stackrel{\text{def}}{=} Q(\omega, r) \cap \mathcal{L}_\omega$ denote the heavy and light queries given proof ω and randomness r, respectively. Consider the verifier's view given input Π and the honest proof $\mathcal{W}(\Pi)$:

$$\begin{aligned} X &\stackrel{\text{def}}{=} \big(\mathcal{W}(\Pi), \Pi\big(Q(\mathcal{W}(\Pi), R)\big), R\big) \\ &\equiv \big(\mathcal{W}(\Pi), \Pi\big(Q_H(\mathcal{W}(\Pi), R)\big), \Pi\big(Q_L(\mathcal{W}(\Pi), R)\big), R\big) \end{aligned} \tag{36}$$

Similarly, consider the verifier's view given input $G = G(\Pi, F)$ and proof $\mathcal{W}(\Pi)$:

$$\begin{aligned} X' &\stackrel{\text{def}}{=} \big(\mathcal{W}(\Pi), G\big(Q(\mathcal{W}(\Pi), R)\big), R\big) \\ &\equiv \big(\mathcal{W}(\Pi), \Pi\big(Q_H(\mathcal{W}(\Pi), R)\big), F\big(Q_L(\mathcal{W}(\Pi), R)\big), R\big) \end{aligned} \tag{37}$$

Since G is ϵ-far from PERM with high probability, the completeness and soundness conditions imply that the verifier must distinguish between X and X' (cf. Claim 5.2). However, we will show that these views are indistinguishable unless $q^3 \cdot c = \Omega(n)$.

Claim D.4. *If $q \cdot c = O(n)$, then $\Delta(X, X') = O\big(\big(\frac{q^3 \cdot c}{n}\big)^{1/4}\big)$.*

Proof: As in Claim 6.3 we focus on bounding $\mathrm{D}(X \,||\, X')$. For every $\omega \in \{0,1\}^c$, let X_ω and X'_ω denote the views X and X' conditioned on $\mathcal{W}(\Pi)$ equaling ω; that is, $X_\omega = (\Pi(Q_H(\omega, R)), \Pi(Q_L(\omega, R)), R \,|\, \mathcal{W}(\Pi) = \omega)$ and similarly for X'_ω. Note that

$$\mathrm{D}(X \,||\, X') = \mathbb{E}_{\omega \sim \mathcal{W}(\Pi)} \left[\mathrm{D}(X_\omega \,||\, X'_\omega) \right]$$

Consider an arbitrary proof string ω. We can bound $\mathrm{D}(X_\omega \,||\, X'_\omega)$ analogously to the way we have bounded $\mathrm{D}(Y \,||\, Y')$ in Claim 6.3 up to Eq. (30). The only difference is that: (1) instead of a "general" condition on $T(\Pi)$ (i.e., an expectation over all possible fixed values for $T(\Pi)$), we have a specific condition on $\mathcal{W}(\Pi) = \omega$, and (2) the queries and the light locations now depend on this fixed ω. In more detail, letting $X_{\omega,r}$ and $X'_{\omega,r}$ denote the views X_ω and X'_ω when fixing R to r and following analogously to the analysis in Eq. (25) through (26), we get

$$\mathrm{D}(X_{\omega,r} \,||\, X'_{\omega,r}) \leq \sum_{i \in Q_L(\omega,r)} \Big(\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(Q(\omega, r) \setminus \{i\}), \mathcal{W}(\Pi) = \omega\big) \Big) \tag{38}$$

For each $i \in [n]$, we define

$$S_i^\omega \stackrel{\text{def}}{=} \arg \max_{S \subseteq [n] \setminus \{i\}, |S| < q} \big\{ \mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S), \mathcal{W}(\Pi) = \omega\big) \big\} \tag{39}$$

and get

$$\mathrm{D}(X_{\omega,r} \,||\, X'_{\omega,r}) \leq \sum_{i \in Q_L(\omega,r)} \Big(\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i^\omega), \mathcal{W}(\Pi) = \omega\big)\Big) \tag{40}$$

Analogously to Eq. (29), we have

$$\begin{aligned}\mathrm{D}(X_\omega \,||\, X'_\omega) &= \mathbb{E}_{r\sim R}\left[\mathrm{D}(X_{\omega,r} \,||\, X'_{\omega,r})\right] \\ &\leq \sum_{i \in \mathcal{L}_\omega} \Pr_R[i \in Q(\omega, R)] \cdot \Big(\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i^\omega), \mathcal{W}(\Pi) = \omega\big)\Big)\end{aligned}$$

