arXiv:2611.05303v1 [quant-ph] 3 Nov 2026

Quantum Advantage has Few Complexity Implications

Eli Goldin
Princeton University

Alex Lombardi
Princeton University

Tomoyuki Morimae
Yukawa Institute for Theoretical Physics
Kyoto University

November 3, 2026

Abstract

We consider the complexity implications of proofs of quantumness and (quantum verifier) oblivious state preparation. We show that both of these objects can coexist with the easiness of nondeterministic computation. That is, we give evidence that proofs of quantumness can exist when $\mathsf{P} = \mathsf{NP}$ and that (quantum verifier) oblivious state preparation can exist when $\mathsf{BQP} = \mathsf{QCMA}$.

In particular, we give a classical oracle relative to which

1. $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$
2. $\mathsf{BQP}^{\mathcal{O}} = \mathsf{PSPACE}^{\mathcal{O}}$
3. Proofs of quantumness exist. In particular, there exists a language in $\mathsf{IP}^{\mathcal{O}} \subseteq \mathsf{BQP}^{\mathcal{O}}$ which is hard on average against $\mathsf{P}^{\mathcal{O}}/poly$.

We give a unitary oracle such that

1. $\mathsf{BQP}^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}}$
2. Quantum verifier oblivious state preparation exists.

These oracle separations directly imply a number of related consequences. For example, our first result implies that proofs of quantumness cannot be used to construct classical or quantum cryptography with a relativizing proof. Our second result implies that quantum computation/classical communication oblivious transfer cannot be used to construct post-quantum one-way functions with a relativizing proof.

Contents

1 Introduction **1**
- 1.1 Our Results and Corollaries . . . 2
- 1.2 Related Works . . . 4

2 Technical Overview **5**
- 2.1 A classical oracle relative to which $\mathsf{P} = \mathsf{NP}$ but proofs of quantumness exist . . . 5
- 2.2 An oracle relative to which $\mathsf{BQP} = \mathsf{QCMA}$ but oblivious state preparation exists . . . 8

3 Preliminaries **11**
- 3.1 Notation . . . 11
- 3.2 Complexity Preliminaries . . . 12
- 3.3 Techniques for Simulating Oracles . . . 15
- 3.4 Cryptographic Preliminaries . . . 19

4 A classical oracle with quantum advantage but no cryptography **20**
- 4.1 Our Oracle . . . 21
- 4.2 Complexity classes collapse under our oracle . . . 22
- 4.3 Representing oracle Turing machines as circuits . . . 24
- 4.4 Forrelation is useless for classical computation . . . 24
- 4.5 Majority is average case hard . . . 26
- 4.6 Applications . . . 30

5 A quantum oracle where $\mathsf{BQP} = \mathsf{QCMA}$ **33**
- 5.1 The oracle . . . 33
- 5.2 Simulating the oracle with random oracle points . . . 34
- 5.3 Showing $\mathsf{BQP}^{\mathcal{O},\mathsf{PSPACE}} = \mathsf{QCMA}^{\mathcal{O},\mathsf{PSPACE}}$ . . . 41
- 5.4 Building oblivious state preparation . . . 42

6 AI Disclosure **46**

A From state preparation channels to unitaries **52**

B EV-OWPuzzs from OSP **53**
- B.1 Commitments and the implication from OSP . . . 53
- B.2 The transcript determines the committed bit . . . 54
- B.3 Construction and proof of security . . . 56

1 Introduction

Quantum advantage, the ability of a quantum computer to perform a task that no efficient classical computer can, is the defining promise of quantum computation. For this promise to be checkable, the task should admit efficient classical verification. Sampling output distributions of quantum computers is believed to be classically hard [AA11, BJS11], but it is not efficiently verifiable. In contrast, a *proof of quantumness* (PoQ) [BCM+18] is an interactive protocol between a probabilistic polynomial-time (PPT) verifier and a prover such that an honest quantum polynomial-time (QPT) prover makes the verifier accept with high probability, while no PPT prover can.

In this paper, we study PoQ as well as both stronger and weaker forms of quantum advantage. On the stronger side, we consider the notion of oblivious state preparation (OSP) [BK25]. OSP is a two-party protocol with classical communication in which a classical verifier can remotely generate a quantum state in the prover's register in such a way that only the verifier knows its classical description. This can be seen as a very strong form of quantum advantage: a prover who can obliviously prepare arbitrary states can demonstrate quantum advantage by giving the result of a related quantum measurement. In particular, this is a form of advantage which retains some notion of hardness even against quantum adversaries.

Beyond the fact that OSP can be used to construct PoQ [BK25], it has a number of other powerful applications. For example, OSP can be used to construct (single-qubit) proofs of quantum memory (PoQM) [BK25, HMOY25]. A proof of quantum memory is a proof of quantumness where not only is a classical prover unable to make the verifier accept, but even a quantum prover with insufficient quantum memory cannot succeed. This means that an honest prover demonstrates the ability to maintain quantum memory of a specified size for a specified period during the protocol. In addition, OSP can be used to construct a powerful quantum cryptographic object known as search-secure quantum-computation-classical-communication (QCCC) oblivious transfer [BK25].

On the weaker side, we consider quantum-verifier proofs of quantumness (qv-PoQ) and quantum-verifier oblivious state preparation (qv-OSP), where the honest verifier in the protocol also has quantum capabilities. Note that these are particularly interesting in the latter setting, where the adversaries are also allowed to be quantum. In particular, as the corresponding protocols allow for quantum verification, qv-OSP is sufficient to instantiate the constructions of (quantumly verifiable) PoQM as well as search-secure QCCC oblivious transfer.

Constructions of PoQ and OSP. Since unconditionally sound proofs of quantumness are not known to exist (indeed, there is a complexity theoretic barrier, see below), PoQ are constructed and proved sound under computational hardness assumptions; this means that the quantum advantage is "lost" if the assumptions turn out to be false. The simplest PoQs arise from problems in $\mathsf{NP} \cap \mathsf{BQP}$ that are believed to be classically hard on average. The verifier samples an instance of the problem and the prover produces a witness. Examples of this formula include factoring and discrete log [Sho94]. Proofs of quantumness can also be built from cryptographic assumptions such as the learning with errors (LWE) assumption [BCM+18], quantum fully homomorphic encryption [KLVY23], and indistinguishability obfuscation (iO) plus one-way functions (OWFs) [MY23, SZ25]. Post-quantum versions of the above listed assumptions can all also be used to achieve OSP through similar constructions [BK25, SZ25, MY23, CWX26]. Unlike the $\mathsf{NP} \cap \mathsf{BQP}$ style of construction, these protocols are inherently interactive.

As far as the authors are aware, all known candidate constructions of PoQ rely on assumptions

which themselves imply that classically-secure one-way functions exist, and so $\mathsf{NP} \not\subset \mathsf{BPP}$. OSP itself directly implies that $\mathsf{BQP} \neq \mathsf{QCMA}$ (see Appendix B) in the classical verifier setting. This implication is not known to hold for qv-OSP.

Known complexity-theoretic/cryptographic lower bounds Despite the fact that all constructions of PoQ imply $\mathsf{NP} \not\subset \mathsf{BPP}$, it is open whether such an implication holds for PoQ generically. Likewise, it is not known whether the existence of qv-OSP implies $\mathsf{BQP} \neq \mathsf{QCMA}$. Nevertheless, complexity theoretic lower bounds are known for both primitives. It is known that if $\mathsf{BPP} = \mathsf{PP}$, then PoQ cannot exist [MSY25, GMMY24]. Intuitively, this is because a classical prover with access to a PP oracle can simulate arbitrary quantum computation, even conditioned on partial outputs. Thus, a PPT algorithm can simulate the honest quantum prover. Similarly, if $\mathsf{BQP} = \mathsf{PP}$, then qv-OSP cannot exist [BK25, HMOY25, GMMY24].

We remark that for PoQ, stronger complexity lower bounds are known in restricted settings. If $\mathsf{BPP} = \mathsf{NP}$, then *constant-round* PoQ cannot exist[1]. Furthermore, if one assumes a particular structure on the proof of security of the PoQ (namely that it is a black-box reduction to a quantumly-secure falsifiable assumption), then it is necessary that $\mathsf{BQP} \neq \mathsf{PP}$ [TZ26, GMMY24].

Conjectural lower bounds In this work, we investigate the following conjectural lower bounds on PoQ and qv-OSP.

Conjecture 1.1. *If PoQ exist, then* $\mathsf{BPP} \neq \mathsf{NP}$.

Conjecture 1.2. *If qv-OSP exist, then* $\mathsf{BQP} \neq \mathsf{QCMA}$.

In particular, we will give oracle evidence that neither of these conjectures hold. This opens up the door to constructing PoQ and qv-OSP from much weaker assumptions than previously expected.

Our oracle separations are significantly stronger than simply resolving the above conjectures, and thus imply a series of independently interesting related separations.

1.1 Our Results and Corollaries

PoQ can exist even if $\mathsf{BPP} = \mathsf{NP}$. We construct a classical oracle relative to which proofs of quantumness exist, but $\mathsf{BPP} = \mathsf{NP}$. Relative to our oracle, it will also be the case that $\mathsf{BQP} = \mathsf{PSPACE}$. In particular, we show

Theorem 1.3. *There exists a classical oracle relative to which*

1. $\mathsf{P} = \mathsf{NP}$
2. $\mathsf{BQP} = \mathsf{PSPACE}$
3. *There is a problem in* $\mathsf{IP} \subseteq \mathsf{BQP}$ *which is hard-on-average for* $\mathsf{P/poly}$.
4. *a polynomial-round public-coin (publicly-verifiable) PoQ exists*
5. *a two-message qv-PoQ exists*

[1] We assumed this was a known result, but were unable to find a citation. As such, we have included a proof in Corollary 4.16.

Note that nearly all forms of classical cryptography (in particular, everything implying one-way functions) cannot exist if $\mathsf{P} = \mathsf{NP}$. Similarly, nearly all forms of quantum cryptography (in particular, everything implying quantum commitments) cannot exist if $\mathsf{BQP} = \mathsf{PSPACE}$.[2]

Corollary 1.4. *There is no relativizing construction of (most forms of) classical or quantum cryptographic primitives from polynomial-round PoQ or two-message qv-PoQ.*

Notably, there is no relativizing construction of OSP or PoQM from PoQ, even in the setting of inefficient verification.

As $\mathsf{P} = \mathsf{NP}$, constant-round proofs of quantumness similarly do not exist. This gives a barrier towards round reduction for PoQ as well as a partial barrier for constructing classically verifiable PoQ from qv-PoQ.

Corollary 1.5. *For every fixed constant r, there is no relativizing construction of an r-round PoQ from polynomial-round PoQ or two-message qv-PoQ.*

We remark that for fully black-box constructions, a negative answer for round-reduction is implicit in prior work. By combining two papers [MY23, SZ25] one can construct polynomial-round PoQ from indistinguishability obfuscation and one-way functions. On the other hand, Tomer and Zhandry [TZ26] rule out fully black-box constructions of constant-round PoQ from these same primitives.

We also get a partial negative answer to the important open question of whether PoQ implies classical verification of quantum computation (CVQC) [Mah18]. In particular, if a constant-round CVQC protocol existed relative to our oracle, then applying the CVQC to the hard problem for $\mathsf{P}/poly$ gives a constant-round PoQ, which does not exist relative to our oracle.

Corollary 1.6. *For every fixed constant r, there is no relativizing construction of an r-round CVQC from polynomial-round PoQ or two-message qv-PoQ.*

We remark that the way we prove that PoQ exists relative to our oracle is by demonstrating a hard problem in $\mathsf{BQP}^{\mathcal{O}}$ which has an interactive proof whose prover is implementable in $\mathsf{BQP}^{\mathcal{O}}$. Extracting this fact from our proof gives the following independently interesting corollary.

Corollary 1.7. *Relative to the oracle $\mathcal{O}$ from Theorem 1.3, there exists a language $L^{\mathcal{O}}$ such that*

1. *There is a classically verifiable, quantum computable interactive proof for $L^{\mathcal{O}}$.*
2. *$L^{\mathcal{O}}$ is hard on average for $\mathsf{P}^{\mathcal{O}}/poly$*

In particular, since $\mathsf{PH}^{\mathcal{O}} \subseteq \mathsf{P}^{\mathcal{O}}/poly$ this gives an oracle relative to which there is a problem outside of PH which can be decided with classical verification of quantum computation.

qv-OSP can exist even if $\mathsf{BQP} = \mathsf{QCMA}$ We construct a quantum oracle relative to which qv-OSP exists, but $\mathsf{BQP} = \mathsf{QCMA}$.

Theorem 1.8. *There exists a quantum oracle relative to which*

1. $\mathsf{BQP} = \mathsf{QCMA}$

[2] This follows from a PSPACE-effective variant of OpenAI's unitary synthesis algorithm [Ope26].

2. *a four message qv-OSP exists*

The question of $\mathsf{BQP} = \mathsf{QCMA}$ is also closely tied to a quantum cryptographic primitive known as efficiently-verifiable one-way puzzles (EV-OWPuzz) [CGG24]. The existence of EV-OWPuzzs implies $\mathsf{BQP} \neq \mathsf{QCMA}$. Furthermore, EV-OWPuzzs are implied by a wide variety of quantum cryptographic primitives, including all of post-quantum cryptography.

Goldin, Morimae, Mutreja, and Yamakawa [GMMY24] previously showed that there is no relativizing construction of EV-OWPuzzs from either QCCC commitments or QCCC key exchange. Since qv-OSP implies both of these primitives, Corollary 1.9 strengthens these separations by showing that even qv-OSP does not suffice for a relativizing construction of EV-OWPuzzs.

Note that many quantum cryptographic primitives are known to imply EV-OWPuzzs [CGG24]. Selecting interesting examples from this set, we get the following corollary

Corollary 1.9. *For all A in*

1. *EV-OWPuzzs*
2. *post-quantum one-way functions*
3. *classically implementable oblivious transfer*
4. *quantum public key encryption with classical keys*
5. *classically verifiable OSP*

for all B in

1. *four message qv-OSP*
2. *qv-PoQM*
3. *search-secure QCCC oblivious transfer*

there is no relativizing construction of A from B.

Note that the fact that classically verifiable OSP implies EV-OWPuzzs is new. For details, see Appendix B.

1.2 Related Works

[MSY25] showed that PoQ imply either $\mathsf{NP} \not\subseteq \mathsf{BQP}$ or $\mathsf{SampBQP} \neq \mathsf{SampBPP}$. A natural question is whether the first alternative alone is necessary: does PoQ imply $\mathsf{NP} \not\subseteq \mathsf{BQP}$? Our first result provides an oracle counterexample to this stronger implication, because relative to our oracle, PoQ exist but $\mathsf{NP} = \mathsf{P} \subseteq \mathsf{BQP}$. Thus, the disjunctive necessary condition of [MSY25] cannot be strengthened to $\mathsf{NP} \not\subseteq \mathsf{BQP}$ by a relativizing argument.

[TZ26] showed that constant-round PoQ imply either two-message PoQ, quantumly-secure OWFs, or (weak) signature token schemes. They also show that if polynomial-round PoQ can be constructed from a quantumly-secure falsifiable assumption in a black-box way then quantumly-secure OWPuzz exist. Although these results are the first derivations of quantum cryptographic primitives from PoQ, the former is disjunctive and the latter is the derivation from the way of constructing PoQ, not PoQ itself.

The concurrent work [Man26] showed that relative to a classical oracle there is a language L such that $L \in \mathsf{EQP} \setminus \mathsf{PH}$ and L has an interactive proof system with a PPT verifier and an honest QPT prover. Our results yield an analogous phenomenon for BQP, while additionally providing a PoQ secure against nonuniform classical provers in a world where $\mathsf{P} = \mathsf{NP}$ and $\mathsf{BQP} = \mathsf{PSPACE}$.

[BHN+26] showed a classical oracle relative to which $\mathsf{BQP} \not\subseteq \mathsf{IP}$, establishing a relativization barrier to information-theoretically sound, single-prover classical verification of quantum computation. Our result is complementary: we construct a classical oracle relative to which $\mathsf{P} = \mathsf{NP}$ and $\mathsf{BQP} = \mathsf{PSPACE}$, yet polynomial-round proofs of quantumness with efficient classical verification exist.

2 Technical Overview

2.1 A classical oracle relative to which $\mathsf{P} = \mathsf{NP}$ but proofs of quantumness exist

We begin with an overview of Theorem 1.3. Prior work on quantum advantage relative to oracles [Sim97, BBBV97, CCD+03, Aar10, AA15, RT19, AIK22, YZ22, BHN+26] typically proceeds in two steps:

- Identify an oracle $\mathcal{A}$ relative to which a non-trivial quantum algorithm solves an apparently challenging computational problem.
- Prove (or conjecture) that classical algorithms relative to that oracle – potentially augmented by a more powerful oracle such as $\mathsf{PH}^{\mathcal{A}}$ – cannot solve the task.

One notable extension of this methodology is an approach of Aaronson, Ingram, and Kretschmer [AIK22], who use a so-called *Forrelation encoding* to embed the answers to powerful (in their case, $\mathsf{P}^{\#\mathsf{P}}$) computations into an oracle in a form that quantum algorithms can efficiently decode, while the encoding looks random to algorithms in the polynomial hierarchy.

Our methodology: prover-efficient interactive proofs. Our proof relies on this idea from [AIK22] but diverges from the above high-level template. We construct a proof of quantumness by giving an efficient prover interactive proof for a decision problem that is hard for BPP relative to our oracle. That is, the honest prover in our interactive proof can be implemented with a QPT algorithm, and soundness of the interactive proof is information-theoretic.

Let us see under what precise conditions this suffices to get a proof of quantumness. As a warm-up, first suppose that we only want a protocol with soundness error of $1/2 + \epsilon$ against *uniform i.o.*BPP adversaries. Then, it suffices to work with a *unary* language $L \subseteq \{1\}^*$ outside $\mathsf{BPP}^{\mathcal{O}}$. To see this, suppose that we have such a language where for each bit b, the claim $b = \mathbf{1}_L(1^n)$ has an interactive proof with a QPT honest prover. Then, one gets a proof of quantumness in which the honest quantum prover (on input 1^n) send the bit b declaring whether $1^n \in L$ and then gives an interactive proof for this fact. The protocol is complete by construction, and satisfies soundness (against classical randomized adversaries) because, by the information theoretical soundness of the interactive proof, any adversary passing the PoQ is deciding membership in L.

To achieve a full-fledged protocol with (standard) soundness against $\mathsf{P/poly}$, we instead work with a (binary) language L that is *non-uniformly hard on average* for a uniformly sampled input x. In this case, the proof of quantumness is as follows:

- The verifier samples a uniformly random input x.
- The prover decides whether $x \in L$, sending the corresponding bit b to the verifier.
- The prover and verifier execute the interactive proof for $x \in L_b$ (either L or $\overline{L}$).

Again, soundness follows because a prover passing the PoQ is correctly deciding whether a random $x \in L$ by the soundness of the interactive proof.

Constructing an appropriate oracle with a hard unary language. Next, we describe a full solution to the warm-up problem. As mentioned earlier, we deviate from the template of most quantum advantage oracle separations. Why? Because relative to most of the oracles considered in prior work, it is not clear that the problems separating BQP from classical computation admit efficient-prover interactive proofs. It is actually not clear that they admit interactive proofs of any kind![3] Indeed, relative to a random oracle it is known that even simple classes like coNP do not have interactive proofs [FS88, CCG+94], while a recent work [BHN+26] proves the same for a variant of the Forrelation problem.

In other words, natural choices of oracle such as a random oracle or an oracle for the Forrelation problem do not seem "IP-friendly." On the other hand, we know that in the *unrelativized* setting, $\mathsf{IP} = \mathsf{PSPACE}$, so this problem arises because this result (famously) does not relativize.

Our solution is to use Algebrization [AW08] and build our oracle by the following process

- First, construct an oracle $\mathcal{A}$ relative to which $\mathsf{IP} = \mathsf{PSPACE} \supsetneq \mathsf{PH}$, so that there are problems outside of the polynomial hierarchy with interactive proofs.
- Then, introduce an oracle $\mathcal{B}$ so that $\mathsf{PSPACE}^{\mathcal{A}} \subset \mathsf{BQP}^{\mathcal{B}}$. This means that every problem in $\mathsf{PSPACE}^{\mathcal{A}}$ has an interactive proof *with a* $\mathsf{BQP}^{\mathcal{B}}$-efficient prover!
- Finally, prove that there are still problems in $\mathsf{PSPACE}^{\mathcal{A}}$ that are hard relative to $\mathsf{PH}^{\mathcal{A},\mathcal{B}}$.

