Electronic Colloquium on Computational Complexity, Report No. 181 (2025)

On Cryptography and Distribution Verification, with Applications to Quantum Advantage

Bruno Cavalar[1], Eli Goldin[2], Matthew Gray[1], Taiga Hiroka[3,4], Tomoyuki Morimae[4,3]

[1]University of Oxford, UK
{bruno.cavalar,matthew.gray}@cs.ox.ac.uk
[2]New York University, USA
eli.goldin@nyu.edu
[3]Hon-Hai Research Institute, Taiwan
taiga.hirooka@foxconn.com
[4]Yukawa Institute for Theoretical Physics, Kyoto University, Japan
tomoyuki.morimae@yukawa.kyoto-u.ac.jp

Abstract

One of the most fundamental problems in the field of hypothesis testing is the identity testing problem: whether samples from some unknown distribution $\mathcal{G}$ are actually from some explicit distribution $\mathcal{D}$. It is known that when the distribution $\mathcal{D}$ has support $[N]$, the optimal sample complexity for the identity testing problem is roughly $O(\sqrt{N})$. However, many distributions of interest, including those which can be sampled efficiently, have exponential support size, and therefore the optimal identity tester also requires exponential samples. In this paper, we bypass this lower bound by considering restricted settings. The above $O(\sqrt{N})$ sample complexity identity tester is constructed so that it is not fooled by any (even inefficiently-sampled) distributions. However, in most applications, the distributions under consideration are efficiently samplable, and therefore it is enough to consider only identity testers that are not fooled by efficiently-sampled distributions. In this setting we can hope to construct efficient identity testers. We investigate relations between efficient verification of classical/quantum distributions with classical/quantum cryptography, showing the following results:

- Classically efficiently samplable distributions are verifiable if and only if one-way functions do not exist.
- Quantumly efficiently samplable distributions are verifiable by $\mathbf{P^{PP}}$ with a polynomial number of samples.
- Sampling-based quantum advantage can be verified quantumly (with a polynomial number of samples) if one-way puzzles do not exist.
- If QEFID pairs exist, then some quantumly efficiently samplable distributions are not verifiable.

To obtain these results, we introduce a quantum variant of time-bounded Kolmogorov complexity, and show a coding theorem and incompressibility for it. These technical contributions may be of independent interest.

ISSN 1433-8092

Contents

1 Introduction

One of the most fundamental problems in the field of hypothesis testing is the following: is it possible to check whether samples from some unknown distribution $\mathcal{G}$ are actually from some explicit distribution $\mathcal{D}$? This is often known as the identity testing problem. The identity testing problem has been extremely well-studied in the statistical setting ([BFR+00, BFF+01, DKN15, Pan08, VV17] for some examples). Ignoring error terms, it is known that when the distribution $\mathcal{D}$ has support $[N]$, the optimal sample complexity for the identity testing problem is roughly $O(\sqrt{N})$ [BFF+01, Pan08, VV17].

However, many distributions of interest have exponential support size. In particular, distributions which can be sampled by efficient algorithms can have support exponential in their size. When this is the case, the optimal identity tester also requires exponential samples, and so is untenable in practice.

One may hope to bypass this lower bound by considering restricted settings. The above optimal identity tester is constructed so that it is not fooled by any (even inefficiently-sampled) distributions. However, in most applications, the distributions under consideration are efficiently-samplable, and therefore it is enough to consider only identity testers that are not fooled by efficiently-sampled distributions. In that case, we can focus on efficient identity testers. Let us call a distribution efficiently verifiable if there exists an efficient identity tester which cannot be fooled by efficiently-sampled distributions. We then have the following natural question, which is the main question of this work:

Which distributions are efficiently verifiable, and under what assumptions?

Verification of Sampling-Based Quantum Advantage The identity testing problem is also important in the verification of quantum advantage. Quantum advantage refers to the existence of computational tasks that are easy for quantum computers but hard for classical ones. Sampling-based quantum advantage [BJS11, AA11] is one of the most well-studied frameworks for demonstrating quantum advantage. Several sub-universal quantum computing models have been shown to exhibit sampling-based quantum advantage: their output probability distributions cannot be classically efficiently sampled under some computational assumptions.[1] For example, the Boson sampling model [AA11] (quantum computing with non-interacting photons), the IQP model [BJS11, BMS16] (quantum computing with commuting gates), the one-clean-qubit model [FKM+18] (quantum computing with a single pure qubit), and the random-circuit model [BFNV19] (quantum computing with random gates) are well-studied sub-universal models.

One advantage of sampling-based quantum advantage is that the experimental implementations seem plausible to achieve with near-future devices. On the other hand, one often-claimed disadvantage of sampling-based quantum advantage compared with more sophisticated frameworks, such as proofs of quantumness [BCM+18], is the lack of efficient (or even inefficient) verification.

This disadvantage of sampling-based quantum advantage has been justified by appealing to the intuition that verifying distributions is impossible (even inefficiently). For example, [HKEG19] showed that verifying distributions requires exponentially many samples. However, in their impossibility result, unknown distributions can be any (even inefficiently-sampled) distributions, while, in the realistic setting, considering only efficiently-sampled distributions is enough.

Hence, in the context of the verification of sampling-based quantum advantage, we have the following specific questions:

Is there a notion of verifying distributions appropriate for the setting of sampling-based quantum advantage? If so, when can this notion be achieved?

[1] The assumptions are that the polynomial-time hierarchy will not collapse to the third (or second) level, and some additional newly-introduced assumptions called the "anticoncentration property" and "average-case $\sharp\mathbf{P}$-hardness".

1.1 Our Contributions

In this paper, we answer those questions. Our results are summarized as follows.

1. We provide a formal definition of verification of distributions.
2. Every quantumly (resp. classically) efficiently-samplable distribution is verifiable with a classical deterministic polynomial-time algorithm querying to a $\mathbf{PP}$ oracle (resp. a probabilistic polynomial-time algorithm querying to an $\mathbf{NP}$ oracle). In particular, this means that sampling-based quantum advantage can be verified with a *polynomial* number of samples by a classical *polynomial*-time algorithm querying to a $\mathbf{PP}$ oracle.
3. If one-way functions do not exist, then every classically-efficiently-samplable distribution is efficiently verifiable.
4. If one-way puzzles do not exist, then sampling-based quantum advantage can be verified with an efficient quantum algorithm.
5. If one-way functions exist, then any classically-efficiently-samplable distribution with high entropy is not efficiently verifiable.
6. If QEFID pairs exist, then there exists a quantumly-efficiently-samplable distribution which is not efficiently verifiable.
7. Every classically (resp. quantumly) efficiently-samplable distribution with small entropy can be verified with an efficient classical (resp. quantum) algorithm.

In the following, we explain each contribution in detail.

Defining efficiently verifiable distributions. Our first contribution is a formal definition of verification of distributions.

A previous definition of verification of distributions that is often studied is the following [Can20]: Let $\mathcal{D}$ be a (not-necessarily-efficiently-samplable) distribution. We say that $\mathcal{D}$ is verifiable if there exists a (not-necessarily-efficient) algorithm Ver and a (not-necessarily-polynomial) function s such that

- $\Pr[\top \leftarrow \mathsf{Ver}(x_1, ..., x_s) : (x_1, ..., x_s) \leftarrow \mathcal{D}^{\otimes s}]$ is large.[2]
- For any (not-necessarily-efficient) adversary $\mathcal{A}$, if $\Delta(\mathcal{A}, \mathcal{D})$[3] is large then

$$\Pr[\top \leftarrow \mathsf{Ver}(x_1, ..., x_s) : (x_1, ..., x_s) \leftarrow \mathcal{A}^{\otimes s}]$$

  is small.

Intuitively, this means that samples sampled from the ideal distribution $\mathcal{D}$ are accepted with high probability, but samples sampled from distributions that are far from $\mathcal{D}$ are rejected with high probability.

One issue of using this definition for our purpose is that $\mathcal{D}$, $\mathcal{A}$, and Ver are not necessarily efficient, and s is not necessarily polynomial. In a realistic setting of verification of distributions, such as verification of

[2]Here, $(x_1, ..., x_s) \leftarrow \mathcal{D}^{\otimes s}$ means that the collection $(x_1, ..., x_s)$ is the result of s independent samples from $\mathcal{D}$, and x_i is the result of the ith sampling.

[3]Here, $\Delta(\mathcal{A}, \mathcal{D})$ is the statistical distance between the output ditribution of $\mathcal{A}$ and $\mathcal{D}$.

sampling-based quantum advantage, $\mathcal{D}$ should be quantum polynomial-time samplable (QPT-samplable) and s should be polynomial. Moreover, as already discussed, for many applications we only need to consider probabilistic-polynomial time (PPT) classical algorithms $\mathcal{A}$.

We therefore consider the following definition, and call it *selective-verifiability*: Let $\mathcal{D}$ be any distribution. We say that $\mathcal{D}$ is selectively-verifiable with a PPT verification algorithm if there exists a PPT algorithm Ver and a polynomial s such that

- $\Pr[\top \leftarrow \mathsf{Ver}(x_1, ..., x_s) : (x_1, ..., x_s) \leftarrow \mathcal{D}^{\otimes s}]$ is large.
- For any PPT adversary $\mathcal{A}$, if $\Delta(\mathcal{A}, \mathcal{D})$ is large then

$$\Pr[\top \leftarrow \mathsf{Ver}(x_1, ..., x_s) : (x_1, ..., x_s) \leftarrow \mathcal{A}^{\otimes s}]$$

  is small.

The definition is, however, still not satisfactory, because the adversary's output is assumed to be independently and identically (i.i.d.) distributed. This is insufficient for the purpose of verifying quantum advantage, for example.

In particular, let us consider the situation where someone with a quantum computer wants to prove to the public that they have a quantum computer. They would like to use an efficiently verifiable sampling-based advantage protocol to do this. In particular, they will run their sampling-based advantage protocol many times, publish their outputs, and then the public can use the classical verifier to check that the outputs really came from a quantum distribution. Selective-verifiability will be insufficient in practice for this scenario. In particular, even if there is no single PPT-samplable distribution such that s i.i.d. samples pass verification, there may still be a sampler which outputs s correlated samples passing verification.

We therefore introduce the following definition, which we call *adaptive-verifiability*: Let $\mathcal{D}$ be any distribution. We say that $\mathcal{D}$ is *adaptively-verifiable* with a PPT verification algorithm if there exists a PPT algorithm Ver and a polynomial s such that

- $\Pr[\top \leftarrow \mathsf{Ver}(x_1, ..., x_s) : (x_1, ..., x_s) \leftarrow \mathcal{D}^{\otimes s}]$ is large.
- For any PPT adversary $\mathcal{A}$ that outputs correlated bit strings $(x_1, ..., x_s)$, if $\Delta(\mathsf{Marginal}_{\mathcal{A}}, \mathcal{D})$ is large then

$$\Pr[\top \leftarrow \mathsf{Ver}(x_1, ..., x_s) : (x_1, ..., x_s) \leftarrow \mathcal{A}]$$

  is small, where $\mathsf{Marginal}_{\mathcal{A}}$ is the distribution defined as follows:

  1. Run $(x_1, ..., x_s) \leftarrow \mathcal{A}$.
  2. Choose $i \leftarrow [s]$.
  3. Output x_i.

Intuitively, in this definition, Ver can verify that a randomly-chosen marginal distribution of $\mathcal{A}$ is close to the ideal distribution $\mathcal{D}$.

Complexity upper bound on verification. Our second contribution is showing the following upper bounds of the complexity of the verification of distributions.

Theorem 1.1. *Every PPT-samplable distribution is adaptively-verifiable with polynomially-many samples and with a classical probabilistic polynomial-time algorithm querying an* $\mathbf{NP}$ *oracle.*[4]

Theorem 1.2. *Every QPT-samplable distribution is adaptively-verifiable with polynomially-many samples and with a classical deterministic polynomial-time algorithm querying a* $\mathbf{PP}$ *oracle.*

Let us clarify the significance of Theorem 1.2 in the context of sampling-based quantum advantage. It seems that the following is widely believed:

- Verification of distributions sampled with quantum computers requires unbounded time.
- Verification of distributions sampled with quantum computers requires exponentially-many samples. (For example, see [HKEG19].)

However, these beliefs are often based on the intuition implicitly assuming that $\mathcal{D}$ is not necessarily efficiently-samplable and $\mathcal{A}$ is not necessarily efficient. In Theorem 1.2, on the other hand, we focus on QPT-samplable $\mathcal{D}$ and PPT $\mathcal{A}$. Because of the restriction, we can show from Theorem 1.2 that

- Verification of QPT-samplable distributions can be done with a classical deterministic *polynomial*-time algorithm querying to $\mathbf{PP}$ oracle.
- Verification of QPT-samplable distributions can be done with only *polynomially*-many samples.

To our knowledge, this is the first time that verification of quantum distributions is shown to be possible with a classical *efficient* algorithm (querying to $\mathbf{PP}$ oracle) and with *polynomially*-many samples.

Several search-type problems have been introduced that exhibit quantum advantage and that can be verified by a classical exponential time verifier. For instance, [AC17] introduced the Heavy Output Generation (HOG) problem and argued for its classical hardness via the QAUTH conjecture. [NRK+18, BIS+18, AAB+19] studied cross-entropy benchmarking, and [AG20] supported its hardness under the XQUATH conjecture. It is often claimed that these search-type quantum advantage are better than sampling-based quantum advantage, because the former can be verified at least inefficiently, while the latter cannot. However, our result shows that sampling-based quantum advantage is also "inefficiently-verifiable". One advantage of sampling-based quantum advantage over these search problems is that the assumptions have been better studied.[5]

Cryptographic upper bound on verification. In Theorems 1.1 and 1.2, we showed that efficient verification is generically possible in the classical and quantum settings if $\mathbf{BPP} \supseteq \mathbf{NP}$ or $\mathbf{BQP} = \mathbf{PP}$ respectively. We may then ask, can we weaken these assumptions?

The key idea behind the proof of Theorem 1.2 is that the task of verification can be reduced to the following task: given an output x of an efficiently-samplable distribution $\mathcal{D}$, estimate the probability that $\mathcal{D}$ outputs x up to multiplicative error. It is known that if one-way functions (OWFs) do not exist, then this task is possible *on average* for classically-efficiently-samplable distributions [IL90, IRS22, CHK25]. Recently,

[4] We will not include a direct proof of this statement, since it follows directly from Theorem 1.3. A direct argument can also be made using the fact that probability estimation can be solved in the worst-case using an $\mathbf{NP}$ oracle [Sto83].

[5] Indeed, recent works [OJF23, TGB24, GKC+24] have raised doubts about those conjectures.

[CGGH25, HM25, KT25] showed that if one-way puzzles (OWPuzzs) do not exist, then this task is possible on average for quantumly-efficiently-samplable distributions.[6]

Using this idea, we can directly show the following result in the classical setting.

Theorem 1.3. *Assume that infinitely-often OWFs do not exist. Then every PPT-samplable distribution is adaptively-verifiable with a PPT verification algorithm.*

Unfortunately, due to certain technical difficulties, the same technique does not apply in the quantum setting (for details, see the Technical Overview section). However, we can show at least that the non-existence of OWPuzzs can be used to efficiently verify strong quantum advantage samplers (QASs) [MSY25] with a *QPT* verification algorithm.

Theorem 1.4. *If there do not exist infinitely-often OWPuzzs, then every strong QAS is adaptively-verifiable with a QPT algorithm against PPT adversaries.*

A QAS [MSY25] is a QPT-samplable distribution whose statistical distance is far from any PPT-samplable distribution. More formally, a QAS is a QPT-samplable distribution $\mathcal{Q}$ such that $\Delta(\mathcal{Q}(1^n), \mathcal{D}(1^n)) \geq 1/\mathrm{poly}(n)$ for any PPT-samplable distribution $\mathcal{D}$. As is shown in [MSY25], we can construct QASs based on standard assumptions in the field of sampling-based quantum advantage.[7] In this work, we consider strong QAS, which is a QPT-samplable distribution $\mathcal{Q}$ such that $\Delta(\mathcal{Q}(1^n), \mathcal{D}(1^n)) \geq 1 - \mathsf{negl}(n)$ for any PPT algorithm $\mathcal{D}$. Although we do not know how to construct a strong QAS from a QAS, we can construct it from a plausible cryptographic assumption.[8] Theorem 1.4 can be applied to only strong QASs, and we do not know how to extend it to (not-necessarily-strong) QASs. For details, see the Technical Overview section.

Theorem 1.4 is incomparable with Theorem 1.2: In terms of the assumption, Theorem 1.4 is better than Theorem 1.2, because the non-existence of OWPuzzs is weaker than $\mathbf{PP} = \mathbf{BQP}$. However, Theorem 1.4 can verify only QASs, and the verification algorithm is quantum.

This result has the following interesting corollary.

Corollary 1.5. *If strong QASs exist and infinitely-often OWPuzz do not, then there exists an efficiently-verifiable non-interactive proof of quantumness with uniform security, where the verifier is quantum and knows an upper bound on the runtime of the adversary.*

Furthermore, it is known that if OWPuzzs secure against PPT adversaries exist but infinitely-often OWFs do not, then a strong QAS exists (with uniform security) [MSY25][9]. Thus, we have the following curious result:

[6] OWPuzzs are a natural quantum analogue of OWFs. A OWPuzz is a pair $(\mathsf{Samp}, \mathsf{Ver})$ of algorithms. $\mathsf{Samp}(1^n) \to (\mathsf{puzz}, \mathsf{ans})$ is a QPT algorithm that, on input the security parameter, outputs two bit strings puzz and ans. $\mathsf{Ver}(\mathsf{puzz},) \to \top/\bot$ is an unbounded algorithm that, on input puzz and ans, outputs $\top/\bot$. The security requires that no QPT adversary, given puzz, cannot output ans' such that $\mathsf{Ver}(\mathsf{puzz}, \mathsf{ans}') \to \top$ except for a negligible probability.

[7] The assumption that the polynomial-hierarchy will not collapse to the third level (or the second level), the anti-concentration property, and so-called average-case $\sharp\mathbf{P}$-hardness.

[8] A strong QAS can be obtained from a QPRG secure against a QPT algorithm querying to an $\mathbf{NP}$ oracle. A QPRG is a QPT algorithm that outputs a classical bit string on input the security parameter such that its output probability distribution is statistically far but computationally indistinguiable from the uniform distribution. We consider security against a QPT algorithm querying to an $\mathbf{NP}$ oracle. For completeness, we describe the proof in Appendix B.

[9] See Theorem 1.7. Note that the authors only show that standard QAS exist. However, the argument trivially extends to strong QAS. In particular, the sampler for the OWPuzz will itself be a strong QAS, since if not then it can be sampled up to 1/poly error by a classical machine, and thus that classical machine is a weak, distributional, infinitely-often, classically secure one-way function (from which it is known how to construct a full one-way function).

Corollary 1.6. *If one-way puzzles secure against classical adversaries exist but quantum secure infinitely-often one-way puzzles and classically secure infinitely-often one-way functions do not, then there exists an efficiently-verifiable non-interactive proof of quantumness with uniform security, where the verifier is quantum and knows an upper bound on the runtime of the adversary.*

Proofs of quantumness [BCM+21] studied in the literature almost always have classical verification. However, it is reasonable to consider the case of quantum verification. In particular, such a protocol would allow two quantum computers to verify each other, thus diversifying trust. Furthermore, non-interactive proofs of quantumness (a primitive only possible in a uniform security model) are especially interesting, even if they only allow for quantum verification. In particular, with a non-interactive proof of quantumness, a quantum prover can generate some string using their machine, post it publicly, and then all other quantum computer owners can verify that the string was really generated quantumly.

Cryptographic lower bound on verification. We also show that if cryptography exists, then efficiently-samplable distributions are not efficiently verifiable.

In the classical setting, we show that if (infinitely-often) OWFs exist,[10] then any PPT-samplable distribution with high entropy is unverifiable.

Theorem 1.7. *If infinitely-often OWFs exist, then any classically-efficiently-samplable distribution $\mathcal{D}(1^n)$ with $H(\mathcal{D}(1^n)) = n^{\Omega(1)}$ is not selectively-verifiable with a PPT verification algorithm. Here, $H(\mathcal{D}(1^n))$ represents the Shannon entropy of $\mathcal{D}(1^n)$.*

In the quantum setting, we instead observe that if quantum cryptography exists, then there is some unverifiable distribution.

Theorem 1.8. *If infinitely-often QEFID pairs exist, then there exists a QPT-samplable distribution that is not selectively-verifiable with a QPT verification algorithm.*

Here, QEFID pairs [CGG24, KT24] are one of the natural quantum analogues of pseudorandom generators (PRGs). Formally, a QEFID pair is a QPT algorithm G that takes the security parameter 1^n and a bit $b \in \{0,1\}$ as input and outputs a classical bit string, such that $G(1^n, 0)$ and $G(1^n, 1)$ are statistically far but computationally indistinguishable. QEFID pairs are implied by pseudorandom state generators (PRSGs) [JLS18][11] and imply OWPuzzs [CGG24]. The above result is obtained by considering the verification of either distribution in a QEFID pair. It is clear that this cannot be efficiently verified, since the other distribution in the QEFID pair will also pass verification.

Although it is widely believed that QPT-samplable distributions (including those that exhibit sampling-based quantum advantage) cannot be efficiently verified, as far as we know, there was no formal proof. Theorem 1.8 shows that, based on a plausible assumption on the existence of a quantum cryptographic primitive, the verification of some QPT-samplable distributions cannot be done efficiently.

Theorem 1.8 also suggests that the upperbound of $\mathbf{PP}$ in Theorem 1.2 cannot be improved to $\mathbf{QMA}$. This is because if it were the case, then QEFID would be broken by $\mathrm{PPT}^{\mathbf{QMA}}$, which contrasts with the known fact that PRSGs (and therefore QEFID) can exist even if $\mathbf{BQP} = \mathbf{QMA}$ [Kre21] (under some oracle).

[10] Infinitely-often OWFs are OWFs whose security hold for infinitely-many security parameters. For the definition, see Definition 2.1.

[11] The construction of QEFID from PRSGs is known as a folklore, but we do not know any paper that explicitly shows it. By using shadow tomography [HKP20], QEFID can be obtained from PRFSGs, which can be constructed from PRSGs [AQY22].

Unconditional Verification of Distributions with Small Entropy. Finally, we also study which distributions can be efficiently verified without relying on any computational assumptions.

Theorem 1.9. *Every PPT (resp. QPT)-samplable distribution $\mathcal{D}(1^n)$ with $H(\mathcal{D}(1^n)) = O(\log n)$ is adaptively-verifiable with a PPT (resp. QPT) algorithm. Here, $H(\mathcal{D}(1^n))$ represents the Shannon entropy of $\mathcal{D}(1^n)$.*

Remark that it is arguably impossible to extend Theorem 1.9 to higher-entropy distributions without any computational assumptions. This is because if a PPT-samplable distribution has high entropy, it cannot be verifiable assuming the existence of OWFs as shown in Theorem 1.7. Therefore, the entropy condition in Theorem 1.9 is almost optimal. [12]

1.2 Technical Overview

In this subsection, we provide a high level overview of our main results.