By the definition of $\mathcal{L}_\omega$, for each $i \in \mathcal{L}_\omega$ we have that $\Pr_R[i \in Q(\omega, R)] < \frac{2q}{n}$. Therefore, analogously to Eq. (30), we have:[16]

$$\mathrm{D}(X_\omega \,||\, X'_\omega) < \frac{2q}{n} \cdot \sum_{i \in [n]} \Big(\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i^\omega), \mathcal{W}(\Pi) = \omega\big)\Big)$$

At this point, we take expectation over $\omega \sim \mathcal{W}(\Pi)$:

$$\begin{aligned}\mathrm{D}(X \,||\, X') &= \mathbb{E}_{\omega\sim\mathcal{W}(\Pi)}\left[\mathrm{D}(X_\omega \,||\, X'_\omega)\right] \\ &< \frac{2q}{n} \cdot \mathbb{E}_{\omega\sim\mathcal{W}(\Pi)}\Big[\sum_{i\in[n]} \Big(\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i^\omega), \mathcal{W}(\Pi) = \omega\big)\Big)\Big] \\ &= \frac{2q}{n} \cdot \sum_{i\in[n]} \mathrm{I}\big(\Pi(i); \Pi(S_i^{\mathcal{W}(\Pi)}), \mathcal{W}(\Pi)\big)\end{aligned} \tag{41}$$

Assuming that $q \cdot c = O(n)$, we can apply the generalized Lemma 6.2 to obtain

$$\sum_{i\in[n]} \mathrm{I}\big(\Pi(i); \Pi(S_i^{\mathcal{W}(\Pi)}), \mathcal{W}(\Pi)\big) = O(\sqrt{q \cdot c \cdot n}).$$

Combining this with Eq. (41) we get that $\mathrm{D}(X \,||\, X') = O\Big(\sqrt{\frac{q^3 \cdot c}{n}}\Big)$. Recalling that $\Delta(X, X') \leq \sqrt{\frac{1}{2}\,\mathrm{D}(X \,||\, X')}$, it follows that $\Delta(X, X') = O\big(\big(\frac{q^3 \cdot c}{n}\big)^{1/4}\big)$, as claimed. □

D.2.3 Alternative proof

We present an alternative proof of Theorem D.2 that follows a similar approach to the alternative proof of Theorem 6.1 (which was presented in Section 6.2). Similarly to Eq. (39), for each $\omega \in \{0,1\}^c$, for each $i \in [n]$, define

$$S_i^\omega = \arg \max_{S \subseteq [n]\setminus\{i\}, |S|<q} \big\{ \mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S), \mathcal{W}(\Pi) = \omega\big)\big\}.$$

Let $\alpha > 0$ be a parameter we will set shortly. We change the definition of ω-heavy locations to:

$$\mathcal{H}_\omega \stackrel{\text{def}}{=} \big\{ i \in [n] : \mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i), \mathcal{W}(\Pi) = \omega\big) \geq \alpha \big\}$$

[16] Note that we also rely on the fact that for each $i \in [n]$ the term $\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i^\omega), \mathcal{W}(\Pi) = \omega\big)$ is non-negative, since $\mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i^\omega), \mathcal{W}(\Pi) = \omega\big)$ can be at most the entropy of the uniform random variable over support of size n, i.e., $\log(n) = \mathrm{H}\big(\Pi(i)\big)$.

We set α so as to ensure that $\mathbb{E}_{\omega\sim\mathcal{W}(\Pi)}[|\mathcal{H}_\omega|] \leq n/\ell$, where $\ell > 0$ is a sufficiently large constant (the reason for not taking $l = 2$ will become clear in the proof of Claim D.6):

$$\begin{aligned}\alpha &\stackrel{\text{def}}{=} \frac{\ell}{n} \cdot \mathbb{E}_{\omega\sim\mathcal{W}(\Pi)}\Big[\sum_{i\in[n]}\Big(\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i)\,\big|\,\Pi(S_i),\mathcal{W}(\Pi)=\omega\big)\Big)\Big] \\ &= \frac{\ell}{n}\cdot\sum_{i\in[n]}\mathrm{I}\big(\Pi(i);\Pi(S_i^{\mathcal{W}(\Pi)}),\mathcal{W}(\Pi)\big)\end{aligned}$$