Then, by appending an appropriately defined "$\mathsf{PH}^{\mathcal{A},\mathcal{B}}$ oracle," we obtain an oracle $\mathcal{O} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$ relative to which $\mathsf{P} = \mathsf{NP}$ but a BQP prover-efficient interactive proof for a classically hard language exists.[4]

This is an outline of our approach. To complete the warm-up, the final question is how to instantiate $\mathcal{A}$ and $\mathcal{B}$. First, we take $\mathcal{A} = (f, \oplus\mathsf{P}^f)$, where f is a random function. It follows that

$$\mathsf{PSPACE}^{\mathcal{A}} = \mathsf{PSPACE}^{f} \subseteq \mathsf{IP}^{\mathcal{A}} \subseteq \mathsf{PSPACE}^{\mathcal{A}}$$

using the Goldwasser-Kalai-Rothblum variant of the $\mathsf{IP} = \mathsf{PSPACE}$ theorem [Sha90, GKR08, AW08, AB17], which builds an IP for PSPACE using only low-degree extensions of Boolean functions over characteristic 2 fields. This is extremely convenient because we know (see, e.g., [For09, Section 3]) that

$$\mathsf{PH}^{\mathcal{A}} = \mathsf{PH}^{\oplus\mathsf{P}^f} \subseteq \mathsf{BPP}^{\oplus P^f}.$$

[3] For the complement of Simon's problem, an interactive proof based on approximate counting is known, but Aaronson [Aar21, Section 7] proposed it as a candidate for lacking an interactive proof with a QPT honest prover.

[4] This can be strengthened to an oracle relative to which $\mathsf{BQP} = \mathsf{PSPACE}$ by recursively defining $\mathcal{B}$ and $\mathcal{C}$ so that BQP can solve all of $\mathsf{PSPACE}^{\mathcal{O}}$.

This gives a hard unary language in $\mathsf{PSPACE}^{\mathcal{A}}$ that is hard for $\mathsf{PH}^{\mathcal{A}}$ using the majority of the function f over all of its n-length inputs. This language is hard for $\mathsf{PH}^{\mathcal{A}}$ by the $\mathsf{AC}^0[\oplus]$ lower bound for majority [Raz87].

However, at this point we have an IP for a language outside PH, but it does not have an *efficient* quantum prover. To do this, we employ an [AIK22] Forrelation encoding of $\mathsf{PSPACE}^{\mathcal{A}}$, and call this oracle $\mathcal{B}$. Then, we have that $\mathsf{PSPACE}^{\mathcal{A}}$ is contained in $\mathsf{IP}^{\mathcal{A}}$ with a $\mathsf{BQP}^{\mathcal{B}}$-efficient prover. On the other hand, by the security of the Forrelation encoding, we have that $\mathsf{PH}^{\mathcal{A},\mathcal{B}}$ algorithms cannot solve majority on f with probability much better than a PH algorithm relative to $\mathcal{A}$ along with an independent random oracle. This completes the warm-up.

Extending to non-uniform hardness. Of course, the unary language used in our warm-up PoQ is not hard for non-uniform classical adversaries, which can have memorized the language in their advice strings. To get around this, we naturally turn to *salting*: for a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, we define the language

$$L^f = \{x \in \{0,1\}^n : \sum_{y \in \{0,1\}^n} f(x,y) > 2^{n-1}\}.$$

For a random function f, this gives 2^n independent majority instances at each input length, indexed by a uniformly sampled x. Then, by constructing the same oracle $\mathcal{O} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$ we get our candidate oracle relative to which $\mathsf{P} = \mathsf{NP}$ and there is an efficient-prover interactive proof for the language L^f.

The remaining question is how to show that deciding L^f is almost everywhere hard-on-average for $(\mathsf{PH}/\mathsf{poly})^{\mathcal{A},\mathcal{B}}$. Fix a universal oracle Turing machine M and an advice length $s(n) = \mathsf{poly}(n)$. We want to upper bound the quantity

$$\rho^{\delta}(\mathcal{A},\mathcal{B}) = \Pr_{\mathcal{A},\mathcal{B}}\left[\exists z \in \{0,1\}^{s(n)} : \quad \Pr_{x \leftarrow \{0,1\}^n}[M^{\mathcal{O}}(x,z) = \mathrm{maj}(f_x)] \geq 1 - \delta\right].$$

Our argument proceeds as follows. Suppose instead that this quantity is large.

1. *Carefully invoke security of the Forrelation encoding.* As a first step (which turns out to be the hardest part of the proof), we would like to convert this attack into one where the Forrelated PSPACE oracle $\mathcal{B}$ has been replaced by a random oracle, invoking the classical (PH) hardness of Forrelation. Compared to before, the challenge is handling advice strings z that can depend on the oracles $\mathcal{A}, \mathcal{B}$; since these can reveal individual bits of the Forrelation encoding, it's not immediately clear how to invoke the hardness of Forrelation.

   In other words, we would like to argue that the quantity $\rho(\mathcal{A},\mathcal{B})$ does not change by much when we replace $\mathcal{B}$ by a random oracle. If $\rho(\mathcal{A},\mathcal{B})$ were computable by a $\mathsf{PH}^{\mathcal{A},\mathcal{B}}$ machine, then we would be done. Unfortunately, this doesn't seem to be the case:

   - The verification itself involves the quantity $\mathrm{maj}(f_x)$, which is hard to compute, and
   - The expectation over x is also not naturally a PH computation.

   To fix this, we introduce a *proxy*

$$\rho_t^{\delta}(\mathcal{A},\mathcal{B},x_1,\ldots,x_t,y_1,\ldots,y_t) = \Pr_{\mathcal{A},\mathcal{B}}\left[\exists z \in \{0,1\}^{s(n)} : \quad \left|i : M^{\mathcal{O}}(x_i,z) = y_i\right| \geq 1 - \delta\right].$$

By standard concentration arguments, if $t \gg s$ then we have that

$$\mathop{\mathbb{E}}_{x_1,\ldots,x_t} \rho_t^{\delta/2}(\mathcal{A}, \mathcal{B}, x_1, \ldots, x_t, \mathrm{maj}(f_{x_1}), \ldots, \mathrm{maj}(f_{x_t})) < \rho^{\delta}(\mathcal{A}, \mathcal{B}) + \epsilon$$

and

$$\rho^{\delta}(\mathcal{A}, \mathcal{B}) < \mathop{\mathbb{E}}_{x_1,\ldots,x_t} \rho_t^{2\delta}(\mathcal{A}, \mathcal{B}, x_1, \ldots, x_t, \mathrm{maj}(f_{x_1}), \ldots, \mathrm{maj}(f_{x_t})) + \epsilon$$

for a small error ϵ. That is, if we sample random $x_1, \ldots, x_t$ and precompute the corresponding majority values $y_i = \mathrm{maj}(f_{x_i})$, then the quantity $\rho(\mathcal{A}, \mathcal{B})$ that we care about can be approximated by the proxy with small losses in the parameters. Crucially, the proxy is efficiently computable by a $\mathsf{PH}^{\mathcal{A},\mathcal{B}}$ machine.

Finally, we invoke Forrelation hardness to argue that the quantity

$$\mathop{\mathbb{E}}_{x_1,\ldots,x_t} \rho_t(\mathcal{A}, \mathcal{B}, x_1, \ldots, x_t, \mathrm{maj}(f_{x_1}), \ldots, \mathrm{maj}(f_{x_t}))$$

is approximately preserved when switching $\mathcal{B}$ to a random oracle.[5] Combining these arguments, we can conclude that if $\rho^{\delta}(\mathcal{A}, \mathcal{B})$ is noticeably large in the case where $\mathcal{B}$ is Forrelated, then $\rho^{O(\delta)}(\mathcal{A}, \mathcal{B})$ is noticeably large when $\mathcal{B}$ is a random oracle.

2. *Reuse the advice to solve many instances.* Now that $\mathcal{B}$ has been replaced by a random oracle, relatively standard techniques for handling adversaries with advice can be applied. Specifically, suppose that a non-uniform $\mathsf{PH}^{\mathcal{A},\mathcal{B}}$ adversary computes $\mathrm{maj}(f_x)$ with probability $1 - \delta$. Then, if we ran the adversary – reusing the advice string z – on independent inputs $x_1, \ldots, x_t$, we would immediately obtain that the adversary succeeds on all t inputs with probability $(1-\delta)^t$. In turn, this implies a *uniform* PH algorithm for the t-fold problem that succeeds with probability $2^{-s} \cdot (1-\delta)^t$ by guessing the advice string z uniformly.

3. *Uniform hardness using a direct-product theorem.* In the random oracle-circuit complexity correspondance, the above algorithm gives a quasi-polynomial size $\mathsf{AC}^0[\oplus]$ circuit that computes t simultaneous majorities on independent random inputs with probability $2^{-s} \cdot (1-\delta)^t$. The fact that such circuits do not exist follows from Razborov's majority lower bound combined with an AC^0-friendly direct-product theorem [Raz87, IJKW10].

These steps, combined with standard union bound arguments, yields a proof that salted majority is weakly hard-on-average, which gives us a proof of quantumness with constant soundness error. This soundness error can then be amplified to negligible by standard repetition.

2.2 An oracle relative to which $\mathsf{BQP} = \mathsf{QCMA}$ but oblivious state preparation exists

For our second result, we construct a quantum (unitary) oracle relative to which $\mathsf{BQP} = \mathsf{QCMA}$, but the (very strong) quantum primitive of quantum verifier oblivious state preparation (qv-OSP) exists. Similarly to [Kre21, GMMY24], we will construct an oracle $\mathcal{O}$ relative to which it is easy to construct qv-OSP secure against inefficient, query-bounded adversaries, but for which efficiently

[5] This is arguing that the Forrelation encoding can be used to hide a "message" sampled from a fixed distribution depending on $\mathcal{A}$ but not the randomness of the Forrelation oracle.

identifiable information about the oracle cannot be encoded classically. In particular, it will be the case that for any function $c(\mathcal{O})$ output a polynomial length classical string, it will be impossible for an oracle algorithm $\mathcal{A}^{(\cdot),\mathsf{PSPACE}}(c(\mathcal{O}))$ to tell if it is interacting with the oracle $\mathcal{O}$ or a freshly sampled oracle $\widetilde{\mathcal{O}}$[6]. i.e. with high probability over the choice of $\mathcal{O}$,

$$\left|\Pr\left[\mathcal{A}^{\mathcal{O},\mathsf{PSPACE}}(c(\mathcal{O}))\to 1\right]-\Pr_{\widetilde{\mathcal{O}}}[\mathcal{A}^{\widetilde{\mathcal{O}},\mathsf{PSPACE}}(c(\mathcal{O}))\to 1]\right|=\mathsf{negl}$$

As such, for PSPACE an oracle to a PSPACE complete problem, relative to $\mathcal{O}, \mathsf{PSPACE}$ qv-OSP exists. This is because the PSPACE oracle is independent of $\mathcal{O}$, and so cannot help inefficient, query-bounded algorithms. However, any $\mathsf{QCMA}^{\mathcal{O},\mathsf{PSPACE}}$ verifer $\mathsf{Ver}^{\mathcal{O},\mathsf{PSPACE}}$ will have approximately the same behavior on all inputs when $\mathcal{O}$ is replaced with a fresh random oracle $\widetilde{\mathcal{O}}$ sampled by the verifier itself. As such, for all $\mathsf{Ver}^{\mathcal{O},\mathsf{PSPACE}}$ there is an equivalent verifier $\widetilde{\mathsf{Ver}}^{\mathsf{PSPACE}}$ deciding the same language and making no queries to $\mathcal{O}$. Therefore, the language corresponding to $\mathsf{Ver}^{\mathcal{O},\mathsf{PSPACE}}$ can be decided in $\mathsf{BQP}^{\mathsf{PSPACE}}$ by checking if the modified $\widetilde{\mathsf{Ver}}^{\mathsf{PSPACE}}$ accepts.

For simplicity, we will first construct a quantum channel oracle, and then describe how we lift this to a unitary oracle.

Our quantum channel oracle Our oracle will be indexed by collections of Haar random states $|\phi_x\rangle$ for $x\in\{0,1\}^n$, as well as a random oracle $\mathrm{RO}_n:\{0,1\}^n\times\{0,1\}^{4n}\to\{0,1\}^{6n}\times\{0,1\}^{6n}$. For an input x,y, write $\mathrm{RO}_n(x,y)=u_{x,y},v_{x,y}$.

1. SG_n: Samples a random $x\in\{0,1\}^n$. Outputs $(x,|\phi_x\rangle)$.
2. $\mathsf{Expand}_n(x)$: Samples a random $y\in\{0,1\}^{4n}$. Outputs y as well as the "claw state"
$$|c_{x,y}\rangle\propto|u_{x,y}\rangle+|v_{x,y}\rangle$$
3. $\mathsf{Mix}_n(x,y,|\psi\rangle)$: If $|\psi\rangle=|\phi_x\rangle$, outputs $|\mathrm{RO}_n(x,y)\rangle=|u_{x,y}\rangle|v_{x,y}\rangle$.

Constructing qv-OSP To construct qv-OSP, it suffices to construct a "claw-state generator" [BK25, Theorem 4.7]. This is a classical communication interactive protocol where at the end of the protocol the verifier receives two strings a,b while the prover receives the "claw-state" $\frac{1}{\sqrt{2}}|a\rangle+\frac{1}{\sqrt{2}}|b\rangle$. Security says that at the end of the protocol, a malicious prover cannot predict both a and b with better than negligible probability.

Constructing a claw-state generator relative to our oracle is straightforward. The protocol is as follows

1. The verifier will run $\mathsf{SG}_n\to(x,|\phi_x\rangle)$ and sends x to the prover
2. The prover will run $\mathsf{Expand}_n(x)\to(y,|c_{x,y}\rangle)$. It will send y to the verifier, and output the claw state
$$|c_{x,y}\rangle\propto|u_{x,y}\rangle+|v_{x,y}\rangle\,.$$
3. Finally, the verifier will query $\mathsf{Mix}_n(x,y,|\phi_x\rangle)\to(u_{x,y},v_{x,y})$ and output $a=u_{x,y},b=v_{x,y}$.

[6]Note that this will only hold for the portions of the oracle acting on sufficiently large inputs. Oracles with small inputs will be handled using the same techniques as [Kre21].

Correctness of the protocol is clear. Security comes from the fact that the prover does not see $|\phi_x\rangle$, and so cannot access $\mathsf{Mix}(x,\cdot)$. As such, its only access to $\mathrm{RO}_n(x,\cdot)$ is through $\mathsf{Expand}_n(x)$. In particular, it can only see one copy of $|c_{x,y}\rangle$ for each y randomly sampled by $\mathsf{Expand}_n(x)$. Thus, the only way it can predict both $u_{x,y}, v_{x,y}$ simultaneously for any y is if it guesses $|\phi_x\rangle$, guesses $u_{x,y}, v_{x,y}$ directly, or Expand_n samples the same y twice. These events occur with at most negligible probability.

In addition, we prove that security holds against non-uniform adversaries with classical advice on the oracle. This will follow by essentially repeating the same techniques used to show $\mathsf{BQP}^{\mathcal{O},\mathsf{PSPACE}} = \mathsf{QCMA}^{\mathcal{O},\mathsf{PSPACE}}$, which essentially says that classical information on the oracle is useless.

Proving $\mathsf{BQP}^{\mathcal{O},\mathsf{PSPACE}} = \mathsf{QCMA}^{\mathcal{O},\mathsf{PSPACE}}$ As described above, this proof boils down to showing with high probability over the choice of $\mathcal{O}$,

$$\left|\Pr\left[\mathcal{A}^{\mathcal{O},\mathsf{PSPACE}}(c(\mathcal{O})) \to 1\right] - \Pr_{\widetilde{\mathcal{O}}}[\mathcal{A}^{\widetilde{\mathcal{O}},\mathsf{PSPACE}}(c(\mathcal{O})) \to 1]\right| = \mathsf{negl} \tag{1}$$

That is, it is impossible to describe any efficiently identifiable information about $\mathcal{O}$ using polynomially many classical bits.

Note that there are two components of $\mathcal{O}$ one could hope to describe: the quantum states $|\phi_x\rangle$ and the random oracle RO. It is well known that a short amount of classical information cannot encode any identifiable information about a Haar random state [Kre21]. This is known as superconcentration of the Haar distribution.

And so, it remains to show that a short classical string cannot give any identifiable information about the random oracle. Just as in [GMMY24], the key here is that any algorithm $\mathcal{A}$ can only access the random oracle at *random points*. In particular, at points $\mathrm{RO}_n(x,\cdot)$ for x sampled by SG_n or at points $\mathrm{RO}_n(\cdot,y)$ for y sampled by Expand_n. However, by the presampling technique [CDGS18], any classical string describing the random oracle can only describe a polynomial number of its inputs. Thus, any classical advice $c(\mathcal{O})$ is unlikely to carry any information about $\mathrm{RO}_n(x,\cdot), \mathrm{RO}_n(\cdot,y)$ for sampled x, y. The claim in Equation (1) follows.

Lifting to a unitary oracle In general, while unitary oracle separations can be difficult to reason about, there are a number of "stronger" oracle types which are easier to work with. One particularly useful oracle formulation are state preparation channels. These are oracles defined by some family of mixed states σ_x. When queried on an input x, the oracle will return a copy of σ_x. Such oracles have recently been used to achieve a number of separations in quantum cryptography [AGL24, CCS25].

One can also consider state preparation *unitaries*. A state preparation unitary for a mixed state σ is any unitary S_σ such that

$$\sigma_{\mathbf{A}} = \mathrm{Tr}_{\mathbf{B}}(S_\sigma |0\rangle\langle 0|_{\mathbf{AB}} S_\sigma^\dagger)$$

A series of recent works has given techniques for simulating state preparation unitaries with state preparation channels [BMM+25, GZ25, GMMY24, AG25, TWZ26]. The culmination of these results gives that for any mixed state σ, there exists a distribution over state preparation unitaries S_σ that can be efficiently simulated using copies of σ (i.e. by querying a state preparation channel). More formally, there exists simulator $\mathsf{Sim}(\sigma^{\otimes L})$ taking in a polynomial number L of copies of σ

such that for all polynomial-query oracle algorithms $\mathcal{A}$ outputting a single bit,

$$\left|\Pr_{S_\sigma}[\mathcal{A}^{S_\sigma} \to 1] - \Pr\left[\mathcal{A}^{\mathsf{Sim}(\sigma^{\otimes L})} \to 1\right]\right| \leq \frac{1}{\mathsf{poly}}$$

An immediate consequence of this is that if a complexity collapse holds relative to a random S_σ, then it also holds relative to a state preparation channel for σ. It is tempting to then claim that anytime we achieve an oracle separation relative to a state preparation oracle, we can translate it to a unitary oracle separation in this way. However, this is false. In particular, the issue lies in the other direction. If a complexity collapse holds relative to a state preparation oracle for σ, it need not necessarily also hold relative to S_σ^U.

The key observation implicit in [GMMY24] is that this reverse direction does hold specifically for the case of $\mathsf{BQP}^{\mathcal{O},\mathsf{PSPACE}} = \mathsf{QCMA}^{\mathcal{O},\mathsf{PSPACE}}$. The argument takes advantage of the fact that breaking QCMA is a non-interactive game using only classical communication.

For a family $\vec{\sigma} = \{\sigma_t\}_{t\in\mathbb{N}}$ of t-qubit mixed states, write $\mathsf{BQP}^{\vec{\sigma}}, \mathsf{QCMA}^{\vec{\sigma}}$ for the corresponding classes with access to a state preparation channel for each σ_t. Write $S = \{S_{\sigma_t}^{U_t}\}_{t\in\mathbb{N}}$, where the U_t are independent Haar unitaries on t qubits. It turns out that by combining the ability to simulate S with the techniques of [Kre21], one can show $\mathsf{QCMA}^S \subseteq \mathsf{BQP}^{S,\mathsf{QCMA}^{\vec{\sigma}}}$. As an immediate corollary, if $\mathsf{BQP}^{\vec{\sigma}} = \mathsf{QCMA}^{\vec{\sigma}}$, then with probability 1, $\mathsf{BQP}^S = \mathsf{QCMA}^S$. For full formality, we include a proof of this in Appendix A.