Inefficient verification with polynomially-many samples. Aaronson [Aar14] showed the equivalence between $\mathbf{SampBQP} = \mathbf{SampBPP}$ and $\mathbf{FBQP} = \mathbf{FBPP}$. His proofs are based on Kolmogorov complexity. By using a similar argument, one can construct an *inefficient* adaptive-verification of distributions with *polynomially-many* samples as follows: Let $\mathcal{D}$ be an algorithm that takes 1^n as input and outputs $y \in \{0,1\}^{m(n)}$, where m is a polynomial. Let s be a polynomial. We define a verification algorithm Ver as follows: On input $(y_1, ..., y_{s(n)})$, it first computes both $K\left(y_1, \dots, y_{s(n)}\right)$ and $\Pr\left[(y_1, \dots, y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]$. It then outputs $\top$ if

$$K\left(y_1, \dots, y_{s(n)}\right) \geq \log_2\left(\frac{1}{\Pr\left[(y_1, ..., y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]}\right) - (\log n)^2,$$

and otherwise, outputs $\bot$. Here, $K(x)$ is the Kolmogorov complexity of a bit string x, which, roughly speaking, is the length of the shortest binary program that outputs x.

We can show that the following two properties are satisfied:

- **Correctness.** $\mathsf{Ver}\left(y_1, \dots, y_{s(n)}\right) = \top$ with high probability when $(y_1, \dots, y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}$.
- **Security.** For any uniform adversary $\mathcal{A}$ that outputs $(y_1, ..., y_{s(n)})$ such that the marginal distribution $\mathsf{Marginal}_{\mathcal{A}}$ is statistically far from $\mathcal{D}$, $\mathsf{Ver}\left(y_1, \dots, y_{s(n)}\right) = \bot$ with high probability. Here, $\mathsf{Marginal}_{\mathcal{A}}(1^n)$ denotes the algorithm that samples $y_1, \dots, y_{s(n)} \leftarrow \mathcal{A}(1^n)$, chooses $i \leftarrow [s(n)]$, and outputs y_i.

Correctness and security can be shown based on the two key properties of Kolmogorov complexity, namely, incompressibility and the coding theorem. The incompressibility property guarantees that, for every algorithm $\mathcal{D}$, we have

$$K\left(y_1, \dots, y_{s(n)}\right) \geq \log_2\left(\frac{1}{\Pr\left[(y_1, \dots, y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]}\right) - (\log n)^2$$

[12] Strictly speaking, it is not prohibited to slightly improve the entropy parameter in Theorem 1.9 because Theorem 1.7 only prohibits a verification of PPT-samplable distribution $\mathcal{D}(1^n)$ with $H(\mathcal{D}(1^n)) = n^{\Omega(1)}$.

with high probability over $(y_1, \dots, y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}$, which directly implies the correctness of Ver. The coding theorem guarantees that for every algorithm $\mathcal{A}$ there exists a constant C such that, for all $y_1, \dots, y_{s(n)}$,

$$K\left(y_1, \dots, y_{s(n)}\right) \leq \log_2\left(\frac{1}{\Pr\left[(y_1, \dots, y_{s(n)}) \leftarrow \mathcal{A}(1^n)\right]}\right) + C.$$

We can show that the coding theorem implies the security of Ver as follows: Intuitively, if $\mathsf{Marginal}_{\mathcal{A}}$ is statistically far from $\mathcal{D}$, then with high probability over $(y_1, \dots, y_{s(n)}) \leftarrow \mathcal{A}(1^n)$, we have

$$\Pr\left[(y_1, ..., y_{s(n)}) \leftarrow \mathcal{A}(1^n)\right] \gg \Pr\left[(y_1, ..., y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right].$$

By the coding theorem, this implies

$$K\left(y_1, ..., y_{s(n)}\right) \leq \log_2\left(\frac{1}{\Pr\left[(y_1, ..., y_{s(n)}) \leftarrow \mathcal{A}(1^n)\right]}\right) \ll \log_2\left(\frac{1}{\Pr\left[(y_1, ..., y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]}\right)$$

with high probability over $(y_1, \dots, y_{s(n)}) \leftarrow \mathcal{A}(1^n)$. Since Ver rejects any $y_1, \dots, y_{s(n)}$ with such small Kolmogorov complexity, the security follows.

In this construction, although the number of samples is polynomial, one crucial issue of the construction is that Ver is uncomputable, since Kolmogorov complexity $K(x)$ is generally uncomputable. To overcome this issue, we introduce a quantum variant of time-bounded Kolmogorov complexity, and show that it satisfies incompressibility and the coding theorem as is shown in the next paragraph.[13]

Quantum imcompressibility and quantum coding theorem. We introduce a quantum time-bounded Kolmogorov complexity, $quK^t(x)$, which is a quantum variant of (classical) time-bounded Kolmogorov complexity ($uK^t(x)$) [LV93, HN23][14]. To define $quK^t(x)$, we first introduce *the universal quantum probability distribution* $\mathcal{Q}^t(1^n)$, defined as follows. Let $\mathcal{U}$ be a universal Turing machine, and denote by $\mathcal{U}^t(\Pi, 1^n)$ the output tape of $\mathcal{U}$ on input Π and 1^n after t steps. $\mathsf{QU}^t(1^n, c)$ is a quantum algorithm treating a binary string c as an encoding of an output length m, an input size s, a $t(n)$-depth quantum circuit C, and then it runs $C|0^s\rangle$, measures the first m-bits, and outputs the resulting m bits. The *universal quantum probability distribution* $\mathcal{Q}^t(1^n)$ samples $\Pi \leftarrow \{0,1\}^t$ uniformly at random, obtains $c \leftarrow \mathcal{U}^t(\Pi, 1^n)$, runs the algorithm $x \leftarrow \mathsf{QU}^t(1^n, c)$, and then outputs x.[15] We define $quK^t(x|1^n)$ as

$$quK^t(x|1^n) := \log_2\left(\frac{1}{\Pr[x \leftarrow \mathcal{Q}^t(1^n)]}\right).$$

We can show that $quK^t(x|1^n)$ satisfies the coding theorem. In other words, there exists a constant C such that

$$quK^t(x|1^n) \leq \log_2\left(\frac{1}{\Pr[x \leftarrow \mathcal{D}(1^n)]}\right) + C$$

[13] [Aar14] already observed the issue, and pointed out that by considering space-bounded Kolmogorov complexity, at least **PSPACE** verification is possible.

[14] [HN23] uses $q^t(x)$ to mean $uK^t(x)$. To avoid the possibility for confusion that the q in q^t [HN23] might stand for "quantum" we relabel the notion as uK^t with the u standing for "universal".

[15] In quantum computing, the standard model of computation is the quantum circuit model rather than the quantum Turing machine. Therefore, we define $\mathcal{Q}^t(1^n)$ by using quantum circuits instead of a universal quantum Turing machine.

for any $x \in \{0,1\}^*$ and any QPT algorithm $\mathcal{D}$ for all sufficiently large t. Intuitively, this is because $\mathcal{Q}^t(1^n)$ emulates every QPT algorithm, and hence we have

$$\Pr\left[x \leftarrow \mathcal{Q}^t(1^n)\right] \geq \frac{\Pr[x \leftarrow \mathcal{D}(1^n)]}{C}$$

for some constant C, for any QPT algorithm $\mathcal{D}$, any $x \in \{0,1\}^*$, and for all sufficiently large t.

We can also show that $quK^t(x)$ is incompressible. This means that, for any algorithm $\mathcal{D}$, the inequality

$$quK^t(y_1,\dots,y_{s(n)}|1^n) < \log_2\left(\frac{1}{\Pr\left[(y_1,\dots,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]}\right) - (\log n)^2$$

holds only with negligible probability over $(y_1,\dots,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}$. The reason is as follows. For any tuple $y_1,\dots,y_{s(n)}$ with

$$quK^t(y_1,\dots,y_{s(n)}|1^n) < \log_2\left(\frac{1}{\Pr\left[(y_1,\dots,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]}\right) - (\log n)^2,$$

we have

$$\Pr\left[(y_1,\dots,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right] \leq n^{-\log(n)} \cdot 2^{-quK^t(y_1,\dots,y_{s(n)}|1^n)}.$$

Moreover, for each $i \in \mathbb{N}$, the number of tuples $y_1,\dots,y_{s(n)}$ with $quK^t(y_1,\dots,y_{s(n)}|1^n) = i$ is at most 2^i, so we have

$$\Pr_{(y_1,\dots,y_s)\leftarrow\mathcal{D}(1^n)^{\otimes s(n)}}\left[\begin{array}{c} quK^t(y_1,\dots,y_{s(n)}|1^n) = i \\ \wedge \\ quK^t(y_1,\dots,y_{s(n)}|1^n) < \log_2\left(\frac{1}{\Pr\left[(y_1,\dots,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]}\right) - (\log n)^2 \end{array}\right] \leq n^{-\log(n)}.$$

Since $quK^t(y_1,\dots,y_{s(n)}|1^n)$ ranges between 0 and $\mathrm{poly}(n)$, the total probability of the above event is bounded by $\mathrm{poly}(n)n^{-\log(n)}$, which is negligible. This establishes the incompressibility of quK^t.

Upper bound. Because quK^t satisfies incompressibility and the coding theorem, the following verification algorithm (Ver) works: Output $\top$ if

$$quK^t\left(y_1,...,y_{s(n)}|1^n\right) \geq \log_2\left(\frac{1}{\Pr\left[(y_1,...,y_{s(n)}) \leftarrow \mathcal{D}(1^n)\right]}\right) - (\log n)^2,$$

and otherwise, output $\bot$.

Although this Ver is still not efficient, we can show that Ver can be executed at least in classical deterministic polynomial-time querying to $\mathbf{PP}$ oracle, because a classical deterministic polynomial-time algorithm querying to $\mathbf{PP}$ oracle can compute both $\Pr\left[(y_1,\dots,y_s) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]$ and $quK^t(y_1,\dots,y_{s(n)}|1^n)$ [FR99].

Lower bound. A QEFID is a QPT algorithm G that takes a security parameter 1^n and a bit $b \in \{0,1\}$ as inputs, and outputs a classical binary string such that $G(1^n,0)$ and $G(1^n,1)$ are statistically far but

computationally indistinguishable. We claim that the distribution $G(1^n,0)$ is not selectively-verifiable because of the following reason: From the computational indistinguishability, we have

$$\left|\Pr_{(y_1,\dots,y_{s(n)})\leftarrow G(1^n,0)^{\otimes s(n)}}[\top\leftarrow \mathsf{Ver}(1^n,y_1,\dots,y_{s(n)})]\right.$$
$$\left.-\Pr_{(y_1,\dots,y_{s(n)})\leftarrow G(1^n,1)^{\otimes s(n)}}[\top\leftarrow \mathsf{Ver}(1^n,y_1,\dots,y_{s(n)})]\right|\leq \mathsf{negl}(n)$$

for any polynomial s and any QPT algorithm Ver. Therefore, if

$$\Pr_{(y_1,\dots,y_{s(n)})\leftarrow G(1^n,0)^{\otimes s(n)}}[\top\leftarrow \mathsf{Ver}(1^n,y_1,\dots,y_{s(n)})]$$

is large, $\Pr_{(y_1,\dots,y_{s(n)})\leftarrow G(1^n,1)^{\otimes s(n)}}[\top\leftarrow \mathsf{Ver}(1^n,y_1,\dots,y_{s(n)})]$ also must be large. This means that $G(1^n,0)$ is not selectively-verifiable.

Verification of PPT-sampleable distributions based on the non-existence of OWFs. We now use $uK^t(x)$ [LV93, HN23], which is a classical analogue of $quK^t(x)$. Similar to $quK^t(x)$, the complexity measure $uK^t(x)$ also satisfies incompressibility and a coding theorem [LV93, HN23]. Therefore, we can verify whether a marginal $\mathsf{Marginal}_{\mathcal{A}}(1^n)$ of a PPT algorithm $\mathcal{A}(1^n)$ is close to $\mathcal{D}(1^n)$ by checking

$$uK^t\left(y_1,...,y_{s(n)}|1^n\right)\geq \log_2\left(\frac{1}{\Pr\left[(y_1,...,y_{s(n)})\leftarrow\mathcal{D}(1^n)\right]}\right)-(\log n)^2.$$

To compute $uK^t\left(y_1,...,y_{s(n)}|1^n\right)$ and $\Pr\left[(y_1,...,y_{s(n)})\leftarrow\mathcal{D}(1^n)\right]$, we use the non-existence of OWFs. More concretely, if OWFs do not exist, then for any PPT algorithm $\mathcal{A}$, there exists a PPT algorithm $\mathcal{M}$ such that

$$uK^t\left(y_1,...,y_{s(n)}|1^n\right)-1\leq\mathcal{M}\left(y_1,...,y_{s(n)}\right)\leq uK^t\left(y_1,...,y_{s(n)}|1^n\right)+1$$

with high probability over $(y_1,...,y_{s(n)})\leftarrow\mathcal{A}(1^n)$ [IL90, HN23]. There also exists another PPT algorithm Approx such that, for any PPT algorithm $\mathcal{D}$,

$$\frac{\Pr\left[(y_1,...,y_{s(n)})\leftarrow\mathcal{D}(1^n)^{\otimes s(n)}\right]}{2}\leq\mathsf{Approx}\left(y_1,...,y_{s(n)}\right)$$

with high probability over $(y_1,...,y_{s(n)})\leftarrow\mathcal{D}(1^n)^{\otimes s(n)}$. Furthermore, and crucially, Approx never overestimates $\Pr\left[(y_1,...,y_{s(n)})\leftarrow\mathcal{D}(1^n)^{\otimes s(n)}\right]$, i.e.,

$$\mathsf{Approx}\left(y_1,...,y_{s(n)}\right)\leq\Pr\left[(y_1,...,y_{s(n)})\leftarrow\mathcal{D}(1^n)^{\otimes s(n)}\right]$$

for all $y_1,\dots,y_{s(n)}$, with high probability over the internal randomness of Approx [IL90, IRS22, CHK25].

Our Ver algorithm outputs $\top$ if

$$\mathcal{M}\left(y_1,...,y_{s(n)}\right)\geq\log_2\left(\frac{1}{\mathsf{Approx}\left(y_1,...,y_{s(n)}\right)}\right)-(\log n)^2,$$

and otherwise outputs $\bot$. The correctness holds because, with high probability over $(y_1,...,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}$, the algorithms $\mathcal{M}$ and Approx do not fail to estimate $uK^t\left(y_1,...,y_{s(n)}\right)$ and $\Pr\left[(y_1,...,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]$, respectively.

The security also holds for the following reasons. In order for $\mathsf{Ver}\left(y_1,...,y_{s(n)}\right)$ to output $\top$, the quantity $\log_2\left(\frac{1}{\mathsf{Approx}\left(y_1,...,y_{s(n)}\right)}\right)$ would need to be incorrectly small compared to $\log_2\left(\frac{1}{\Pr\left[(y_1,...,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]}\right)$. This would mean that $\mathsf{Approx}\left(y_1,...,y_{s(n)}\right)$ must be incorrectly large. However, by the definition of Approx, $\mathsf{Approx}\left(y_1,...,y_{s(n)}\right)$ does not overestimate $\Pr\left[(y_1,...,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]$ for all $y_1,\dots,y_{s(n)}$. Furthermore, with high probability over $(y_1,...,y_{s(n)}) \leftarrow \mathcal{A}(1^n)$, $\mathcal{M}$ does not fail to estimate $uK^t\left(y_1,...,y_{s(n)}\right)$. Therefore, the security of Ver holds.

Hardness of verification of PPT-sampleable distributions with high entropy from OWFs. We can show that if OWFs exist, then every PPT-sampleable distribution $\mathcal{D}(1^n)$ with $H(\mathcal{D}(1^n)) \geq n^{1/c}$ for any constant c is not selectively-verifiable, where $H(\mathcal{D}(1^n))$ denotes the Shannon entropy of $\mathcal{D}(1^n)$. For this, it is sufficient to show that for every PPT-sampleable distribution $\mathcal{D}(1^n)$ with $H(\mathcal{D}(1^n)) \geq n^{1/c}$, there exists another PPT-sampleable distribution $\mathcal{D}^*(1^n)$ such that $\mathcal{D}^*(1^n)$ is statistically far but computationally indistinguishable from $\mathcal{D}(1^n)$.

Since $\mathcal{D}(1^n)$ is a PPT algorithm, it takes in $\ell(n)$ bits of randomness for some polynomial ℓ. Our algorithm $\mathcal{D}^*(1^n)$ samples $r \leftarrow \{0,1\}^{n^{1/c}/4}$, runs $G(r)$, and outputs $\mathcal{D}(1^n;G(r))$. Here, G is a PRG which takes $r \in \{0,1\}^{n^{1/c}/4}$ as input, and outputs $x \in \{0,1\}^{\ell(n)}$. From the construction, $\mathcal{D}^*(1^n)$ is computationally indistinguishable from $\mathcal{D}(1^n)$. From the construction, we have $H(\mathcal{D}^*(1^n)) \leq n^{1/c}/4$. From Fannes inequality, we have

$$\Delta(\mathcal{D}(1^n),\mathcal{D}^*(1^n)) \geq \frac{|H(\mathcal{D}(1^n)) - H(\mathcal{D}^*(1^n))|}{\mathrm{poly}(n)}.$$

This means that $\mathcal{D}^*(1^n)$ is statistically far from $\mathcal{D}(1^n)$.

One-way puzzles and verification of quantum advantage sampler. In Theorem 1.2, we obtained a $\mathbf{PP}$ upper bound for verifying quantum distributions. Can we improve this upper bound? In the proof of Theorem 1.2, we used a $\mathbf{PP}$ oracle to estimate $\Pr\left[(y_1,\dots,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]$ and $quK^t(y_1,\dots,y_{s(n)}|1^n)$. Recently, [CGGH25, HM25, KT25] showed that if no OWPuzz's exist, then, for every QPT algorithm $\mathcal{D}$, a QPT algorithm Approx can estimate $\Pr\left[(y_1,\dots,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]$ with high probability *on average over* $(y_1,\dots,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}$. Because of this, one might think that we can improve the $\mathbf{PP}$ oracle to the non-existence of OWPuzzs. However, we do not know how to use the non-existence of OWPuzzs, because there is no guarantee that Approx succeeds in estimating $\Pr\left[(y_1,\dots,y_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right]$ when $y_1,\dots,y_{s(n)}$ is not necessarily sampled from $\mathcal{D}(1^n)^{\otimes s(n)}$.

In this way, we do not know how to use the non-existence of OWPuzzs to efficiently verify QPT-samplable distributions. However, we can show at least that the non-existence of OWPuzzs can be used to efficiently verify strong quantum advantage samplers (QAS) with a *QPT* verification algorithm. Our construction of the verification algorithm Ver is as follows: On input 1^n and x, output $\top$ if

$$uK^t(x|1^n) - quK^t(x|1^n) > \omega(\log(n)), \tag{1}$$

and otherwise outputs $\bot$. If there is no OWPuzz, then a QPT algorithm can estimate $quK^t(x|1^n)$ on average *over any QPT algorithm* $\mathcal{Q}$ and can estimate $uK^t(x|1^n)$ with one-side error on average *over any PPT algorithm*. Therefore, Ver algorithm can check Equation (1).

Finally, let us show correctness and security of Ver. First we show correctness. If $\mathcal{D}_{\mathcal{Q}}(1^n)$ is a strong QAS, then, with overwhelming probability over $x \leftarrow \mathcal{D}_{\mathcal{Q}}(1^n)$, x satisfies Equation (1). [16] Intuitively, this is because, for any strong QAS $\mathcal{D}_{\mathcal{Q}}$ and any PPT algorithm $\mathcal{D}_{\mathcal{C}}$, with high probability over $x \leftarrow \mathcal{D}_{\mathcal{Q}}(1^n)$, $\Pr[x \leftarrow \mathcal{D}_{\mathcal{C}}(1^n)] \ll \Pr[x \leftarrow \mathcal{D}_{\mathcal{Q}}(1^n)]$, and the coding theorem and incompressibility imply that $quK^t(x) \approx -\log(\Pr[x \leftarrow \mathcal{D}_{\mathcal{Q}}(1^n)])$ and $uK^t(x) \approx -\log(\Pr[x \leftarrow \mathcal{D}_{\mathcal{C}}(1^n)])$.

Next let us show security. Any PPT algorithm $\mathcal{A}(1^n)$ cannot output x such that Equation (1) holds. The intuitive reason is as follows: For any PPT algorithm $\mathcal{D}$, incompressibility and coding theorem respectively implies that $quK^t(x) \gg -\log(\Pr[x \leftarrow \mathcal{D}(1^n)])$ and $uK^t(x) \ll -\log(\Pr[x \leftarrow \mathcal{D}(1^n)])$ with high probability over $x \leftarrow \mathcal{D}(1^n)$. Therefore, $\mathcal{A}$ cannot output x, which satisfies Equation (1).

1.3 Related Works

Verification of distributions. Verification of distributions (a.k.a. the identity testing problem) has a long history [BFR+00, BFF+01, DKN15, Pan08, VV17, Can20]. However, to the best of our knowledge, few works have studied the setting in which the unknown distribution is promised to be efficiently samplable. Therefore, here we will only mention the closely related result of Aaronson [Aar14].

Aaronson [Aar14] showed that $\mathbf{SampBQP} \neq \mathbf{SampBPP}$ if and only if $\mathbf{FBQP} \neq \mathbf{FBPP}$. In proving this result, Aaronson implicitly showed that every efficiently-samplable distribution is adaptively-verifiable with *only polynomially many samples*. However, the paper does not give a formal definition of verification, and the verification algorithm presented requires exponential time. [17] In fact, it is often claimed that verification of sampling-based quantum advantage requires exponentially many samples [HKEG19]; hence the quantum-advantage community may not be aware that polynomially many samples suffice for distribution verification under a reasonable definition. By contrast, we formally define distribution verification and improve the exponential-time verification algorithm.

Identity testing beyond independent and identically distributed samples. Before our work, only a few works studied the identity testing problem when the samples are not independent and identically distributed (i.i.d.). Here, we only mention the related result of Garg et al. [GPSV23]. Their work also considers identity testing beyond the i.i.d. case. Their definition is different from ours, and we now explain the differences.

First, in their definition, an unknown algorithm is not allowed to output correlated samples. In their definition, an unknown algorithm $\mathcal{A}$ is allowed to generate samples $x_1, \dots, x_s$ from not necessarily identical distributions $\mathcal{D}_1, \dots, \mathcal{D}_s$, but these are not allowed to be correlated. More specifically, the distribution $\mathcal{D}_{i+1}$ cannot depend on the output of $\mathcal{D}_1, \dots, \mathcal{D}_i$. On the other hand, our definition does not have such a restriction, and thus our setting is more general. Second, in their definition, the unknown algorithm $\mathcal{A}$ is not promised to be efficiently-samplable. Hence, their protocol requires exponentially many samples to verify the unknown distribution. On the other hand, in our definition, an unknown algorithm is promised to be efficiently-samplable, and thus we are able to show that polynomially-many samples suffice.