Note that by Lemma D.3, if $q\cdot c = O(n)$, then we have that $\sum_{i\in[n]}\mathrm{I}\big(\Pi(i);\Pi(S_i^{\mathcal{W}(\Pi)}),\mathcal{W}(\Pi)\big) = O(\sqrt{q\cdot c\cdot n})$. Therefore, if $q\cdot c = O(n)$, then $\alpha = O\left(\sqrt{\frac{q\cdot c}{n}}\right)$.

Claim D.5. $\mathbb{E}_{\omega\sim\mathcal{W}(\Pi)}[|\mathcal{H}_\omega|] \leq n/\ell$.

Proof: For each $\omega\in\{0,1\}^c$, it holds that

$$\begin{aligned}|\mathcal{H}_\omega|\cdot\alpha &\leq \sum_{i\in\mathcal{H}_\omega}\Big(\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i)\,\big|\,\Pi(S_i^\omega),\mathcal{W}(\Pi)=\omega\big)\Big) \\ &\leq \sum_{i\in[n]}\Big(\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i)\,\big|\,\Pi(S_i^\omega),\mathcal{W}(\Pi)=\omega\big)\Big)\end{aligned}$$

Therefore,

$$\mathbb{E}_{\omega\sim\mathcal{W}(\Pi)}\left[|\mathcal{H}_\omega|\right]\cdot\alpha \leq \mathbb{E}_{\omega\sim\mathcal{W}(\Pi)}\Big[\sum_{i\in[n]}\Big(\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i)\,\big|\,\Pi(S_i),\mathcal{W}(\Pi)=\omega\big)\Big)\Big] = \frac{n}{\ell}\cdot\alpha$$

by definition of α. The claim follows. □

We define $G = G(\Pi, F)$ as in the previous proof, only now with the new definition of the heavy locations. We claim that G is still far from PERM with high probability (assuming ℓ is sufficiently large).

Claim D.6. *For all sufficiently small ϵ, the probability that G is ϵ-close to* PERM *is at most* $2/\ell + \exp(-\Omega(n))$.

Proof: By Claim D.5 and Markov's inequality, the probability that $|\mathcal{H}_{\mathcal{W}(\Pi)}| > n/2$ is at most $2/\ell$. On the other hand, by Claim 4.1, for every fixed permutation π for which $|\mathcal{L}_{\mathcal{W}(\pi)}| \geq n/2$ (i.e., $|\mathcal{H}_{\mathcal{W}(\pi)}| \leq n/2$), the probability that $G(\pi, F)$ is ϵ-close to PERM is at most $\exp(-\Omega(n))$. Combining both cases, the claim follows. □

Consider again the honest and cheating views X and X' from Equations (36) and (37), but now under the new definition of the heavy locations. As before, let $X_{\omega,r}$ and $X'_{\omega,r}$ denote the views X and X' when fixing $\mathcal{W}(\Pi)$ to ω and R to r. Since G remains far from PERM with high probability, we still have that the verifier must distinguish between X and X'. On the other hand, we will show that the views X and X' are indistinguishable unless $q^3\cdot c = \Omega(n)$. We will establish the indistinguishability claim even for $X_{\omega,r}$ and $X'_{\omega,r}$ for any ω and r.

Claim D.7. *Let ω be an arbitrary proof string, and let r be an arbitrary randomness string. If $q\cdot c = O(n)$, then $\Delta(X_{\omega,r}, X'_{\omega,r}) = O\big(\big(\frac{q^3\cdot c}{n}\big)^{1/4}\big)$.*

Proof: Notice that the analysis in Claim D.4 leading to Eq. (40) is independent of the definition of the heavy locations, hence Eq. (40) still holds:

$$\mathrm{D}(X_{\omega,r} \,||\, X'_{\omega,r}) \leq \sum_{i \in Q_L(\omega,r)} \Big(\mathrm{H}\big(\Pi(i)\big) - \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i), \mathcal{W}(\Pi) = \omega\big) \Big) \tag{42}$$