As such, these techniques show that any oracle where $\mathsf{BQP} = \mathsf{QCMA}$ and some cryptographic object exists[7] relative to a state preparation channel immediately lifts to a unitary oracle. Note that our oracle is not a state preparation channel oracle. However, our oracle morally makes controlled queries to a channel oracle, and combining with the fact that the individual prepared states are themselves simulatable, similar techniques apply.

3 Preliminaries

3.1 Notation

For $m \in \mathbb{N}$, write $[m] = \{1, \ldots, m\}$. For a finite set X, $x \leftarrow X$ denotes a uniformly random element of X; for a distribution $\mathcal{D}$, $x \leftarrow \mathcal{D}$ denotes a sample from $\mathcal{D}$. All logarithms are base 2 unless written as ln. For a finite set X, let $\mathcal{H}(X)$ be the Hilbert space with orthonormal basis $\{|x\rangle : x \in X\}$. Write $\mathrm{U}(D)$ for the group of unitaries on a D-dimensional space. We use $\mathsf{Haar}(X)$ for the Haar distribution on unitaries on $\mathcal{H}(X)$, and $\mathsf{HaarSt}(X)$ for the Haar distribution on pure states in $\mathcal{H}(X)$.

We write $\mathsf{poly}(n) = n^{O(1)}$, $\mathsf{polylog}(n) = (\log n)^{O(1)}$, and $\mathsf{quasipoly}(n) = 2^{(\log n)^{O(1)}}$ for unspecified bounds of these forms. A nonnegative function $\nu(n)$ is negligible if for every constant $c > 0$, $\nu(n) \leq n^{-c}$ for all sufficiently large n; we write $\mathsf{negl}(n)$ for an unspecified negligible function.

In acceptance-probability bounds, algorithms output a single bit, and the probability includes all internal randomness and measurements. For a discrete random variable X, its min-entropy is $H_\infty(X) = -\log \max_x \Pr[X = x]$.

[7] Technically, this argument only applies to "one-stage" cryptographic games, i.e. games where there is only one stateful adversary.

3.2 Complexity Preliminaries

3.2.1 Algorithms and circuits

PPT and QPT denote probabilistic and quantum polynomial-time algorithms, respectively. Quantum algorithms use a fixed finite universal gate set, computational basis measurements, and ancillas initialized to $|0\rangle$. Uniform circuit families have descriptions generated by a deterministic polynomial-time algorithm. Nonuniform algorithms are polynomial-size circuit families; their descriptions, or equivalently their classical advice, may depend on the input length and the oracle. We do not allow quantum advice unless stated explicitly. Query-bounded algorithms need not be time bounded unless this is stated.

We measure the size of a circuit by its description length in bits. Under our convention, there are at most 2^ℓ circuits of size ℓ.

3.2.2 Oracle Algorithms

A classical oracle is a function f on bit strings. A classical oracle algorithm may submit an input x and receive $f(x)$, choosing each query using the answers to previous queries. Preparing and storing queries and reading answers count towards the algorithm's time and space bounds.

A quantum oracle is a family of unitary operators. We allow controlled queries and queries to both a unitary and its inverse. Components and levels may be queried in superposition, using zero padding for unused registers. A query to level t takes at least t time. A classical function f is accessed quantumly through the map $|x, z\rangle \mapsto |x, z \oplus f(x)\rangle$. All random choices in our oracle constructions are independent unless a dependence is specified.

3.2.3 Oracle complexity classes

We identify a language $L \subseteq \{0,1\}^*$ with its characteristic function $L : \{0,1\}^* \to \{0,1\}$. A superscript denotes oracle access. In the classical definitions below, $\mathcal{O}$ is a classical oracle. In the quantum definitions, $\mathcal{U}$ is a quantum oracle with the query conventions of Section 3.2.2, following [Kre21]. Classical oracles are included as a special case, and multiple oracles may be listed in a superscript.

Definition 3.1 ($\mathsf{P}^{\mathcal{O}}$)**.** A language L is in $\mathsf{P}^{\mathcal{O}}$ if there is a deterministic polynomial-time oracle Turing machine $M^{\mathcal{O}}$ such that, for every $x \in \{0,1\}^*$,

$$M^{\mathcal{O}}(x) = L(x).$$

Definition 3.2 ($\mathsf{NP}^{\mathcal{O}}$)**.** A language L is in $\mathsf{NP}^{\mathcal{O}}$ if there are a deterministic polynomial-time oracle verifier $\mathsf{Ver}^{\mathcal{O}}$ and a polynomial p such that, for every $x \in \{0,1\}^*$,

$$x \in L \iff \exists w \in \{0,1\}^{p(|x|)}, \quad \mathsf{Ver}^{\mathcal{O}}(x, w) = 1.$$

Definition 3.3 ($\oplus\mathsf{P}^{\mathcal{O}}$)**.** A language L is in $\oplus\mathsf{P}^{\mathcal{O}}$ if there are a deterministic polynomial-time oracle verifier $\mathsf{Ver}^{\mathcal{O}}$ and a polynomial p such that, for every $x \in \{0,1\}^*$,

$$L(x) = \bigoplus_{w \in \{0,1\}^{p(|x|)}} \mathsf{Ver}^{\mathcal{O}}(x, w).$$

Thus, $x \in L$ exactly when the number of accepting witnesses is odd.

Definition 3.4 ($\mathsf{BPP}^{\mathcal{O}}$)**.** A language L is in $\mathsf{BPP}^{\mathcal{O}}$ if there is a PPT oracle algorithm $\mathcal{A}^{\mathcal{O}}$ such that, for every $x \in \{0,1\}^*$,

$$\Pr\left[\mathcal{A}^{\mathcal{O}}(x) \to L(x)\right] \geq 2/3.$$

Definition 3.5 ($\mathsf{PSPACE}^{\mathcal{O}}$)**.** A language L is in $\mathsf{PSPACE}^{\mathcal{O}}$ if there is a deterministic oracle Turing machine $M^{\mathcal{O}}$ which halts on every input, uses polynomial space including its query tape, and satisfies $M^{\mathcal{O}}(x) = L(x)$ for every $x \in \{0,1\}^*$.

Definition 3.6 ($\mathsf{BQP}^{\mathcal{U}}$)**.** A language L is in $\mathsf{BQP}^{\mathcal{U}}$ if there is a QPT oracle algorithm $\mathcal{A}^{\mathcal{U}}$ such that, for every $x \in \{0,1\}^*$,

1. If $x \in L$, then $\Pr\left[\mathcal{A}^{\mathcal{U}}(x) \to 1\right] \geq 2/3$.
2. If $x \notin L$, then $\Pr\left[\mathcal{A}^{\mathcal{U}}(x) \to 1\right] \leq 1/3$.

Definition 3.7 ($\mathsf{QCMA}^{\mathcal{U}}$)**.** A language L is in $\mathsf{QCMA}^{\mathcal{U}}$ if there are a QPT oracle verifier $\mathsf{Ver}^{\mathcal{U}}$ and a polynomial p such that, for every $x \in \{0,1\}^*$,

1. **Completeness.** If $x \in L$, there is a classical witness $w \in \{0,1\}^{p(|x|)}$ such that $\Pr\left[\mathsf{Ver}^{\mathcal{U}}(x,w) \to 1\right] \geq 2/3$.
2. **Soundness.** If $x \notin L$, then for every $w \in \{0,1\}^{p(|x|)}$, $\Pr\left[\mathsf{Ver}^{\mathcal{U}}(x,w) \to 1\right] \leq 1/3$.

Definition 3.8 ($\mathsf{IP}^{\mathcal{O}}$)**.** A language L is in $\mathsf{IP}^{\mathcal{O}}$ if there is a PPT oracle verifier $V^{\mathcal{O}}$ in a polynomial-round protocol with classical communication such that, for every $x \in \{0,1\}^*$,

1. **Completeness.** If $x \in L$, there is an unbounded prover $P^{\mathcal{O}}$ such that $\Pr\left[(V^{\mathcal{O}} \leftrightarrows P^{\mathcal{O}})(x) \to 1\right] \geq 2/3$.
2. **Soundness.** If $x \notin L$, then for every unbounded prover $P^{*\mathcal{O}}$, $\Pr\left[(V^{\mathcal{O}} \leftrightarrows P^{*\mathcal{O}})(x) \to 1\right] \leq 1/3$.

Both parties have access to $\mathcal{O}$.

Omitting the oracle gives the corresponding unrelativized classes. A complexity class in an oracle superscript denotes a fixed complete problem for that class. For a promise problem $(L_{\text{yes}}, L_{\text{no}})$, the same acceptance and verification conditions are required only on promised inputs: replace $x \in L$ by $x \in L_{\text{yes}}$ and $x \notin L$ by $x \in L_{\text{no}}$. We specify explicitly when complexity classes refer to promise problems.

3.2.4 Circuit classes

Definition 3.9 (Circuit classes)**.** We write $\mathsf{AC}^0(S,d)$ for the class of Boolean circuits of size at most S and depth at most d, using unbounded fan-in AND and OR gates and fan-in one NOT gates. The class $\mathsf{AC}^0[\oplus](S,d)$ also allows unbounded fan-in parity gates. The classes AC^0 and $\mathsf{AC}^0[\oplus]$ consist of nonuniform families of polynomial-size, constant-depth circuits of these forms. Size follows the convention above, and depth is the largest number of gates on an input-to-output path. An oracle gate evaluates the indicated oracle on the bits supplied to that gate.

3.2.5 Multilinear Extensions

We first define an important concept known as the multilinear extension of a boolean function to a finite field. In particular, let $f : \{0,1\}^t \to \{0,1\}$ be any function. We define the multilinear extension of f over $\mathbb{F}_{2^d}^t$ by

$$\mathsf{ML}_{2^d}[f](x) \coloneqq \sum_{u \in \{0,1\}^t} f(u)\chi_u(x) \quad \chi_u(x) \coloneqq \prod_{i=1}^{t}(u_i x_i + (1-u_i)(1-x_i)) \tag{2}$$

Note that $\mathsf{ML}_{2^d}[f]$ is a multilinear function (i.e. it is of degree at most 1 in each variable and thus of total degree at most t). Furthermore, viewing $\{0,1\}^t$ as a subset of $\mathbb{F}_{2^d}^t$ under the natural inclusion, we have for all $x \in \{0,1\}^t$ that $\chi_u(x) = 1$ if $x = u$ and $\chi_u(x) = 0$ if $x \neq u$. Thus, for all $x \in \{0,1\}^t$, $\mathsf{ML}_{2^d}[f](x) = f(x)$.

3.2.6 Circuit Lower Bounds

Forrelation We give the following statement, which says that there exists a problem (Forrelation) which can be efficiently solved quantumly but which cannot be solved by quasipolynomial size AC^0 circuits.

Lemma 3.10 (Hardness of Forrelation, Theorem 1.2 of [RT19])**.** *There exists an explicit distribution $\mathcal{F}_N$ (Forrelation) over $\{0,1\}^N$ such that*

1. *there exists a quantum algorithm $\mathcal{A}$ making $\mathsf{polylog}(N)$ queries to its input and running in time $\mathsf{polylog}(N)$ such that:*

$$\left| \Pr_{x \leftarrow \mathcal{F}_N}[\mathcal{A}(x) \to 1] - \Pr_{x \leftarrow \{0,1\}^N}[\mathcal{A}(x) \to 1] \right| \geq 1 - \frac{1}{N^2}$$

2. *For any $C \in \mathsf{AC}^0(\mathsf{quasipoly}(N), O(1))$:*

$$\left| \Pr_{x \leftarrow \mathcal{F}_N}[C(x) \to 1] - \Pr_{x \leftarrow \{0,1\}^N}[C(x) \to 1] \right| \leq \frac{\mathsf{polylog}(N)}{\sqrt{N}}$$

Note that the hardness of Forrelation also applies even when C is additionally given as input samples from any distribution independent of x.

Hardness of Majority Let $\mathsf{MAJ} : \{0,1\}^* \to \{0,1\}$ be the function such that $\mathsf{MAJ}(x)$ is 1 if and only if more than half the bits of x are 1.

We state the following lower bound on the ability for AC^0 circuits to compute the majority function. This lower bound comes originally from [Raz87]. For an explanation, see [KPG13]. The precise statement we use appears as Theorem 3 of [Hoz26].

Theorem 3.11 (Majority lower bound)**.** *Let $\mathsf{MAJ}(x)$ be the majority function. Then for any $C \in \mathsf{AC}^0[\oplus](S, O(1))$*

$$\Pr_{x \leftarrow \{0,1\}^N}[C(x) = \mathsf{MAJ}(x)] \leq \frac{1}{2} + \frac{\mathsf{polylog}(S)}{\sqrt{N}}$$

Direct product theorem for circuits

Theorem 3.12 (Direct product theorem for circuits (Theorem 1.2 of [IJKW10]))**.** *There exists a universal constant $c > 0$ such that the following holds. Let $\delta \in (0,1)$, $f : \{0,1\}^n \to \{0,1\}$ be any boolean function, and* Samp *any distribution over $\{0,1\}^n$. Let $\mathcal{A}$ be any algorithm such that*

$$\Pr_{\mathsf{Samp} \to x_1,\ldots,x_t}[\mathcal{A}(x_1,\ldots,x_t) = (f(x_1),\ldots,f(x_t))] \geq e^{-c\delta t}$$

Then there exists an AC^0 circuit C with oracle gates of size $\mathsf{poly}(n, t, \log(1/\delta), e^{c\delta t})$ such that

$$\Pr_{\mathsf{Samp} \to x}[C^{\mathcal{A}}(x) = f(x)] \geq 1 - \delta$$

3.3 Techniques for Simulating Oracles

3.3.1 Unitary Tomography

Theorem 3.13 (Unitary tomography [HKOT23])**.** *Let U be a unitary on m qubits and let $\epsilon, \delta \in (0,1)$. There is a quantum algorithm with controlled query access to U which runs in time $\mathsf{poly}(2^m, 1/\epsilon, \log(1/\delta))$ and outputs a classical description of a unitary circuit $\widetilde{U}$. With probability at least $1-\delta$, for every $q \in \mathbb{N}$ and every oracle algorithm $\mathcal{A}^{(\cdot)}$ making at most q queries and outputting a single bit,*

$$\left|\Pr\left[\mathcal{A}^{U} \to 1\right] - \Pr\left[\mathcal{A}^{\widetilde{U}} \to 1\right]\right| \leq q\epsilon.$$

Queries may be controlled, forwards or inverse, and the bound allows arbitrary auxiliary registers and fixed side oracles.

Remark 3.14. Unitary tomography as stated by [HKOT23] specifically says that one can learn a description of a unitary $\widetilde{U}$ which is close to U under a metric known as the diamond norm. A basic induction then gives that $\widetilde{U}$ is indistinguishable from U by bounded query algorithms, where the error blows up linearly in the number of queries.

A second subtlety emerges in that the guarantee only applies to algorithms which do not make controlled queries to their oracle. However, applying tomography instead to the controlled version of U

$$|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$$

gives the above theorem.

3.3.2 Reprogramming Unitaries

We will use as well a simplified statement of the unitary reprogramming lemma from [ÇGR26, HY25]. This says that unitary oracles which differ on unpredictable points are indistinguishable.

Lemma 3.15 (Unitary reprogramming lemma)**.** *Let $\epsilon \in (0,1)$. Let $\Pi \leftarrow S$ be a distribution over projectors such that for any fixed state $|\phi\rangle$,*

$$\mathop{\mathbb{E}}_{\Pi \leftarrow S}\left[\|\Pi|\phi\rangle\|^2\right] \leq \epsilon$$

Let $\mathcal{O}$ be any unitary oracles, and let $\overline{\mathcal{O}}_\Pi$ be any family of oracles such that

$$\mathcal{O}(I-\Pi) = \overline{\mathcal{O}}_\Pi(I-\Pi)$$

Then for all q query algorithms $\mathcal{D}^{(\cdot)}$, we have

$$\left|\Pr\left[\mathcal{D}^{\mathcal{O}} \to 1\right] - \Pr_{\Pi \leftarrow S}[\mathcal{D}^{\overline{\mathcal{O}}_\Pi} \to 1]\right| \leq 2q\sqrt{\epsilon}$$

Theorem 3.16 (Random unitaries are useless [Kre21, Mec19])**.** *Let $D \in \mathbb{N}$, $\epsilon \in (0,1)$ and let $\mathcal{A}^{(\cdot)}$ be any algorithm making q queries to its oracle. Then*

$$\Pr_{U \leftarrow \mathsf{Haar}([D])}\left[\left|\Pr\left[\mathcal{A}^{U} \to 1\right] - \Pr_{U' \leftarrow \mathsf{Haar}([D])}[\mathcal{A}^{U'} \to 1]\right| \geq \epsilon\right] \leq 2\exp\left(-\frac{(D-2)\epsilon^2}{96q^2}\right)$$

3.3.3 Simulating State Preparation

Lemma 3.17 (Replacing unitaries with states [GMMY24, AGL24])**.** *Let $N < M \in \mathbb{N}$. Let ρ be the mixed state of dimension M^N consisting of N random elements of $[M]$. That is,*

$$\rho = \mathop{\mathbb{E}}_{k_i \leftarrow [M], i \in [N]}\left[\bigotimes_{i=1}^{N} |k_i\rangle\langle k_i|\right]$$

Let ρ' be the same but we replace $|k_i\rangle$ with $|\phi_i\rangle$, a Haar random state subject to the restriction that for each $i \neq j$, $|\phi_i\rangle, |\phi_j\rangle$ are orthogonal. That is,

$$\rho' = \mathop{\mathbb{E}}_{U \leftarrow \mathsf{Haar}([M])}\left[\bigotimes_{i=1}^{N} U|i\rangle\langle i|U^\dagger\right]$$

Then, for every quantum algorithm $\mathcal{A}$ outputting a single bit,

$$\left|\Pr[\mathcal{A}(\rho) \to 1] - \Pr\left[\mathcal{A}(\rho') \to 1\right]\right| \leq \frac{N(N+1)}{2M}.$$

Remark 3.18. Although Lemma 2.24 of [GMMY24] uses independent Haar random states, since a single copy of a Haar random state is maximally mixed it is sufficient to consider random strings instead.

Definition 3.19 (State preparation for pure states)**.** Let $|\phi\rangle$ be a pure state. The state preparation unitary for $|\phi\rangle$ is the map which swaps $|0\rangle|0\rangle$ and $|1\rangle|\phi\rangle$ defined by

$$S_{|\phi\rangle} = I - |1\rangle\langle 1| \otimes |\phi\rangle\langle\phi| - |0\rangle\langle 0| \otimes |0\rangle\langle 0| + |0\rangle\langle 1| \otimes |0\rangle\langle\phi| + |1\rangle\langle 0| \otimes |\phi\rangle\langle 0|$$

Definition 3.20 (State preparation for mixed states)**.** Let σ be a mixed state. We say that a map S_σ is a state preparation unitary for σ if there exists a pure state $|\phi\rangle_{AB}$ such that $\sigma = \mathrm{Tr}_B(|\phi\rangle\langle\phi|_{AB})$ and $S_\sigma = S_{|\phi\rangle}$.