[16] Note that we do not know how to extend this into weak QAS. In other words, we do no know how to prove that the output x of any weak QAS $\mathcal{D}_{\mathcal{Q}}$ satisfies Equation (1) with non-negligible probability. Because of this, we cannot extend our results into the verification of weak QAS.

[17] [Aar14] already observed the issue, and pointed out that by considering space-bounded Kolmogorov complexity, at least $\mathbf{PSPACE}$ verification is possible.

Distribution Learning. It is sometimes claimed that verification of distributions is possible via distribution learning [BFKL94]. In distribution learning, one is given independent samples $x_1, \ldots, x_t$ drawn from $\mathcal{D}(z)$ (with unknown parameter z) and must output a hypothesis z^* such that the statistical distance $\Delta(\mathcal{D}(z), \mathcal{D}(z^*))$ is small. Here $\mathcal{D}$ denotes a PPT or QPT sampler and $\Delta(\mathcal{D}(z)), \mathcal{D}(z^*)$ denotes the statistical distance between $\mathcal{D}(z)$ and $\mathcal{D}(z^*)$. It is known that we can learn PPT- (resp. QPT-) samplable distributions on average assuming the non-existence of OWFs (resp. OWPuzzs) [HN23, HHM25].

Therefore, one might think that we can reduce verification to the non-existence of cryptography via learning: given samples $x_1, \ldots, x_t$ from $\mathcal{D}(z)$, run a learning algorithm to obtain z^* with $\Delta(\mathcal{D}(z), \mathcal{D}(z^*))$ small, and then check whether $\mathcal{D}(z^*)$ is close to some target distribution $\mathcal{D}(a)$. However, this approach faces two serious obstacles. First, making the strategy work requires an efficient procedure that decides whether $\mathcal{D}(a)$ and $\mathcal{D}(z^*)$ are close — i.e., an efficient closeness test — and such a decision procedure is not guaranteed to exist assuming the non-existence of cryptographic primitives. Second, and more fundamentally, the samples used in our definition of distribution verification are not necessarily drawn i.i.d. from $\mathcal{D}(z)$. Since learning guarantees typically require i.i.d. samples, the non-i.i.d. nature of verification samples prevents a straightforward derivation of verification results from learning results.

Consequently, although distribution learning and verification are related, existing learning results do not immediately yield verification procedures in our model. Hence, we rely on a different approach to obtain our result.

Meta-Complexity Characterization of OWFs. A recent line of research [LP20, LP21, IRS22, HN23] shows that the existence of OWFs is equivalent to several average-case hardness notions of Kolmogorov complexity, i.e., the average-case hardness of computing the minimum description length of a given string. In this paper, we have shown the equivalence between the feasibility of verifying PPT-samplable distributions and the existence of OWFs. Therefore, by combining it with [LP20, LP21, IRS22, HN23], we also obtain the equivalence between the feasibility of verifying PPT-samplable distributions and these classical average-case hardness notions of Kolmogorov complexity.

Cryptographic Perspective on Quantum Advantage Verification. In a recent work [CFSY23], Chia et al. study a model for verifying sampling-based quantum advantage. In their protocol the verifier first publishes a list of circuits; the honest quantum sampler then selects one circuit, publishes that choice together with a set of i.i.d. samples from the circuit, and a classical adversary subsequently publishes a classical algorithm and a set of i.i.d. samples from that algorithm. A distribution is said to have quantum-verifiable advantage if there exists a verifier that can distinguish the samples produced by the honest quantum sampler from those produced by any efficient classical adversary. Our framework differs in two respects: Chia et al. focus on non-uniform algorithms while we consider uniform algorithms, and — more importantly — they consider that the verifier knows the code of the classical adversary whereas we consider that the verifier has only black-box access to the adversary. Because their verifier sees the adversary's code, their notion is much closer to QEFID, where the distinguisher's task is to tell apart two known distributions.

1.4 Open Problems

Our research raises several important open questions that remain to be explored. Below, we highlight a particularly interesting one.

In Theorem 1.8, we have shown that, assuming the existence of QEFID, there exists a QPT-samplable distribution that is hard to verify. This result does not exclude the possibility that certain QPT-samplable

distributions with high entropy might still be efficiently verifiable. Therefore, a natural question arises: is every QPT-samplable distribution with high entropy hard to verify?

More concretely, we ask:

For any QPT-samplable distribution $\mathcal{Q}$ with sufficiently large entropy, does there exist a PPT-samplable distribution $\mathcal{C}$ such that no PPT algorithm can distinguish whether samples come from $\mathcal{Q}$ or $\mathcal{C}$?

An affirmative answer would suggest that quantum sampling advantage cannot be detected by any PPT algorithm. Conversely, a negative answer would indicate that some form of quantum sampling advantage can, in fact, be detected by classical verification.

In the classical case, we already establish an affirmative answer in Theorem 1.7. Specifically, we show that for any PPT-samplable distribution $\mathcal{Q}$ with sufficiently large entropy, there exists another PPT-samplable distribution $\mathcal{C}$ that is statistically far from $\mathcal{Q}$ yet indistinguishable by any PPT algorithm. However, our proof technique there crucially relies on the fact that the output of a PPT algorithm can be described by its internal randomness, making it unclear how to extend the argument to the quantum setting.

2 Preliminaries

2.1 Notations

Here we introduce basic notations and mathematical tools used in this paper. We denote by $x \leftarrow X$ an element from a finite set X chosen uniformly at random, and $y \leftarrow \mathsf{A}(x)$ denotes assigning to y the output of a probabilistic or deterministic algorithm A on an input x. When we want to explicitly denote that A uses randomness r, we write $y \leftarrow \mathsf{A}(x;r)$. When D is a distribution, $x \leftarrow D$ denotes sampling an element from D. The notation $\{y_i\}_{i\in[N]} \leftarrow \mathcal{A}(x)^{\otimes N}$ means that $\mathcal{A}$ is run on input x independently N times, and y_i is the ith result. Let $[\ell]$ denote the set of integers $\{1,\cdots,\ell\}$, and $y := z$ denote that y is set, defined, or substituted by z. For a string $s \in \{0,1\}^\ell$, $s[i]$ and s_i denotes i-th bit of s. QPT stands for quantum polynomial time. PPT stands for (classical) probabilistic polynomial time. A function $f : \mathbb{N} \to \mathbb{R}$ is a negligible function if for any constant c, there exists $n_0 \in \mathbb{N}$ such that for any $n > n_0$, $f(n) < n^{-c}$. We write $f(n) \leq \mathsf{negl}(n)$ to denote $f(n)$ being a negligible function. The statistical distance between two distributions $\mathcal{D}$ and $\mathcal{C}$ is given by $\Delta(\mathcal{D},\mathcal{C}) := \frac{1}{2}\sum_x |\Pr[x \leftarrow \mathcal{D}] - \Pr[x \leftarrow \mathcal{C}]|$. The Shannon entropy of a distribution $\mathcal{D}$ is given by $H(\mathcal{D}) := \sum_x \Pr[x \leftarrow \mathcal{D}] \log\left(\frac{1}{\Pr[x\leftarrow\mathcal{D}]}\right)$.

2.2 Cryptography

Definition 2.1 (Infinitely-Often One-Way Functions (OWFs)). *A function $f : \{0,1\}^* \to \{0,1\}^*$ that is computable in classical deterministic polynomial-time is an infinitely-often one-way function (OWF) if, for any PPT adversary $\mathcal{A}$ and for any polynomial p, we have*

$$\Pr[f(x') = f(x) : x \leftarrow \{0,1\}^n, x' \leftarrow \mathcal{A}(1^n, f(x))] \leq \frac{1}{p(n)}$$

for infinitely many $n \in \mathbb{N}$.

Definition 2.2 (Infinitely-Often QEFID). *An infinitely-often QEFID is a QPT algorithm $\mathsf{Gen}(1^n, b)$ that takes a security parameter 1^n and $b \in \{0,1\}$ as input, and outputs a classical bit string $x \in \{0,1\}^n$. We require the following two conditions.*

Statistically far:

$$\Delta(\mathsf{Gen}(1^n,0),\mathsf{Gen}(1^n,1)) \geq 1-\mathsf{negl}(n).$$

Computationally indistinguishable: *For any QPT algorithm $\mathcal{A}$ and any polynomials p,t,*

$$\Big|\Pr\Big[1\leftarrow\mathcal{A}(1^n,\{x_i\}_{i\in[t(n)]}):\{x_i\}_{i\in[t(n)]}\leftarrow\mathsf{Gen}(1^n,0)^{\otimes t(n)}\Big] - \Pr\Big[1\leftarrow\mathcal{A}(1^n,\{x_i\}_{i\in[t(n)]}):\{x_i\}_{i\in[t(n)]}\leftarrow\mathsf{Gen}(1^n,1)^{\otimes t(n)}\Big]\Big| \leq \frac{1}{p(n)},$$

for infinitely many $n\in\mathbb{N}$.

2.3 Classical Meta-Complexity

Throughout this work, we consider a fixed universal Turing machine $\mathcal{U}$. Let $\mathcal{U}^t$ refer to the execution of $\mathcal{U}$ for t steps. When r is a bit string sampled uniformly at random, we write $\mathcal{U}^t(y;r)$ to mean that we run a universal Turing machine $\mathcal{U}$ on input y and r, halting after t steps. We will call the output distribution of $\mathcal{U}^t(y;r)$ the *universal (classical) time bounded distribution*.

Definition 2.3 (Universal time bounded complexity [LV93, HN23]). *For any bit strings $x,y\in\{0,1\}^*$ and for any $t\in\mathbb{N}$, we define*

$$uK^t(x|y) := -\log_2 \Pr_{r\leftarrow\{0,1\}^t}\left[x\leftarrow\mathcal{U}^t(y;r)\right].$$

Remark 2.4. [HN23] uses $q^t(x)$ to mean $uK^t(x)$. To avoid the possibility for confusion that the q in q^t [HN23] might stand for "quantum" we relabel the notion as uK^t with the u standing for "universal".

The following Theorem 2.5 is a small modification of a theorem shown in [HN23], where we use $uK^t(x|1^n)$ instead of $uK^t(x)$. For completeness, we provide a proof.

Theorem 2.5 (Coding Theorem for $uK^t(x|1^n)$ [HN23]). *There exist a universal constant c and a polynomial t_0 such that, for any t and for every $t(n)$-time probabilistic algorithm $\mathcal{D}$, which takes 1^n as input and outputs $x\in\{0,1\}^{m(n)}$ for any polynomial m, we have*

$$uK^T(x|1^n) \leq \log_2\frac{1}{\Pr[x\leftarrow\mathcal{D}(1^n)]} + 2|\mathcal{D}| + c$$

for every $T>t_0(t(n))$. Here, $|\mathcal{D}|$ is the description length of the algorithm $\mathcal{D}$.

Proof of Theorem 2.5. Let $\mathcal{M}\in\{0,1\}^{|\mathcal{D}|}$ be an encoding of $\mathcal{D}$. Let $K\in\{0,1\}^{|K|}$ be an encoding of algorithm that represents that "the first $|\mathcal{D}|$ bits are regarded as the description of the program, and the remaining bits are regarded as the program's $t(n)$ bits of randomness followed by 1^n". There exists a constant c such that $|K|\leq|\mathcal{D}|+c$. Conditioned on that the first $|\mathcal{D}|+|K|$ bits of r is equivalent to $K\|\mathcal{M}$, $U^T(r,1^n)$ behaves in the same way as $\mathcal{D}(1^n)$. This event occurs with probability at least $2^{-|K|+|\mathcal{D}|}$. Hence, we have

$$\begin{aligned}\Pr_{r\leftarrow\{0,1\}^T}[x\leftarrow\mathcal{U}^T(r,1^n)] &\geq 2^{-|K|+|\mathcal{D}|}\Pr[x\leftarrow\mathcal{D}(1^n)]\\ &\geq 2^{-2|\mathcal{D}|+c}\Pr[x\leftarrow\mathcal{D}(1^n)].\end{aligned}$$

This implies that

$$uK^T(x|1^n) \leq \log_2 \frac{1}{\Pr[x \leftarrow \mathcal{D}(1^n)]} + 2|\mathcal{D}| + c. \qquad \square$$

We also use the following Theorem 2.6. It was shown in [LV93] for the prefix Kolmogorov complexity, but here we consider $uK^t(x|1^n)$ for our purpose, which needs some modifications of a proof. For clarity, we provide a proof.

Theorem 2.6 (Incompressibility for $uK^t(x)$). *There exists a constant c such that, for any $n \in \mathbb{N}$, any algorithm distribution $\mathcal{D}$, which takes 1^n as input and outputs $x \in \{0,1\}^{m(n)}$ for any polynomial m, any $t > n$, and for any $\alpha > 0$ we have*

$$\Pr\left[uK^t(x|1^n) \leq -\log_2 \Pr[x \leftarrow \mathcal{D}(1^n)] - \alpha : x \leftarrow \mathcal{D}(1^n)\right] \leq (m(n)+c) \cdot 2^{-\alpha+1}.$$

Proof of Theorem 2.6. There exists a constant c such that for any $t > n$, and $x \in \{0,1\}^{m(n)}$

$$uK^t(x|1^n) \leq m(n) + c$$

for all $n \in \mathbb{N}$. This directly follows from Theorem 2.5 by considering a probabilistic algorithm $\mathcal{D}$, which takes 1^n and uniformly randomly output $x \in \{0,1\}^{m(n)}$.

For any $n \in \mathbb{N}$, any $x \in \{0,1\}^{m(n)}$, and any $s \in [m(n)+c]$, we define

$$\mathcal{S}_{n,t}(s) := \{x \in \{0,1\}^{m(n)} : s-1 < uK^t(x|1^n) \leq s\}$$

and

$$\mathcal{H}_{n,t} := \{x \in \{0,1\}^{m(n)} : uK^t(x|1^n) \leq -\log\left(\Pr[x \leftarrow \mathcal{D}(1^n)]\right) - \alpha\}.$$

Note that, from the definition of $\mathcal{S}_{n,t}(s)$ and $\mathcal{H}_{n,t}$, for all $x \in \mathcal{S}_{n,t}(s) \cap \mathcal{H}_{n,t}$, we have

$$\Pr[x \leftarrow \mathcal{D}(1^n)] \leq 2^{-s-\alpha+1}.$$

Furthermore, because $\Pr_{r \leftarrow \{0,1\}^t}[x \leftarrow U^t(r|1^n)] \geq 2^{-s}$ for all $x \in \mathcal{S}_{n,t}(s)$ and $\sum_{x \in \{0,1\}^n} \Pr_{r \leftarrow \{0,1\}^t}[x \leftarrow U^t(r)] \leq 1$, we have

$$|\mathcal{S}_{n,t}(s)| \leq 2^s.$$

Therefore, we have

$$\begin{aligned}
&\Pr_{x \leftarrow \mathcal{D}(1^n)}[uK^t(x|1^n) \leq -\log(\Pr[x \leftarrow \mathcal{D}(1^n)]) - \alpha] \\
&= \sum_{i \in [m(n)+c]} \sum_{x \in \mathcal{S}_{n,t}(i) \cap \mathcal{H}_{n,t}} \Pr[x \leftarrow \mathcal{D}(1^n)] \\
&\leq \sum_{i \in [m(n)+c]} \sum_{x \in \mathcal{S}_{n,t}(i) \cap \mathcal{H}_{n,t}} 2^{-i-\alpha+1} \\
&\leq \sum_{i \in [m(n)+c]} |\mathcal{S}_{n,t}(i)| 2^{-i-\alpha+1} \leq \sum_{i \in [m(n)+c]} 2^{-\alpha+1} \leq (m(n)+c) \cdot 2^{-\alpha+1}.
\end{aligned}$$

$\square$

We use the following Theorem 2.7 for showing the security of distribution verification. Theorem 2.7 is based on Lemma 19 of [Aar14], but we cannot directly use it and we need some non-trivial modifications of the proof, because [Aar14] considers a prefix-free unbounded Kolmogorov complexity while we consider $uK^t(x)$. We give a proof of Theorem 2.7 in Appendix A.

Theorem 2.7 ([Aar14] Adapted to uK^t). *Let ϵ be any function and let s, m, t be any polynomials, and let t_0 be a polynomial given in Theorem 2.5. Let $\mathcal{D}$ be any algorithm that takes 1^n as input and outputs $x \in \{0,1\}^{m(n)}$. Let $\mathcal{G}$ be a classical algorithm that takes 1^n and outputs $(x_1, \dots, x_{s(n)}) \in (\{0,1\}^{m(n)})^{\times s(n)}$ running in time $t(n)$. Define $\mathsf{Marginal}_{\mathcal{G}}(1^n)$ to be the distribution defined as follows:*

1. *Sample $(y_1, \dots, y_{s(n)}) \leftarrow \mathcal{G}(1^n)$.*
2. *Sample $i \leftarrow [s(n)]$ uniformly at random.*
3. *Output y_i*

Define $p_y := \Pr[y \leftarrow \mathcal{D}(1^n)]$. Define $A^{\mathcal{D}}_{n,s,\alpha}$ to be the set

$$A^{\mathcal{D}}_{n,s,\alpha} := \left\{ (y_1, \dots, y_{s(n)}) : \log_2 \frac{1}{p_{y_1} \dots p_{y_{s(n)}}} \leq uK^{t_0(t(n))}(y_1, \dots, y_{s(n)} | 1^n) + \alpha(n) \right\}.$$

There exists a constant C such that for all sufficiently large $n \in \mathbb{N}$, if

$$\Pr_{y_1,\dots,y_{s(n)} \leftarrow \mathcal{G}(1^n)} \left[(y_1, \dots, y_{s(n)}) \in A^{\mathcal{D}}_{n,s,\alpha} \right] \geq 1 - \epsilon(n),$$

then we have

$$\Delta(\mathcal{D}(1^n), \mathsf{Marginal}_{\mathcal{G}}(1^n)) \leq \epsilon(n) + \sqrt{\frac{\log_2 \frac{1}{1-\epsilon(n)} + \alpha(n) + C}{s(n)}}.$$

3 Quantum Meta-Complexity

In this section, we introduce $quK^t(x)$, which is a quantum analog of $uK^t(x)$, and show several properties of $quK^t(x)$.

3.1 Definition of $quK^t(x)$

Let QU^t be a quantum algorithm that takes 1^n and $c \in \{0,1\}^*$ as input. The algorithm considers c as an encoding of an output length m, an input size s, a $t(n)$-depth quantum circuit C. Then, it runs $C|0^s\rangle$, measures the first m-bits, and outputs the resulting m bits. If c is not a valid encoding of a quantum circuit, then we simply output $\perp$.

Let $\mathcal{A}$ be a quantum algorithm that takes 1^n as input and outputs $x \in \{0,1\}^{m(n)}$, where m is a polynomial. We say that $\mathcal{A}$ is a t-time quantum algorithm if it can be run by first running a classical Turing machine $c \leftarrow \mathcal{M}^t(1^n)$ running in t-steps, then running $x \leftarrow \mathsf{QU}^t(1^n, c)$, and outputting x.

We define $quK^t(x|1^n)$ as follows.

Definition 3.1 ($quK^t(x|1^n)$). *For any $n, t \in \mathbb{N}$, let us define a quantum algorithm $\mathcal{Q}^t(1^n)$ as follows:*

$\mathcal{Q}^t(1^n)$**:**

1. *Sample* $\Pi \leftarrow \{0,1\}^t$.
2. *Run* $c \leftarrow \mathcal{U}^t(\Pi, 1^n)$.
3. *Run* $x \leftarrow \mathsf{QU}^t(1^n, c)$.
4. *Output* x.

For any $t \in \mathbb{N}$, $x \in \{0,1\}^*$ *and any* $n \in \mathbb{N}$, *let us define* $quK^t(x|1^n)$ *as follows:*

$$quK^t(x|1^n) := -\log_2 \Pr\Big[x \leftarrow \mathcal{Q}^t(1^n)\Big].$$

Remark 3.2. This notion is both a quantum generalization of uK^t [LV93, HN23] and a restriction of the notion of quantum Kolmogorov complexity introduced by Gács [Gác01] to time bounded algorithms outputting classical states. While Gács' notion is defined slightly differently, the notions are equivalent and the invariance of Gács' notion applies to ours.

3.2 Properties of $quK^t(x)$

To prove our main results, we need quantum analogues of many of the theorems and lemmas in Section 2.3. The next three theorems are specifically quantum analogues of Theorem 2.5, Theorem 2.6, and Theorem 2.7. In all three cases their proofs go almost identically to their classical analogues so we choose to omit them.

Theorem 3.3 (Coding Theorem for $quK^t(x)$**).** *There exist a universal constant* c *and polynomial* t_0 *such that, for every* $t(n)$*-time quantum algorithm* $\mathcal{D}$*, which takes* 1^n *as input and outputs* $x \in \{0,1\}^{m(n)}$ *for any polynomial* m*, we have*

$$quK^T(x|1^n) \leq \log_2 \frac{1}{\Pr[x \leftarrow \mathcal{D}(1^n)]} + 2|\mathcal{D}| + c$$

for every $T > t_0(t(n))$*. Here,* $|\mathcal{D}|$ *is the description length of the algorithm* $\mathcal{D}$*.*

Theorem 3.4 (Incompressibility for $quK^t(x)$**).** *There exists a constant* c *such that, for any* $n \in \mathbb{N}$*, any algorithm* $\mathcal{D}$*, which takes* 1^n *as input and outputs* $x \in \{0,1\}^{m(n)}$ *for any polynomial* m*, any* $t > n$*, and for any* $\alpha > 0$*, we have*

$$\Pr\Big[quK^t(x|1^n) \leq -\log_2 \Pr[x \leftarrow \mathcal{D}(1^n)] - \alpha : x \leftarrow \mathcal{D}(1^n)\Big] \leq (m(n) + c) \cdot 2^{-\alpha+1}.$$

Theorem 3.5. *Let* ϵ *be any function and let* s*,* m *and* t *be any polynomials. Let* $\mathcal{D}$ *be any algorithm that takes* 1^n *as input, and outputs* $x \in \{0,1\}^{m(n)}$*. Let* $\mathcal{G}$ *be a* $t(n)$*-time quantum algorithm that takes* 1^n *and outputs* $x \in \{0,1\}^{s(n)\cdot m(n)}$*. Define* $\mathsf{Marginal}_{\mathcal{G}}(1^n)$ *to be the distribution defined as follows:*

1. *Sample* $(y_1, \ldots, y_s) \leftarrow \mathcal{G}(1^n)$.
2. *Sample* $i \leftarrow [s]$ *uniformly at random.*
3. *Output* y_i.