By our new definition of the heavy locations, each term in the summation in Eq. (42) is smaller than α, since we sum over light locations only. If $q \cdot c = O(n)$ we have that $\alpha = O\left(\sqrt{\frac{q \cdot c}{n}}\right)$. Thus, we have

$$\mathrm{D}(X_{\omega,r} \,||\, X'_{\omega,r}) < q \cdot \alpha = O\left(\sqrt{\frac{q^3 \cdot c}{n}}\right)$$

Since $\Delta(X_{\omega,r}, X'_{\omega,r}) \leq \sqrt{\frac{1}{2}\,\mathrm{D}(X_{\omega,r} \,||\, X'_{\omega,r})}$, the claim follows. □

D.3 A lower bound for a hybrid model

We can extend the non-adaptive lower bounds presented for isolated IPPs and MAPs to a model of IPPs that extends both MAPs and isolated IPPs under a single model. This model is identical to the isolated model, except that we assume the prover sends the first message and that the querying module gets access to this first message. (Except for this message, there is no information flow between the querying and interacting modules.) We refer to this model as the hybrid model.

Theorem D.8. *If* PERM *can be verified by a computationally-sound IPP in the hybrid model that uses non-adaptive queries and has query complexity $q > 0$ and communication complexity $c > 0$, then $q^3 \cdot c = \Omega(n)$.*

Proof: Given the separate proofs for the isolated model and for MAPs (i.e., Theorem 6.1 and Theorem D.2), the proof for the hybrid model follows naturally. We briefly describe how to combine the previous proofs to get the proof for the hybrid model.

Let R_Q and R_I denote the randomness of the querying and the interacting modules, respectively. Using the notation $T(\Pi, R_I)$ from the proof of Theorem 6.1, we decompose it to $T(\Pi, R_I) = (\mathcal{W}(\Pi), T'(\Pi, R_I))$, where $\mathcal{W}(\Pi)$ is the first prover message and $T'(\Pi, R_I)$ is the transcript of the rest of the interaction. We define the ω-heavy locations and the function G in the same way as in the proof of Theorem D.2, except that R is replaced by R_Q. We consider the same cheating provers $\{P_\pi\}_\pi$ as in the proof of Theorem 6.1, and analyze the verifier's view when interacting with the honest prover on a random permutation Π and its view when interacting with P_Π on G.

We fix R_I to an arbitrary randomness string, and denote the transcript $T'(\Pi, R_I)$ with this fixed randomness by $T'(\Pi)$. We then apply the analysis from the proof of Theorem D.2 as is, except that we add conditioning on $T'(\Pi)$ in every place where there is a condition on $\mathcal{W}(\Pi) = \omega$ (i.e., we condition on $(T'(\Pi), \mathcal{W}(\Pi) = \omega)$). Consequently, in Eq. (41) the term $\mathrm{I}\big(\Pi(i); \Pi(S_i^{\mathcal{W}(\Pi)}), \mathcal{W}(\Pi)\big)$ is replaced with $\mathrm{I}\big(\Pi(i); \Pi(S_i^{\mathcal{W}(\Pi)}), T'(\Pi), \mathcal{W}(\Pi)\big)$, and the proof follows by applying Lemma D.3 on

$$\sum_{i \in [n]} \mathrm{I}\big(\Pi(i); \Pi(S_i^{\mathcal{W}(\Pi)}), T'(\Pi), \mathcal{W}(\Pi)\big)$$

where the pair $(T'(\Pi), \mathcal{W}(\Pi))$ take the place of the term $\mathcal{W}(\Pi)$ appearing in the lemma. ■

We remark that the alternative proof of Theorem D.2 can similarly be adapted to the hybrid model by adding conditioning on $T'(\Pi, R_I)$ in every place where there is a condition on $\mathcal{W}(\Pi) = \omega$.[17] (As in the alternative proof of Theorem 6.1, we need to make the assumption that R_I is included in $T'(\Pi, R_I)$.)

D.4 Emulation of the hybrid model by MAPs

In [6, Thm. 1.2] it was shown that *statistically* sound IPPs in the isolated model can be efficiently emulated by testers. In this section we show that the same emulation technique implies that statistically sound IPPs in the hybrid model can be efficiently emulated by MAPs.