Definition 3.21 (Random state preparation)**.** Let σ be a mixed state on $\mathcal{H}(\{0,1\}^n)$. Fix any purification $|\phi\rangle \in \mathcal{H}(\{0,1\}^n) \otimes \mathcal{H}(\{0,1\}^m)$ with $\sigma = \mathrm{Tr}_B(|\phi\rangle\langle\phi|_{AB})$. Let $S_\sigma = S_{|\phi\rangle}$ be the state preparation unitary for σ. Let $U \in \mathrm{U}(2^m)$. We define S_σ^U to be the map

$$S_\sigma^U = S_{(I_{\mathcal{H}(\{0,1\}^n)} \otimes U)|\phi\rangle} = C_U S_\sigma C_U^\dagger,$$
$$C_U = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes (I_{\mathcal{H}(\{0,1\}^n)} \otimes U).$$

Lemma 3.22 (Simulating state preparation unitaries [TWZ26])**.** *There exists a universal constant $c > 0$ such that the following holds. Let $q, n, m \in \mathbb{N}$, let $\epsilon \in (0,1)$, and let $L \geq cq^2/\epsilon$. Then there exists a stateful algorithm* Sim *with the following property. Let σ be any mixed state on $\mathcal{H}(\{0,1\}^n)$ with purification $|\phi\rangle \in \mathcal{H}(\{0,1\}^n) \otimes \mathcal{H}(\{0,1\}^m)$ and $S_\sigma = S_{|\phi\rangle}$. Then for all oracle algorithms $\mathcal{A}^{(\cdot)}$ making at most q (possibly controlled, forwards or inverse) queries and outputting a single bit, we have*

$$\left| \Pr_{U \leftarrow \mathsf{Haar}(\{0,1\}^m)} [\mathcal{A}^{S_\sigma^U} \to 1] - \Pr\left[\mathcal{A}^{\mathsf{Sim}(\sigma^{\otimes L})} \to 1\right] \right| \leq \epsilon$$

Furthermore, there is an efficient circuit implementing Sim *of size* $\mathsf{poly}(L, n, m)$*.*

Lemma 3.23 (Simulating families of state preparation unitaries)**.** *There exists a universal constant $c > 0$ such that the following holds. Let $q, \ell, n, m \in \mathbb{N}$, let $\epsilon \in (0,1)$, and let $L \geq cq^2/\epsilon$. Then there exists a stateful algorithm* Sim *with the following property. Let $\{\sigma_x\}_{x\in[\ell]}$ be a family of mixed states on $\mathcal{H}(\{0,1\}^n)$ with purifications $|\phi_x\rangle \in \mathcal{H}(\{0,1\}^n) \otimes \mathcal{H}(\{0,1\}^m)$ and $S_{\sigma_x} = S_{|\phi_x\rangle}$. Define $\Sigma = \bigotimes_{x\in[\ell]} \sigma_x^{\otimes L}$. For a family $\{U_x\}_{x\in[\ell]}$, define $S^{\{U_x\}_{x\in[\ell]}} = \sum_x |x\rangle\langle x| \otimes S_{\sigma_x}^{U_x}$. Then for all oracle algorithms $\mathcal{A}^{(\cdot)}$ making at most q (possibly controlled, forwards or inverse) queries and outputting a single bit, we have*

$$\left| \Pr_{U_x \leftarrow \mathsf{Haar}(\{0,1\}^m),\, x\in[\ell]} [\mathcal{A}^{S^{\{U_x\}}} \to 1] - \Pr\left[\mathcal{A}^{\mathsf{Sim}(\Sigma)} \to 1\right] \right| \leq \epsilon$$

Furthermore, there is an efficient circuit implementing Sim *of size* $\mathsf{poly}(L, \ell, n, m)$*.*

Proof. As in the proof of Lemma 3.22, apply the procedures from [TWZ26, Lemmas 2.3 and 2.11] to the copies of each σ_x. This gives L copies of a random pure state $|\eta_x\rangle$ such that

$$S_{\sigma_x}^{U_x} = I - 2P_x, \qquad P_x = |\eta_x\rangle\langle\eta_x| \,.$$

Here the U_x are independent Haar unitaries, with the same U_x used for all copies of $|\eta_x\rangle$. The random phases introduced by the procedures can be absorbed into the U_x. Define

$$H = \sum_x |x\rangle\langle x| \otimes P_x,$$

so the oracle we want to simulate is $I - 2H = e^{-i\pi H}$.

Let B be the target register of a query. Take one unused copy of each P_x in registers E_x, and define

$$\omega_E = \bigotimes_x P_x, \qquad V = \sum_x |x\rangle\langle x| \otimes \mathrm{SWAP}_{B,E_x}.$$

Thus, V swaps B with the copy indexed by x, without measuring x. Applying e^{-ihV} and tracing out E gives the channel

$$\Phi_h(\rho) = \mathrm{Tr}_E \left[e^{-ihV} (\rho \otimes \omega_E) e^{ihV} \right] = \rho - ih[H, \rho] + O(h^2).$$

Since $\|V\|_\infty = 1$ and $\|H\|_\infty \leq 1$, the error term changes acceptance probabilities by at most $O(h^2)$, even in the presence of other registers. The map $\rho \mapsto e^{-ihH} \rho e^{ihH}$ has the same first-order expansion, so one such operation simulates it up to error $O(h^2)$. This bound also holds when the label register is in superposition or entangled with other registers.

For each query, repeat this operation $r = \lfloor L/q \rfloor$ times using unused copies and $h = \pi/r$. The error for one query is $O(rh^2) = O(1/r)$, and so the total error over q queries is $O(q/r) = O(q^2/L) \leq \epsilon$ for sufficiently large c. This uses at most $qr \leq L$ copies for each x. For controlled queries, replace H and V by $|1\rangle\langle 1| \otimes H$ and $|1\rangle\langle 1| \otimes V$. Inverse queries are the same as forward queries since the oracle is a reflection. Finally, averaging over the U_x preserves the error bound, and the procedures above can be implemented to the required accuracy using $\mathsf{poly}(L, \ell, n, m)$ gates. □

3.3.4 Presampling Lemmas

Theorem 3.24 (Presampling [CDGS18])**.** *Let $\mathcal{Y}$ be any finite set. We define a partial assignment $P = \{(x_1, y_1), \ldots, (x_L, y_L)\}$ to be a subset of $\{0,1\}^n \times \mathcal{Y}$ such that the x_i are distinct. Write $\mathsf{Dom}(P) = \{x : \exists y,\ (x, y) \in P\}$. For a function f, define a new function $f[P]$ by $f[P](x) = y$ for all $(x, y) \in P$, otherwise $f[P](x) = f(x)$.*

Let $n, q, L, s \in \mathbb{N}$ with $1 \leq L \leq 2^n$, and let $\epsilon \in (0, 1)$. Let $\mathcal{D}^{(\cdot)}$ be an oracle algorithm making at most q classical queries to its oracle. Let $c : \{f : \{0,1\}^n \to \mathcal{Y}\} \to \{0,1\}^s$ be any function, and let $f, \widetilde{f}$ be independent uniformly random functions. Then there exists a family $\{P_z\}_{z \in \{0,1\}^s}$ of distributions over partial assignments of length L such that

$$\left| \Pr_f[\mathcal{D}^f(c(f)) \to 1] - \Pr_{\substack{f,\widetilde{f} \\ P \leftarrow P_{c(f)}}}[\mathcal{D}^{\widetilde{f}[P]}(c(f)) \to 1] \right| \leq \frac{(s + \log(1/\epsilon)) \cdot q}{L} + \epsilon$$

Corollary 3.25 (Advice can't distinguish exponentially many random query points)**.** *Let $\mathcal{Y}$ be any finite set, and let $f, \widetilde{f} : \{0,1\}^n \to \mathcal{Y}$ be independent uniformly random functions. Let $X_1, \ldots, X_q$ be random variables over $\{0,1\}^n$, jointly independent of $f, \widetilde{f}$, such that for each i, $H_\infty(X_i) \geq \gamma$. Let $c : \{f : \{0,1\}^n \to \mathcal{Y}\} \to \{0,1\}^s$ be any function. Let $\mathcal{A}$ be any (potentially inefficient or quantum) randomized algorithm. Then*

$$\left| \Pr_f[\mathcal{A}(c(f), f(X_1), \ldots, f(X_q)) \to 1] - \Pr_{f,\widetilde{f}}[\mathcal{A}(c(f), \widetilde{f}(X_1), \ldots, \widetilde{f}(X_q)) \to 1] \right| \leq 2^{-n} + \frac{(s + n + 1)q}{2^{\gamma/2}}$$

The same bound holds if $\mathcal{A}$ also receives the sampled points $X_1, \ldots, X_q$.

Proof. The claim is trivial if the stated bound is at least 1, so assume otherwise; in particular, $s + n \geq 2$. Consider the distinguisher $\mathcal{D}^f(c(f))$ which samples $X_1, \ldots, X_q$, queries $f(X_1), \ldots, f(X_q)$, and then outputs $\mathcal{A}(c(f), f(X_1), \ldots, f(X_q))$. This is an algorithm making at most q classical queries to its oracle. By setting $\epsilon = 2^{-n}$ and $L = \lceil 2^{\gamma/2} \rceil$, we have

$$\left| \Pr_f[\mathcal{A}(c(f), f(X_1), \ldots, f(X_q)) \to 1] - \Pr_{\substack{f,\widetilde{f} \\ P \leftarrow P_{c(f)}}}[\mathcal{A}(c(f), \widetilde{f}[P](X_1), \ldots, \widetilde{f}[P](X_q)) \to 1] \right|$$
$$\leq \frac{(s + n)q}{L} + 2^{-n}$$

But note that if none of $X_1, \ldots, X_q$ lie in $\mathsf{Dom}(P)$, then $\widetilde{f}[P](X_i) = \widetilde{f}(X_i)$ is a fresh uniformly random value. And by the definition of min-entropy and the union bound, since P is of size L,

we have that the probability that any X_i lies in $\mathsf{Dom}(P)$ is $\leq \frac{qL}{2^\gamma}$. Since $2^{\gamma/2} \leq L \leq 2 \cdot 2^{\gamma/2}$ and $s+n \geq 2$, we have

$$\frac{s+n}{L} + \frac{L}{2^\gamma} \leq \frac{s+n+1}{2^{\gamma/2}}.$$

And so

$$\left| \Pr_f[\mathcal{A}(c(f), f(X_1), \ldots, f(X_q)) \to 1] - \Pr_{f,\widetilde{f}}[\mathcal{A}(c(f), \widetilde{f}(X_1), \ldots, \widetilde{f}(X_q)) \to 1] \right| \leq 2^{-n} + \frac{(s+n+1)q}{2^{\gamma/2}}$$

Giving the sampled points to $\mathcal{A}$ does not change the number of queries made by $\mathcal{D}$ or the probability of hitting P, so the same proof applies. □

3.4 Cryptographic Preliminaries

3.4.1 Proofs of Quantumness

Definition 3.26 (Proof of quantumness [BCM+18, AGKZ20])**.** A proof of quantumness relative to $\mathcal{O}$ is a protocol with classical communication between a QPT prover $P^{\mathcal{O}}$ and a PPT verifier $V^{\mathcal{O}}$, on common input 1^n. The verifier outputs a bit, and the following properties hold:

1. **Completeness.** The honest prover makes the verifier accept with probability

$$\Pr\left[(V^{\mathcal{O}} \leftrightarrows P^{\mathcal{O}})(1^n) \to 1\right] \geq 1 - \mathsf{negl}(n).$$

2. **Classical soundness.** For every nonuniform PPT classical prover $P^{*\mathcal{O}}$,

$$\Pr\left[(V^{\mathcal{O}} \leftrightarrows P^{*\mathcal{O}})(1^n) \to 1\right] \leq \mathsf{negl}(n).$$

The probabilities are over the parties' randomness and measurements. Allowing $V^{\mathcal{O}}$ to be QPT instead gives a quantumly-verifiable proof of quantumness.

Remark 3.27. A proof of quantumness with a classical verifier is *public coin* if every verifier message consists of fresh uniformly random bits, and the verifier's final decision is a deterministic function of 1^n, the transcript, and $\mathcal{O}$ [AGKZ20, Definition 7.1].

3.4.2 Oblivious State Preparation

Definition 3.28 (Oblivious state preparation [BK25])**.** Oblivious state preparation relative to $\mathcal{O}$ is a protocol with classical communication between a PPT verifier $V^{\mathcal{O}}$ and a QPT prover $P^{\mathcal{O}}$, on common input 1^n. The verifier also has an input bit b, outputs a bit s, and the prover outputs one qubit. It satisfies:

1. **Correctness.** For each $b \in \{0,1\}$, measuring the honest prover's output in the computational basis if $b = 0$, or the Hadamard basis if $b = 1$, gives s with probability at least $1 - \mathsf{negl}(n)$.

2. **Security.** For every nonuniform QPT malicious prover $P^{*\mathcal{O}}$ outputting a bit,

$$\left|\Pr\left[P^{*\mathcal{O}} \to 1 \text{ in } \langle V^{\mathcal{O}}(1^n, 0), P^{*\mathcal{O}}(1^n)\rangle\right] - \Pr\left[P^{*\mathcal{O}} \to 1 \text{ in } \langle V^{\mathcal{O}}(1^n, 1), P^{*\mathcal{O}}(1^n)\rangle\right]\right| \leq \mathsf{negl}(n).$$

Allowing the verifier to be QPT instead gives quantum verifier oblivious state preparation (qv-OSP).

[BK25] defines a notion of claw state generators, which are sufficient to construct oblivious state preparation. Intuitively, a claw state generator lets a (classical) verifier produce two strings a, b while giving a quantum prover the claw $\frac{1}{\sqrt{2}}(|a\rangle + |b\rangle)$. Security says that the quantum prover should not be able to produce *both* a and b. We thus define an oracle version of claw state generators, and show how to construct them relative to our oracle. Note that unlike [BK25], we allow the verifier to be quantum, although we restrict communication to be classical.

Definition 3.29 (Claw state generator)**.** A *claw state generator* relative to $\mathcal{O}$ is a protocol with classical communication between quantum polynomial-time parties $\mathsf{V}^{\mathcal{O}}$ and $\mathsf{P}^{\mathcal{O}}$ on input 1^n. The verifier outputs $(a, b) \in \{0,1\}^{6n} \times \{0,1\}^{6n}$ or aborts, and the honest prover outputs a $6n$-qubit register. The protocol satisfies:

1. **Correctness.** Let ρ be the joint output state of an honest execution, and define

$$|\chi_{a,b}\rangle = \frac{|a\rangle + |b\rangle}{\sqrt{2}} \qquad (a \neq b),$$

$$\Pi_{\mathrm{CSG}} = \sum_{a,b \in \{0,1\}^{6n},\ a \neq b} |a,b\rangle\langle a,b| \otimes |\chi_{a,b}\rangle\langle\chi_{a,b}| .$$

   Then $\mathrm{Tr}(\Pi_{\mathrm{CSG}}\rho) \geq 1 - \mathsf{negl}(n)$.

2. **Search security.** For every nonuniform quantum polynomial-time malicious prover $\mathsf{P}^{*\mathcal{O}}$,

$$\Pr\left[\{a', b'\} = \{a, b\} \ : \ \big((a,b),(a',b')\big) \leftarrow \langle \mathsf{V}^{\mathcal{O}}(1^n), \mathsf{P}^{*\mathcal{O}}(1^n)\rangle\right] \leq \mathsf{negl}(n).$$

   The event is false if the verifier aborts; the probability includes all protocol randomness and measurements.

Theorem 3.30 (Adapted from Theorem 4.7 of [BK25])**.** *For all oracles $\mathcal{O}$, if a claw-state generator exists relative to $\mathcal{O}$, then qv-OSP exists relative to $\mathcal{O}$.*

Remark 3.31. Theorem 4.7 of [BK25] states that if claw-state generators with classical verification imply oblivious state preparation with classical verification. We remark that their proof immediately extends to quantum verifiers when we allow the oblivious state preparation protocol to use quantum verification. Their argument is black-box and thus relativizes.

4 A classical oracle with quantum advantage but no cryptography

It is known that there exists an oracle where $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$ and $\mathsf{BQP}^{\mathcal{O}} = P^{\mathsf{PP}^{\mathcal{O}}}$ [AIK22]. The oracle consists of random oracle, a (recursively defined) NP oracle, as well as a (recursively defined) quantum accessible P^{PP} oracle. In particular, classical queries to the latter oracle are prevented by encoding the oracle under the Forrelation problem. For outputs where the oracle would output 1, instead a pair of Forrelated random oracles are provided, and for outputs where the oracle would output 0, instead a pair of independent random oracles are provided.

By replacing P^{PP} with PSPACE, the same proof gives an oracle where $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$, $\mathsf{BQP}^{\mathcal{O}} = \mathsf{PSPACE}^{\mathcal{O}}$, and $\mathsf{NP}^{\mathcal{O}} \subsetneq \mathsf{BQP}^{\mathcal{O}}$. If these complexity collapses held in the standard model (i.e. without oracles), we would get the following template for a proof of quantumness:

Pick a hard problem in BQP. Since $\mathsf{BQP} \subseteq \mathsf{IP}$, there is an interactive proof for this problem. Since $\mathsf{BQP} = \mathsf{PSPACE}$, the interactive proof can be implemented by a QPT algorithm. And so the proof of quantumness will consist of the verifier choosing a problem instance, sending it to the prover, who then answers the problem and gives an interactive proof that its answer is correct. This can be implemented quantumly, but since the problem is hard and the proof is sound it cannot be implemented classically.

There are two main problems with this template. The first issue is that $\mathsf{PSPACE} \subseteq \mathsf{IP}$ (or even $\mathsf{BQP} \subseteq \mathsf{IP}$) does not relativize [BHN+26]. In particular, it may be the case that $\mathsf{IP}^{\mathcal{O}} \subsetneq \mathsf{PSPACE}^{\mathcal{O}}$ relative to the oracle defined by [AIK22]. The second problem is that the hard problem given by [AIK22] is only infinitely often worst-case hard against uniform adversaries. If we want a proof of quantumness to be cryptographically secure, we need a much stronger notion: average case hardness against non-uniform adversaries for all sufficiently large security parameters. In this section, we give an oracle resolving both of these issues.

Theorem 4.1. *There exists a distribution over triples of classical oracles $\mathcal{O} = (\mathsf{A}, \mathsf{B}, \mathsf{C}) \leftarrow \mathcal{D}$ such that with probability 1,*

1. $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$
2. $\mathsf{BQP}^{\mathcal{O}} = \mathsf{PSPACE}^{\mathcal{O}}$
3. $\mathsf{IP}^{\mathsf{A}} = \mathsf{PSPACE}^{\mathsf{A}}$
4. *There exists a constant $\delta \in (0,1)$ and a boolean function $f : \{0,1\}^* \to \{0,1\}$ such that*
   (a) *for all non-uniform PPT algorithms $\mathcal{A}$, for all sufficiently large n,*
   $$\Pr_{x \leftarrow \{0,1\}^n}[\mathcal{A}^{\mathcal{O}}(x) = f(x)] \leq 1 - \delta$$
   (b) *the language $L^f = \{x \in \{0,1\}^* : f(x) = 1\}$ is in $\mathsf{PSPACE}^{\mathsf{A}} \subseteq \mathsf{IP}^{\mathcal{O}}$*

4.1 Our Oracle

Similarly to the proof of Theorem 32 in [AIK22], we define a distribution $\mathcal{D}$ over classical oracles $\mathcal{O} = (\mathsf{A}, \mathsf{B}, \mathsf{C}) \leftarrow \mathcal{D}$ consisting of three parts. We will have $\mathsf{A} = \{\mathsf{A}_t\}_{t\in\mathbb{N}}, \mathsf{B} = \{\mathsf{B}_t\}_{t\in\mathbb{N}}, \mathsf{C} = \{\mathsf{C}_t\}_{t\in\mathbb{N}}$, where B_t and C_t will be defined inductively. We will write $\mathcal{O}_t = (\mathsf{A}_t, \mathsf{B}_t, \mathsf{C}_t)$.

Let $\mathrm{RO}_t : \{0,1\}^t \to \{0,1\}$ be a random oracle. We will write $\mathrm{RO} = \{\mathrm{RO}_t\}$. Define $\mathsf{A}_t(d, x) = \mathsf{ML}_{2^d}[\mathrm{RO}_t](x)$ to be the multilinear extension of RO.

Now, we will describe B_t and C_t, which will enforce that $\mathsf{BQP}^{\mathcal{O}} = \mathsf{PSPACE}^{\mathcal{O}}$ and $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$ respectively.

Denote S_t^{PSPACE} to be the set of ordered pairs (M, x) such that

1. M is an oracle Turing machine and x is an input to M.
2. (M, x) takes less than t bits to specify

3. M is syntactically restricted in the following three ways
   (a) M runs in at most 2^t steps
   (b) M uses space at most t
   (c) M queries only the oracles $\{\mathcal{O}_1, \ldots, \mathcal{O}_{\lfloor\sqrt{t}\rfloor}\}$

For each $(M, x) \in S_t^{\mathsf{PSPACE}}$, sample the function $\mathsf{B}_t((M, x), \cdot) : \{0,1\}^{t^2} \to \{0,1\}$ as follows

1. If $M^{\mathcal{O}}(x)$ accepts, then sample $\mathsf{B}_t((M, x), \cdot)$ from the Forrelation distribution $\mathcal{F}_{2^{t^2}}$
2. If $M^{\mathcal{O}}(x)$ rejects, then sample $\mathsf{B}_t((M, x), \cdot)$ uniformly at random

Thus, $\mathsf{B}_t : S_t^{\mathsf{PSPACE}} \times \{0,1\}^{t^2} \to \{0,1\}$.