Define $p_y \coloneqq \Pr[y \leftarrow \mathcal{D}(1^n)]$. *Define* $A^{\mathcal{D}}_{n,s,\alpha}$ *to be the set*

$$A^{\mathcal{D}}_{n,s,\alpha} \coloneqq \left\{(y_1,\dots,y_{s(n)}) : \log_2 \frac{1}{p_{y_1}\cdots p_{y_{s(n)}}} \leq quK^{t(n)}(y_1,\dots,y_{s(n)}|1^n) + \alpha(n)\right\}$$

There exists a constant C *such that for all sufficiently large* $n \in \mathbb{N}$*, if*

$$\Pr_{y_1,\dots,y_{s(n)}\leftarrow\mathcal{G}(1^n)}\left[(y_1,\dots,y_{s(n)}) \in A^{\mathcal{D}}_{n,s,\alpha}\right] \geq 1-\epsilon(n),$$

then we have

$$\Delta(\mathcal{D}(1^n), \mathsf{Marginal}_{\mathcal{G}}(1^n)) \leq \epsilon(n) + \sqrt{\frac{\log_2 \frac{1}{1-\epsilon(n)} + \alpha(n) + C}{s(n)}}.$$

4 Definition of Verification of Distributions

In this section, we introduce two definitions of verification of distributions both for classical and quantum cases.

Definition 4.1 (Selective-Verification of Distributions). *Let* $\mathcal{D}$ *be an algorithm that takes* 1^n *as input, and outputs* $x \in \{0,1\}^{m(n)}$ *where* m *is an arbitrary polynomial. Let us denote* X *to mean that* $X \in \{$*PPT, QPT, deterministic polynomial-time algorithm querying to* PP *oracle*$\}$.

We say that the algorithm $\mathcal{D}$ *is selectively-verifiable with an* X*-algorithm if for any polynomial* t*, any function* $\epsilon : \mathbb{N} \to (0,1)$*, and any constant* $c > 0$*, there exist a polynomial* s *and an* X*-algorithm* Ver *such that the following holds.*

Correctness: *We have*

$$\Pr[\top \leftarrow \mathsf{Ver}(1^n, x_1,\dots,x_s) : (x_1,\dots,x_s) \leftarrow \mathcal{D}(1^n)^{\otimes s}] \geq 1 - n^{-c},$$

for all sufficiently large $n \in \mathbb{N}$*. Here,* $s = s(n^c, t(n), 1/\epsilon(n))$.

Selective-Security: *For any* $t(n)$*-time uniform quantum adversary* $\mathcal{A}$*, which takes* 1^n *as input and outputs* $x \in \{0,1\}^{m(n)}$*, such that*

$$\Delta(\mathcal{A}(1^n), \mathcal{D}(1^n)) \geq \epsilon(n),$$

we have

$$\Pr[\top \leftarrow \mathsf{Ver}(1^n, x_1,\dots,x_s) : (x_1,\dots,x_s) \leftarrow \mathcal{A}(1^n)^{\otimes s}] \leq n^{-c},$$

for all sufficiently large $n \in \mathbb{N}$*. Here,* $s = s(n^c, t(n), 1/\epsilon(n))$.

Definition 4.2 (Adaptive-Verification of Distributions). *Let* $\mathcal{D}$ *be an algorithm that takes* 1^n *as input, and outputs* $x \in \{0,1\}^{m(n)}$*, where* m *is an arbitrary polynomial. Let us denote* X *to mean that* $X \in \{$*PPT, QPT, deterministic polynomial-time algorithm querying to* PP *oracle*$\}$.

We say that the algorithm $\mathcal{D}$ *is adaptively-verifiable with an* X*-algorithm if, for any polynomial* t*, any function* $\epsilon : \mathbb{N} \to (0,1)$*, and any constant* $c > 0$*, there exist a polynomial* s *and an* X*-algorithm* Ver *such that the following hold.*

Correctness: *We have*

$$\Pr[\top \leftarrow \mathsf{Ver}(1^n, x_1, \dots, x_s) : x_1, \dots, x_s \leftarrow \mathcal{D}(1^n)^{\otimes s}] \geq 1 - n^{-c}$$

for all sufficiently large $n \in \mathbb{N}$. Here, $s = s(n^c, t(n), 1/\epsilon(n))$.

Adaptive-Security: *For any algorithm $\mathcal{A}$ that takes 1^n as input and outputs $(x_1, \dots, x_{s(n)}) \in \left(\{0,1\}^{m(n)}\right)^{\times s(n)}$, we define an algorithm $\mathsf{Marginal}_{\mathcal{A}}(1^n)$ as follows:*

$\mathsf{Marginal}_{\mathcal{A}}(1^n)$**:**

1. *Sample $(x_1, \dots, x_s) \leftarrow \mathcal{A}(1^n)$.*
2. *Sample $i \leftarrow [s]$.*
3. *Output x_i.*

For any $t(n)$-time uniform quantum adversary $\mathcal{A}$ with $\Delta(\mathsf{Marginal}_{\mathcal{A}}(1^n), \mathcal{D}(1^n)) \geq \epsilon(n)$ for all $n \in \mathbb{N}$, we have

$$\Pr[\top \leftarrow \mathsf{Ver}(1^n, x_1, \dots, x_s) : (x_1, ..., x_s) \leftarrow \mathcal{A}(1^n)] \leq (1 - \epsilon(n)) + n^{-c}$$

for all sufficiently large $n \in \mathbb{N}$. Here, $s = s(n^c, t(n), 1/\epsilon(n))$.

Remark 4.3. Note that in the above two definitions, Definitions 4.1 and 4.2, the adversary $\mathcal{A}$ is quantum even if X or $\mathcal{D}$ is classical. When we want to consider uniform classical probabilistic $\mathcal{A}$, we explicitly say that $\mathcal{D}$ is selectively/adaptively-verifiable with an X-algorithm with *classical-security*.

We can show that adaptive-verifiability implies selective-verifiability as follows.

Lemma 4.4. *If an algorithm $\mathcal{D}$ is adaptively-verifiable, then it is selectively-verifiable.*

Proof. In the following, we show that $\mathcal{D}$ is selectively-verifiable with a QPT algorithm if $\mathcal{D}$ is adaptively-verifiable with a QPT algorithm. In a similar way, we can show that $\mathcal{D}$ is selectively-verifiable with an X algorithm if $\mathcal{D}$ is selectively-verifiable with an X algorithm for

$$X \in \{\text{PPT},\ \text{QPT},\ \text{deterministic polynomial-time algorithm querying to PP oracle}\}.$$

Let s be a polynomial and Ver be a QPT algorithm such that

$$\Pr\left[\top \leftarrow \mathsf{Ver}(1^n, x_1, \dots, x_{s(n)}) : x_1, \dots, x_s \leftarrow \mathcal{D}(1^n)^{\otimes s}\right] \geq 1 - \frac{\epsilon(n)}{n}$$

and

$$\Pr\left[\top \leftarrow \mathsf{Ver}(1^n, x_1, \dots, x_{s(n)}) : x_1, \dots, x_s \leftarrow \mathcal{B}(1^n)\right] \leq 1 - \left(1 - \frac{1}{n}\right)\epsilon(n)$$

for any QPT algorithm $\mathcal{B}$ such that $\Delta(\mathsf{Marginal}_{\mathcal{B}}(1^n), \mathcal{D}(1^n)) \geq \epsilon(n)$ for all sufficiently large $n \in \mathbb{N}$.

Let $s^*(n) := s(n) \cdot \frac{4n^2}{\epsilon(n)^2}$. We consider the following QPT algorithm Ver^*.

Ver^***:**

1. Take $x_1, \dots, x_{s^*(n)}$ as input.
2. For all $i \in [\frac{4n^2}{\epsilon(n)^2}]$, run $b_i \leftarrow \mathsf{Ver}(x_{is+1}, \dots, x_{(i+1)s})$. Let $\mathsf{Count}_{x_1,\dots,x_{s^*(n)}}$ be the number of times such that $b_i = \top$.
3. Output $\top$ if $\frac{\epsilon(n)^2}{4n^2}\mathsf{Count}_{x_1,\dots,x_{s^*(n)}} \geq \left(1 - \frac{\epsilon(n)}{2}\right)$. Otherwise, output $\bot$.

Correctness: From Hoeffding's inequality,

$$\Pr_{x_1,\ldots,x_{s^*(n)}\leftarrow\mathcal{D}(1^n)^{\otimes s^*(n)}}\left[\frac{\epsilon(n)^2}{4n^2}\mathsf{Count}_{x_1,\ldots,x_{s^*(n)}}-\left(1-\frac{\epsilon(n)}{n}\right)\leq-\frac{\epsilon(n)}{4}\right]\leq\exp\left(-\frac{n^2}{2}\right).$$

This implies that

$$\Pr_{x_1,\ldots,x_{s^*(n)}\leftarrow\mathcal{D}(1^n)^{\otimes s^*(n)}}\left[\frac{\epsilon(n)^2}{4n^2}\mathsf{Count}_{x_1,\ldots,x_{s^*(n)}}\leq\left(1-\frac{\epsilon(n)}{2}\right)\right]\leq\exp\left(-\frac{n^2}{2}\right).$$

Selective Security: Let $\mathcal{A}$ be an algorithm which takes 1^n as input and outputs $x\in\{0,1\}^{m(n)}$ such that

$$\Delta(\mathcal{A}(1^n),\mathcal{D}(1^n))\geq\epsilon(n).$$

The definition of Ver guarantees that

$$\Pr\left[\top\leftarrow\mathsf{Ver}(1^n,x_1,\ldots,x_{s(n)}):x_1,\ldots,x_s\leftarrow\mathcal{A}(1^n)^{\otimes s(n)}\right]\leq1-\left(1-\frac{1}{n}\right)\epsilon(n).$$

From Hoeffding's inequality, for any QPT algorithm such that $\Delta(\mathcal{A}(1^n),\mathcal{D}(1^n))\geq\epsilon(n)$, we have

$$\Pr_{x_1,\ldots,x_{s^*(n)}\leftarrow\mathcal{A}(1^n)^{\otimes s^*(n)}}\left[\frac{\epsilon(n)^2}{4n^2}\mathsf{Count}_{x_1,\ldots,x_{s^*(n)}}-\left(1-\left(1-\frac{1}{n}\right)\epsilon(n)\right)\geq\frac{\epsilon(n)}{4}\right]\leq\exp\left(-\frac{n^2}{2}\right).$$

This implies that

$$\Pr_{x_1,\ldots,x_{s^*(n)}\leftarrow\mathcal{A}(1^n)^{\otimes s^*(n)}}\left[\frac{\epsilon(n)^2}{4n^2}\mathsf{Count}_{x_1,\ldots,x_{s^*(n)}}\geq\left(1-\frac{\epsilon(n)}{2}\right)\right]\leq\exp\left(-\frac{n^2}{2}\right)$$

for all sufficiently large $n\in\mathbb{N}$. □

5 Classical Distribution Verification

In this section, we show an equivalence between the existence of OWFs and hardness of classical distribution verification.

5.1 OWFs from Hardness of Classical Distribution Verification

Theorem 5.1. *Assume that infinitely-often OWFs do not exist. Then every uniform PPT algorithm $\mathcal{D}$, which takes 1^n as input and outputs $x\in\{0,1\}^{m(n)}$, where m is a polynomial, is adaptively-verifiable with a PPT algorithm with classical-security.*

For showing Theorem 5.1, we will use the following Theorems 5.2 to 5.4.

Theorem 5.2 (No i.o.-OWFs implies one-sided error probability estimation [IL90, IRS22, CHK25]). *If there do not exist infinitely-often OWFs, then for any polynomial m, for any PPT algorithm $\mathcal{D}$, which takes 1^n*

as input and outputs $x \in \{0,1\}^{m(n)}$, and for any constant $c \in \mathbb{N}$, there exists a PPT algorithm Approx *such that for all sufficiently large n,*

$$\Pr_{\substack{x \leftarrow \mathcal{D}(1^n) \\ \mathsf{Approx}}}\left[\frac{1}{2}\Pr[x \leftarrow \mathcal{D}(1^n)] \leq \mathsf{Approx}(x, 1^n) \leq \Pr[x \leftarrow \mathcal{D}(1^n)]\right] \geq 1 - n^{-c}.$$

Furthermore, for all sufficiently large n, for all $x \in \{0,1\}^{m(n)}$,

$$\Pr_{\mathsf{Approx}}[\mathsf{Approx}(x, 1^n) \leq \Pr[x \leftarrow \mathcal{D}(1^n)]] \geq 1 - n^{-c}.$$

Theorem 5.3 (No i.o.-OWFs implies uK^t estimation on all PPT distributions (Theorem 7.1 of [HN23])). *If there do not exist infinitely-often OWFs, then there exists a PPT algorithm $\mathcal{M}$ such that for any polynomial m and for any PPT algorithm $\mathcal{D}$, which takes 1^n as input and outputs $x \in \{0,1\}^{m(n)}$, there exists a polynomial t_0, such that for every $t > t_0(n)$ and every $\delta^{-1} \in \mathbb{N}$, for all sufficiently large n,*

$$\Pr_{\mathcal{M}, x \leftarrow \mathcal{D}(1^n)}[uK^t(x) - 1 \leq \mathcal{M}(x, 1^t, 1^{\delta^{-1}}) \leq uK^t(x) + 1] \geq 1 - \delta.$$

Theorem 5.4 (Length conditional version of uK^t estimation). *If there do not exist infinitely-often OWFs, then there exists a PPT algorithm $\mathcal{M}$ such that for any PPT algorithm $\mathcal{D}$, which takes 1^n as input and outputs $x \in \{0,1\}^{m(n)}$, there exists a polynomial t_0, such that for every $t > t_0(n)$ and every $\delta^{-1} \in \mathbb{N}$, for all sufficiently large n,*

$$\Pr_{\mathcal{M}, x \leftarrow \mathcal{D}(1^n)}[uK^t(x|1^n) - 1 \leq \mathcal{M}(x, 1^{n,t,\delta^{-1}}) \leq uK^t(x|1^n) + 1] \geq 1 - \delta.$$

The proof of Theorem 5.4. is the same as that of Theorem 5.3, and hence we omit it.

Proof of Theorem 5.1. In the following, assume that infinitely-often OWFs do no exist. Then, we give a polynomial s and a PPT algorithm Ver which satisfies the correctness and adaptive security in Definition 4.2. For describing Ver, let us introduce several notations below.

Notations: We set $s(n, t(n), \epsilon(n)) = n^{4c}\left(\log_2 \frac{1}{1-\epsilon(n)} + 2(\log(n))^2\right)$. For simplicity, we often denote $s = n^{4c}\left(\log_2 \frac{1}{1-\epsilon(n)} + 2(\log(n))^2\right)$ below. Let t_0 be a polynomial given in Theorem 2.5.

We also set $\alpha = (\log(n))^2$.

Approx is a PPT algorithm given in Theorem 5.2 such that

$$\Pr_{x \leftarrow \mathcal{D}(1^n)^{\otimes s}}\left[\frac{1}{2}\Pr[x \leftarrow \mathcal{D}(1^n)^{\otimes s}] \leq \mathsf{Approx}(x, 1^n) \leq \Pr[x \leftarrow \mathcal{D}(1^n)^{\otimes s}]\right] \geq 1 - n^{-4c}$$

and

$$\Pr[\mathsf{Approx}(x, 1^n) \leq \Pr[x \leftarrow \mathcal{D}(1^n)^{\otimes s}]] \geq 1 - n^{-4c}$$

for all $x \in \{0,1\}^{s \cdot m(n)}$ and all sufficiently large $n \in \mathbb{N}$.

$\mathcal{M}$ is a PPT algorithm given in Theorem 5.4 such that, for any PPT algorithm $\mathcal{A}$, we have

$$\Pr_{x \leftarrow \mathcal{A}(1^n)}[uK^{t_0(t(n))}(x|1^n) - 1 \leq \mathcal{M}(x, 1^t) \leq uK^{t_0(t(n))}(x|1^n) + 1] \geq 1 - n^{-4c}$$

for all sufficiently large $n \in \mathbb{N}$.

Construction: We give a construction of Ver.

Ver**:**

1. Take 1^n and $y_1, \dots, y_{s(n)}$ as input.
2. Compute $\mathcal{M}(y_1, \dots, y_{s(n)}, 1^t) \to k$
3. Compute $\mathsf{Approx}(y_1, \dots, y_{s(n)}) \to p$
4. Output $\top$ if $-\log_2 p \leq k + \alpha$. Otherwise, output $\bot$.

In the following, we show that Ver satisfies correctness and security.

Correctness: In the following, we show that

$$\Pr[\top \leftarrow \mathsf{Ver}(1^n, y_1, \dots, y_s) : y_1, \dots, y_s \leftarrow \mathcal{D}(1^n)^{\otimes s}] \geq 1 - \frac{1}{n^c}$$

for all sufficiently large $n \in \mathbb{N}$.

In the following, we use the following Claims 5.5 to 5.7. These directly follows from probabilistic argument. For clarity, we describe the proof in the end of proof of correctness.

Claim 5.5. Let us denote

$$\mathcal{I}_n := \left\{ y_1, \dots, y_s \in (\{0,1\}^{m(n)})^{\times s} : \Pr\left[\mathsf{Approx}(y_1, \dots, y_s) \geq \frac{\Pr[y_1, \dots, y_s \leftarrow \mathcal{D}(1^n)^{\otimes s}]}{2} \right] \geq 1 - n^{-2c} \right\}.$$

Then, we have

$$\Pr_{y \leftarrow \mathcal{D}(1^n)^{\otimes s}}[y_1, \dots, y_s \in \mathcal{I}_n] \geq 1 - n^{-2c}$$

for all sufficiently large $n \in \mathbb{N}$.

Claim 5.6. Let us denote

$$\mathcal{J}_n := \left\{ (y_1, \dots, y_s) \in (\{0,1\}^{m(n)})^{\times s} : \Pr\left[\mathcal{M}(y_1, \dots, y_s) \geq uK^{t_0(t(n))}(y_1, \dots, y_s | 1^n) - 1 \right] \geq 1 - n^{-2c} \right\}.$$

Then, we have

$$\Pr_{y_1, \dots, y_s \leftarrow \mathcal{D}(1^n)^{\otimes s}}[y_1, \dots, y_s \in \mathcal{J}_n] \geq 1 - n^{-2c}$$

for all sufficiently large $n \in \mathbb{N}$.

Claim 5.7. Let us denote

$$\mathcal{K}_n := \Bigg\{ (y_1, \dots, y_s) \in (\{0,1\}^{m(n)})^{\times s} : \\ -\log_2(\Pr[y_1, \dots, y_s \leftarrow \mathcal{D}(1^n)^{\otimes s}]) \geq uK^{t_0(t(n))}(y_1, \dots, y_s | 1^n) + \alpha - 2 \Bigg\}.$$

Then, we have

$$\Pr_{y_1, \dots, y_s \leftarrow \mathcal{D}(1^n)^{\otimes s}}[y_1, \dots, y_s \in \mathcal{K}_n] \leq \frac{1}{n^{2c}}$$

for all sufficiently large $n \in \mathbb{N}$.

From union bound, we have

$$
\begin{aligned}
&\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}[\bot\leftarrow\mathsf{Ver}(1^n,y_1,\dots,y_s)]\\
&\le\sum_{y_1,\dots,y_s\notin\mathcal{I}_n\cap\mathcal{J}_n}\Pr\left[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s(n)}\right]\\
&+\sum_{y_1,\dots,y_s\in\mathcal{I}_n\cap\mathcal{J}_n}\Pr\left[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s(n)}\right]\Pr_{\mathsf{Approx},\mathcal{M}}[-\log_2\left(\mathsf{Approx}(y_1,\dots,y_s)\right)\ge\mathcal{M}(y_1,\dots,y_s)+\alpha]\\
&\le\frac{2}{n^{2c}}\\
&+\sum_{y_1,\dots,y_s\in\mathcal{I}_n\cap\mathcal{J}_n}\Pr\left[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s(n)}\right]\Pr_{\mathsf{Approx},\mathcal{M}}[-\log_2\left(\mathsf{Approx}(y_1,\dots,y_s)\right)\ge\mathcal{M}(y_1,\dots,y_s)+\alpha]\\
&\le\frac{2}{n^{2c}}+\frac{2}{n^{2c}}\\
&+\sum_{y_1,\dots,y_s\in\mathcal{I}_n\cap\mathcal{J}_n}\Pr\left[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s(n)}\right]\\
&\qquad\Pr\left[-\log_2\left(\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}]\right)\ge uK^{t_0(t(n))}(y_1,\dots,y_s|1^n)+\alpha-2\right]\\
&\le\frac{4}{n^{2c}}+\sum_{y_1,\dots,y_s\in\mathcal{I}_n\cap\mathcal{J}_n\cap\mathcal{K}_n}\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}]\\
&\le\frac{5}{n^{2c}}
\end{aligned}
$$

for all sufficiently large $n\in\mathbb{N}$, which completes the proof of the correctness. Here, in the second inequality, we have used Claims 5.5 and 5.6, in the third inequality, we have used the definition of $\mathcal{I}_n$ and $\mathcal{J}_n$ and in the final inequality, we have used Claim 5.7.

Proof of Claim 5.5. We have

$$
\begin{aligned}
1-n^{-4c}&\le\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}\left[\frac{\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}]}{2}\le\mathsf{Approx}(y_1,\dots,y_s)\right]\\
&=\sum_{y_1,\dots,y_s\in\mathcal{I}_n}\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}]\Pr\left[\frac{\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}]}{2}\le\mathsf{Approx}(y_1,\dots,y_s)\right]\\
&+\sum_{y_1,\dots,y_s\notin\mathcal{I}_n}\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}]\Pr\left[\frac{\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}]}{2}\le\mathsf{Approx}(y_1,\dots,y_s)\right]\\
&\le\sum_{y_1,\dots,y_s\in\mathcal{I}_n}\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}]+\sum_{y_1,\dots,y_s\notin\mathcal{I}_n}\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}](1-n^{-2c})
\end{aligned}
$$

for all sufficiently large $n\in\mathbb{N}$. This implies that

$$
1-n^{-2c}\le\sum_{y_1,\dots,y_s\in\mathcal{I}_n}\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}]
$$

for all sufficiently large $n\in\mathbb{N}$. □

Proof of Claim 5.6. The proof is the same as that of Claim 5.5. Therefore, we omit it. □

Proof of Claim 5.7. From Theorem 2.6, we have

$$\begin{aligned}&\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}[-\log_2(\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}])\geq uK^{t_0(t(n))}(y_1,\dots,y_s|1^n)+\alpha-2]\\&\leq(m(n)\cdot s+C)2^{-\alpha+3}\leq 8(m(n)\cdot s+C)n^{-\log(n)}\leq n^{-2c}\end{aligned}$$

for all sufficiently large $n\in\mathbb{N}$. This implies that

$$\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}[y_1,\dots,y_s\in\mathcal{K}_n]\leq\frac{1}{n^{2c}}$$

for all sufficiently large $n\in\mathbb{N}$. □

Adaptive Security: In the following, we prove the security. In other words, for any t-time probabilistic adversary $\mathcal{A}$, which takes 1^n as input and outputs $x\in\{0,1\}^{m(n)\cdot s}$, and satisfies $\Delta(\mathsf{Marginal}_{\mathcal{A}}(1^n),\mathcal{D}(1^n))\geq\epsilon(n)$, we show that

$$\Pr[\top\leftarrow\mathsf{Ver}(1^n,y_1,\dots,y_s):y_1,\dots,y_s\leftarrow\mathcal{A}(1^n)]\leq 1-\epsilon(n)+n^{-c}$$

for all sufficiently large $n\in\mathbb{N}$.