We define an isolated protocol to be a protocol as in an isolated IPP, but lacking the completeness and soundness conditions (i.e., with no conditions on the acceptance probability of any input). The value of such a protocol on input f is defined to be the probability that the verifier accepts f when interacting with the optimal prover (i.e., the one that maximizes the acceptance probability of f). The following theorem follows immediately from the proof of [6, Thm. 1.2].

Theorem D.9 (implicit in the proof of [6, Thm. 1.2])**.** *There exists a constant $\alpha > 0$ such that the value of any isolated protocol that uses q queries and c bits of communication on input f can be approximated to within an additive deviation of $1/6$ with probability of at least $2/3$, using only $\alpha \cdot q \cdot c$ queries to f.*

As shown in [6, Thm. 1.2], the above theorem leads to an efficient emulation of the isolated model by testers: For any function f, the value of the isolated IPP protocol on f is at least $2/3$ if f is in the property and at most $1/3$ if f is far from the property. An approximation of the value to within an additive deviation of $1/6$ allows for distinguishing these two cases. We use a similar approach to derive the emulation of the hybrid model by MAPs.

Theorem D.10 (efficient emulation of the hybrid model by MAPs)**.** *Suppose that Π is a property that can be verified in the hybrid model with query complexity q, a first message of length $c_P > 0$ and subsequent messages of total length $c_I > 0$. Then, Π has a MAP with query complexity $O(q \cdot c_I)$ and proof length c_P.*

Proof: We construct the MAP as follows: The honest MAP-proof is the first message that the honest prover sends in the hybrid IPP. Given input f and proof ω, the MAP-verifier approximates the value of the isolated protocol on f that occurs in the hybrid IPP after fixing the first prover message to ω.[18] Specifically, the verifier will follow the approximation procedure guaranteed by Theorem D.9, where the approximation is within additive deviation of $1/6$ with probability of at least $2/3$, and uses $\alpha \cdot q \cdot c_I$ queries to f (where α is the constant guaranteed by Theorem D.9). The verifier accepts if and only if the resulting approximated value is greater than $1/2$.

If f is in Π, then the honest proof leads to an isolated protocol on f with a value of at least $2/3$. On the other hand, if f is ϵ-far from Π, then any first prover message leads to an isolated protocol on f with a value of at most $1/3$. Thus, a successful approximation to within additive deviation of $1/6$ will correctly determine whether $f \in \Pi$ or f is ϵ-far from Π. ■

[17] Here we use Lemma D.3 with $(T'(\Pi, R_I), \mathcal{W}(\Pi))$ taking the place of the term $\mathcal{W}(\Pi)$ in the lemma. Note that this relies on the fact that in the lemma the term $\mathcal{W}(\Pi)$ is allowed to be any random variable (not only a deterministic function of Π).

[18] Note that unlike in an isolated IPP (i.e., an isolated protocol with soundness and completeness guarantees), in this isolated protocol there could be inputs in Π for which the value is less than $2/3$ (including values between $1/3$ and $2/3$). While every input in Π must have at least one first prover message that leads to an isolated protocol with a value of at least $2/3$, there could be other messages that lead to an isolated protocol with an arbitrary value.

Theorem D.10 implies that, in the case of statistical soundness, we can get a lower bound on the hybrid model for PERM by first applying the MAP emulation and then applying a known lower bound on MAPs for PERM. Specifically, by using the lower bound of [8, Lem. 4.3], which states that any (possibly adaptive) MAP for PERM with query complexity $q > 0$ and proof length $c > 0$ must satisfy $q \cdot c = \Omega(\sqrt{n})$, we get the following corollary:

Corollary D.11. *If* PERM *can be verified by an IPP in the hybrid model that has query complexity $q > 0$, a first message of length $c_P > 0$ and subsequent messages of total length $c_I > 0$, then $q \cdot c_I \cdot c_P = \Omega(\sqrt{n})$.*

Note that, as opposed to the lower bound established in Theorem D.8, this lower bound holds also for adaptive queries.

E Proof of Lemma D.3

In this section we prove Lemma D.3. The proof closely follows that of Lemma 6.2, which handles a special case in which the sets S_i are not dependent on the proof/transcript.

We begin with a technical claim that we will need in the proof. This claim generalizes the property that conditioning reduces entropy.