Define S_t^{NP} to be the set of ordered pairs (M, x) such that

1. M is a nondeterministic Turing machine and x is an input to M.
2. (M, x) takes less than t bits to specify
3. M is syntactically restricted to run in at most t steps and query only the oracles $(\mathcal{O}_1, \ldots, \mathcal{O}_{\lfloor\sqrt{t}\rfloor})$

Define $\mathsf{C}_t : S_t^{\mathsf{NP}} \to \{0,1\}$ to be

$$\mathsf{C}_t(M, x) = \begin{cases} 1 & M^{\mathcal{O}}(x) \text{ accepts} \\ 0 & \text{otherwise} \end{cases}$$

In order to enforce that a query to $\mathsf{B}_t, \mathsf{C}_t$ requires algorithms to run in time t, we will have $\mathsf{B}_t, \mathsf{C}_t$ ignore padding as well as inputs of shorter than t bits. A query to $\mathsf{B}_t, \mathsf{C}_t$ on an input shorter than t bits we will implicitly take to mean a query on that input padded to t bits.

4.2 Complexity classes collapse under our oracle

We will show the following three important complexity collapses relative to our oracle (or some subset thereof).

Lemma 4.2. *Let $\mathcal{O} = (\mathsf{A}, \mathsf{B}, \mathsf{C})$ be any oracle in the support of $\mathcal{D}$, then*

$$\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$$

Proof. Given an oracle NP machine M running in time at most some computable polynomial $p(|x|)$ and querying only the oracles $(\mathcal{O}_1, \ldots, \mathcal{O}_{p(|x|)})$, a $\mathsf{P}^{\mathcal{O}}$ machine can produce $M^{\mathcal{O}}(x)$ by outputting $\mathsf{C}_t(M, x)$ for $t = \max(|M, x| + 1, p(|x|)^2)$ (where $|M, x|$ is the number of bits needed to specify M and x). □

Lemma 4.3. *With probability 1 over $\mathcal{O} = (\mathsf{A}, \mathsf{B}, \mathsf{C}) \leftarrow \mathcal{D}$,*

$$\mathsf{BQP}^{\mathcal{O}} = \mathsf{PSPACE}^{\mathcal{O}}$$

Proof. We follow the proof of Claim 30 in [AIK22]. Let $Q^{\mathsf{B}_t}(M,x)$ be the quantum algorithm from Lemma 3.10 applied to $\mathsf{B}_t((M,x),\cdot)$, with output 1 corresponding to the Forrelation distribution. For every $(M,x) \in S_t^{\mathsf{PSPACE}}$, its error averaged over $\mathcal{O} \leftarrow \mathcal{D}$ is at most 2^{-2t^2}. By Markov's inequality and a union bound over the at most 2^t pairs in S_t^{PSPACE},

$$\Pr_{\mathcal{O}\leftarrow\mathcal{D}}\left[\exists (M,x) \in S_t^{\mathsf{PSPACE}} : \Pr_Q[Q^{\mathsf{B}_t}(M,x) \neq M^{\mathcal{O}}(x)] > 1/3\right] \leq 3 \cdot 2^{t-2t^2}.$$

Since $\sum_t 3 \cdot 2^{t-2t^2} < \infty$, Borel-Cantelli gives that with probability 1, for all sufficiently large t, Q has error at most $1/3$ on every pair in S_t^{PSPACE}.

Fix such an oracle and a $\mathsf{PSPACE}^{\mathcal{O}}$ machine M. Choose a polynomial p such that $M^{\mathcal{O}}(x)$ uses at most $p(|x|)$ space, runs in time at most $2^{p(|x|)}$, queries only $\mathcal{O}_1,\ldots,\mathcal{O}_{p(|x|)}$, and $|M,x| < p(|x|)$. Set $t = p(|x|)^2$. Then $(M,x) \in S_t^{\mathsf{PSPACE}}$ and $Q^{\mathsf{B}_t}(M,x)$ runs in polynomial time and computes $M^{\mathcal{O}}(x)$ with error at most $1/3$ for all sufficiently long x. Hardcoding the answers on the finitely many remaining inputs gives $\mathsf{PSPACE}^{\mathcal{O}} \subseteq \mathsf{BQP}^{\mathcal{O}}$. The reverse inclusion holds for every classical oracle. □

$\mathsf{PSPACE}^{\mathsf{A}} = \mathsf{IP}^{\mathsf{A}}$ Aaronson and Wigderson [AW08] show that for any oracle R, relative to a low-degree encoding $\widetilde{R}$ of R over a sufficiently large prime field, $\mathsf{PSPACE}^R \subseteq \mathsf{IP}^{\widetilde{R}}$. The way this argument works is by "algebrizing" the existing argument that $\mathsf{PSPACE} = \mathsf{IP}$ proved by Shamir [Sha90]. In particular, Shamir's argument gives an interactive proof for the PSPACE complete language Quantified Boolean Formula (QBF) by utilizing the low-degree extension of the corresponding formulae. Since QBF^R is complete for PSPACE^R, one would like to similarly utilize low-degree extensions of formulae querying the oracle R. Since the verifier can compute these using access to the low-degree extension of R, we get $\mathsf{PSPACE}^R \subseteq \mathsf{IP}^{\widetilde{R}}$.

While [AW08] does not immediately apply to our oracle, since it requires the field to have large prime characteristic, Aydınlıoğlu and Bach [AB17] consider fields of characteristic 2 directly. They show that $\mathsf{PSPACE} \subseteq \mathsf{IP}$ holds relative to every affine oracle. These are oracles of the same form as A, giving access to the multilinear extensions of a family of Boolean functions over $\mathbb{F}_{2^d}$ for every $d \geq 1$, with field elements represented by bit strings.

Theorem 4.4 (Corollary 34 of [AB17])**.** *Let $\{f_n : \{0,1\}^n \to \{0,1\}\}_{n\in\mathbb{N}}$ be any family of functions. Define $\widetilde{f} = \{\mathsf{ML}_{2^d}[f_n]\}_{d,n\in\mathbb{N}}$ to be the collection of multilinear extensions of f_n over $\mathbb{F}_{2^d}$, with field elements represented by bit strings. Each extension has total degree at most n. Then*

$$\mathsf{PSPACE}^{\widetilde{f}} \subseteq \mathsf{IP}^{\widetilde{f}}$$

As our oracle A is an oracle of this form and as $\mathsf{IP} \subseteq \mathsf{PSPACE}$ relativizes, we get the following corollary.

Corollary 4.5. *Let $\mathcal{O} = (\mathsf{A},\mathsf{B},\mathsf{C})$ be any oracle in the support of $\mathcal{D}$, then*

$$\mathsf{IP}^{\mathsf{A}} = \mathsf{PSPACE}^{\mathsf{A}}$$

Remark 4.6. We emphasize that we explicitly do not show that $\mathsf{IP}^{\mathcal{O}} = \mathsf{PSPACE}^{\mathcal{O}}$ relativizes to the entire oracle $\mathcal{O}$; it may be the case that $\mathsf{IP}^{\mathcal{O}} \subsetneq \mathsf{PSPACE}^{\mathcal{O}}$.

4.3 Representing oracle Turing machines as circuits

We begin by describing how to represent polynomial time oracle machines by constant depth circuits. We first restate Lemma 35 from [AIK22], which shows how to represent $\mathsf{NP}^{\mathcal{O}}$ Turing machines by AC^0 circuits. Although their choice of oracle differs slightly from ours (B encodes PSPACE instead of $\mathsf{P}^{\#\mathsf{P}}$ and A gives multilinear extensions instead of direct access), the lemma only relies on the recursive definition of C in terms of A and B, and so can be adapted to our oracle without changing the proof.

Lemma 4.7 (Representing oracle Turing machines as AC^0 circuits)**.** *Fix $t, d \in \mathbb{N}$ and let $t' \leq t^{2^d}$. For each $(M, x) \in S^{\mathsf{NP}}_{t'}$, there exists an $\mathsf{AC}^0(2^{1+t^{2^d}}, 2d)$ circuit that takes as input x, $(\mathsf{A}_1, \ldots, \mathsf{A}_{t^{2^{d-1}}})$, $(\mathsf{B}_1, \ldots, \mathsf{B}_{t^{2^{d-1}}})$, $(\mathsf{C}_1, \ldots, \mathsf{C}_t)$, and outputs $M^{\mathcal{O}}(x)$ for all choices of oracle $\mathcal{O} = (\mathsf{A}, \mathsf{B}, \mathsf{C})$ in the support of $\mathcal{D}$.*

Proof. This follows from the exact same argument as Lemma 35 in [AIK22]. □

We note that since $\mathsf{AC}^0[\oplus]$ circuits can compute A from RO, we get the following corollary.

Corollary 4.8. *Fix $t, d \in \mathbb{N}$ and let $t' \leq t^{2^d}$. For each $(M, x) \in S^{\mathsf{NP}}_{t'}$, there exists an $\mathsf{AC}^0[\oplus](2^{1+t^{2^d}}, 2d+2)$ circuit that takes as input x, $(\mathrm{RO}_1, \ldots, \mathrm{RO}_{t^{2^{d-1}}})$, $(\mathsf{B}_1, \ldots, \mathsf{B}_{t^{2^{d-1}}})$, $(\mathsf{C}_1, \ldots, \mathsf{C}_t)$, and outputs $M^{\mathcal{O}}(x)$ for all choices of oracle $\mathcal{O}$ in the support of $\mathcal{D}$.*

Proof. Note that for each $x \in \mathbb{F}^j_{2^i}$ there is a depth 1 $\mathsf{AC}^0[\oplus]$ circuit which computes $\mathsf{ML}_{2^i}[\mathrm{RO}_j](x)$ when given input RO_j. This is because each output bit of $\mathsf{ML}_{2^i}[\mathrm{RO}_j](x)$ is a linear function of the coordinates of RO_j, and so can be computed by taking a parity of some subset of the oracle positions.

Replacing the A_i input wires of the circuit from Lemma 4.7 with the results of this computation gives an $\mathsf{AC}^0[\oplus]$ circuit with the input scheme given in the lemma. Note that this increases the depth by at most 1 □

4.4 Forrelation is useless for classical computation

In this section, we will prove the following main theorem, which states that from the perspective of any classical algorithm the oracle B can be replaced with truly random functions.

Fix any $n \in \mathbb{N}$. Define $\mathcal{D}^{(n)}$ to be the distribution over oracles defined as follows:

1. For $t < n$, sample $\mathsf{A}_t, \mathsf{B}_t, \mathsf{C}_t$ the same way as in $\mathcal{D}$
2. For $t \geq n$, produce $\mathsf{A}_t, \mathsf{B}_t, \mathsf{C}_t$ exactly the same as when sampling from $\mathcal{D}_t$ with one key difference: always sample $\mathsf{B}_t((M, x), \cdot)$ from the uniform distribution.

Note that the proofs of Lemma 4.7 and Corollary 4.8 do not depend on the choice of RO, B, but only on the structure of A and C. And so they apply just as well to oracles sampled from $\mathcal{D}^{(n)}$.

Lemma 4.9 (Forrelated oracle is useless to classical algorithms)**.** *Let $p(n), q(n), s(n)$ be three polynomials. Let $f^{\mathsf{A}}_n : \{0,1\}^{s(n)} \to \{0,1\}^{q(n)}$ be any family of functions dependent only on the A portion of the oracle. Let $\mathsf{Samp}^{\mathsf{A}}(1^n) \to x \in \{0,1\}^{s(n)}$ be any distribution dependent only on the A*

portion of the oracle. Let $M^{\mathcal{O}}$ be any oracle Turing machine running in time $p(|x|)$. Let $n \in \mathbb{N}$. Then

$$\left| \Pr_{\substack{\mathcal{O}=(\mathsf{A},\mathsf{B},\mathsf{C})\leftarrow\mathcal{D} \\ x\leftarrow\mathsf{Samp}^{\mathsf{A}}(1^n)}}[M^{\mathcal{O}}(x) = f_n^{\mathsf{A}}(x)] - \Pr_{\substack{\mathcal{O}=(\mathsf{A},\mathsf{B},\mathsf{C})\leftarrow\mathcal{D}^{(n)} \\ x\leftarrow\mathsf{Samp}^{\mathsf{A}}(1^n)}}[M^{\mathcal{O}}(x) = f_n^{\mathsf{A}}(x)] \right| \leq 2^{-\Omega(n^2)}$$

Proof. Let $T(n) = p(s(n))$. This is a polynomial bound on the running time of $M^{\mathcal{O}}$ on inputs of length $s(n)$.

First, observe that

$$\Pr_{\substack{\mathcal{O}=(\mathsf{A},\mathsf{B},\mathsf{C})\leftarrow\mathcal{D} \\ x\leftarrow\mathsf{Samp}^{\mathsf{A}}(1^n)}}[M^{\mathcal{O}}(x) = f_n^{\mathsf{A}}(x)] = \Pr_{\substack{\mathcal{O}=(\mathsf{A},\mathsf{B},\mathsf{C})\leftarrow\mathcal{D}^{(T(n)+1)} \\ x\leftarrow\mathsf{Samp}^{\mathsf{A}}(1^n)}}[M^{\mathcal{O}}(x) = f_n^{\mathsf{A}}(x)]$$

since $M^{\mathcal{O}}$ does not run in time large enough to query $\mathcal{O}_{T(n)+1}$. We will then proceed to show that for all $i \in [n, T(n)]$,

$$\Pr_{\substack{\mathcal{O}=(\mathsf{A},\mathsf{B},\mathsf{C})\leftarrow\mathcal{D}^{(i)} \\ x\leftarrow\mathsf{Samp}^{\mathsf{A}}(1^n)}}[M^{\mathcal{O}}(x) = f_n^{\mathsf{A}}(x)] \approx \Pr_{\substack{\mathcal{O}=(\mathsf{A},\mathsf{B},\mathsf{C})\leftarrow\mathcal{D}^{(i+1)} \\ x\leftarrow\mathsf{Samp}^{\mathsf{A}}(1^n)}}[M^{\mathcal{O}}(x) = f_n^{\mathsf{A}}(x)]$$

For $i \in [n, T(n)]$, we enumerate all pairs in S_i^{PSPACE} as $(M_{i,1}, x_{i,1}), \ldots, (M_{i,s_i}, x_{i,s_i})$, where $s_i \leq 2^i$. Define $H_j^{(i)}$ to be the same as $\mathcal{D}^{(i+1)}$ but where $\mathsf{B}_i((M_{i,1}, x_{i,1}), \cdot), \ldots, \mathsf{B}_i((M_{i,j}, x_{i,j}), \cdot)$ are sampled uniformly at random instead of from the Forrelation distribution.

Now, fix any i, j. Let $t = i$ and $t' = T(i)^2$. Note that there exists a constant d such that $t' \leq t^{2^d}$. Set $N = 2^{i^2}$. By Lemma 4.7 there is an $\mathsf{AC}^0(\mathsf{quasipoly}(N), O(1))$ circuit $\mathcal{C}$ which takes in as input x, $(\mathsf{A}_1, \ldots, \mathsf{A}_{t^{2^{d-1}}})$, $(\mathsf{B}_1, \ldots, \mathsf{B}_{t^{2^{d-1}}})$, $(\mathsf{C}_1, \ldots, \mathsf{C}_t)$ and outputs $M^{\mathcal{O}}(x)$. Now consider the circuit $\mathcal{C}'$ which additionally takes as input $y = f_n^{\mathsf{A}}(x)$ and outputs 1 if and only if $M^{\mathcal{O}}(x) = y$. This can also be implemented in $\mathsf{AC}^0(\mathsf{quasipoly}(N), O(1))$ by running $\mathcal{C}$ and comparing the output with the input y.

Conditioned on $M_{i,j}^{\mathcal{O}}(x_{i,j}) = 0$, $H_j^{(i)}$ and $H_{j-1}^{(i)}$ are identically distributed. And so define H_1^*, H_2^* to be $H_j^{(i)}$ and $H_{j-1}^{(i)}$ conditioned on $M_{i,j}^{\mathcal{O}}(x_{i,j}) = 1$. Observe that the only difference between H_1^* and H_2^* is that in H_1^*, $\mathsf{B}_i((M_{i,j}, x_{i,j}), \cdot)$ is uniformly random and in H_2^*, $\mathsf{B}_i((M_{i,j}, x_{i,j}), \cdot)$ is sampled from $\mathcal{F}_N$.

Further observe that since $i = t > \sqrt{t}$, when drawn from either of H_1^*, H_2^* all portions of the

input to $\mathcal{C}'$ besides $\mathsf{B}_i((M_{i,j}, x_{i,j}), \cdot)$ are independent of $\mathsf{B}_i((M_{i,j}, x_{i,j}), \cdot)$. And so by Lemma 3.10,

$$\begin{aligned}
&\left| \Pr_{\mathcal{O} \leftarrow H_j^{(i)}}[M^{\mathcal{O}}(x) = f_n^{\mathsf{A}}(x)] - \Pr_{\mathcal{O} \leftarrow H_{j-1}^{(i)}}[M^{\mathcal{O}}(x) = f_n^{\mathsf{A}}(x)] \right| \\
\leq &\left| \Pr_{\mathcal{O} \leftarrow H_1^*}[M^{\mathcal{O}}(x) = f_n^{\mathsf{A}}(x)] - \Pr_{\mathcal{O} \leftarrow H_2^*}[M^{\mathcal{O}}(x) = f_n^{\mathsf{A}}(x)] \right| \\
\leq &\left| \Pr_{\mathcal{O} \leftarrow H_1^*}[\mathcal{C}'(x, f_n^{\mathsf{A}}(x), (\mathsf{A}_1, \ldots, \mathsf{A}_{t^{2^{d-1}}}), (\mathsf{B}_1, \ldots, \mathsf{B}_{t^{2^{d-1}}}), (\mathsf{C}_1, \ldots, \mathsf{C}_t)) \to 1] \right. \\
&\left. - \Pr_{\mathcal{O} \leftarrow H_2^*}[\mathcal{C}'(x, f_n^{\mathsf{A}}(x), (\mathsf{A}_1, \ldots, \mathsf{A}_{t^{2^{d-1}}}), (\mathsf{B}_1, \ldots, \mathsf{B}_{t^{2^{d-1}}}), (\mathsf{C}_1, \ldots, \mathsf{C}_t)) \to 1] \right| \\
\leq &\frac{\mathsf{polylog}(N)}{\sqrt{N}} = \frac{\mathsf{poly}(i)}{2^{i^2/2}}
\end{aligned} \tag{3}$$

Summing over all $i \in [n, T(n)], j \in [1, s_i]$, we get

$$\begin{aligned}
&\left| \Pr_{\substack{\mathcal{O}=(\mathsf{A},\mathsf{B},\mathsf{C}) \leftarrow \mathcal{D} \\ x \leftarrow \mathsf{Samp}^{\mathsf{A}}(1^n)}}[M^{\mathcal{O}}(x) = f_n^{\mathsf{A}}(x)] - \Pr_{\substack{\mathcal{O}=(\mathsf{A},\mathsf{B},\mathsf{C}) \leftarrow \mathcal{D}^{(n)} \\ x \leftarrow \mathsf{Samp}^{\mathsf{A}}(1^n)}}[M^{\mathcal{O}}(x) = f_n^{\mathsf{A}}(x)] \right| \\
\leq &\sum_{i=n}^{T(n)} 2^i \cdot \frac{\mathsf{poly}(i)}{2^{i^2/2}} \leq \sum_{i=n}^{T(n)} \frac{\mathsf{poly}(n)}{2^{n^2/2-n}} = \frac{1}{2^{\Omega(n^2)}}
\end{aligned} \tag{4}$$

□

Remark 4.10. Note that the same theorem applies for M a randomized Turing machine by augmenting the output of $\mathsf{Samp}^{\mathsf{A}}$ with random coins ignored by f_n^{A}.

4.5 Majority is average case hard

We will show that the problem of salted majority is hard on average, even against non-uniform adversaries. In particular, let $\mathcal{O} = (\mathsf{A}, \mathsf{B}, \mathsf{C})$ be an oracle in the support of $\mathcal{D}$ with A representing the multilinear extension of $\{\mathrm{RO}_n\}_{n \in \mathbb{N}}$. We will define the salted majority problem to be the following function $h^{\mathrm{RO}} : \{0,1\}^* \to \{0,1\}$ defined by

$$h^{\mathrm{RO}}(x) = 1 \iff |\{y \in \{0,1\}^{|x|} : \mathrm{RO}_{2|x|}(x, y) = 1\}| > 2^{|x|}/2$$

In particular, $h^{\mathrm{RO}}(x)$ is the majority problem on the function $\mathrm{RO}_{2|x|}(x, \cdot)$.