We have the following Claims 5.8 and 5.9. The proof of Claim 5.8 is similar to that of Claim 5.6. Claim 5.9 directly follows from Theorem 2.5 and $s=n^{4c}\left(\log_2\frac{1}{1-\epsilon(n)}+2(\log(n))^2\right)$. Therefore, we omit the proofs.

Claim 5.8. Let us denote $\mathcal{L}_n$

$$\mathcal{L}_n:=\left\{y_1,\dots,y_s\in\{0,1\}^{m(n)\cdot s}:\Pr\left[\mathcal{M}(y_1,\dots,y_s)\leq uK^{t_0(t(n))}(y_1,\dots,y_s|1^n)+1\right]\geq 1-n^{-2c}\right\}.$$

Then, we have

$$\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}[y_1,\dots,y_s\in\mathcal{L}_n]\geq 1-n^{-2c}$$

for all sufficiently large $n\in\mathbb{N}$.

Claim 5.9. Let us denote

$$\mathcal{N}_n:=\left\{y_1,\dots,y_s\in\{0,1\}^{m(n)\cdot s}:-\log_2(\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}])\leq uK^{t_0(t(n))}(y_1,\dots,y_s|1^n)+1+\alpha\right\}.$$

Then, for any PPT algorithm $\mathcal{A}$ such that

$$\Delta(\mathsf{Marginal}_{\mathcal{A}}(1^n),\mathcal{D}(1^n))\geq\epsilon(n),$$

we have

$$\Pr_{y_1,\dots,y_s\leftarrow\mathcal{A}(1^n)}[y_1,\dots,y_s\in\mathcal{N}_n]\leq 1-\epsilon(n)+n^{-2c}$$

for all sufficiently large $n\in\mathbb{N}$.

From union bound,

$$
\begin{aligned}
&\Pr_{y_1,\ldots,y_s\leftarrow\mathcal{A}(1^n)}[\top\leftarrow\mathsf{Ver}(1^n,y_1,\ldots,y_s)]\\
&\leq \sum_{y_1,\ldots,y_s\notin\mathcal{L}_n}\Pr[y_1,\ldots,y_s\leftarrow\mathcal{A}(1^n)]\\
&+\sum_{y_1,\ldots,y_s\in\mathcal{L}_n}\Pr[y_1,\ldots,y_s\leftarrow\mathcal{A}(1^n)]\Pr[-\log_2\left(\mathsf{Approx}(y_1,\ldots,y_s)\right)\leq\mathcal{M}(y_1,\ldots,y_s)+\alpha]\\
&\leq n^{-2c}\\
&+\sum_{y_1,\ldots,y_s\in\mathcal{L}_n}\Pr[y_1,\ldots,y_s\leftarrow\mathcal{A}(1^n)]\Pr[-\log_2\left(\mathsf{Approx}(y_1,\ldots,y_s)\right)\leq\mathcal{M}(y_1,\ldots,y_s)+\alpha]\\
&\leq 3n^{-2c}\\
&+\sum_{y_1,\ldots,y_s\in\mathcal{L}_n}\Pr[y_1,\ldots,y_s\leftarrow\mathcal{A}(1^n)]\Pr\left[-\log_2\left(\Pr[y_1,\ldots,y_S\leftarrow\mathcal{D}(1^n)]\right)\leq uK^{t_0(t(n))}+\alpha+1\right]\\
&\leq n^{-2c}+\sum_{y_1,\ldots,y_s\in\mathcal{L}_n\cap\mathcal{N}_n}\Pr[y_1,\ldots,y_s\leftarrow\mathcal{A}(1^n)]\\
&\leq 1-\epsilon(n)+n^{-2c}
\end{aligned}
$$

for all sufficiently large $n\in\mathbb{N}$. Here, in the second inequality, we have used Claim 5.8, in the third inequality, we have used the definition of $\mathcal{L}_n$ and the definition of Approx, and in the final inequality, we have used Claim 5.9. This completes the security. □

5.2 Hardness of Classical Distribution Verification from OWFs

Theorem 5.10. *Consider any PPT algorithm $\mathcal{D}$ that takes 1^n as input and outputs an $x\in\{0,1\}^{m(n)}$ with a polynomial m. Suppose that $H(\mathcal{D}(1^n))\geq n^{1/c}$ for a constant $c>0$. If OWFs exist, then $\mathcal{D}$ is not selectively-verifiable with a PPT algorithm with classical-security.*

For showing Theorem 5.10, we will use the following Lemma 5.11.

Lemma 5.11 (Fannes-Inequality [NC11]). *For any pair of random variables X and Y over domain U*

$$|H(X)-H(Y)|\leq\Delta(X,Y)\cdot\log(|U|)+1/e.$$

Proof of Theorem 5.10. Let $\mathcal{D}$ be a PPT algorithm, which takes 1^n as input and outputs $x\in\{0,1\}^{m(n)}$ such that $H(D(1^n))\geq n^{1/c}$ for some constant c.

In the following, we construct a PPT algorithm $\mathcal{A}$ such that

$$\Delta(\mathcal{A}(1^n),\mathcal{D}(1^n))\geq 1/\mathrm{poly}(n)$$

and for every PPT algorithm Ver, every polynomial s and every polynomial p,

$$
\begin{aligned}
&\Big|\Pr\Big[\top\leftarrow\mathsf{Ver}(x_1,\ldots,x_s):x_1,\ldots,x_s\leftarrow\mathcal{D}^{\otimes s(n)}(1^n)\Big]\\
&\qquad-\Pr\Big[\top\leftarrow\mathsf{Ver}(x_1,\ldots,x_s):x_1,\ldots,x_s\leftarrow\mathcal{A}(1^n)^{\otimes s(n)}\Big]\Big|\leq\frac{1}{p(n)}
\end{aligned}
\tag{2}
$$

for infinitely many $n\in\mathbb{N}$.

Because $\mathcal{D}$ is a PPT algorithm, $\mathcal{D}$ internally takes $\ell(n)$ random bits as input for some polynomial ℓ. Let G be an infinitely-often PRG, which takes $n^{1/c}/4$ bits of randomness to $\ell(n)$ bits. Our construction of $\mathcal{A}$ is as follows:

$\mathcal{A}(1^n)$**:**

1. Sample $r \leftarrow \{0,1\}^{\frac{n^{1/c}}{4}}$.
2. Output $\mathcal{D}(1^n; G(r))$.

Statistically Far: From Lemma 5.11, we have

$$\begin{aligned} |H(\mathcal{A}(1^n)) - H(\mathcal{D}(1^n))| &\leq \Delta(\mathcal{A}(1^n), \mathcal{D}(1^n)) \cdot m(n) + 1/e \\ \frac{3}{4}n^{1/c} - 1/e &\leq \Delta(\mathcal{A}(1^n), \mathcal{D}(1^n)) \cdot m(n). \end{aligned}$$

This implies that

$$\Delta(\mathcal{A}(1^n), \mathcal{D}(1^n)) \geq \frac{1}{m(n)} \left(\frac{3n^{1/c}}{4} - 1/e \right).$$

Security: For contradiction, suppose that there exists a PPT adversary Ver that breaks Equation (2). Then, we construct a PPT adversary $\mathcal{B}$ that breaks the security of PRGs.

$\mathcal{B}(1^n)$**:**

1. Receive $r_1, ..., r_s$.
2. Run $x_i \leftarrow \mathcal{D}(1^n; r_i)$ for all $i \in [s]$.
3. Run $b \leftarrow \mathsf{Ver}(x_1, \ldots, x_s)$.

For contradiction, suppose that Ver breaks the Equation (2). This directly implies that $\mathcal{B}$ breaks the security of PRGs. □

6 Quantum Distribution Verification

In this section, we show an upperbound and a lowerbound of the complexity of verifying quantum distributions.

6.1 Efficient Verification of Quantum Distributions with PP Oracle

Theorem 6.1. *Every QPT algorithm $\mathcal{D}$ is adaptively-verifiable with a classical deterministic polynomial-time algorithm querying to* PP *oracle.*

Although the proof is similar to that of Theorem 5.1, we describe the proof for clarity. For showing Theorem 6.1, we use the following Theorems 6.2 and 6.3.

Theorem 6.2 (Worst-case probability estimation with PP oracle [FR99]). *For any constant $c \in \mathbb{N}$, and for any QPT algorithm $\mathcal{D}$, which takes 1^n as input and outputs $x \in \{0,1\}^{m(n)}$, where m is a polynomial, there exists a deterministic polynomial-time algorithm* Approx *querying to* PP *oracle such that*

$$\Pr\left[\mathsf{Approx}^{\mathsf{PP}}(x, 1^n) = \Pr[x \leftarrow \mathcal{D}(1^n)]\right] = 1$$

for all $x \in \{0,1\}^{m(n)}$ and for all sufficiently large $n \in \mathbb{N}$.

Theorem 6.2 directly implies the following Theorem 6.3.

Theorem 6.3. *For any constant $c \in \mathbb{N}$ and polynomial t, there exists a deterministic polynomial-time algorithm $\mathcal{M}$ querying to* PP *oracle such that*

$$\Pr\left[\mathcal{M}^{\mathsf{PP}}(x, 1^n) = quK^{t(|x|)}(x|1^n)\right] = 1$$

for all $x \in \{0,1\}^*$ and all sufficiently large $n \in \mathbb{N}$.

Proof of Theorem 6.1. In the following, let us introduce notations to describe Ver.

Notations. We set $s(n, t(n), \epsilon(n)) = n^{4c}\left(\log_2 \frac{1}{1-\epsilon(n)} + 2(\log(n))^2\right)$. For simplicity, we often denote $s = n^{4c}\left(\log_2 \frac{1}{1-\epsilon(n)} + 2(\log(n))^2\right)$ below.

We also set $\alpha = (\log(n))^2$.

Approx is a deterministic polynomial-time algorithm querying to PP oracle given in Theorem 6.2 such that

$$\Pr\left[\mathsf{Approx}^{\mathsf{PP}}(x, 1^n) = \Pr[x \leftarrow \mathcal{D}(1^n)^{\otimes s}]\right] = 1$$

for all $x \in \{0,1\}^{s \cdot m(n)}$ and all sufficiently large $n \in \mathbb{N}$.

$\mathcal{M}$ is a deterministic polynomial-time algorithm querying to PP oracle given in Theorem 6.3 such that

$$\Pr\left[\mathcal{M}^{\mathsf{PP}}(x, 1^t) = quK^{t(1^n)}(x|1^n)\right] = 1$$

for all $x \in \{0,1\}^{s \cdot m(n)}$ and for all sufficiently large $n \in \mathbb{N}$.

Construction: We give a construction of Ver.

Ver**:**

1. Take 1^n and $y_1, \ldots, y_{s(n)}$ as input.
2. Compute $k \leftarrow \mathcal{M}^{\mathsf{PP}}(y_1, \ldots, y_{s(n)}, 1^t)$.
3. Compute $p \leftarrow \mathsf{Approx}^{\mathsf{PP}}(y_1, \ldots, y_{s(n)})$.
4. Output $\top$ if $-\log_2 p \leq k + \alpha$. Otherwise, output $\bot$.

In the following, we show that Ver satisfies the correctness and security.

Correctness: From union bound, and the definition of $\mathcal{M}$ and Approx, we have

$$\begin{aligned}
&\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}[\bot\leftarrow\mathsf{Ver}(1^n,y_1,\dots,y_s)]\\
&=\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}[-\log_2\left(\mathsf{Approx}^{\mathsf{PP}}(y_1,\dots,y_s)\right)\geq\mathcal{M}^{\mathsf{PP}}(y_1,\dots,y_s)+\alpha]\\
&=\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}\left[-\log_2\left(\Pr[\mathcal{D}(1^n)^{\otimes s}\to y_1,\dots,y_s]\right)\geq quK^{t(n)}(y_1,\dots,y_s|1^n)+\alpha\right]
\end{aligned}$$

for all sufficiently large $n\in\mathbb{N}$. From Theorem 3.4, we have

$$\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}\left[-\log_2(\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}])\geq quK^{t(n)}(y_1,\dots,y_s|1^n)+\alpha\right]\leq\frac{s\cdot m(n)}{n^{\log(n)}}\leq\frac{1}{n^{2c}}$$

for all sufficiently large $n\in\mathbb{N}$. This concludes the correctness.

Adaptive-Security: From union bound and the definition of $\mathcal{M}$ and Approx, we have

$$\begin{aligned}
&\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}[\mathsf{Ver}(1^n,y_1,\dots,y_s)\to\top]\\
&\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}[-\log_2\left(\mathsf{Approx}^{\mathsf{PP}}(y_1,\dots,y_s)\right)\leq\mathcal{M}^{\mathsf{PP}}(y_1,\dots,y_s)+\alpha]\\
&=\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}\left[-\log_2\left(\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}]\right)\leq quK^{t(n)}(y_1,\dots,y_s|1^n)+\alpha\right]
\end{aligned}$$

for all sufficiently large $n\in\mathbb{N}$.

Furthermore, from Theorem 3.5, for any $t(n)$-time quantum adversary $\mathcal{A}$, which takes 1^n and outputs strings of length $m(n)\cdot s$, and satisfies $\Delta(\mathsf{Marginal}_{\mathcal{A}}(1^n),\mathcal{D}_n)\geq\epsilon(n)$, we have

$$\begin{aligned}
&\Pr_{y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}}\left[-\log_2\left(\Pr[y_1,\dots,y_s\leftarrow\mathcal{D}(1^n)^{\otimes s}]\right)\leq quK^{t(n)}(y_1,\dots,y_s|1^n)+\alpha\right]\\
&\leq 1-\epsilon(n)+\sqrt{\frac{\log_2\frac{1}{1-\epsilon(n)}+(\log(n))^2+C}{n^{4c}\left(\log_2\frac{1}{1-\epsilon(n)}+2(\log(n))^2\right)}}\\
&\leq 1-\epsilon(n)+n^{-2c}
\end{aligned}$$

for all sufficiently large $n\in\mathbb{N}$. □

6.2 Hardness of Quantum Distribution Verification from QEFID

Theorem 6.4. *Assume that infinitely-often QEFID exists. Then, there exists a QPT algorithm $\mathcal{Q}$, which is not selectively-verifiable with a QPT algorithm.*

Proof of Theorem 6.4. Let us consider a infinitely-often QEFID Gen. Then, from the definition, we have

$$\mathsf{SD}(\mathsf{Gen}(1^n,0),\mathsf{Gen}(1^n,1))\geq 1-\mathsf{negl}(n)$$

and for any QPT algorithm Ver, any polynomial s, and any polynomial p, we have

$$\begin{aligned}
&\Pr[\top\leftarrow\mathsf{Ver}(x_1,\dots,x_s):(x_1,\dots,x_s)\leftarrow\mathsf{Gen}(1^n,0)^{\otimes s}]\\
&\qquad-\Pr[\top\leftarrow\mathsf{Ver}(x_1,\dots,x_s):(x_1,\dots,x_s)\leftarrow\mathsf{Gen}(1^n,1)^{\otimes s}]\leq\frac{1}{p(n)}
\end{aligned}$$

for infinitely many $n\in\mathbb{N}$. This means that, for each $b\in\{0,1\}$, $\mathsf{Gen}(1^n,b)$ is not selectively-verifiable in the sense of Definition 4.2. □

7 One-Way Puzzles from Hardness of Verifying Quantum Advantage

7.1 Definition

In the following, we introduce the definitions of strong quantum advantage samplers and verification of quantum advantage samplers.

Definition 7.1 (Quantum Advantage Samplers (QASs) [MSY25]). *A strong quantum advantage sampler (QAS) is a QPT algorithm $\mathcal{D}_Q$ taking in 1^n and outputting strings in $\{0,1\}^{m(n)}$, where m is a polynomial, such that for all PPT algorithms $\mathcal{D}_C$ that take 1^n as input and output $x \in \{0,1\}^{m(n)}$:*

$$\Delta(\mathcal{D}_Q(1^n), \mathcal{D}_C(1^n)) \geq 1 - \mathsf{negl}(n).$$

Remark 7.2. In the original definition of QASs [MSY25], the statistical distance between $\mathcal{D}_Q$ and $\mathcal{D}_C$ is lowerbounded by $1/p(n)$ for a polynomial p. Here, we consider the stronger lowerbound, $1 - \mathsf{negl}(n)$, and therefore we call them *strong* QASs. Note that we do not know how to construct a strong QAS from a weak QAS (i.e. QAS according to the original definition of QASs [MSY25]). However, we can obtain a strong QAS from a plausible cryptographic assumption. More specifically, if a quantum pseudorandom generators (QPRG) secure against a QPT algorithm querying to an $\mathbf{NP}$ oracle, then a strong QAS exists. Please, refer the proof to Appendix B.

Definition 7.3 (Verification of Quantum Advantage Samplers). *Let $\mathcal{D}_Q$ be a strong QAS that takes 1^n as input, and outputs $x \in \{0,1\}^{m(n)}$ for some polynomial m. We say that $\mathcal{D}_Q$ is adaptively-verifiable with a QPT algorithm if, for any polynomial t, and any constant $c > 0$, there exists a polynomial s and a QPT algorithm Ver such that the following holds.*

Correctness: *We have*

$$\Pr[\top \leftarrow \mathsf{Ver}(1^n, x_1, \ldots, x_s) : (x_1, \ldots, x_s) \leftarrow \mathcal{D}_Q(1^n)^{\otimes s}] \geq 1 - \mathsf{negl}(n).$$

Security against adaptive classical adversaries: *For any $t(n)$-time classical adversary $\mathcal{A}$, which takes 1^n as input and outputs $(x_1, ..., x_{s(n)}) \in (\{0,1\}^{m(n)})^{\times s(n)}$, we have*

$$\Pr[\top \leftarrow \mathsf{Ver}(1^n, x_1, \ldots, x_s) : (x_1, \ldots, x_s) \leftarrow \mathcal{A}(1^n)] \leq \mathsf{negl}(n).$$

7.2 Result

In this subsection, we prove the following theorem.

Theorem 7.4. *If there do not exist infinitely-often OWPuzzs, then every strong QAS is adaptively-verifiable with a QPT algorithm.*

To show Theorem 7.4, we use the following Lemmata 7.5 and 7.6 and Theorems 7.7 to 7.9.

Lemma 7.5. *For any strong QAS $\mathcal{D}_Q$, constant c, and polynomial $t(n)$ greater than the runtime of $\mathcal{D}_Q$, there exists a function $f(n) = \omega(\log(n))$ such that with probability at least $1 - 1/n^c$ over $x \leftarrow \mathcal{D}_Q(1^n)$,*

$$quK^t(x|1^n) + f(n) \leq uK^t(x|1^n).$$

for sufficiently large n.

Proof of Lemma 7.5. By the definitions of quK^t and uK^t, we know that $quK^t(x|1^n) \leq -\log(\Pr[x \leftarrow \mathcal{D}_Q(1^n)]) + O(1)$ and $uK^t(x|1^n) = -\log(\Pr[x \leftarrow \mathcal{U}^t(r,1^n)])$. So it suffices to prove the lemma to show that there exists a function $f(n) = \omega(\log(n))$ such that with probability at least $1 - 1/n^c$ over $x \leftarrow \mathcal{D}_Q(1^n)$,

$$-\log(\Pr[x \leftarrow \mathcal{D}_Q(1^n)]) + f(n) \leq -\log\Big(\Pr\Big[x \leftarrow \mathcal{U}^t(r,1^n)\Big]\Big)$$

or equivalently that for all c' and sufficiently large n

$$\Pr\Big[x \leftarrow \mathcal{U}^t(r,1^n)\Big] / \Pr[x \leftarrow \mathcal{D}_Q(1^n)] \leq \frac{1}{n^{c'}}. \tag{3}$$

For convenience of notation we define $Q(x) = \Pr[x \leftarrow \mathcal{D}_Q(1^n)] - \Pr[x \leftarrow \mathcal{U}^t(r,1^n)]$, $R(x) = \Pr[x \leftarrow \mathcal{U}^t(r,1^n)] / \Pr[x \leftarrow \mathcal{D}_Q(1^n)]$, $S_D = \{x : Q(x) \geq 0\}$, and $S_{B,c} = \{x : R(x) \geq 1/n^c\}$. For the sake of contradiction with Equation (3) we assume that there exists c and c' such that $\Pr_{x \leftarrow \mathcal{D}_Q}[x \in S_{B,c}] \geq 1/n^{c'}$. By the definition of $S_{B,c}$ for all $x \in S_{B,c}$

$$\Pr\Big[x \leftarrow \mathcal{U}^t(r,1^n)\Big] \geq \frac{1}{n^{c'}} \Pr[x \leftarrow \mathcal{D}_Q(1^n)].$$

Subtracting the above equation from $\Pr[x \leftarrow \mathcal{D}_Q(1^n)] = \Pr[x \leftarrow \mathcal{D}_Q(1^n)]$ we get that for all $x \in S_{B,c}$

$$\begin{aligned} Q(x) = \Pr[x \leftarrow \mathcal{D}_Q(1^n)] - \Pr\Big[x \leftarrow \mathcal{U}^t(r,1^n)\Big] &\leq \Pr[x \leftarrow \mathcal{D}_Q(1^n)] - \frac{1}{n^{c'}} \Pr[x \leftarrow \mathcal{D}_Q(1^n)] \\ &= \left(1 - \frac{1}{n^{c'}}\right) \Pr[x \leftarrow \mathcal{D}_Q(1^n)] \end{aligned}$$

Because $\mathcal{D}_Q$ is a QAS

$$\begin{aligned} 1 - \mathsf{negl}(n) \leq \Delta(D_Q(1^n), \mathcal{U}^t(r,1^n)) &= \sum_{x \in S_D} Q(x) \\ &= \sum_{x \in S_D / S_{B,c}} Q(x) + \sum_{x \in S_D \cap S_{B,c}} Q(x) \\ &\leq \sum_{x \in S_D / S_{B,c}} \Pr[x \leftarrow \mathcal{D}_Q(1^n)] + \sum_{x \in S_D \cap S_{B,c}} Q(x) \\ &= \Pr_{x \leftarrow \mathcal{D}_Q(1^n)}[x \in S_D / S_{B,c}] + \sum_{x \in S_D \cap S_{B,c}} Q(x) \\ &\leq \Pr_{x \leftarrow \mathcal{D}_Q(1^n)}[x \in S_D / S_{B,c}] + \Pr_{x \leftarrow \mathcal{D}_Q(1^n)}[x \in S_D \cap S_{B,c}](1 - \frac{1}{n^{c'}}) \\ &= 1 - \Pr_{x \leftarrow \mathcal{D}_Q(1^n)}[x \in S_D \cap S_{B,c}] - \Pr_{x \leftarrow \mathcal{D}_Q(1^n)}[x \notin S_D] \\ &\quad + \Pr_{x \leftarrow \mathcal{D}_Q(1^n)}[x \in S_D \cap S_{B,c}](1 - \frac{1}{n^{c'}}) \\ &= 1 - \frac{1}{n^{c'}} \Pr_{x \leftarrow \mathcal{D}_Q(1^n)}[x \in S_D \cap S_{B,c}] - \mathsf{negl}(n) \\ &\leq 1 - \frac{1}{n^{c+c'}} - \mathsf{negl}(n) < 1 - \mathsf{negl}(n), \end{aligned}$$

which is a contradiction. □

Lemma 7.6 (No OWPuzzs implies average-case probability estimation [CGGH25, HM25, KT25]). *Assume that there do not exist infinitely-often OWPuzzs. Then, for any constant $c \in \mathbb{N}$, and for any QPT algorithm $\mathcal{D}$, which takes 1^n as input and outputs $x \in \{0,1\}^{m(n)}$, where m is a polynomial, there exists a QPT algorithm* Approx *such that*

$$\Pr_{\substack{x \leftarrow \mathcal{D}(1^n) \\ \mathsf{Approx}}}\left[\frac{1}{2}\Pr[x \leftarrow \mathcal{D}(1^n)] \leq \mathsf{Approx}(x) \leq \Pr[x \leftarrow \mathcal{D}(1^n)]\right] \geq 1 - n^{-c}$$

for all sufficiently large $n \in \mathbb{N}$.