Claim E.1 (generalized conditioning reduces entropy)**.** *For any random variable X and any collection of random variables $\{Y_x\}_x$, it holds that*

$$\sum_{x,y} \Pr[X = x, Y_x = y] \cdot \log\left(\frac{1}{\Pr[X = x | Y_x = y]}\right) \leq \mathrm{H}(X)$$

Proof: Rearranging, we have

$$\begin{aligned}
&\mathrm{H}(X) - \sum_{x,y} \Pr[X = x, Y_x = y] \cdot \log\left(\frac{1}{\Pr[X = x | Y_x = y]}\right) \\
&= \sum_{x,y} \Pr[X = x, Y_X = y] \cdot \log\left(\frac{\Pr[X = x, Y_X = y]}{\Pr[X = x] \cdot \Pr[Y_x = y]}\right) \\
&\geq \left(\sum_{x,y} \Pr[X = x, Y_X = y]\right) \cdot \log\left(\frac{\left(\sum_{x,y} \Pr[X = x, Y_X = y]\right)}{\left(\sum_{x,y} \Pr[X = x] \cdot \Pr[Y_x = y]\right)}\right) \\
&= 1 \cdot \log\left(\frac{1}{1}\right) = 0
\end{aligned}$$

where the first inequality is by the log sum inequality, and in the last equality we use that $\sum_{x,y} \Pr[X = x] \cdot \Pr[Y_x = y] = \sum_x \big(\Pr[X = x] \cdot \sum_y \Pr[Y_x = y]\big) = \sum_x \Pr[X = x] = 1$. ■

Proof of Lemma D.3: First, it holds that

$$\begin{aligned}
\sum_{i \in [n]} \mathrm{I}\big(\Pi(i); \Pi(S_i^{\mathcal{W}(\Pi)}), \mathcal{W}(\Pi)\big) &= \sum_{i \in [n]} \mathrm{H}\big(\Pi(i)\big) - \sum_{i \in [n]} \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i^{\mathcal{W}(\Pi)}), \mathcal{W}(\Pi)\big) \\
&= n \cdot \log(n) - \mathbb{E}_{\omega \sim \mathcal{W}(\Pi)} \left[\sum_{i \in [n]} \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i^{\omega}), \mathcal{W}(\Pi) = \omega\big) \right]
\end{aligned} \tag{43}$$

Consider any $\omega \in \{0,1\}^c$. Let k be a parameter as in the proof of Lemma 6.2. Let P_ω be a partition of $[n]$ define analogously to the partition P in the proof of Lemma 6.2, such that (1) P_ω has at most $n/k + O(q) = O(n/k)$ parts each of size at most k, and (2) for each part $B \in P_\omega$ and each $i \in B$ it holds that S_i^ω does not intersect B. For each $B \in P_\omega$, define $S_B^\omega = \bigcup_{i \in B} S_i^\omega$. Note that $|S_B^\omega| \leq |B| \cdot \max_{i \in B}\{|S_i^\omega|\} \leq k \cdot (q-1)$, and that B and S_B^ω do not intersect. Analogously to Eq. (20) in Lemma 6.2, we have

$$\sum_{i \in [n]} \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i^\omega), \mathcal{W}(\Pi) = \omega\big) \geq \sum_{B \in P_\omega} \mathrm{H}\big(\Pi(B) \,\big|\, \Pi(S_B^\omega), \mathcal{W}(\Pi) = \omega\big) \tag{44}$$

The next steps are analogous to Eq. (21) in Lemma 6.2, but are more complex due to the fact that we condition on a specific proof string $\mathcal{W}(\Pi) = \omega$. For each part $B \in P_\omega$, it holds that

$$\begin{aligned}
&\mathrm{H}\big(\Pi(B) \,\big|\, \Pi(S_B^\omega), \mathcal{W}(\Pi) = \omega\big) \\
&= \sum_{a,b} \Pr[\Pi(B) = a, \Pi(S_B^\omega) = b | \mathcal{W}(\Pi) = \omega] \cdot \log\left(\frac{1}{\Pr[\Pi(B) = a | \Pi(S_B^\omega) = b, \mathcal{W}(\Pi) = \omega]}\right) \\
&\geq \sum_{a,b} \Pr[\Pi(B) = a, \Pi(S_B^\omega) = b | \mathcal{W}(\Pi) = \omega] \cdot \log\left(\frac{\Pr[\mathcal{W}(\Pi) = \omega | \Pi(S_B^\omega) = b]}{\Pr[\Pi(B) = a | \Pi(S_B^\omega) = b]}\right) \\
&= \sum_{a,b} \Pr[\Pi(B) = a, \Pi(S_B^\omega) = b | \mathcal{W}(\Pi) = \omega] \cdot \log\left(\frac{1}{\Pr[\Pi(B) = a | \Pi(S_B^\omega) = b]}\right) - \\
&\qquad \sum_{b} \Pr[\Pi(S_B^\omega) = b | \mathcal{W}(\Pi) = \omega] \cdot \log\left(\frac{1}{\Pr[\mathcal{W}(\Pi) = \omega | \Pi(S_B^\omega) = b]}\right)
\end{aligned} \tag{45}$$