We will also consider the parallel repetition of the salted majority problem. In particular, define

$$H_r^{\mathrm{RO}}(x_1, \ldots, x_r) = (h^{\mathrm{RO}}(x_1), \ldots, h^{\mathrm{RO}}(x_r))$$

For $\mathcal{O}$ defined using RO, we also write $h^{\mathcal{O}} := h^{\mathrm{RO}}$ and $H_r^{\mathcal{O}} := H_r^{\mathrm{RO}}$.

Note that for AC^0 circuits taking as input just RO, H^{RO} is very hard.

Lemma 4.11. *There is a universal constant $\beta > 0$ such that, for all fixed d and polynomials p, q, the following holds. Let $r \le p(n)$ and let $\mathcal{C}$ be a distribution over $\mathsf{AC}^0[\oplus](2^{q(n)}, d)$ circuits. Then for all sufficiently large n,*

$$\Pr_{C \leftarrow \mathcal{C},\, \mathrm{RO},\, x_1,\ldots,x_r \leftarrow \{0,1\}^n} [C(x_1, \ldots, x_r, \mathrm{RO}) = H_r^{\mathrm{RO}}(x_1, \ldots, x_r)] \le e^{-\beta r}$$

Proof. Let

$$\epsilon = \Pr_{C \leftarrow \mathcal{C},\, \mathrm{RO},\, x_1,\ldots,x_r \leftarrow \{0,1\}^n} [C(x_1, \ldots, x_r, \mathrm{RO}) = H_r^{\mathrm{RO}}(x_1, \ldots, x_r)]$$

and assume towards contradiction that $\epsilon > e^{-\beta r}$.

We can transform $\mathcal{C}$ into a distribution over circuits C' which solves $\mathsf{MAJ}^{\otimes \lceil r/2 \rceil}$ with probability at least $\epsilon - (er/2^n)^{\lceil r/2 \rceil}$ as follows.

1. On input $Y_1, \ldots, Y_{\lceil r/2 \rceil} \in \{0,1\}^{2^n}$
2. Sample $C \leftarrow \mathcal{C}$. Sample $x_1, \ldots, x_r \in \{0,1\}^n$ uniformly at random
3. If there are less than $\lceil r/2 \rceil$ distinct values of x_i, reject.
4. Let $\{x'_1, \ldots, x'_{\lceil r/2 \rceil}\}$ be the first $\lceil r/2 \rceil$ distinct values of $x_1, \ldots, x_r$.
5. For all $t \neq 2n$, sample RO_t uniformly at random.
6. For all $x' \notin \{x'_1, \ldots, x'_{\lceil r/2 \rceil}\}$, sample $\mathrm{RO}_{2n}(x', \cdot)$ uniformly at random.
7. For all $i \in [\lceil r/2 \rceil]$, set $\mathrm{RO}_{2n}(x'_i, \cdot) = Y_i$
8. Run C on $(x_1, \ldots, x_r, \mathrm{RO})$ and output the response on $x'_1, \ldots, x'_{\lceil r/2 \rceil}$.

This can be implemented by an $\mathsf{AC}^0[\oplus]$ circuit of the same size and depth as C.

Let $m = \lceil r/2 \rceil$ and let K be the number of distinct sampled salts. We have

$$\begin{aligned}
&\Pr_{C', Y_1, \ldots, Y_m} [C'(Y_1, \ldots, Y_m) = (\mathsf{MAJ}(Y_1), \ldots, \mathsf{MAJ}(Y_m))] \\
&\quad \ge \Pr_{C \leftarrow \mathcal{C},\, \mathrm{RO},\, x_1,\ldots,x_r} [C(x_1, \ldots, x_r, \mathrm{RO}) = H_r^{\mathrm{RO}}(x_1, \ldots, x_r) \text{ and } K \ge m] \\
&\quad \ge \epsilon - \left(\frac{er}{2^n}\right)^m.
\end{aligned}$$

where $\left(\frac{er}{2^n}\right)^{\lceil r/2 \rceil}$ is an upper bound on the probability that there are fewer than $\lceil r/2 \rceil$ distinct samples.

Set $\delta = 1/4$ and choose a universal $0 < \beta < c\delta/2$, where c is from Theorem 3.12. Since $r \le p(n)$, the collision term is at most $e^{-\Omega(nm)}$. As $r \le 2m$, for all sufficiently large n we have

$$\epsilon - \left(\frac{er}{2^n}\right)^m > e^{-\beta r} - \left(\frac{er}{2^n}\right)^m \ge e^{-c\delta m}.$$

Fix the randomness of C' so that its success probability is at least this bound. Applying Theorem 3.12 with $t = m$ and input length $N = 2^n$, then substituting C' for its oracle gates, gives a constant-depth $\mathsf{AC}^0[\oplus]$ circuit for majority with agreement at least $3/4$ and size at most

$$S' = 2^{q(n)} \cdot \mathsf{poly}(2^n, m, \log(1/\delta), e^{c\delta m}) = 2^{\mathsf{poly}(n)}.$$

Since $\mathsf{polylog}(S') = \mathsf{poly}(n)$, by Theorem 3.11 every circuit of this size has agreement at most $\frac{1}{2} + \frac{\mathsf{poly}(n)}{2^{n/2}} < \frac{3}{4}$ for sufficiently large n. This contradicts the agreement of the reconstructed circuit. □

Theorem 4.12 (Weak hardness of salted majority)**.** *There exists a universal constant δ such that with probability 1 over $\mathcal{O} \leftarrow \mathcal{D}$ the following holds. Let $p(n)$ be a polynomial. Let $\{C_n^{\mathcal{O}}\}_{n\in\mathbb{N}}$ be a family of circuits of size $p(n)$. Then, for all sufficiently large n,*

$$\Pr_{x\leftarrow\{0,1\}^n}[C_n^{\mathcal{O}}(x) = h^{\mathcal{O}}(x)] \leq 1 - \delta$$

Proof. Fix some polynomial $p(n)$. Let $BAD^{\mathcal{O}}_{n,p,\delta}$ be the event (over randomness the choice in oracle) that for some circuit $C^{\mathcal{O}}$ of size $\leq p(n)$ we have

$$\Pr_{x\leftarrow\{0,1\}^n}[C^{\mathcal{O}}(x) = h^{\mathcal{O}}(x)] \geq 1 - \delta$$

By padding, we can assume without loss of generality that the size of $C^{\mathcal{O}}$ is $p(n)$. For simplicity, we will say that the number of circuits of size $p(n)$ is at most $2^{p(n)}$. In fact, this is true for a larger polynomial $p'(n)$, and the same proof will follow by replacing $p(n)$ with $p'(n)$.

It now suffices to show that there exists some constant δ such that for all polynomials p, the probability over $\mathcal{O} \leftarrow \mathcal{D}$ that $BAD^{\mathcal{O}}_{n,p,\delta}$ occurs infinitely often is 0.

Removing the Forrelated part of the oracle Sample a new oracle $\mathcal{O}' \leftarrow \mathcal{D}^{(2n)}$. We will first show that for all sufficiently large n,

$$\Pr_{\mathcal{O}\leftarrow\mathcal{D}}[BAD^{\mathcal{O}}_{n,p,\delta}] \leq \Pr_{\mathcal{O}\leftarrow\mathcal{D}^{(2n)}}[BAD^{\mathcal{O}}_{n,p,3\delta}] + 2^{-\Omega(n^2)} \tag{5}$$

In particular, let $\ell \in \mathbb{N}$ be values to be set later and define $\mathsf{Test}^{\mathcal{O}}$ to be the following Turing machine: "On input $(k_1, \ldots, k_\ell, s_1, \ldots, s_\ell)$, output 1 if and only if there is an oracle circuit $C^{\mathcal{O}}$ of size $p(n)$ such that $C^{\mathcal{O}}(k_i) = s_i$ for at least $1 - 2\delta$ fraction of $i \in [\ell]$."

Note that since $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$, this Turing machine can run in time $\mathsf{poly}(n, \ell)$. Let $\mathsf{Samp}^{\mathsf{A}}(1^n)$ be the distribution which samples $(k_1, \ldots, k_\ell)$ uniform random distinct elements from $\{0,1\}^n$, and sets $s_i = h^{\mathcal{O}}(k_i)$.

For an oracle $\mathcal{O}$ and an oracle circuit $C^{(\cdot)}$, define

$$a(\mathcal{O}, C^{(\cdot)}) = \Pr_{k\leftarrow\{0,1\}^n}[C^{\mathcal{O}}(k) = h^{\mathcal{O}}(k)]$$

For a sequence $\vec{k} = (k_1, \ldots, k_\ell)$, define

$$\widetilde{a}(\mathcal{O}, C^{(\cdot)}, \vec{k}) = \Pr_{i\leftarrow[\ell]}[C^{\mathcal{O}}(k_i) = h^{\mathcal{O}}(k_i)]$$

Hoeffding's inequality gives that for all oracles $\mathcal{O}$, circuits $C^{(\cdot)}$,

$$\Pr_{(k_1,\ldots,k_\ell)\text{ distinct}}[|a(\mathcal{O}, C^{(\cdot)}) - \widetilde{a}(\mathcal{O}, C^{(\cdot)}, \vec{k})| > \delta] \leq 2e^{-2\ell\delta^2}$$

Union bound then says that with probability $\geq 1 - 2^{p(n)+1}e^{-2\ell\delta^2}$ over distinct. random $(k_1, \ldots, k_\ell)$, for all circuits $C^{(\cdot)}$ of size at most $p(n)$,

$$|a(\mathcal{O}, C^{(\cdot)}) - \widetilde{a}(\mathcal{O}, C^{(\cdot)}, \vec{k})| \leq \delta$$

Let $\epsilon = 2^{p(n)+1}e^{-2\ell\delta^2}$. If $BAD^{\mathcal{O}}_{n,p,\delta}$ is true, then there exists a circuit such that $a(\mathcal{O}, C^{(\cdot)}) \geq 1-\delta$. With probability $\geq 1-\epsilon$ over the choice of distinct $(k_1, \ldots, k_\ell)$, for that $C^{\mathcal{O}}$ we have

$$\widetilde{a}(\mathcal{O}, C^{(\cdot)}, \vec{k}) \geq 1 - 2\delta$$

Thus, $\Pr_{X \leftarrow \mathsf{Samp}^{\mathsf{A}}}[\mathsf{Test}^{\mathcal{O}}(X) \to 1] \geq 1 - \epsilon$.

Similarly, if $BAD^{\mathcal{O}}_{n,p,3\delta}$ is false, then for all circuits $C^{(\cdot)}$ of size at most $p(n)$, $a(\mathcal{O}, C^{(\cdot)}) < 1 - 3\delta$. So with probability $\geq 1-\epsilon$ over the choice of distinct $(k_1, \ldots, k_\ell)$, for all circuits $C^{\mathcal{O}}$ we have

$$\widetilde{a}(\mathcal{O}, C^{(\cdot)}, \vec{k}) < 1 - 2\delta$$

Thus, $\Pr_{X \leftarrow \mathsf{Samp}^{\mathsf{A}}}[\mathsf{Test}^{\mathcal{O}}(X) \to 1] \leq \epsilon$.

In particular, we have for all distributions $\mathcal{R}$ over oracles,

$$\begin{aligned}\Pr_{\mathcal{O} \leftarrow \mathcal{R}}[BAD^{\mathcal{O}}_{n,p,\delta}] - \epsilon &\leq \Pr_{\mathcal{O} \leftarrow \mathcal{R},\, X \leftarrow \mathsf{Samp}^{\mathsf{A}}(1^n)}[\mathsf{Test}^{\mathcal{O}}(X) = 1] \\ &\leq \Pr_{\mathcal{O} \leftarrow \mathcal{R}}[BAD^{\mathcal{O}}_{n,p,3\delta}] + \epsilon.\end{aligned}$$

By Lemma 4.9 with parameter $2n$,

$$\left| \Pr_{\substack{\mathcal{O} \leftarrow \mathcal{D} \\ X \leftarrow \mathsf{Samp}^{\mathsf{A}}(1^n)}}[\mathsf{Test}^{\mathcal{O}}(X) = 1] - \Pr_{\substack{\mathcal{O} \leftarrow \mathcal{D}^{(2n)} \\ X \leftarrow \mathsf{Samp}^{\mathsf{A}}(1^n)}}[\mathsf{Test}^{\mathcal{O}}(X) = 1] \right| \leq 2^{-\Omega(n^2)}$$

Taking $\ell = \Omega(n^2 p(n)/\delta^2)$ gives $\epsilon = 2^{-\Omega(n^2)}$. And so we get Equation (5).

Bounding $\Pr_{\mathcal{O} \leftarrow \mathcal{D}^{(2n)}}[BAD_{n,p,3\delta}]$ Let $\mathcal{O}$ be such that $BAD^{\mathcal{O}}_{n,p,3\delta}$ holds. Define a Turing machine M solving $H^{\mathcal{O}}_r$ (for some r to be set later) as follows

1. On input $x_1, \ldots, x_r \in \{0,1\}^n$
2. Pick a circuit $C^{\mathcal{O}}$ of size $p(n)$ uniformly at random
3. For each $i \in [r]$, run $C^{\mathcal{O}}(x_i) \to b_i$
4. Output $(b_1, \ldots, b_r)$

Note that if $BAD^{\mathcal{O}}_{n,p,3\delta}$ holds, then with probability at least $2^{-p(n)}$ $C^{\mathcal{O}}$ will solve salted majority with probability $1 - 3\delta$. Since the x_is are sampled independently, we get

$$\Pr_{x_1,\ldots,x_r \leftarrow \{0,1\}^n}[M^{\mathcal{O}}(x_1, \ldots, x_r) = H^{\mathcal{O}}_r(x_1, \ldots, x_r)] \geq 2^{-p(n)} \cdot (1-3\delta)^r$$

And so in particular we get

$$\Pr_{\substack{\mathcal{O} \leftarrow \mathcal{D}^{(2n)} \\ x_1,\ldots,x_r \leftarrow \{0,1\}^n}}[M^{\mathcal{O}}(x_1, \ldots, x_r) = H^{\mathcal{O}}_r(x_1, \ldots, x_r)] \geq 2^{-p(n)} \cdot (1-3\delta)^r \cdot \Pr_{\mathcal{O} \leftarrow \mathcal{D}^{(2n)}}[BAD^{\mathcal{O}}_{n,p,3\delta}]$$

Now, we apply Corollary 4.8 to get an $\mathsf{AC}^0[\oplus]$ circuit solving salted majority. In particular, for $\mathcal{O} = (\mathsf{A}, \mathsf{B}, \mathsf{C}) \leftarrow \mathcal{D}^{(2n)}$, with A the multilinear extension of RO, there exists some constant d (depending on $p(n)$) such that we have an $\mathsf{AC}^0[\oplus](2^{1+n^{2^d}}, 2d+2)$ circuit C' taking in input

$$x_1, \ldots, x_r, (\mathrm{RO}_1, \ldots, \mathrm{RO}_{n^{2^{d-1}}}), (\mathsf{B}_1, \ldots, \mathsf{B}_{n^{2^{d-1}}}), (\mathsf{C}_1, \ldots, \mathsf{C}_n)$$

such that

$$\Pr_{\substack{\mathcal{O} \leftarrow \mathcal{D}^{(2n)} \\ x_1, \ldots, x_r \leftarrow \{0,1\}^n}} [C'(x_1, \ldots, x_r) = H_r^{\mathcal{O}}(x_1, \ldots, x_r)] \geq 2^{-p(n)} \cdot (1-3\delta)^r \cdot \Pr_{\mathcal{O} \leftarrow \mathcal{D}^{(2n)}} [BAD^{\mathcal{O}}_{n,p,3\delta}]$$

But note that since $\mathcal{O}$ is drawn from $\mathcal{D}^{(2n)}$, the auxiliary input of C' (corresponding to the desciption of the oracles B and C) is independent of RO_{2n}. And so by Lemma 4.11 we get that for all sufficiently large n,

$$2^{-p(n)} \cdot (1-3\delta)^r \cdot \Pr_{\mathcal{O} \leftarrow \mathcal{D}^{(2n)}} [BAD_{n,p,3\delta}] \leq e^{-\beta r}$$

Therefore,

$$\Pr_{\mathcal{O} \leftarrow \mathcal{D}^{(2n)}} [BAD^{\mathcal{O}}_{n,p,3\delta}] \leq 2^{p(n)} \cdot (1-3\delta)^{-r} \cdot e^{-\beta r}$$

Setting $r = \lceil 2n^2 \cdot p(n)/\beta \rceil$ and δ such that $(1-3\delta) = e^{-\beta/2}$, we get

$$\Pr_{\mathcal{O} \leftarrow \mathcal{D}} [BAD^{\mathcal{O}}_{n,p,\delta}] \leq \Pr_{\mathcal{O} \leftarrow \mathcal{D}^{(2n)}} [BAD^{\mathcal{O}}_{n,p,3\delta}] + 2^{-\Omega(n^2)} \leq 2^{p(n)} \cdot e^{-\beta r/2} + 2^{-\Omega(n^2)} = 2^{-\Omega(n^2)}$$

We then have

$$\sum_{n=1}^{\infty} \Pr_{\mathcal{O} \leftarrow \mathcal{D}} [BAD^{\mathcal{O}}_{n,p,\delta}] = \sum_{n=1}^{\infty} 2^{-\Omega(n^2)} < \infty$$

and so by Borel-Cantelli, for all polynomials p,

$$\Pr_{\mathcal{O} \leftarrow \mathcal{D}} [BAD^{\mathcal{O}}_{n,p,\delta} \text{ occurs infinitely often}] = 0$$

Note that every polynomial p is bounded by $(n+2)^c$ for some constant c. And so taking the union bound over the countable set of all such polynomials we get that with probability 1 over $\mathcal{O} \leftarrow \mathcal{D}$, for all polynomials p, $BAD^{\mathcal{O}}_{n,p,\delta}$ does not occur infinitely often. □

4.6 Applications

Proofs of Quantumness

Corollary 4.13. *With probability* 1 *over* $\mathcal{O} \leftarrow \mathcal{D}$*, there exists a public coin, efficiently verifiable proof of quantumness.*

Proof. Let δ be as in Theorem 4.12.

Consider the following construction

1. Repeat the following trial n times. If all trials pass, the verifier accepts, otherwise it rejects.
2. In trial i, the verifier samples $k_i \leftarrow \{0,1\}^n$ and sends k_i to the prover.

3. The prover responds with $z_i = h^{\mathcal{O}}(k_i)$.

4. Then the prover and verifier engage in a public-coin interactive proof that $z_i = h^{\mathcal{O}}(k_i)$, with a $\mathsf{BQP}^{\mathcal{O}}$ honest prover, amplified to make its soundness error at most $\delta/2$ and its completeness error negligible.

5. If the verifier accepts the proof, then the trial is passed. Otherwise, the verifier rejects.

Correctness follows from the fact that $h^{\mathcal{O}} \in \mathsf{PSPACE}^{\mathrm{RO}} \subseteq \mathsf{IP}^{\mathsf{A}}$. Note that all languages in IP^{A} have an interactive proof with a prover implementable in $\mathsf{PSPACE}^{\mathsf{A}}$ that is public-coin, has negligible completeness error, and arbitrary constant soundness error [GS86]. Since $\mathsf{PSPACE}^{\mathsf{A}} \subseteq \mathsf{BQP}^{\mathcal{O}}$, the honest prover can be implemented by a QPT oracle machine.

By viewing the previous rounds and the prover's state as classical advice, Theorem 4.12 gives that, conditioned on the previous trials passing, a classical prover finds $z_i = h^{\mathcal{O}}(k_i)$ with probability at most $1 - \delta$. If $z_i \neq h^{\mathcal{O}}(k_i)$, the interactive proof accepts with probability at most $\delta/2$. Thus each trial passes with probability at most $1 - \delta/2$, conditioned on the previous trials passing, and the probability that all trials pass is at most $(1 - \delta/2)^n = \mathsf{negl}(n)$. □

Theorem 4.14 (Stockmeyer's approximate counting [Sto83])**.** *Let $\mathcal{O}$ be a classical oracle such that $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$, let $R^{\mathcal{O}}(x, y)$ be a deterministic polynomial-time oracle predicate, and let m be a polynomial. Define*

$$N(x) = \left|\{y \in \{0,1\}^{m(|x|)} : R^{\mathcal{O}}(x, y) = 1\}\right|.$$

For any inverse-polynomial accuracy parameter $\alpha \in (0, 1)$, there exists a deterministic polynomial-time algorithm with oracle access to $\mathcal{O}$ outputting an estimate $\widehat{N}(x)$ satisfying

$$(1 - \alpha)N(x) \leq \widehat{N}(x) \leq (1 + \alpha)N(x)$$

for all x. In particular, zero counts are recognized exactly.