Because we can run the probability estimator above against the universal time bounded distribution $\mathcal{U}^t(r, 1^n)$, and every t time classical and quantum distribution makes up a constant fraction of $\mathcal{Q}^t(1^n)$, Lemma 7.6 directly implies the following theorem. For clarity, we describe the proof.

Theorem 7.7. *Let $\mathcal{D}$ be a QPT algorithm running in time at most $t(n)$ that takes 1^n as input, and outputs $x \in \{0,1\}^{m(n)}$, where m is an arbitrary polynomial. Assume that there do not exist infinitely-often OWPuzzs. Then, for any constant $c \in \mathbb{N}$ and polynomial t, there exists a polynomial t_0 and a QPT algorithm $\mathcal{M}$ such that*

$$\Pr_{\substack{x \leftarrow \mathcal{D}(1^n) \\ \mathcal{M}}}[quK^{t_0(t(n))}(x|1^n) \leq \mathcal{M}(x) \leq quK^{t_0(t(n))}(x|1^n) + 1] \geq 1 - n^{-c}$$

for all sufficiently large $n \in \mathbb{N}$.

Proof of Theorem 7.7. Let t_0 be a polynomial given in Theorem 3.3.

Assume that infinitely-often OWPuzzs do not exist. Then, from Lemma 7.6, there exists a QPT algorithm Approx

$$\Pr_{\substack{x \leftarrow \mathcal{Q}^t(1^n) \\ \mathsf{Approx}}}\left[\frac{1}{2}\Pr\left[x \leftarrow \mathcal{Q}^{t_0(t(n))}(1^n)\right] \leq \mathsf{Approx}(x) \leq \Pr\left[x \leftarrow \mathcal{Q}^{t_0(t(n))}(1^n)\right]\right] \geq 1 - n^{-4c}$$

for all sufficiently large $n \in \mathbb{N}$. We let

$$\mathcal{S}_n := \left\{x \in \{0,1\}^* : \Pr_{\mathsf{Approx}}\left[\frac{1}{2}\Pr\left[x \leftarrow \mathcal{Q}^{t_0(t(n))}(1^n)\right] \leq \mathsf{Approx}(x) \leq \Pr\left[x \leftarrow \mathcal{Q}^{t_0(t(n))}(1^n)\right]\right] \leq 1 - n^{-2c}\right\}.$$

From Markov inequality, we have

$$\Pr_{x \leftarrow \mathcal{Q}^{t_0(t(n))}(1^n)}[x \in \mathcal{S}_n] \leq n^{-2c}.$$

This implies that, for any QPT algorithm $\mathcal{D}$ that takes 1^n as input, and outputs $x \in \{0,1\}^{m(n)}$ running in at most $t(n)$ times, there exists a constant D such that

$$\begin{aligned} n^{-2c} &\geq \Pr_{x \leftarrow \mathcal{Q}^{t_0(t(n))}(1^n)}[x \in \mathcal{S}_n] \\ &= \sum_{x \in \mathcal{S}_n} \Pr\left[x \leftarrow \mathcal{Q}^{t_0(t(n))}(1^n)\right] \\ &\geq \sum_{x \in \mathcal{S}_n} \frac{\Pr[x \leftarrow \mathcal{D}(1^n)]}{D} \end{aligned}$$

for all sufficiently large $n \in \mathbb{N}$. Here, in the last inequality, we have used Theorem 3.3. Therefore, we have

$$\Pr_{x \leftarrow \mathcal{D}(1^n)}[x \in \mathcal{S}_n] \leq D \cdot n^{-2c}$$

for all sufficiently large $n \in \mathbb{N}$. This implies that

$$\Pr_{x \leftarrow \mathcal{D}(1^n)}\left[\Pr_{\mathsf{Approx}}\left[\frac{1}{2}\Pr\left[x \leftarrow \mathcal{Q}^{t_0(t(n))}(1^n)\right] \leq \mathsf{Approx}(x) \leq \Pr\left[x \leftarrow \mathcal{Q}^{t_0(t(n))}(1^n)\right]\right] \geq 1 - n^{-2c}\right] \geq 1 - D \cdot n^{-2c}$$

for all sufficiently large $n \in \mathbb{N}$. Hence, we have

$$\begin{aligned}\Pr_{\substack{x \leftarrow \mathcal{D}(1^n)\\ \mathsf{Approx}}}\left[\frac{1}{2}\Pr\left[x \leftarrow \mathcal{Q}^{t_0(t(n))}(1^n)\right] \leq \mathsf{Approx}(x) \leq \Pr\left[x \leftarrow \mathcal{Q}^{t_0(t(n))}(1^n)\right]\right] &\geq 1 - (D+1) \cdot n^{-2c}\\ &\geq 1 - n^{-c}\end{aligned}$$

for all sufficiently large $n \in \mathbb{N}$. □

If OWPuzz do not exist, then post-quantum OWFs also do not exist.[18] Therefore Theorem 5.2 and Theorem 5.3 give us the following results.

Theorem 7.8 (No i.o.-pq-OWFs implies one-sided error probability estimation [IL90, IRS22, CHK25]). *If there do not exist infinitely-often post-quantum OWFs, then for any polynomial m, for any PPT algorithm $\mathcal{D}$, which takes 1^n as input and outputs $x \in \{0,1\}^{m(n)}$, and for any constant $c \in \mathbb{N}$, there exists a QPT algorithm* Approx *such that for all sufficiently large n,*

$$\Pr_{\substack{x \leftarrow \mathcal{D}(1^n)\\ \mathsf{Approx}}}\left[\frac{1}{2}\Pr[x \leftarrow \mathcal{D}(1^n)] \leq \mathsf{Approx}(x, 1^n) \leq \Pr[x \leftarrow \mathcal{D}(1^n)]\right] \geq 1 - n^{-c}.$$

Furthermore, for all sufficiently large n, for all $x \in \{0,1\}^{m(n)}$,

$$\Pr_{\mathsf{Approx}}[\mathsf{Approx}(x, 1^n) \leq \Pr[x \leftarrow \mathcal{D}(1^n)]] \geq 1 - n^{-c}.$$

Theorem 7.9 (No i.o.-pq-OWFs implies uK^t estimation on all PPT distributions). *Let $\mathcal{D}$ be a PPT algorithm running in time at most t that takes 1^n as input, and outputs $x \in \{0,1\}^{m(n)}$, where m is an arbitrary polynomial. Assume that there do not exist infinitely-often post-quantum OWFs. Then, for any constant $c \in \mathbb{N}$ and polynomial t, there exists a QPT algorithm $\mathcal{M}$ such that for all sufficiently large $n \in \mathbb{N}$*

$$\Pr_{\substack{x \leftarrow \mathcal{D}(1^n)\\ \mathcal{M}}}[uK^{t(n)}(x|1^n) \leq \mathcal{M}(x) \leq uK^{t(n)}(x|1^n) + 1] \geq 1 - n^{-c}$$

Furthermore, for all sufficiently large n and for all $x \in \{0,1\}^{m(n)}$,

$$\Pr_{\substack{x \leftarrow \mathcal{D}(1^n)\\ \mathcal{M}}}[\mathcal{M}(x) \geq uK^{t(n)}(x|1^n)] \geq 1 - n^{-c}$$

In the following, we describe a proof of Theorem 7.4.

[18] The reason is as follows. Let f be a post-quantum OWF. Then $(\mathsf{ans}, \mathsf{puzz}) := (x, f(x))$ serves as a secure OWPuzz.

Proof of Theorem 7.4. Let c be some constant such that $n^{-c} \cdot m(n) \leq 1/3$ where $m(n)$ is the length of the outputs of the quantum advantage sampler $\mathcal{D}_Q(1^n)$ that we are constructing a verifier for.

$\mathcal{M}_Q$ is the QPT algorithm given in Theorem 7.7 such that for all sufficiently large n

$$\Pr_{\substack{x\leftarrow\mathcal{D}(1^n)\\ \mathcal{M}_Q}}[quK^{t(n)}(x|1^n) \leq \mathcal{M}_Q(x) \leq quK^{t(n)}(x|1^n)+1] \geq 1-n^{-3c}$$

$\mathcal{M}_C$ is the QPT algorithm given in Theorem 7.9 such that for all sufficiently large n

$$\Pr_{\substack{x\xleftarrow{\$}\mathcal{D}_C(1^n)\\ \mathcal{M}_C}}[\mathcal{M}_C(x,1^n) = uK^{t(n}(x)] \geq 1-n^{-3c},$$

and

$$\Pr_{\mathcal{M}_C}[\mathcal{M}_C(x,1^n) \geq uK^{t(n}(x)] \geq 1-n^{-3c}.$$

Construction: We give a construction of Ver.

Ver**:**

1. Take 1^n and $y_1, ...y_n$ as input.
2. Compute $\mathcal{M}_C(y_1), ...\mathcal{M}_C(y_n) \to k_{c,1}, ..., k_{c,n}$.
3. Compute $\mathcal{M}_Q(y_1), ...\mathcal{M}_Q(y_n) \to k_{q,1}, ..., k_{q,n}$.
4. Let $\mathsf{Count}_{y_1,...,y_n}$ be the number of i's for which $k_{q,i} \leq k_{c,i} - 3c\log n$.
5. Output $\top$ if $\mathsf{Count}_{y_1,...,y_n} \geq n/2$. Otherwise, output $\bot$.

In the following, we show that Ver satisfies correctness and security.

Correctness: In the following, we show that

$$\Pr[\top \leftarrow \mathsf{Ver}(1^n, y_1, \ldots, y_n) : y_1, \ldots, y_n \leftarrow \mathcal{D}_Q(1^n)^{\otimes n}] \geq 1-\mathsf{negl}(n)$$

for all sufficiently large $n \in \mathbb{N}$.

From Lemma 7.5 we get that

$$\Pr_{x\leftarrow\mathcal{D}_Q}[quK^t(x|1^n)+1+3c\log(n) < uK^t(x|1^n)] \geq 1-n^{-3c},$$

from Theorem 7.7 we get that

$$\Pr_{\substack{x\leftarrow\mathcal{D}(1^n)\\ \mathcal{M}_Q}}[quK^{t(n)}(x|1^n) \leq \mathcal{M}_Q(x) \leq quK^{t(n)}(x|1^n)+1] \geq 1-n^{-3c}$$

and from Theorem 7.9 we get that

$$\Pr_{\mathcal{M}_C}[\mathcal{M}_C(x) \leq uK^t(x|1^n)] \geq 1-n^{-3c}.$$

By union bound over the above three conditions we get that

$$\Pr_{\substack{x\leftarrow\mathcal{D}_Q(1^n)\\ \mathcal{M}_Q\\ \mathcal{M}_C}}[\mathcal{M}_Q(x)+3c\log(n)\leq\mathcal{M}_C(x)]\geq 1-3n^{-3c}$$

for all sufficiently large $n\in\mathbb{N}$.
From Hoeffding's inequality,

$$\Pr_{\substack{x_1,..x_n\leftarrow\mathcal{D}_Q^{\otimes n}\\ \mathcal{M}_Q\\ \mathcal{M}_C}}[\mathsf{Count}_{x_1,\ldots,x_n}\leq n/2]$$

$$\leq\Pr_{\substack{x_1,..x_n\leftarrow\mathcal{D}_Q(1^n)^{\otimes n}\\ \mathcal{M}_Q\\ \mathcal{M}_C}}\left[|\mathsf{Count}_{x_1,\ldots,x_n}-(1-3n^{-3c})n|\geq n/3\right]\leq 2\exp\left(-\frac{n^2}{9}\right).$$

Adaptive Security: In the following, we prove security.

Assume for the sake of contradiction that there does exist some constant c', and t-time probabilistic classical adversary $\mathcal{A}$, which takes 1^n as input and outputs strings of length $m(n)\cdot n$ such that

$$\Pr_{\substack{y_1,\ldots,y_n\leftarrow\mathcal{A}(1^n)^{\otimes n}\\ \mathsf{Ver}}}[\top\leftarrow\mathsf{Ver}(1^n,y_1,\ldots,y_n)]\geq n^{-c'}.$$

This implies that

$$\Pr_{\substack{y_1,\ldots,y_n\leftarrow\mathcal{A}(1^n)^{\otimes n}\\ \mathcal{M}_C\\ \mathcal{M}_Q}}[\mathsf{Count}_{y_1,\ldots,y_n}\geq n/2]\geq n^{-c'}.$$

$$\Pr_{\substack{y_1,\ldots,y_n\leftarrow\mathcal{A}(1^n)^{\otimes n}\\ \mathcal{M}_C\\ \mathcal{M}_Q}}[\exists J\subseteq[n]:|J|\geq n/2\text{ and }\mathcal{M}_Q(y)\leq\mathcal{M}_C(y)-3c\log n]\geq n^{-c'}.$$

$$\Pr_{\substack{y\leftarrow\mathsf{Marginal}_{\mathcal{A}}(1^n)\\ \mathcal{M}_C\\ \mathcal{M}_Q}}[\mathcal{M}_Q(y)\leq\mathcal{M}_C(y)-3c\log n]\geq n^{-c'}/2 \tag{4}$$

However by Theorem 7.7, Theorem 7.9, and a union bound we know that for sufficiently large n

$$\Pr_{\substack{y\leftarrow\mathsf{Marginal}_{\mathcal{A}}(1^n)\\ \mathcal{M}_C\\ \mathcal{M}_Q}}[\mathcal{M}_Q(y)\leq\mathcal{M}_C(y)-3c\log n]\leq$$

$$2n^{-3c}+\Pr_{y\leftarrow\mathsf{Marginal}_{\mathcal{A}}(1^n)}[quK^t(y|1^n)\leq uK^t(y|1^n)-3c\log n]\leq$$

$$2n^{-3c}+\Pr_{y\leftarrow\mathsf{Marginal}_{\mathcal{A}}(1^n)}[quK^t(y|1^n)\leq\Pr[y\leftarrow\mathsf{Marginal}_{\mathcal{A}}(1^n)]+|\mathsf{Marginal}_{\mathcal{A}}(1^n)|-3c\log n] \tag{5}$$

Which (if we label $c'' := |\mathsf{Marginal}_{\mathcal{A}}|$, and consequently get that $|\mathsf{Marginal}_{\mathcal{A}}(1^n)| \leq c'' + \log(n)$) by Theorem 3.4 we know is less than

$$\begin{aligned}(5) &\leq 2n^{-3c} + (m(n)+c')2^{-3c\log n + c'' + \log n} \leq 2n^{-3c} + (m(n)+c')2^{-2c\log n} \\ &\leq 2n^{-3c} + 2m(n)\cdot n^{-2c} \leq 2n^{-3c} + \frac{2}{3}n^{-c} \leq n^{-c}\end{aligned}$$

which contradicts Equation (4). The second to last inequality above follows from the condition that $n^{-c}\cdot m(n) \leq 1/3$. □

8 Unconditional Verification of Distributions with Small Entropy

In this section, we show that every PPT (resp. QPT) algorithm with small entropy can be verified by a PPT (resp. QPT) algorithm. The security is slightly different from the previous definitions. First, the parameter of the security is weak. Second, we consider the security against any adversary instead of PPT or QPT adversary. For clarity, we describe the definition in the following:

Definition 8.1 (Adaptive-Verification of Distributions). *Let $\mathcal{D}$ be an algorithm that takes 1^n as input, and outputs $x \in \{0,1\}^{m(n)}$, where m is an arbitrary polynomial. Let us denote X to mean that $X \in \{$PPT, QPT$\}$.*

We say that the algorithm $\mathcal{D}$ is adaptively-verifiable with an X-algorithm if, for any function $\epsilon : \mathbb{N} \to (0,1)$, and any constant $c, d > 0$, there exist a polynomial s and an X-algorithm Ver such that the following holds.

Correctness: *We have*

$$\Pr[\top \leftarrow \mathsf{Ver}(1^n, x_1, \ldots, x_s) : x_1, \ldots, x_s \leftarrow \mathcal{D}(1^n)^{\otimes s}] \geq 1 - n^{-c}$$

for all sufficiently large $n \in \mathbb{N}$. Here, $s = s(n^c, d, 1/\epsilon(n))$.

Adaptive-Security: *For any algorithm $\mathcal{A}$ that takes 1^n as input and outputs $(x_1, \ldots, x_{s(n)}) \in \left(\{0,1\}^{m(n)}\right)^{\times s(n)}$, we define an algorithm $\mathsf{Marginal}_{\mathcal{A}}(1^n)$ as follows:*

$\mathsf{Marginal}_{\mathcal{A}}(1^n)$**:**

1. *Sample $(x_1, \ldots, x_s) \leftarrow \mathcal{A}(1^n)$.*
2. *Sample $i \leftarrow [s]$.*
3. *Output x_i.*

For any adversary $\mathcal{A}$ with $\Delta(\mathsf{Marginal}_{\mathcal{A}}(1^n), \mathcal{D}(1^n)) \geq \epsilon(n)$ for all $n \in \mathbb{N}$, we have

$$\Pr[\top \leftarrow \mathsf{Ver}(1^n, x_1, \ldots, x_s) : (x_1, ..., x_s) \leftarrow \mathcal{A}(1^n)] \leq (1 - \epsilon(n)) + \frac{1}{d}$$

for all sufficiently large $n \in \mathbb{N}$. Here, $s = s(n^c, d, 1/\epsilon(n))$.

In the following, we prove Theorems 8.2 and 8.3. The proof of Theorem 8.3 is the same as Theorem 8.2, and hence we omit the proof.

Theorem 8.2. *Let $\mathcal{D}$ be a PPT algorithm that takes 1^n as input and outputs $x \in \{0,1\}^{m(n)}$ for some polynomial m. If $H(\mathcal{D}(1^n)) \leq c\log(n)$ for some constant c, then $\mathcal{D}$ is adaptively-verifiable with a PPT algorithm.*

Theorem 8.3. *Let $\mathcal{D}$ be a QPT algorithm that takes 1^n as input, and outputs $x \in \{0,1\}^{m(n)}$ for some polynomial m. If $H(\mathcal{D}(1^n)) \leq c\log(n)$ for some constant c, then $\mathcal{D}$ is adaptively-verifiable with a QPT algorithm.*

Proof of Theorem 8.2. For any constant $c, d > 0$, and any PPT algorithm $\mathcal{D}$ with $H(\mathcal{D}(1^n)) \leq c\log(n)$, we construct a PPT algorithm Ver such that the following is satisfied for some polynomial s.

- $$\Pr\left[\top \leftarrow \mathsf{Ver}(x_1, \dots x_{s(n)}) : (x_1, \dots, x_{s(n)}) \leftarrow \mathcal{D}(1^n)^{\otimes s(n)}\right] \geq 1 - \mathsf{negl}(n)$$
  for all sufficiently large $n \in \mathbb{N}$.
- For any PPT algorithm $\mathcal{A}$ with $\Delta(\mathsf{Marginal}_{\mathcal{A}}(1^n), \mathcal{D}(1^n)) \geq \epsilon(n)$, we have
  $$\Pr\left[\top \leftarrow \mathsf{Ver}(x_1, \dots x_{s(n)}) : (x_1, \dots, x_{s(n)}) \leftarrow \mathcal{A}(1^n)\right] \leq 1 - \epsilon(n) + \frac{1}{d}$$
  for all sufficiently large $n \in \mathbb{N}$.

Let m be a polynomial such that $\mathcal{D}(1^n)$ outputs $m(n)$-bit strings. Let us describe our Ver algorithm.

Ver**:**

1. Receive $x_1, \dots, x_{n^{1000cd}}$.
2. Let $A^*_{x_1,\dots,x_{n^{1000cd}}}$ be a distribution that samples $i \leftarrow [n^{1000cd}]$, and outputs x_i.
3. Run $X_i \leftarrow \mathcal{D}(1^n)$ for all $i \in [m(n) \cdot n^{1000cd}]$. For each X_i, let $\mathsf{Count}_{X_1,\dots,X_{m(n)\cdot n^{1000cd}}}(X_i)$ be the number of $j \in [m(n) \cdot n^{1000cd}]$ such that $X_i = X_j$.
4. Let List_n be a set of $X \in \{X_i\}_{i\in[m(n)\cdot n^{1000cd}]}$ such that $\frac{\mathsf{Count}_{X_1,\dots,X_{m(n)\cdot n^{1000cd}}}(X)}{m(n)\cdot n^{1000cd}} \geq n^{-200cd}$.
5. Output $\top$ if $\left|\frac{\mathsf{Count}_{X_1,\dots,X_{m(n)\cdot n^{1000cd}}}(X_i)}{m(n)\cdot n^{1000cd}} - \Pr\left[X_i \leftarrow A^*_{x_1,\dots,x_{n^{1000cd}}}\right]\right| \leq n^{-100cd}$ for all $X_i \in \mathsf{List}_n$. Otherwise, output $\bot$.

Correctness. Let

$$\mathcal{S}_{\mathcal{D}}(1^n, 250cd) := \left\{\Pr[x \leftarrow \mathcal{D}(1^n)] \geq n^{-250cd}\right\}.$$

We use the following Claim 8.4.

Claim 8.4. With probability at least $1 - \mathsf{negl}(n)$ over $\{X_i\}_{i\in[m(n)\cdot n^{1000cd}]} \leftarrow \mathcal{D}(1^n)^{\otimes m(n)\cdot n^{1000cd}}$, we have

$$x \notin \mathsf{List}_n$$

for all $x \notin \mathcal{S}_{\mathcal{D}}(1^n, 250cd)$, and

$$\left|\frac{\mathsf{Count}_{X_1,\dots,X_{m(n)\cdot n^{1000cd}}}(X)}{m(n)\cdot n^{1000cd}} - \Pr[X \leftarrow \mathcal{D}(1^n)]\right| \leq n^{-250cd}$$

for all $X \in \mathsf{List}_n$.