where the inequality uses that $\Pr[\Pi(B) = a | \Pi(S_B^\omega) = b, \mathcal{W}(\Pi) = \omega] = \frac{\Pr[\Pi(B)=a, \mathcal{W}(\Pi)=\omega | \Pi(S_B^\omega)=b]}{\Pr[\mathcal{W}(\Pi)=\omega | \Pi(S_B^\omega)=b]} \leq \frac{\Pr[\Pi(B)=a | \Pi(S_B^\omega)=b]}{\Pr[\mathcal{W}(\Pi)=\omega | \Pi(S_B^\omega)=b]}$. Consider the first term of Eq. (45). Analogously to Eq. (22) in Lemma 6.2, for each a and b we have:

$$\begin{aligned}
\log\left(\frac{1}{\Pr[\Pi(B) = a | \Pi(S_B^\omega) = b]}\right) &= \log\left(\prod_{i \in [|B|]} (n - |S_B^\omega| + 1 - i)\right) \\
&\geq |B| \cdot \left(\log(n) - O\left(\frac{k \cdot q}{n}\right)\right)
\end{aligned} \tag{46}$$

Denoting the second term of Eq. (45) by ψ_ω, and combining Equations (44), (45) and (46), we get:

$$\begin{aligned}
\sum_{i \in [n]} \mathrm{H}\big(\Pi(i) \,\big|\, \Pi(S_i^\omega), \mathcal{W}(\Pi) = \omega\big) &\geq \sum_{B \in P_\omega} |B| \cdot \left(\log(n) - O\left(\frac{k \cdot q}{n}\right)\right) - |P_\omega| \cdot \psi_\omega \\
&= n \cdot \left(\log(n) - O\left(\frac{k \cdot q}{n}\right)\right) + O\left(\frac{n}{k}\right) \cdot \psi_\omega
\end{aligned}$$

since $|P_\omega| = O(n/k)$. Now, notice that $\mathbb{E}_{\omega \sim \mathcal{W}(\Pi)}[\psi_\omega]$ resembles a conditional entropy of $\mathcal{W}(\Pi)$. By Claim E.1, similarly to conditional entropy we have that

$$\mathbb{E}_{\omega \sim \mathcal{W}(\Pi)}[\psi_\omega] \leq \mathrm{H}(\mathcal{W}(\Pi)) \leq c$$

Hence, we get that

$$\begin{aligned}\mathbb{E}_{\omega\sim\mathcal{W}(\Pi)}\left[\sum_{i\in[n]}\mathrm{H}\big(\Pi(i)\,\big|\,\Pi(S_i^{\omega}),\mathcal{W}(\Pi)=\omega\big)\right] &\geq n\cdot\left(\log(n)-O\left(\frac{k\cdot q}{n}\right)\right)-O\left(\frac{n}{k}\right)\cdot\mathbb{E}_{\omega\sim\mathcal{W}(\Pi)}\left[\psi_{\omega}\right]\\ &\geq n\cdot\left(\log(n)-O\left(\frac{k\cdot q}{n}\right)\right)-O\left(\frac{n}{k}\right)\cdot c\end{aligned}$$

Combining this with Eq. (43), we get

$$\sum_{i\in[n]}\mathrm{I}\big(\Pi(i);\Pi(S_i^{\mathcal{W}(\Pi)}),\mathcal{W}(\Pi)\big)=O\left(k\cdot q+\frac{n\cdot c}{k}\right).$$

Setting $k=\Theta\left(\sqrt{\frac{c\cdot n}{q}}\right)$, the lemma follows. ■

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il