The following corollary is folklore, but we provide the full proof for completeness.

Corollary 4.15 (Classical provers for constant-round protocols)**.** *Let $\mathcal{O}$ be a classical oracle such that $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$. Let $V^{\mathcal{O}}$ be a classical probabilistic polynomial-time verifier in a constant-round protocol with classical communication. Then there exists a deterministic polynomial-time classical prover $\overline{P}^{\mathcal{O}}$ such that, for every input x and every prover $P^{\mathcal{O}}$ (including quantum and inefficient provers),*

$$\Pr\left[(V^{\mathcal{O}} \leftrightarrows \overline{P}^{\mathcal{O}})(x) \to 1\right] \geq \Pr\left[(V^{\mathcal{O}} \leftrightarrows P^{\mathcal{O}})(x) \to 1\right] - 0.01.$$

Proof. Let c be the number of prover messages used in the protocol. Fix a polynomial bound $m(|x|)$ on the randomness of the verifier. Let T_i be a partial transcript after i prover messages and any following verifier message. Say that coins r are consistent with (x, T_i) if $V^{\mathcal{O}}(x, r)$ produces the verifier messages in T_i when receiving the prover messages in T_i. We consider transcripts with at least one consistent choice of coins. Define

$$\mathsf{OPT}(x, T_i) = \max_{P^{\mathcal{O}}} \Pr\left[V^{\mathcal{O}}(x, r) \leftrightarrows P^{\mathcal{O}} \to 1 | r \text{ consistent with } (x, T_i)\right]$$

Here T_i is fixed, and $P^{\mathcal{O}}$ chooses only the remaining messages.

Set $\nu_i = \frac{1}{100 \cdot 3^i}$. We will inductively define deterministic strategies $S_i(x, T_i)$ taking in a transcript for the first i rounds and producing a strategy for the remaining. Let $p_i(x, T_i)$ be the probability over r consistent with (x, T_i) that a verifier receiving the prover messages in T_i for the first i rounds and $S_i(x, T_i)$ for the remaining rounds outputs 1. We will prove that for all $i \in \{0, \ldots, c\}, x, T_i$,

$$p_i(x, T_i) \geq \mathsf{OPT}(x, T_i) - \nu_i$$

Define S_c to do nothing. The inductive hypothesis holds since the prover sends no messages after round c.

Now, for each partial transcript T_i and for each message a sent by the prover in round $i+1$, call r "good" for (x, T_i) if (x, r) is consistent with T_i and $V(x, r)$ accepts when receiving prover messages T_i, then a, then the messages produced by strategy S_{i+1}. Define

$$N_{x,T_i}(a) = \left| \{r \in \{0,1\}^{m(|x|)} | r \text{ is "good"} \} \right|$$

Note that given x, T_i, we can check if r is "good" in polynomial time by simply running the verifier with the corresponding strategy. Here, S_{i+1} runs in polynomial time by the inductive hypothesis.

Define $\widetilde{N}_{x,T_i}(a)$ be the estimate for $N_{x,T_i}(a)$ produced by Theorem 4.14 with parameter ν_{i+1}. Since $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$ and since $i + 1 \leq c$ is a constant, this is computable in deterministic polynomial time and for all a, x, T_i,

$$(1 - \nu_{i+1}) N_{x,T_i}(a) \leq \widetilde{N}_{x,T_i}(a) \leq (1 + \nu_{i+1}) N_{x,T_i}(a)$$

Define a^* be any value maximizing $N_{x,T_i}(a)$ and let $\widetilde{a}^*$ be any value maximizing $\widetilde{N}_{x,T_i}(a)$. Note that since $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$ and $\widetilde{N}_{x,T_i}(a)$ is computable in polynomial time, $\widetilde{a}^*$ is also computable in polynomial time. Furthermore, we can bound $N_{x,T_i}(a^*)$ by $N_{x,T_i}(\widetilde{a}^*)$ as follows

$$\begin{aligned} N_{x,T_i}(a^*) &\leq \frac{1}{1 - \nu_{i+1}} \widetilde{N}_{x,T_i}(a^*) \\ &\leq \frac{1}{1 - \nu_{i+1}} \widetilde{N}_{x,T_i}(\widetilde{a}^*) \\ &\leq \frac{1 + \nu_{i+1}}{1 - \nu_{i+1}} N_{x,T_i}(\widetilde{a}^*) \end{aligned}$$

We then define S_i to output in round $i+1$ message $\widetilde{a}^*$, and then run S_{i+1} for the remaining rounds. By the inductive hypothesis, averaging over the next verifier message, $N_{x,T_i}(a^*)$ divided by the number of consistent coins is at least $\mathsf{OPT}(x, T_i) - \nu_{i+1}$. Thus,

$$\begin{aligned} p_i(x, T_i) &\geq \frac{1 - \nu_{i+1}}{1 + \nu_{i+1}} \big(\mathsf{OPT}(x, T_i) - \nu_{i+1} \big) \\ &\geq \mathsf{OPT}(x, T_i) - 3\nu_{i+1} \\ &= \mathsf{OPT}(x, T_i) - \nu_i \end{aligned} \tag{6}$$

And so setting $\overline{P}^{\mathcal{O}} = S_0$ completes the proof. □

Thus, since with probability 1 over $\mathcal{O} \leftarrow \mathcal{D}$ we have $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$, every constant-round proof of quantumness has an efficient classical prover and so therefore cannot be secure.

Corollary 4.16. *With probability* 1 *over* $\mathcal{O} \leftarrow \mathcal{D}$*, there does not exist a constant-round proof of quantumness relative to* $\mathcal{O}$*.*

Combining Theorems 3.12 and 4.12 we get the following corollary.

Corollary 4.17. *With probability* 1 *over* $\mathcal{O} \leftarrow \mathcal{D}$*, there exists a two message quantumly-verifiable proof of quantumness.*

Proof. The verifier samples $k_1, \ldots, k_n \leftarrow \{0,1\}^n$ and sends them to the prover, who responds with $z_1, \ldots, z_n$. The verifier accepts if and only if for all i, $z_i = h^{\mathcal{O}}(k_i)$. This can be computed efficiently since $h^{\mathcal{O}} \in \mathsf{PSPACE}^{\mathcal{O}} \subseteq \mathsf{BQP}^{\mathcal{O}}$. Repeat each computation enough times that the probability of any incorrect check is negligible.

Suppose a polynomial-size classical prover answers all n instances correctly with non-negligible probability. Then this probability is at least $1/q(n)$ for infinitely many n, for some polynomial $q(n) \geq 2$. Apply Theorem 3.12 with $t = n$ and accuracy parameter $\eta = \ln(q(n))/(cn)$, where c is the constant in that theorem. This gives a classical oracle circuit of polynomial size agreeing with $h^{\mathcal{O}}$ on at least a $1-\eta$ fraction of inputs. Since $\eta \to 0$, this contradicts Theorem 4.12 for sufficiently large n. Thus the prover's success probability, including the verifier's error, is negligible. □

5 A quantum oracle where BQP = QCMA

Theorem 5.1. *There exists a distribution over unitary oracles* $\mathcal{O} \leftarrow \mathcal{D}$ *such that with probability* 1*,*

1. $\mathsf{BQP}^{\mathcal{O},\mathsf{PSPACE}} = \mathsf{QCMA}^{\mathcal{O},\mathsf{PSPACE}}$
2. *Relative to* $\mathcal{O}, \mathsf{PSPACE}$*, quantum verifier oblivious state preparation exists*

5.1 The oracle

We define a distribution $\mathcal{D}$ over unitary oracles $\mathcal{O} = (\mathsf{SG}, \mathsf{Expand}, \mathsf{Mix}) \leftarrow \mathcal{D}$ consisting of three parts. We will have $\mathsf{SG} = \{\mathsf{SG}_t\}_{t\in\mathbb{N}}$, $\mathsf{Expand} = \{\mathsf{Expand}_t\}_{t\in\mathbb{N}}$, $\mathsf{Mix} = \{\mathsf{Mix}_t\}_{t\in\mathbb{N}}$. We will write $\mathcal{O}_t = (\mathsf{SG}_t, \mathsf{Expand}_t, \mathsf{Mix}_t)$.

For each t, let $\mathrm{RO}_t : \{0,1\}^t \times \{0,1\}^{4t} \to \{0,1\}^{6t} \times \{0,1\}^{6t}$ be a random oracle. Let $W_t^{\mathsf{SG}} : \mathcal{H}(\{0,1\}^{4t}) \to \mathcal{H}(\{0,1\}^{4t})$ be random unitaries. For each $x \in \{0,1\}^t$, we will also sample a random state $|\phi_x\rangle \leftarrow \mathsf{HaarSt}(\{0,1\}^{4t})$ and a random unitary $W_{t,x}^{\mathsf{Expand}} : \mathcal{H}(\{0,1\}^{4t+1}) \to \mathcal{H}(\{0,1\}^{4t+1})$.

Fix any $t \in \mathbb{N}$. We will define some important states used in the definition of our oracle.

1. We denote the output of the random oracle on an input $x \in \{0,1\}^t$, $y \in \{0,1\}^{4t}$ by
$$\mathrm{RO}_t(x,y) = (u_{x,y}, v_{x,y})$$
2. For $x \in \{0,1\}^t$, $y \in \{0,1\}^{4t}$, we define the claw state
$$|c_{x,y}\rangle \propto (|u_{x,y}\rangle + |v_{x,y}\rangle)$$
Note that the normalization differs depending on whether $u_{x,y} = v_{x,y}$.

3. We define the random index mixed state σ_t to sample a random key $x \in \{0,1\}^t$ and output $|x\rangle |\phi_x\rangle$. Formally,
$$\sigma_t = \frac{1}{2^t} \sum_{x \in \{0,1\}^t} |x\rangle\langle x| \otimes |\phi_x\rangle\langle\phi_x|$$

4. For $x \in \{0,1\}^t$, we define a mixed state used for preparing random claw states ρ_x to sample a random key $y \in \{0,1\}^{4t}$ and output $|y\rangle |c_{x,y}\rangle$. Formally,
$$\rho_x = \frac{1}{2^{4t}} \sum_{y \in \{0,1\}^{4t}} |y\rangle\langle y| \otimes |c_{x,y}\rangle\langle c_{x,y}|$$

Now that we have these states defined, we can define the component oracles as follows.

1. SG_t will be a state preparation unitary for σ_t indexed by W_t^{SG}. That is,
$$\mathsf{SG}_t = S_{\sigma_t}^{W_t^{\mathsf{SG}}}$$

2. Expand_t will take in an input x and prepare ρ_x, with the corresponding state preparation unitary indexed by $W_{t,x}^{\mathsf{Expand}}$. That is,
$$\mathsf{Expand}_t = \sum_{x \in \{0,1\}^t} |x\rangle\langle x| \otimes S_{\rho_x}^{W_{t,x}^{\mathsf{Expand}}}$$

3. Mix_t will take in inputs x, y, and $|\phi_x\rangle$ and prepare $|\mathrm{RO}_t(x,y)\rangle = |u_{x,y}, v_{x,y}\rangle$. It will act as the identity on all orthogonal inputs. Formally,
$$\mathsf{Mix}_t = \sum_{\substack{x \in \{0,1\}^t \\ y \in \{0,1\}^{4t}}} |x,y\rangle\langle x,y| \otimes \left(|\phi_x\rangle\langle\phi_x| \otimes S_{|u_{x,y},v_{x,y}\rangle} + (I - |\phi_x\rangle\langle\phi_x|) \otimes I \right)$$

Independently of $\mathcal{O}$, fix an oracle PSPACE for a PSPACE-complete problem, with PSPACE_t denoting its restriction to inputs of length t. We write superscripts $\mathcal{O}, \mathsf{PSPACE}$ when an algorithm has access to both. The simulations below also allow access to this fixed oracle, which is independent of all sampled states, unitaries, and random functions.

5.2 Simulating the oracle with random oracle points

We will show that with high probability over the oracle, any algorithm querying the oracle can be simulated by an adversary which just sees samples from the oracle. Notably, this will hold even for adversaries which get both classical advice on the random unitaries and states used by the oracle as well as the entire truth table of the classical random function RO.

In particular, for a given random oracle RO_n and a parameter $L(n)$, define the following mixed state $\mathsf{aux}_{\mathrm{RO}_n} := \mathsf{aux}_{n,L(n)}^{\mathrm{RO}} = (\mathsf{aux}_{\mathsf{SG}}, \mathsf{aux}_{\mathsf{Expand}}, \mathsf{aux}_{\mathsf{Mix}})$ by

1. Sample a list of $L(n)$ points $K = (k_1, \ldots, k_{L(n)}) \leftarrow (\{0,1\}^n)^{L(n)}$. Let $K^* = \{k_i : i \in [L(n)]\}$ be the set of distinct points in K. Set
$$\mathsf{aux}_{\mathsf{SG}} = \bigotimes_{i=1}^{L(n)} |k_i\rangle\langle k_i|$$

2. For each $x \in \{0,1\}^n$, sample $L(n)$ points $J_x = (y_{x,1}, \ldots, y_{x,L(n)}) \leftarrow (\{0,1\}^{4n})^{L(n)}$. Define the state $|\alpha_x\rangle = \bigotimes_{i\in[L(n)]} |y_{x,i}\rangle\,|c_{x,y_{x,i}}\rangle$. Define

$$\mathsf{aux}_{\mathsf{Expand}} = \bigotimes_{x\in\{0,1\}^n} |\alpha_x\rangle\langle\alpha_x|$$

3. For all $x \in \{0,1\}^n$, define $|\beta_x\rangle = \bigotimes_{y\in\{0,1\}^{4n}} |\mathrm{RO}_n(x,y)\rangle$. Set

$$\mathsf{aux}_{\mathsf{Mix}} = |K\rangle\langle K| \otimes \bigotimes_{k\in K^*} |\beta_k\rangle\langle\beta_k|$$

The mixed state $\mathsf{aux}_{\mathrm{RO}_n}$ is the average of $\mathsf{aux}_{\mathsf{SG}} \otimes \mathsf{aux}_{\mathsf{Expand}} \otimes \mathsf{aux}_{\mathsf{Mix}}$ over the joint sampling above. Its three registers may be correlated.

Theorem 5.2. *Let $p(n), q(n) \geq 1$ be any computable functions such that $q(n) < 2^{n/8}$ and $p(n) < 2^{n/8}$. There exists a simulator $\mathsf{Sim}_{\mathsf{aux}}$ and a computable function $L(n)$ such that with probability $\geq 1 - 2^{-\Omega(n^2)}$ over the choice of oracle $\mathsf{SG}_n, \mathsf{Mix}_n, \mathsf{Expand}_n$, the following holds. Let $\mathcal{A}^{(\cdot),\mathsf{PSPACE}}$ be any oracle circuit of size at most $q(n)$ (and thus also making at most $q(n)$ queries to its oracle). Then, for all sufficiently large n,*

$$\left|\Pr\left[\mathcal{A}^{\mathsf{SG}_n,\mathsf{Expand}_n,\mathsf{Mix}_n,\mathsf{PSPACE}}(\mathrm{RO}_n) \to 1\right] - \Pr\left[\mathcal{A}^{\mathsf{Sim}_{\mathsf{aux}}(\mathsf{aux}_{\mathrm{RO}_n}),\mathsf{PSPACE}}(\mathrm{RO}_n) \to 1\right]\right| \leq \frac{1}{p(n)}$$

where RO_n corresponds to the entire truth table of the random function RO_n used by the oracle.

Furthermore, we have that $L(n) < \mathsf{poly}(n, p(n), q(n))$ although $\mathsf{Sim}_{\mathsf{aux}}$ itself is inefficient. Note that by virtue of being a circuit, $\mathcal{A}$ implicitly gets non-uniform classical advice on the oracle.

Proof. Since we will only consider level n of the oracle $\mathcal{O}$, for the duration of this proof we will omit n in relevant subscripts.

The simulator will consist of three components $\mathsf{Sim}_{\mathsf{SG}}, \mathsf{Sim}_{\mathsf{Expand}}, \mathsf{Sim}_{\mathsf{Mix}}$ which will emulate the three components of $\mathcal{O}_n$. On initialization, the simulator will also split its advice into three. $\mathsf{Sim}_{\mathsf{SG}}$ will receive the state $\mathsf{aux}_{\mathsf{SG}}$, $\mathsf{Sim}_{\mathsf{Expand}}$ will receive the state $\mathsf{aux}_{\mathsf{Expand}}$, and $\mathsf{Sim}_{\mathsf{Mix}}$ will receive the state $\mathsf{aux}_{\mathsf{Mix}}$.

Our simulator will be defined as follows:

1. $\mathsf{Sim}_{\mathsf{Init}}$: At initialization, for each distinct $k \in K^*$, the simulator will sample a Haar random state $|r_k\rangle \leftarrow \mathsf{HaarSt}(\{0,1\}^{4n})$. It will provide classical descriptions of these states to the corresponding simulators[8]. It will also split the registers of $\mathsf{aux}_{\mathrm{RO}_n}$ into its parts $(\mathsf{aux}_{\mathsf{SG}}, \mathsf{aux}_{\mathsf{Expand}}, \mathsf{aux}_{\mathsf{Mix}})$ and give each part to the corresponding simulator.

2. $\mathsf{Sim}_{\mathsf{SG}}(\mathsf{aux}_{\mathsf{SG}})$: will parse $\mathsf{aux}_{\mathsf{SG}}$ as $K = (k_1, \ldots, k_{L(n)})$. It will then run the simulator from Lemma 3.22 on initial advice state $\bigotimes_{i=1}^{L(n)} (|k_i\rangle\langle k_i| \otimes |r_{k_i}\rangle\langle r_{k_i}|)$.

3. $\mathsf{Sim}_{\mathsf{Expand}}(\mathsf{aux}_{\mathsf{Expand}})$: will run the simulator from Lemma 3.23 on initial advice state $\mathsf{aux}_{\mathsf{Expand}}$.

[8] Note that a Haar random state does not have a finite classical description. This can be handled by replacing the Haar random states with a t-design for sufficiently large t.

4. $\mathsf{Sim}_{\mathsf{Mix}}(\mathsf{aux}_{\mathsf{Mix}})$: will parse $\mathsf{aux}_{\mathsf{Mix}}$ as K as well as $\{\mathrm{RO}_n(k,y)\}_{k\in K^*,y\in\{0,1\}^{4n}}$. It will then apply the map

$$\mathsf{Mix}_K = I + \sum_{x\in K^*}\sum_{y\in\{0,1\}^{4n}} |x,y\rangle\langle x,y| \otimes |r_x\rangle\langle r_x| \otimes \left(S_{|u_{x,y},v_{x,y}\rangle} - I\right)$$

Towards this end, we will define a few hybrid games. Each game Hyb_i will consist of an initialization phase $\mathsf{Init}^{(i)}$, followed by $\mathcal{A}^{(\cdot),\mathsf{PSPACE}}$ interacting with three (possibly stateful) oracles $\mathsf{SG}^{(i)}, \mathsf{Expand}^{(i)}, \mathsf{Mix}^{(i)}$ and outputting a single bit b, which will be the output of the game. They will be defined as follows:

Hyb_0: This will be the game where $\mathcal{A}$ interacts with $\mathcal{O}$ directly. That is,

1. $\mathsf{Init}^{(0)}$: does nothing
2. $\mathsf{SG}^{(0)}$: queries SG on its input
3. $\mathsf{Expand}^{(0)}$: queries Expand on its input
4. $\mathsf{Mix}^{(0)}$: queries Mix on its input.