From Hoeffding's inequality, for each $X \in \mathcal{S}_{\mathcal{D}}(1^n, 250cd)$, we have

$$\Pr_{x_1,\ldots,x_{n^{1000cd}} \leftarrow \mathcal{D}(1^n)^{\otimes n^{1000cd}}}\left[\left|\Pr\left[X \leftarrow \mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}}\right] - \Pr[X \leftarrow \mathcal{D}(1^n)]\right| \le \frac{1}{n^{250cd}}\right] \ge 1 - 2\exp\left(-\frac{n^{1000cd}}{n^{500cd}}\right) \ge 1 - 2\exp\left(-n^{500cd}\right).$$

This implies that

$$\Pr_{x_1,\ldots,x_{n^{1000cd}} \leftarrow \mathcal{D}(1^n)^{\otimes n^{1000cd}}}\left[\left|\Pr\left[X \leftarrow \mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}}\right] - \Pr[X \leftarrow \mathcal{D}(1^n)]\right| \le \frac{1}{n^{250cd}} \text{ for all } X \in \mathcal{S}_{\mathcal{D}}(1^n, 250cd)\right]$$
$$\ge \left(1 - 2\exp\left(-n^{500cd}\right)\right)^{n^{250cd}} \ge 1 - \mathsf{negl}(n).$$

From Claim 8.4, these imply that with probability at least $1 - \mathsf{negl}(n)$, for all $X \in \mathsf{List}_n$, we have

$$\left|\frac{\mathsf{Count}_{X_1,\ldots,X_{m(n)\cdot n^{1000cd}}}(X)}{m(n)\cdot n^{1000cd}} - \Pr\left[X \leftarrow \mathcal{A}^*_{x_1,\ldots,x^{n^{1000cd}}}\right]\right|$$
$$\le \left|\frac{\mathsf{Count}_{X_1,\ldots,X_{m(n)\cdot n^{1000cd}}}(X)}{m(n)\cdot n^{1000cd}} - \Pr[X \leftarrow \mathcal{D}(1^n)]\right| + \left|\Pr\left[X \leftarrow \mathcal{A}^*_{x_1,\ldots,x^{n^{1000cd}}}\right] - \Pr[X \leftarrow \mathcal{D}(1^n)]\right| \le 2n^{-250cd}.$$

Therefore, we have

$$\Pr_{x_1,\ldots,x_{n^{1000cd}} \leftarrow \mathcal{D}(1^n)^{\otimes n^{1000cd}}}[\top \leftarrow \mathsf{Ver}(x_1,\ldots,x_{n^{1000cd}})] \ge 1 - \mathsf{negl}(n).$$

Proof of Claim 8.4. For each $X \in \{0,1\}^{m(n)}$, we have

$$\Pr_{X_1,\ldots,X_{m(n)\cdot n^{1000cd}} \leftarrow \mathcal{D}(1^n)^{\otimes m(n)\cdot n^{1000cd}}}\left[\left|\frac{\mathsf{Count}_{X_1,\ldots,X_{m(n)\cdot n^{1000cd}}}(X)}{n^{1000cd}} - \Pr[X \leftarrow \mathcal{D}(1^n)]\right| \le n^{-250cd}\right]$$
$$\ge 1 - 2\exp\left(-m(n)\cdot n^{500cd}\right).$$

From union bound, with probability at least $1 - \mathsf{negl}(n)$ over $X_1,\ldots,X_{m(n)\cdot n^{1000cd}} \leftarrow \mathcal{D}(1^n)^{\otimes m(n)\cdot n^{1000cd}}$, we have

$$\left|\frac{\mathsf{Count}_{X_1,\ldots,X_{m(n)\cdot n^{1000cd}}}(X)}{m(n)\cdot n^{1000cd}} - \Pr[X \leftarrow \mathcal{D}(1^n)]\right| \le n^{-250cd}$$

for all $X \notin \{0,1\}^{m(n)}$. □

Security. For showing the security, we use the following Claims 8.5 and 8.6. For showing Claim 8.5, we use the following Claim 8.7. We defer the proof of them.

Claim 8.5. Let $\mathcal{D}$ be an arbitrary PPT algorithm, which takes 1^n as input, outputs $x \in \{0,1\}^{m(n)}$, and satisfies $H(\mathcal{D}(1^n)) \le c\log(n)$. Let

$$\mathcal{S}_{\mathcal{D}}(1^n, 50cd) := \left\{x \in \{0,1\}^* : \Pr[x \leftarrow \mathcal{D}(1^n)] \ge \frac{1}{n^{50cd}}\right\}.$$

Suppose that

$$\left|\Pr\left[x \leftarrow \mathcal{A}^*_{x_1,\dots,x_{n^{1000cd}}}\right] - \Pr[x \leftarrow \mathcal{D}(1^n)]\right| \leq 2n^{-100cd} \text{ for all } x \in \mathcal{S}_{\mathcal{D}}(1^n, 50cd). \tag{6}$$

Then, we have

$$\Delta(A^*_{x_1,\dots,x_{n^{1000cd}}}, \mathcal{D}(1^n)) \leq \frac{1}{49d}$$

for all sufficiently large $n \in \mathbb{N}$.

Claim 8.6. Let

$$\mathcal{S}_{\mathcal{D}}(1^n, 50cd) := \left\{x \in \{0,1\}^* : \Pr[x \leftarrow \mathcal{D}(1^n)] \geq \frac{1}{n^{50cd}}\right\}.$$

With probability at least $1 - \mathsf{negl}(n)$ over $\{X_i\}_{i\in[m(n)\cdot n^{1000cd}]} \leftarrow \mathcal{D}(1^n)^{\otimes m(n)\cdot n^{1000cd}}$, for any $X \in \mathcal{S}_{\mathcal{D}}(1^n, 50cd)$, we have

$$\left|\frac{\mathsf{Count}_{X_1,\dots,X_{m(n)\cdot n^{1000cd}}}(X)}{m(n)\cdot n^{1000cd}} - \Pr[X \leftarrow \mathcal{D}(1^n)]\right| \leq n^{-250cd}.$$

Claim 8.7. Let $\mathcal{D}$ be an arbitrary PPT algorithm, which takes 1^n as input, outputs $x \in \{0,1\}^{m(n)}$, and satisfies $H(\mathcal{D}(1^n)) \leq c\log(n)$. We define $\mathcal{S}_{\mathcal{D}}(1^n, d)$ as follows:

$$\mathcal{S}_{\mathcal{D}}(1^n, d) := \left\{x \in \{0,1\}^{m(n)} : \Pr[x \leftarrow \mathcal{D}(1^n)] \geq \frac{1}{n^d}\right\}.$$

Then, we have

$$\Pr_{x\leftarrow\mathcal{D}(1^n)}[x \in \mathcal{S}_{\mathcal{D}}(1^n, d)] \geq 1 - \frac{c}{d}.$$

For contradiction, suppose that

$$\Pr[\top \leftarrow \mathsf{Ver}(x_1, \dots, x_{n^{1000cd}}) : x_1, \dots, x_{n^{1000cd}} \leftarrow \mathcal{A}(1^n)] \geq 1 - \epsilon(n) + \frac{1}{d}.$$

From Claim 8.6, with probability at least $1 - \mathsf{negl}(n)$ over $\{X_i\}_{i\in[m(n)\cdot n^{1000cd}]} \leftarrow \mathcal{D}(1^n)^{\otimes m(n)\cdot n^{1000cd}}$, for any $X \in \mathcal{S}_{\mathcal{D}}(1^n, 50cd)$,

$$\left|\frac{\mathsf{Count}_{X_1,\dots,X_{m(n)\cdot n^{1000cd}}}(X)}{m(n)\cdot n^{1000cd}} - \Pr[X \leftarrow \mathcal{D}(1^n)]\right| \leq n^{-250cd}$$

and

$$\mathcal{S}_{\mathcal{D}}(1^n, 50cd) \subseteq \mathsf{List}_n.$$

This and Equation (6) imply that

$$\Pr\left[\left|\Pr\left[x \leftarrow \mathcal{A}^*_{x_1,\dots,x_{n^{1000cd}}}\right] - \Pr[x \leftarrow \mathcal{D}(1^n)]\right| \leq 2n^{-100cd} \text{ for all } x \in \mathcal{S}_{\mathcal{D}}(1^n, 50cd) : x_1, \dots, x_{n^{1000cd}} \leftarrow \mathcal{A}(1^n)\right]$$
$$\geq 1 - \epsilon(n) - \mathsf{negl}(n) + \frac{1}{d}.$$

From Claim 8.5, this implies that

$$\Pr\left[\Delta(\mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}},\mathcal{D}(1^n)) \leq \frac{1}{49d} : x_1,\ldots,x_{n^{1000cd}} \leftarrow \mathcal{A}(1^n)\right] \geq 1-\epsilon(n)-\mathsf{negl}(n)+\frac{1}{d}. \quad (7)$$

Let

$$\mathcal{T}_{n,d} := \left\{x_1,\ldots,x_{n^{1000cd}} \in \{0,1\}^* : \Delta(\mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}},\mathcal{D}(1^n)) \leq \frac{1}{49d}\right\}.$$

Then, Equation (7) implies that

$$\begin{aligned}
&\Delta\left(\mathsf{Marginal}_{\mathcal{A}}(1^n),\mathcal{D}(1^n)\right)\\
&\leq \mathbb{E}_{x_1,\ldots,x_{n^{1000cd}}\leftarrow\mathcal{A}(1^n)}\left[\Delta(\mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}},\mathcal{D}(1^n))\right]\\
&= \sum_{x_1,\ldots,x_{n^{1000cd}}} \Pr[x_1,\ldots,x_{n^{1000cd}} \leftarrow \mathcal{A}(1^n)]\Delta(\mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}},\mathcal{D}(1^n))\\
&= \sum_{x_1,\ldots,x_{n^{1000cd}}\in\mathcal{T}_{n,d}} \Pr[x_1,\ldots,x_{n^{1000cd}} \leftarrow \mathcal{A}(1^n)]\Delta(\mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}},\mathcal{D}(1^n))\\
&+ \sum_{x_1,\ldots x_{n^{1000cd}}\notin\mathcal{T}_{n,d}} \Pr[x_1,\ldots,x_{n^{1000cd}} \leftarrow \mathcal{A}(1^n)]\Delta(\mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}},\mathcal{D}(1^n))\\
&\leq \sum_{x_1,\ldots,x_{n^{1000cd}}\in\mathcal{T}_{n,d}} \Pr[x_1,\ldots,x_{n^{1000cd}} \leftarrow \mathcal{A}(1^n)]\frac{1}{49d} + \sum_{x_1,\ldots,x_{n^{1000cd}}\notin\mathcal{T}_{n,d}} \Pr[x_1,\ldots,x_{n^{1000cd}} \leftarrow \mathcal{A}(1^n)]\\
&\leq \frac{1}{49d}\left(1-\epsilon(n)-\mathsf{negl}(n)+\frac{1}{d}\right)+\epsilon(n)+\mathsf{negl}(n)-\frac{1}{d} \leq \epsilon(n)
\end{aligned}$$

for all sufficiently large $n \in \mathbb{N}$. Here, in the second inequality, we have used Equation (7). This is a contradiction.

Proof of Claim 8.5. We have

$$\begin{aligned}
&\Delta(A^*_{x_1,\ldots,x_{n^{1000cd}}},\mathcal{D}(1^n))\\
&= \frac{1}{2}\Bigg(\sum_{x\in\mathcal{S}_{\mathcal{D}}(1^n,50cd)}\left|\Pr\left[x \leftarrow \mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}}\right]-\Pr[x\leftarrow\mathcal{D}(1^n)]\right|\\
&+ \sum_{x\notin\mathcal{S}_{\mathcal{D}}(1^n,50cd)}\left|\Pr\left[x \leftarrow \mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}}\right]-\Pr[x\leftarrow\mathcal{D}(1^n)]\right|\Bigg)\\
&\leq \frac{1}{2}\Bigg(|\mathcal{S}_{\mathcal{D}}(1^n,50cd)|2n^{-100cd}\\
&+ \sum_{x\notin\mathcal{S}_{\mathcal{D}}(1^n,50cd)}\left|\Pr\left[x \leftarrow \mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}}\right]-\Pr[x\leftarrow\mathcal{D}(1^n)]\right|\Bigg)\\
&\leq n^{-50cd}+\frac{1}{2}\sum_{x\notin\mathcal{S}_{\mathcal{D}}(1^n,50cd)}\Pr\left[x \leftarrow \mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}}\right]+\frac{1}{2}\sum_{x\notin\mathcal{S}_{\mathcal{D}}(1^n,50cd)}\Pr[x\leftarrow\mathcal{D}(1^n)]\\
&\leq n^{-50cd}+\frac{1}{100d}+\frac{1}{100d}+n^{-50cd} \leq \frac{1}{49d}
\end{aligned}$$

for all sufficiently large $n \in \mathbb{N}$. Here, in the third inequality, we have used Claim 8.7, and

$$\sum_{x\in\mathcal{S}_{\mathcal{D}}(1^n,50cd)} \Pr\left[x \leftarrow \mathcal{A}^*_{x_1,\ldots,x_{n^{1000cd}}}\right] \geq \sum_{x\in\mathcal{S}_{\mathcal{D}}(1^n,50cd)} (\Pr[x \leftarrow \mathcal{D}(1^n)] - 2n^{-100cd})$$
$$\geq 1 - \frac{1}{50d} - 2n^{-50cd}.$$

□

Proof of Claim 8.6. From Hoeffding's inequality, for each $X \in \mathcal{S}_{\mathcal{D}}(1^n, 50cd)$, we have

$$\Pr_{x_1,\ldots,x_{m(n)\cdot n^{1000cd}}\leftarrow\mathcal{D}(1^n)^{\otimes m(n)\cdot n^{1000cd}}}\left[\left|\frac{\mathsf{Count}_{x_1,\ldots,x_{m(n)}}(X)}{m(n)\cdot n^{1000cd}} - \Pr[X \leftarrow \mathcal{D}(1^n)]\right| \leq \frac{1}{n^{250cd}}\right] \geq 1 - 2\exp\left(-n^{500cd}\right).$$

From union bound, we have

$$\Pr_{x_1,\ldots,x_{m(n)\cdot n^{1000cd}}\leftarrow\mathcal{D}(1^n)^{\otimes m(n)\cdot n^{1000cd}}}\left[\left|\frac{\mathsf{Count}_{x_1,\ldots,x_{m(n)}}(X)}{m(n)\cdot n^{1000cd}} - \Pr[X \leftarrow \mathcal{D}(1^n)]\right| \leq \frac{1}{n^{250cd}} \text{ for all } X \in \mathcal{S}_{\mathcal{D}}(1^n, 50cd)\right]$$
$$\geq \left(1 - 2\exp\left(-n^{500cd}\right)\right)^{n^{50cd}} \geq 1 - \mathsf{negl}(n).$$

□

Proof of Claim 8.7. This immediately holds from the definition of entropy as follows:

$$\begin{aligned} c\log(n) &\geq H(D(1^n)) \\ &= \sum_{x\in\mathcal{S}_{n,d}} p_x \log\left(\frac{1}{p_x}\right) + \sum_{x\notin\mathcal{S}_{n,d}} p_x \log\left(\frac{1}{p_x}\right) \\ &\geq \sum_{x\notin\mathcal{S}_{n,d}} p_x \log\left(\frac{1}{p_x}\right) \geq d\log(n) \sum_{x\notin\mathcal{S}_{n,d}} p_x. \end{aligned}$$

This implies that

$$\Pr_{x\leftarrow\mathcal{D}(1^n)}[x \in \mathcal{S}_{\mathcal{D}}(1^n, d)] \geq 1 - \frac{c}{d}.$$

□

□

Acknowledgements. BC acknowledges support of Royal Society University Research Fellowship URF\R1\211106 and EPSRC project EP/Z534158/1 on "Integrated Approach to Computational Complexity: Structure, Self-Reference and Lower Bounds". TM is supported by JST CREST JPMJCR23I3, JST Moonshot R&D JPMJMS2061-5-1-1, JST FOREST, MEXT QLEAP, the Grant-in Aid for Transformative Research Areas (A) 21H05183, and the Grant-in-Aid for Scientific Research (A) No.22H00522. This work was done in part while Matthew Gray and Eli Goldin were visiting the Simons Institute for the Theory of Computing.

Acknowledgments

References

[AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 333–342. ACM Press, June 2011. (Cited on page 1.)

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, October 2019. (Cited on page 4.)

[Aar14] Scott Aaronson. The equivalence of sampling and searching. *Theor. Comp. Sys.*, 55(2):281–298, August 2014. (Cited on page 7, 8, 12, 17.)

[AC17] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. CCC'17: Proceedings of the 32nd Computational Complexity Conference, 2017. (Cited on page 4.)

[AG20] Scott Aaronson and Sam Gunn. On the classical hardness of spoofing linear cross-entropy benchmarking, 2020. (Cited on page 4.)

[AQY22] Prabhanjan Ananth, Luowen Qian, and Henry Yuen. Cryptography from pseudorandom quantum states. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 208–236. Springer, Cham, August 2022. (Cited on page 6.)

[BCM+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th FOCS*, pages 320–331. IEEE Computer Society Press, October 2018. (Cited on page 1.)

[BCM+21] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. *J. ACM*, 68(5), August 2021. (Cited on page 6.)

[BFF+01] T. Batu, E. Fischer, L. Fortnow, R. Kumar, R. Rubinfeld, and P. White. Testing random variables for independence and identity. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 442–451, 2001. (Cited on page 1, 12.)

[BFKL94] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 278–291. Springer, Berlin, Heidelberg, August 1994. (Cited on page 13.)

[BFNV19] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. On the complexity and verification of quantum random circuit sampling. *Nature Physics*, 15:159–163, 2019. (Cited on page 1.)

[BFR+00] T. Batu, L. Fortnow, R. Rubinfeld, W.D. Smith, and P. White. Testing that distributions are close. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 259–269, 2000. (Cited on page 1, 12.)

[BIS+18] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595–600, April 2018. (Cited on page 4.)

[BJS11] Michael J. Bremner, Richard Jozsa, and Dan J. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467:459–472, 2011. (Cited on page 1.)

[BMS16] Michael J. Bremner, Ashley Montanaro, and Dan J. Shepherd. Average-case complexity versus approximate simulation of commuting quantum computations. *Physical Review Letters*, 117:080501, 2016. (Cited on page 1.)

[Can20] Clément L. Canonne. A survey on distribution testing: Your data is big. but is it blue? *Electron. Colloquium Comput. Complex.*, TR15, 2020. (Cited on page 2, 12.)

[CFSY23] Nai-Hui Chia, Honghao Fu, Fang Song, and Penghui Yao. A cryptographic perspective on the verifiability of quantum advantage. *CoRR*, abs/2310.14464, 2023. (Cited on page 13.)

[CGG24] Kai-Min Chung, Eli Goldin, and Matthew Gray. On central primitives for quantum cryptography with classical communication. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VII*, volume 14926 of *LNCS*, pages 215–248. Springer, Cham, August 2024. (Cited on page 6.)

[CGGH25] Bruno Pasqualotto Cavalar, Eli Goldin, Matthew Gray, and Peter Hall. A meta-complexity characterization of quantum cryptography. LNCS, pages 82–107. Springer, Cham, June 2025. (Cited on page 5, 11, 32.)

[CHK25] Suvradip Chakraborty, James Hulett, and Dakshita Khurana. On weak nizks, one-way functions and amplification. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology – CRYPTO 2025*, pages 580–610, Cham, 2025. Springer Nature Switzerland. (Cited on page 4, 10, 21, 33.)

[DKN15] Ilias Diakonikolas, Daniel M. Kane, and Vladimir Nikishkin. Testing identity of structured distributions. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, page 1841–1854, USA, 2015. Society for Industrial and Applied Mathematics. (Cited on page 1, 12.)

[FKM+18] Keisuke Fujii, Hirotada Kobayashi, Tomoyuki Morimae, Harumichi Nishimura, Seiichiro Tani, and Shuhei Tamate. Impossibility of classically simulating one-clean-qubit model with multiplicative error. *Physical Review Letters*, 120:200502, 2018. (Cited on page 1.)

[FR99] Lance Fortnow and John Rogers. Complexity limitations on quantum computation. *Journal of Computer and System Sciences*, 59(2):240–252, 1999. (Cited on page 9, 28.)

[Gác01] Peter Gács. Quantum algorithmic entropy. *Journal of Physics A: Mathematical and General*, 34(35):6859, August 2001. (Cited on page 18.)

[GKC+24] Xun Gao, Marcin Kalinowski, Chi-Ning Chou, Mikhail D. Lukin, Boaz Barak, and Soonwon Choi. Limitations of linear cross-entropy as a measure for quantum advantage. *PRX Quantum*, 5(1), February 2024. (Cited on page 4.)

[GPSV23] Shivam Garg, Chirag Pabbaraju, Kirankumar Shiragur, and Gregory Valiant. Testing with non-identically distributed samples, 2023. (Cited on page 12.)

[HHM25] Taiga Hiroka, Min-Hsiu Hsieh, and Tomoyuki Morimae. Hardness of quantum distribution learning and quantum cryptography, 2025. (Cited on page 13.)

[HKEG19] Dominik Hangleiter, Martin Kliesch, Jens Eisert, and Christian Gogolin. Sample complexity of device-independently certified "quantum supremacy". *Phys. Rev. Lett.*, 122:210502, May 2019. (Cited on page 1, 4, 12.)

[HKP20] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, June 2020. (Cited on page 6.)

[HM25] Taiga Hiroka and Tomoyuki Morimae. Quantum cryptography and meta-complexity. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology – CRYPTO 2025*, pages 545–574, Cham, 2025. Springer Nature Switzerland. (Cited on page 5, 11, 32.)

[HN23] Shuichi Hirahara and Mikito Nanashima. Learning in pessiland via inductive inference. In *64th FOCS*, pages 447–457. IEEE Computer Society Press, November 2023. (Cited on page 8, 10, 13, 15, 18, 22.)

[IL90] Russell Impagliazzo and Leonid A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *31st FOCS*, pages 812–821. IEEE Computer Society Press, October 1990. (Cited on page 4, 10, 21, 33.)

[IRS22] Rahul Ilango, Hanlin Ren, and Rahul Santhanam. Robustness of average-case meta-complexity via pseudorandomness. In Stefano Leonardi and Anupam Gupta, editors, *54th ACM STOC*, pages 1575–1583. ACM Press, June 2022. (Cited on page 4, 10, 13, 21, 33.)

[JLS18] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 126–152. Springer, Cham, August 2018. (Cited on page 6.)

[Kre21] W. Kretschmer. Quantum pseudorandomness and classical complexity. *TQC 2021*, 2021. (Cited on page 6.)

[KT24] Dakshita Khurana and Kabir Tomer. Commitments from quantum one-wayness. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *56th ACM STOC*, pages 968–978. ACM Press, June 2024. (Cited on page 6.)

[KT25] Dakshita Khurana and Kabir Tomer. Founding quantum cryptography on quantum advantage, or, towards cryptography from #p hardness. In *57th ACM STOC*, pages 178–188. ACM Press, June 2025. (Cited on page 5, 11, 32.)

[LP20] Yanyi Liu and Rafael Pass. On one-way functions and kolmogorov complexity. In *61st FOCS*, pages 1243–1254. IEEE Computer Society Press, November 2020. (Cited on page 13.)

[LP21] Yanyi Liu and Rafael Pass. On the possibility of basing cryptography on $\mathsf{EXP} \neq \mathsf{BPP}$. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 11–40, Virtual Event, August 2021. Springer, Cham. (Cited on page 13.)

[LV93] Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag, Berlin, Heidelberg, 1993. (Cited on page 8, 10, 15, 16, 18.)

[MSY25] Tomoyuki Morimae, Yuki Shirakawa, and Takashi Yamakawa. Cryptographic characterization of quantum advantage. In *57th ACM STOC*, pages 1863–1874. ACM Press, June 2025. (Cited on page 5, 30.)

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011. (Cited on page 26.)

[NRK+18] C. Neill, P. Roushan, K. Kechedzhi, S. Boixo, S. V. Isakov, V. Smelyanskiy, A. Megrant, B. Chiaro, A. Dunsworth, K. Arya, R. Barends, B. Burkett, Y. Chen, Z. Chen, A. Fowler, B. Foxen, M. Giustina, R. Graff, E. Jeffrey, T. Huang, J. Kelly, P. Klimov, E. Lucero, J. Mutus, M. Neeley, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, H. Neven, and J. M. Martinis. A blueprint for demonstrating quantum supremacy with superconducting qubits. *Science*, 360(6385):195–199, April 2018. (Cited on page 4.)

[OJF23] Changhun Oh, Liang Jiang, and Bill Fefferman. Spoofing cross-entropy measure in boson sampling. *Physical Review Letters*, 131(1), July 2023. (Cited on page 4.)

[Pan08] Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Transactions on Information Theory*, 54(10):4750–4755, 2008. (Cited on page 1, 12.)

[Sto83] Larry J. Stockmeyer. The complexity of approximate counting (preliminary version). In *15th ACM STOC*, pages 118–126. ACM Press, April 1983. (Cited on page 4.)

[TGB24] Andrew Tanggara, Mile Gu, and Kishor Bharti. Classically spoofing system linear cross entropy score benchmarking, 2024. (Cited on page 4.)

[VV17] Gregory Valiant and Paul Valiant. An automatic inequality prover and instance optimal identity testing. *SIAM Journal on Computing*, 46(1):429–455, 2017. (Cited on page 1, 12.)

A Proof of Theorem 2.7

Proof. Define the distribution $\mathcal{B}_n$ to be the distribution $\mathcal{G}_n$ conditioned on outputting elements in $A^{\mathcal{D}}_{n,s,\alpha}$.

For any $Y = (y_1, \ldots, y_s)$, define

$$p_Y := \Pr[\mathcal{D}_n^{\otimes s} \to Y]$$
$$q_Y := \Pr[\mathcal{G}_n \to Y]$$

and

$$\widetilde{q}_Y := \Pr[\mathcal{B}_n \to Y]$$

Theorem 2.5 tells us that for all Y,

$$uK^{t_0(t(n))}(Y|1^n) \leq \log_2 \frac{1}{q_Y} + |\mathcal{G}| + O(1). \tag{8}$$

Similarly, for all $Y \in A^{\mathcal{D}}_{n,s,\alpha}$, we have

$$\log_2 \frac{1}{p_{y_1} \cdots p_{y_s}} \leq uK^{t_0(t(n))}(Y|1^n) + \alpha(n). \tag{9}$$

And so Equations (8) and (9) together give that for all $Y \in A^{\mathcal{D}}_{n,s,\alpha}$,

$$\log_2 \frac{q_Y}{p_Y} \leq |\mathcal{G}| + O(1) + \alpha(n)$$

By Bayes law, for all Y,

$$\widetilde{q}_Y \leq \frac{q_Y}{1 - \epsilon(n)}$$

and so we also get

$$\log_2 \frac{\widetilde{q}_Y}{p_Y} \leq \log_2 \frac{1}{1-\epsilon(n)} + \log_2 \frac{q_Y}{p_Y} \leq \log_2 \frac{1}{1-\epsilon(n)} + |\mathcal{G}| + O(1) + \alpha(n)$$

We can explicitly bound $\Delta(\mathcal{B}_n, \mathcal{D}_n^{\otimes t})$ by working with KL-divergence.

$$\begin{aligned} D_{KL}(\mathcal{B}_n || \mathcal{D}_n^{\otimes s}) &= \sum_{Y \in [2^n]^s} \widetilde{q}_Y \log_2 \frac{\widetilde{q}_Y}{p_Y} \\ &\leq \max_{Y \in A^{\mathcal{D}}_{n,s,\delta}} \log_2 \frac{\widetilde{q}_Y}{p_Y} \\ &\leq \log_2 \frac{1}{1-\epsilon(n)} + |\mathcal{G}| + O(1) + \alpha(n) \end{aligned}$$

Let $\mathcal{B}_n^i$ be the marginal distribution of $\mathcal{B}_n$ restricted to the ith coordinate. We have

$$\sum_{i=1}^{s} D_{KL}(\mathcal{B}_n^i || \mathcal{D}_n) \leq D_{KL}(\mathcal{B}_n || \mathcal{D}_n^{\otimes s}) \leq \log_2 \frac{1}{1-\epsilon(n)} + |\mathcal{G}| + O(1) + \alpha(n).$$

From Pinsker's inequality, we have

$$\sum_{i=1}^{s} \frac{1}{2} \Delta(\mathcal{B}_n^i, \mathcal{D}_n)^2 \leq D_{KL}(\mathcal{B}_n || \mathcal{D}_n^{\otimes s}) \leq \log_2 \frac{1}{1-\epsilon(n)} + |\mathcal{G}| + O(1) + \alpha(n).$$

From Jensen's inequality, we have

$$\sum_{i=1}^{s}\Delta(\mathcal{B}_n^i,\mathcal{D}_n)\leq\sqrt{s\sum_{i=1}^{s}\Delta(\mathcal{B}_n^i,\mathcal{D}_n)^2}\leq\sqrt{2s\left(\log_2\frac{1}{1-\epsilon(n)}+|\mathcal{G}|+O(1)+\alpha(n)\right)}.$$

This implies that

$$\Delta(\mathsf{Marginal}_{\mathcal{B}}(1^n),\mathcal{D}_n)=\frac{1}{s}\sum_{i=1}^{s}\Delta(\mathcal{B}_n^i,\mathcal{D}_n)\leq\sqrt{\frac{2}{s}\left(\log_2\frac{1}{1-\epsilon(n)}+|\mathcal{G}|+O(1)+\alpha(n)\right)}.$$

From triangle inequality, we have

$$\begin{aligned}\Delta(\mathsf{Marginal}_{\mathcal{G}}(1^n),\mathcal{D}_n)&\leq\Delta(\mathsf{Marginal}_{\mathcal{G}}(1^n),\mathsf{Marginal}_{\mathcal{B}}(1^n))+\Delta(\mathsf{Marginal}_{\mathcal{B}},\mathcal{D}_n)\\&\leq\epsilon(n)+\sqrt{\frac{2}{s}\left(\log_2\frac{1}{1-\epsilon(n)}+|\mathcal{G}|+O(1)+\alpha(n)\right)}.\end{aligned}$$

□

B Strong Quantum Advantage Sampler from Quantum Cryptography

In this section, we prove the following Theorem B.1.

Theorem B.1. *If there exists a QPRG secure against QPT algorithms querying to an* $\mathbf{NP}$ *oracle, then a strong QAS exists.*

Here, a QPRG secure against QPT algorithms querying to an $\mathbf{NP}$ oracle is defined as follows.

Definition B.2 (Quantum Pseudorandom Generator secure against QPT algorithms querying to an $\mathbf{NP}$ oracle.). *Let* Gen *be a QPT algorithm that takes* 1^n *as input, and outputs* $x\in\{0,1\}^n$*. We say that* Gen *is a quantum pseudorandom generator (QPRG) secure against QPT algorithms querying to an* $\mathbf{NP}$ *oracle if the following holds:*

Statistically far:

$$\Delta(\mathsf{Gen}(1^n),U_n)\geq 1-\mathsf{negl}(n)$$

Computationally indistinguishable: *For any QPT adversary* $\mathcal{A}$ *querying to an* $\mathbf{NP}$ *oracle, we have*

$$\left|\Pr_{x\leftarrow\mathsf{Gen}(1^n)}[1\leftarrow\mathcal{A}^{\mathbf{NP}}(x)]-\Pr_{x\leftarrow U_n}[1\leftarrow\mathcal{A}^{\mathbf{NP}}(x)]\right|\leq\mathsf{negl}(n)$$

For showing Theorem B.1, we use the following Lemma B.3. Lemma B.3 directly follows from a standard padding argument and parallel repetition, and hence we omit the proof.

Lemma B.3. *Suppose that there exists a QPRG secure against QPT algorithms querying to* $\mathbf{NP}$ *oracle. Then, for any* $0<\tau<1$*, there exists a* $(1-2^{-n^\tau})$*-statistically-far QPRG* Gen^* *such that the following holds:*

$(1-2^{-n^\tau})$-statistically far:

$$\Delta(\mathsf{Gen}^*(1^n), U_x) \geq 1 - 2^{-n^\tau}$$

Computationally indistinguishable: *For any QPT adversary $\mathcal{A}$ querying to* $\mathbf{NP}$ *oracle, we have*

$$\left| \Pr_{x \leftarrow \mathsf{Gen}^*(1^n)}[1 \leftarrow \mathcal{A}^{\mathbf{NP}}(x)] - \Pr_{x \leftarrow U_n}[1 \leftarrow \mathcal{A}^{\mathbf{NP}}(x)] \right| \leq \mathsf{negl}(n)$$

Proof of Theorem B.1. From Lemma B.3, it is sufficient to construct a strong QAS from a $(1-2^{-n^\tau})$-statistically-far QPRG secure against QPT algorithms querying to an $\mathbf{NP}$ oracle for some $0 < \tau < 1$.

For contradiction, let us assume that there does not exist a strong QAS. Then, we show that there does not exist a $(1-2^{-n^\tau})$-statistically-far QPRG secure against QPT algorithms querying to an $\mathbf{NP}$ oracle for any $0 < \tau < 1$. More specifically, for any QPT algorithm $\mathcal{Q}$ and any constant $0 < \tau < 1$ such that

$$\Delta(\mathcal{Q}(1^n), U_n) \geq 1 - 2^{-n^\tau}$$

for all $n \in \mathbb{N}$, we construct a QPT algorithm $\mathcal{A}$ querying to an $\mathbf{NP}$ oracle such that

$$\left| \Pr_{x \leftarrow \mathcal{Q}(1^n)}[1 \leftarrow \mathcal{A}^{\mathbf{NP}}(x)] - \Pr_{x \leftarrow U_n}[1 \leftarrow \mathcal{A}^{\mathbf{NP}}(x)] \right| \geq n^{-\alpha}$$

for some constant $0 < \alpha < 1$ for infinitely many $n \in \mathbb{N}$ assuming the non-existence of strong QAS.

For showing this, we use the following Claims B.4 to B.6. We defer the proof of them.

Claim B.4. For any algorithm $\mathcal{Q}$ and any constant $0 < \tau < 1$ such that

$$\Delta(\mathcal{Q}(1^n), U_n) \geq 1 - 2^{-n^\tau}$$

for all sufficiently large $n \in \mathbb{N}$, we have

$$\Pr_{x \leftarrow \mathcal{Q}(1^n)}[\Pr[x \leftarrow \mathcal{Q}(1^n)] \geq 2^{-n+\frac{n^\tau}{2}}] \geq 1 - 2 \cdot 2^{\frac{-n^\tau}{2}}$$

for all sufficiently large $n \in \mathbb{N}$.

Claim B.5. For any constant $c > 0$, and any algorithm $\mathcal{Q}$ and $\mathcal{C}$ such that

$$\Delta(\mathcal{Q}(1^n), \mathcal{C}(1^n)) \leq 1 - n^{-c}$$

for all sufficiently large $n \in \mathbb{N}$, we have

$$\Pr_{x \leftarrow \mathcal{Q}(1^n)}\left[\frac{1}{n^{2c}} \Pr[x \leftarrow \mathcal{Q}(1^n)] \leq \Pr[x \leftarrow \mathcal{C}(1^n)]\right] \geq \frac{n^{-c}}{2}$$

for all sufficiently large $n \in \mathbb{N}$.

Claim B.6. For any PPT algorithm $\mathcal{D}$ and any $c > 0$, there exists a PPT algorithm $\mathsf{Estimate}$ querying to an $\mathbf{NP}$ oracle such that

$$\Pr\left[\frac{1}{2} \Pr[x \leftarrow \mathcal{D}(1^n)] \leq \mathsf{Estimate}^{\mathbf{NP}}(1^n, x) \leq \Pr[x \leftarrow \mathcal{D}(1^n)]\right] \geq 1 - n^{-c}$$

for all $x \in \{0,1\}^*$ and all sufficiently large $n \in \mathbb{N}$.

From the non-existence of strong QAS, there exists a PPT algorithm $\mathcal{C}$ and a constant $c > 0$ such that

$$\Delta(\mathcal{Q}(1^n), \mathcal{C}(1^n)) \leq 1 - n^{-c}$$

for infinitely many $n \in \mathbb{N}$. From Claim B.6, there exists a PPT algorithm $\mathbf{Estimate}$ querying to an $\mathbf{NP}$ oracle such that

$$\Pr\left[\frac{1}{2}\Pr[x \leftarrow \mathcal{C}(1^n)] \leq \mathsf{Estimate}^{\mathbf{NP}}(1^n, x) \leq \Pr[x \leftarrow \mathcal{C}(1^n)]\right] \geq 1 - n^{-2c}$$

for all $x \in \{0,1\}^*$ and all sufficiently large $n \in \mathbb{N}$.

Now, we describe $\mathcal{A}$.

The description of $\mathcal{A}^{\mathbf{NP}}$:

1. Receive $x \in \{0,1\}^n$.
2. Run $p \leftarrow \mathsf{Estimate}^{\mathbf{NP}}(x)$.
3. If $p \geq n^{-2c} \cdot 2^{-n+\frac{n^\tau}{2}-1}$, output 1. Otherwise, sample $b \leftarrow \{0,1\}$ and output b.

Let us define $\mathcal{S}_n(A)$.

$$\mathcal{S}_n(A) := \{x \in \{0,1\}^n : \Pr[x \leftarrow \mathcal{C}(1^n)] \geq A\}.$$

We have

$$\Pr_{x \leftarrow U_n}[x \in \mathcal{S}_n(A)] = \sum_{x \in \mathcal{S}_n(A)} 2^{-n} \leq |\mathcal{S}_n(A)| 2^{-n} \leq \frac{2^{-n}}{A}.$$

Furthermore, from Claims B.4 and B.5 and union bound, we have

$$\Pr_{x \leftarrow \mathcal{Q}(1^n)}[x \in \mathcal{S}_n(n^{-2c} \cdot 2^{-n+\frac{n^\tau}{2}})] \geq \frac{n^{-c}}{2} - 2 \cdot 2^{-\frac{n^\tau}{2}} \geq \frac{49}{100} n^{-c}$$

for all sufficiently large $n \in \mathbb{N}$. Hence,

$$\begin{aligned}
&\Pr_{x \leftarrow \mathcal{Q}(1^n)}[1 \leftarrow \mathcal{A}^{\mathbf{NP}}(x)] - \Pr_{x \leftarrow U_n}[1 \leftarrow \mathcal{A}^{\mathbf{NP}}(x)] \\
&= \sum_{x \in \mathcal{S}_n(n^{-2c} \cdot 2^{-n+\frac{n^\tau}{2}})} \Pr\left[1 \leftarrow \mathcal{A}^{\mathbf{NP}}(x)\right] \Pr[x \leftarrow \mathcal{Q}(1^n)] + \sum_{x \notin \mathcal{S}_n(n^{-2c} \cdot 2^{-n+\frac{n^\tau}{2}})} \Pr\left[1 \leftarrow \mathcal{A}^{\mathbf{NP}}(x)\right] \Pr[x \leftarrow \mathcal{Q}(1^n)] \\
&- \left(\sum_{x \in \mathcal{S}_n(n^{-2c} \cdot 2^{-n+\frac{n^\tau}{2}-2})} \Pr\left[1 \leftarrow \mathcal{A}^{\mathbf{NP}}(x)\right] \Pr[x \leftarrow U_n] + \sum_{x \notin \mathcal{S}_n(n^{-2c} \cdot 2^{-n+\frac{n^\tau}{2}-2})} \Pr\left[1 \leftarrow \mathcal{A}^{\mathbf{NP}}(x)\right] \Pr[x \leftarrow U_n] \right) \\
&\geq \sum_{x \in \mathcal{S}_n(n^{-2c} \cdot 2^{-n+\frac{n^\tau}{2}})} (1 - n^{-2c}) \Pr[x \leftarrow \mathcal{Q}(1^n)] + \sum_{x \notin \mathcal{S}_n(n^{-2c} \cdot 2^{-n+\frac{n^\tau}{2}})} \frac{1}{2} \Pr[x \leftarrow \mathcal{Q}(1^n)] \\
&- \left(\sum_{x \in \mathcal{S}_n(n^{-2c} \cdot 2^{-n+\frac{n^\tau}{2}-2})} \Pr[x \leftarrow U_n] + \sum_{x \notin \mathcal{S}_n(n^{-2c} \cdot 2^{-n+\frac{n^\tau}{2}-2})} \left(\frac{1}{2} + \frac{n^{-2c}}{2}\right) \Pr[x \leftarrow U_n] \right) \\
&\geq \left(1 - n^{-2c}\right) \cdot \frac{49}{100} n^{-c} + \frac{1}{2}\left(1 - \frac{49}{100} n^{-c}\right) - \left(4n^{2c} \cdot 2^{\frac{-n^\tau}{2}} + \left(\frac{1}{2} + \frac{n^{-2c}}{2}\right)(1 - 4n^{2c} \cdot 2^{\frac{-n^\tau}{2}})\right) \\
&\geq \frac{1}{2}\left(1 + \frac{49}{200} n^{-c}\right) - \left(\frac{1}{2} + \frac{n^{-2c}}{2}\right)(1 + \mathsf{negl}(n)) \geq \frac{49}{400} n^{-2c} - \mathsf{negl}(n)
\end{aligned}$$

for all sufficiently large $n \in \mathbb{N}$, which is a contradiction to the security of $\mathcal{Q}$. Here, in the first inequality, we have used that

$$\Pr\left[1 \leftarrow \mathcal{A}^{\mathbf{NP}}(x)\right] \geq 1 - n^{-2c}$$

for all $x \in \mathcal{S}(n^{-2c} \cdot 2^{-n+n^{\frac{\tau}{2}}})$ and

$$\Pr\left[1 \leftarrow \mathcal{A}^{\mathbf{NP}}(x)\right] \leq \frac{1}{2} + \frac{n^{-2c}}{2}$$

for all $x \notin \mathcal{S}(n^{-2c} \cdot 2^{-n+n^{\frac{\tau}{2}}-2})$.

Proof of Claim B.4. We define the following sets.

$$\begin{aligned}
A &:= \{x \in \{0,1\}^n : \Pr[x \leftarrow \mathcal{Q}(1^n)] < 2^{-n}\} \\
B &:= \{x \in \{0,1\}^n : 2^{-n} \leq \Pr[x \leftarrow \mathcal{Q}(1^n)] < 2^{-n+\frac{n^\tau}{2}}\} \\
C &:= \{x \in \{0,1\}^n : 2^{-n+\frac{n^\tau}{2}} \leq \Pr[x \leftarrow \mathcal{Q}(1^n)] \leq 1\}.
\end{aligned}$$

From the definition of $\mathcal{Q}$ and total variation distance, we have

$$\begin{aligned}
1 - 2^{-n^\tau} &\leq \sum_{x \in A} \Pr[x \leftarrow \{0,1\}^n] - \Pr[x \leftarrow \mathcal{Q}(1^n)] \\
&= \sum_{x \in A} 2^{-n} - \Pr[x \leftarrow \mathcal{Q}(1^n)] \\
&\leq |A| 2^{-n}.
\end{aligned}$$

This implies that

$$|B| + |C| \leq 2^{n-n^\tau}.$$

Furthermore, we have

$$\begin{aligned}
1 - 2^{-n^\tau} &\leq \sum_{x \in B} (\Pr[x \leftarrow \mathcal{Q}(1^n)] - 2^{-n}) + \sum_{x \in C} (\Pr[x \leftarrow \mathcal{Q}(1^n)] - 2^{-n}) \\
&\leq \sum_{x \in B} 2^{-n+\frac{n^\tau}{2}} + \sum_{x \in C} \Pr[x \leftarrow \mathcal{Q}(1^n)] \\
&\leq |B| \cdot 2^{-n+\frac{n^\tau}{2}} + \sum_{x \in C} \Pr[x \leftarrow \mathcal{Q}(1^n)].
\end{aligned}$$

This implies that

$$1 - 2 \cdot 2^{-\frac{n^\tau}{2}} \leq \sum_{x \in C} \Pr[x \leftarrow \mathcal{Q}(1^n)].$$

□

Proof of Claim B.5. Let us define

$$A := \left\{ x \in \{0,1\}^n : \frac{1}{n^{2c}} \Pr[x \leftarrow \mathcal{Q}(1^n)] \geq \Pr[x \leftarrow \mathcal{D}(1^n)] \right\}.$$

Then, we have

$$\sum_{x \in A} \Pr[x \leftarrow \mathcal{D}(1^n)] \leq \sum_{x \in A} \frac{1}{n^{2c}} \Pr[x \leftarrow \mathcal{Q}(1^n)]$$

$$\sum_{x \in A} \Pr[x \leftarrow \mathcal{D}(1^n)] - \Pr[x \leftarrow \mathcal{Q}(1^n)] \leq \sum_{x \in A} \left(\frac{1}{n^{2c}} - 1 \right) \Pr[x \leftarrow \mathcal{Q}(1^n)]$$

This implies that

$$\left(1 - \frac{1}{n^{2c}}\right) \sum_{x \in A} \Pr[x \leftarrow \mathcal{Q}(1^n)] \leq \sum_{x \in A} \Pr[x \leftarrow \mathcal{Q}(1^n)] - \Pr[x \leftarrow \mathcal{D}(1^n)] \leq \Delta(\mathcal{D}(1^n), \mathcal{Q}(1^n)).$$

Therefore, we have

$$\sum_{x \in A} \Pr[x \leftarrow \mathcal{Q}(1^n)] \leq \frac{1 - n^{-c}}{1 - n^{-2c}} \leq 1 - \frac{n^{-c}}{2}.$$

This implies that

$$\Pr_{x \leftarrow \mathcal{Q}(1^n)} \left[\frac{1}{n^{2c}} \Pr[x \leftarrow \mathcal{Q}(1^n)] \leq \Pr[x \leftarrow \mathcal{C}(1^n)] \right] \geq \frac{n^{-c}}{2}.$$ □

Proof of Claim B.6. Claim B.6 directly follows from Theorem 5.2. □

□

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il