Hyb_1: This will be the same as Hyb_0, except the unitaries and states used by the oracle will be replaced with fresh random unitaries and states, keeping RO_n fixed. That is,

1. $\mathsf{Init}^{(1)}$: Samples $\widetilde{W}^{\mathsf{SG}} \leftarrow \mathsf{Haar}(\{0,1\}^{4n})$, $\{\widetilde{W}_x^{\mathsf{Expand}}\}_{x\in\{0,1\}^n} \leftarrow \mathsf{Haar}(\{0,1\}^{4n+1})$. For each x samples a Haar random state $\left|\widetilde{\phi}_x\right\rangle \leftarrow \mathsf{HaarSt}(\{0,1\}^{4n})$. Define

$$\widetilde{\sigma} = \frac{1}{2^n}\sum_{x\in\{0,1\}^n} |x\rangle\langle x| \otimes \left|\widetilde{\phi}_x\right\rangle\left\langle\widetilde{\phi}_x\right|$$

2. $\mathsf{SG}^{(1)}$: applies the unitary

$$S_{\widetilde{\sigma}}^{\widetilde{W}^{\mathsf{SG}}}$$

3. $\mathsf{Expand}^{(1)}$: applies the unitary

$$\sum_{x\in\{0,1\}^n} |x\rangle\langle x| \otimes S_{\rho_x}^{\widetilde{W}_x^{\mathsf{Expand}}}$$

4. $\mathsf{Mix}^{(1)}$: applies Mix with $|\phi_x\rangle$ replaced by $\left|\widetilde{\phi}_x\right\rangle$. That is, $\mathsf{Mix}^{(1)}$ applies the map

$$\sum_{x\in\{0,1\}^n}\sum_{y\in\{0,1\}^{4n}} |x,y\rangle\langle x,y| \otimes \left(\left|\widetilde{\phi}_x\right\rangle\left\langle\widetilde{\phi}_x\right| \otimes S_{|u_{x,y},v_{x,y}\rangle} + (I - \left|\widetilde{\phi}_x\right\rangle\left\langle\widetilde{\phi}_x\right|) \otimes I\right)$$

The only difference between Hyb_0 and Hyb_1 is the replacement of the oracle's component unitaries by freshly random ones as well as the random states $|\phi_x\rangle$ by fresh random ones. We will apply Theorem 3.16 individually to each state $|\phi_x\rangle$ (viewed as a random unitary U_k applied to $|0\rangle$) as well as each of $W^{\mathsf{SG}}, W_x^{\mathsf{Expand}}$. In particular, we will set $\epsilon = 2^{-3n/2}$ for each replacement. We then get that for any given adversary $\mathcal{A}$, with probability

$$\geq 1 - 2(2\cdot 2^n + 1)\cdot\exp\left(-\frac{(2^{4n}-2)2^{-3n}}{96(4q(n))^2}\right)$$

$$|\Pr[\mathsf{Hyb}_0 \to 1] - \Pr[\mathsf{Hyb}_1 \to 1]| \leq (2 \cdot 2^n + 1) \cdot 2^{-3n/2}$$

Note that giving $\mathcal{A}$ the description of RO_n does not change the adversaries query efficiency for the unitaries we replace nor its description length, and so Theorem 3.16 still applies.

Taking the union bound over all adversaries $\mathcal{A}$, we get for all sufficiently large n, with probability $\geq 1 - 2^{O(q(n)(n+\log(q(n)+1)))} \cdot \exp\left(-2^{2n/3}\right) = 1 - 2^{-\Omega(n^2)}$, we have

$$|\Pr[\mathsf{Hyb}_0 \to 1] - \Pr[\mathsf{Hyb}_1 \to 1]| \leq 2^{-n/3}$$

Hyb_2: This will be the map which replaces $\mathsf{SG}^{(1)}$ and $\mathsf{Expand}^{(1)}$ with the simulators given by Lemmas 3.22 and 3.23.

1. $\mathsf{Init}^{(2)}$:
   (a) Samples $\left|\widetilde{\phi}_x\right\rangle \leftarrow \mathsf{HaarSt}(\{0,1\}^{4n})$ and
   $$\widetilde{\sigma} = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} |x\rangle\langle x| \otimes \left|\widetilde{\phi}_x\right\rangle\left\langle\widetilde{\phi}_x\right|$$
   (b) Samples $k_1, \ldots, k_{L(n)} \leftarrow \{0,1\}^n$ uniformly at random.
   (c) For each $x \in \{0,1\}^n$, samples $y_{x,1}, \ldots, y_{x,L(n)} \leftarrow \{0,1\}^{4n}$ uniformly at random.
2. $\mathsf{SG}^{(2)}$: Let $\mathsf{Sim}_{\mathsf{SG}}$ be the simulator from Lemma 3.22 applied to $S_{\widetilde{\sigma}}^{\widetilde{W}^{\mathsf{SG}}}$. $\mathsf{SG}^{(2)}$ will run $\mathsf{Sim}_{\mathsf{SG}}$ with initial advice state
   $$\bigotimes_{i=1}^{L(n)} \left(|k_i\rangle\langle k_i| \otimes \left|\widetilde{\phi}_{k_i}\right\rangle\left\langle\widetilde{\phi}_{k_i}\right|\right)$$
3. $\mathsf{Expand}^{(2)}$: Let $\mathsf{Sim}_{\mathsf{Expand}}$ be the simulator from Lemma 3.23 applied to
   $$\sum_{x \in \{0,1\}^n} |x\rangle\langle x| \otimes S_{\rho_x}^{\widetilde{W}_x^{\mathsf{Expand}}}$$
   $\mathsf{Expand}^{(2)}$ will run $\mathsf{Sim}_{\mathsf{Expand}}$ with initial advice
   $$\bigotimes_{x \in \{0,1\}^n} \bigotimes_{i=1}^{L(n)} \left(|y_{x,i}\rangle\langle y_{x,i}| \otimes \left|c_{x,y_{x,i}}\right\rangle\left\langle c_{x,y_{x,i}}\right|\right)$$
4. $\mathsf{Mix}^{(2)}$: this will be the same as $\mathsf{Mix}^{(1)}$.

By instantiating Lemmas 3.22 and 3.23 with $\epsilon = \frac{1}{16p(n)}$, $\quad L(n) = \left\lceil 16c\, p(n) q(n)^2 \right\rceil$, we have that for all choices of oracle $\mathcal{O}$ in the support of $\mathcal{D}$, for all adversaries $\mathcal{A}$,

$$|\Pr[\mathsf{Hyb}_2 \to 1] - \Pr[\mathsf{Hyb}_1 \to 1]| \leq \frac{1}{4p(n)}$$

Hyb_3: this will be the same as Hyb_2, but $\mathsf{Mix}^{(2)}$ ignores all $x \notin K^*$ (i.e. acts the same way as $\mathsf{Sim}_{\mathsf{Mix}}$). Formally,

1. $\mathsf{Init}^{(3)}$: This will be the same as $\mathsf{Init}^{(2)}$. Let $K = (k_1, \ldots, k_{L(n)})$ be the list of random points sampled.
2. $\mathsf{SG}^{(3)}$: This will be the same as $\mathsf{SG}^{(2)}$
3. $\mathsf{Expand}^{(3)}$: This will be the same as $\mathsf{Expand}^{(2)}$
4. $\mathsf{Mix}^{(3)}$: This will apply the map

$$I + \sum_{x \in K^*} \sum_{y \in \{0,1\}^{4n}} |x,y\rangle\langle x,y| \otimes \left|\widetilde{\phi}_x\right\rangle\left\langle\widetilde{\phi}_x\right| \otimes \left(S_{|u_{x,y}, v_{x,y}\rangle} - I\right)$$

Fix K, the states $\left|\widetilde{\phi}_k\right\rangle$ for $k \in K^*$, and all randomness besides the remaining states $\left|\widetilde{\phi}_x\right\rangle$. Define the distribution over projectors $\Pi_{bad} = \sum_{x \in \{0,1\}^n \setminus K^*} |x\rangle\langle x| \otimes \left|\widetilde{\phi}_x\right\rangle\left\langle\widetilde{\phi}_x\right|$. Note that $\mathsf{Mix}^{(3)}$ is now fixed and $\mathsf{Mix}^{(3)}(I - \Pi_{bad}) = \mathsf{Mix}^{(2)}(I - \Pi_{bad})$, and so by Lemma 3.15 it is sufficient to bound $\sup_{|\phi\rangle} \mathbb{E}[\|\Pi_{bad} |\phi\rangle\|^2]$ where the expectation is over the remaining states $\left|\widetilde{\phi}_x\right\rangle$. But note that $\left|\widetilde{\phi}_x\right\rangle$ is Haar random for each $x \notin K^*$, and so

$$\begin{aligned} &\sup_{|\phi\rangle} \mathbb{E}[\|\Pi_{bad} |\phi\rangle\|^2] \\ \leq 2^n \sup_{|\phi\rangle} &\mathop{\mathbb{E}}_{|\psi\rangle \leftarrow \mathsf{HaarSt}(\{0,1\}^{4n})} [|\langle\phi|\psi\rangle|^2] \\ &\leq 2^n \cdot 2^{-4n} \leq 2^{-3n} \end{aligned}$$

Applying Lemma 3.15, we have that for all choices of oracle $\mathcal{O}$ in the support of $\mathcal{D}$, for all adversaries $\mathcal{A}$,

$$|\Pr[\mathsf{Hyb}_2 \to 1] - \Pr[\mathsf{Hyb}_3 \to 1]| \leq 2q(n)\sqrt{2^{-3n}} \leq \frac{1}{4p(n)}$$

for sufficiently large n.

Hyb_f: This will be the game where the adversary interacts directly with the simulator. That is,

1. $\mathsf{Init}^{(f)}$: Uses the mixed state $\mathsf{aux}_{\mathrm{RO}_n}$ defined above. Runs $\mathsf{Sim}_{\mathsf{Init}}$ to generate $\{|r_k\rangle\}_{k \in K^*}$.
2. $\mathsf{SG}^{(f)}$: Runs $\mathsf{Sim}_{\mathsf{SG}}(\mathsf{aux}_{\mathsf{SG}})$.
3. $\mathsf{Expand}^{(f)}$: Runs $\mathsf{Sim}_{\mathsf{Expand}}(\mathsf{aux}_{\mathsf{Expand}})$.
4. $\mathsf{Mix}^{(f)}$: Runs $\mathsf{Sim}_{\mathsf{Mix}}(\mathsf{aux}_{\mathsf{Mix}})$.

Note that the only difference between Hyb_3 and Hyb_f is syntactic, they compute the exact same functionality. And so $\Pr[\mathsf{Hyb}_3 \to 1] = \Pr\left[\mathsf{Hyb}_f \to 1\right]$.

Since $p(n) < 2^{n/8}$, putting this all together we get that with probability $1 - 2^{-\Omega(n^2)}$, for all oracle circuits $\mathcal{A}$ of size at most $q(n)$,

$$\left|\Pr[\mathsf{Hyb}_0 \to 1] - \Pr\left[\mathsf{Hyb}_f \to 1\right]\right| \leq \frac{1}{p(n)}$$

□

Corollary 5.3. *Let $p(n), q(n) \geq 1$ be any computable functions such that $q(n) < 2^{n/16+3}$ and $p(n) < 2^{n/16+3}$. There exists a simulator $\mathsf{Sim}_{\mathsf{aux}}$ and a computable function $L(n) < \mathsf{poly}(n, p(n), q(n))$ such that with probability $\geq 1 - 2^{-\Omega(n)}$ over the choice of oracle $\mathsf{SG}_n, \mathsf{Mix}_n, \mathsf{Expand}_n$, the following holds. Let $\mathcal{A}^{(\cdot),\mathsf{PSPACE}}$ be any oracle circuit of size at most $q(n)$ (and thus also making at most $q(n)$ queries to its oracle). Then, for all sufficiently large n,*

$$\left|\Pr\left[\mathcal{A}^{\mathsf{SG}_n,\mathsf{Expand}_n,\mathsf{Mix}_n,\mathsf{PSPACE}} \to 1\right] - \Pr_{\widetilde{\mathrm{RO}}_n}\left[\mathcal{A}^{\mathsf{Sim}_{\mathsf{aux}}(\mathsf{aux}_{\widetilde{\mathrm{RO}}_n}),\mathsf{PSPACE}} \to 1\right]\right| \leq \frac{1}{p(n)}$$

where $\widetilde{\mathrm{RO}}_n$ is a fresh random oracle.

Proof. Let $L(n)$ be as in Theorem 5.2 with accuracy parameter $4p(n)$ and circuit-size bound $2q(n)+n$.

In particular, define $\widetilde{\mathsf{aux}}_{\mathrm{RO}_n,\widetilde{\mathrm{RO}}_n}$ to be the same as $\mathsf{aux}_{\mathrm{RO}_n}$ where for each $i \in [L(n)]$, $\mathrm{RO}_n(k_i,\cdot)$ is replaced with $\widetilde{\mathrm{RO}}_n(k_i,\cdot)$. Whenever there is an overlap with a sampled pair $(x, y_{x,j})$, use $\mathrm{RO}_n(x, y_{x,j})$ in both places in $\widetilde{\mathsf{aux}}_{\mathrm{RO}_n,\widetilde{\mathrm{RO}}_n}$.

Condition on the sampled y's and the oracle values retained there, then apply Corollary 3.25 to the larger function $x \mapsto \mathrm{RO}_n(x,\cdot)$ with these entries omitted. We get that for all algorithms $\mathcal{A}$ and all advice functions c of output length $q(n)$,

$$\left|\Pr_{\mathrm{RO}_n}\left[\mathcal{A}^{\mathsf{PSPACE}}(c(\mathrm{RO}_n), \mathsf{aux}_{\mathrm{RO}_n}) \to 1\right] - \Pr_{\mathrm{RO}_n,\widetilde{\mathrm{RO}}_n}\left[\mathcal{A}^{\mathsf{PSPACE}}(c(\mathrm{RO}_n), \widetilde{\mathsf{aux}}_{\mathrm{RO}_n,\widetilde{\mathrm{RO}}_n}) \to 1\right]\right| \leq \frac{(q(n)+n+1)L(n)}{2^{n/2}} + 2^{-n} \leq 2^{-n/6}$$

for sufficiently large n. For $K = (k_1, \ldots, k_{L(n)})$, the marginal distribution k_i has full entropy, and so $H_\infty(k_i) = n$.

We can then apply Corollary 3.25 to the remaining $2^n \cdot L(n)$ sampled points $\mathrm{RO}_n(x, y_{x,i})$ to get

$$\left|\Pr_{\mathrm{RO}_n,\widetilde{\mathrm{RO}}_n}\left[\mathcal{A}^{\mathsf{PSPACE}}(c(\mathrm{RO}_n), \mathsf{aux}_{\widetilde{\mathrm{RO}}_n}) \to 1\right] - \Pr_{\mathrm{RO}_n,\widetilde{\mathrm{RO}}_n}\left[\mathcal{A}^{\mathsf{PSPACE}}(c(\mathrm{RO}_n), \widetilde{\mathsf{aux}}_{\mathrm{RO}_n,\widetilde{\mathrm{RO}}_n}) \to 1\right]\right| \leq \frac{(q(n)+5n+1)\cdot 2^n L(n)}{2^{2n}} + 2^{-5n} \leq 2^{-n/2}$$

for sufficiently large n. In particular, each $y_{x,i}$ has full min-entropy $4n$.

Let

$$\Delta(\mathrm{RO}_n) = \max_{\mathcal{A}} \left|\Pr\left[\mathcal{A}^{\mathsf{Sim}_{\mathsf{aux}}(\mathsf{aux}_{\mathrm{RO}_n}),\mathsf{PSPACE}} \to 1\right] - \Pr_{\widetilde{\mathrm{RO}}_n}\left[\mathcal{A}^{\mathsf{Sim}_{\mathsf{aux}}(\mathsf{aux}_{\widetilde{\mathrm{RO}}_n}),\mathsf{PSPACE}} \to 1\right]\right|$$

where the maximum is over circuits of size at most $q(n)$. Choose $c(\mathrm{RO}_n)$ to encode a maximizing circuit and the sign of the gap. By our size convention, this uses at most $q(n)$ bits. The two bounds above give $\mathbb{E}[\Delta(\mathrm{RO}_n)] \leq 2^{-n/6} + 2^{-n/2}$, so Markov's inequality gives

$$\Pr\left[\Delta(\mathrm{RO}_n) > \frac{3}{4p(n)}\right] \leq \frac{4p(n)}{3}\left(2^{-n/6} + 2^{-n/2}\right).$$

Combining this with the error $1/(4p(n))$ and failure probability $2^{-\Omega(n^2)}$ from Theorem 5.2, we get, for sufficiently large n, with probability at least $1-2^{-n/10}$ over the choice of $\mathsf{SG}_n, \mathsf{Mix}_n, \mathsf{Expand}_n$, for all oracle circuits $\mathcal{A}^{(\cdot),\mathsf{PSPACE}}$ of size at most $q(n)$,

$$\left|\Pr\left[\mathcal{A}^{\mathsf{SG}_n,\mathsf{Expand}_n,\mathsf{Mix}_n,\mathsf{PSPACE}}\to 1\right]-\Pr_{\widetilde{\mathrm{RO}}_n}\left[\mathcal{A}^{\mathsf{Sim}_{\mathsf{aux}}(\mathsf{aux}_{\widetilde{\mathrm{RO}}_n}),\mathsf{PSPACE}}\to 1\right]\right|\le\frac{1}{p(n)}$$

□

Corollary 5.4. *Let $p(n), q(n) \ge 1$ be any computable functions bounded above by $2^{n/16}$. With probability $\ge 1-2^{-n/10}$ over the choice of oracle $\mathsf{SG}_n, \mathsf{Mix}_n, \mathsf{Expand}_n$, the following holds. Let $\mathcal{A}^{(\cdot),\mathsf{PSPACE}}$ be any oracle circuit of size at most $q(n)$ (and thus also making at most $q(n)$ queries to its oracle). Then, for all sufficiently large n,*

$$\left|\Pr\left[\mathcal{A}^{\mathsf{SG}_n,\mathsf{Expand}_n,\mathsf{Mix}_n,\mathsf{PSPACE}}\to 1\right]-\Pr_{\widetilde{\mathsf{SG}}_n,\widetilde{\mathsf{Expand}}_n,\widetilde{\mathsf{Mix}}_n}\left[\mathcal{A}^{\widetilde{\mathsf{SG}}_n,\widetilde{\mathsf{Expand}}_n,\widetilde{\mathsf{Mix}}_n,\mathsf{PSPACE}}\to 1\right]\right|\le\frac{1}{2p(n)}$$

where $\widetilde{\mathsf{SG}}_n, \widetilde{\mathsf{Expand}}_n, \widetilde{\mathsf{Mix}}_n$ is a freshly sampled set of oracles, including a fresh random function $\widetilde{\mathrm{RO}}_n$.

Proof. The key idea is that we apply Corollary 5.3 twice. First for $\mathsf{SG}_n, \mathsf{Expand}_n, \mathsf{Mix}_n$ and second for a freshly random sampled $\widetilde{\mathsf{SG}}_n, \widetilde{\mathsf{Expand}}_n, \widetilde{\mathsf{Mix}}_n$. In particular, setting the error parameter as $\frac{1}{6p(n)}$, we get that with probability $\ge 1-2^{-n/10}$ over the choice of $\mathsf{SG}_n, \mathsf{Expand}_n, \mathsf{Mix}_n$,

$$\begin{aligned}&\left|\Pr\left[\mathcal{A}^{\mathsf{SG}_n,\mathsf{Expand}_n,\mathsf{Mix}_n,\mathsf{PSPACE}}\to 1\right]-\Pr_{\widetilde{\mathsf{SG}}_n,\widetilde{\mathsf{Expand}}_n,\widetilde{\mathsf{Mix}}_n}\left[\mathcal{A}^{\widetilde{\mathsf{SG}}_n,\widetilde{\mathsf{Expand}}_n,\widetilde{\mathsf{Mix}}_n,\mathsf{PSPACE}}\to 1\right]\right|\\ &\qquad\le\frac{1}{3p(n)}+2^{-n/10}\le\frac{1}{2p(n)}\end{aligned}$$

for all sufficiently large n. Here the extra $2^{-n/10}$ term comes from the probability that $\widetilde{\mathsf{SG}}_n, \widetilde{\mathsf{Expand}}_n, \widetilde{\mathsf{Mix}}_n$ fails to satisfy the conditions of Corollary 5.3. Both applications are within its parameter range since $6p(n) < 2^{n/16+3}$. □

Corollary 5.5. *Let $q(n), p(n) \ge 1$ be any computable functions bounded above by $2^{n/16}$. There exists a simulator $\mathsf{Sim}_{\mathrm{eff}}$ such that with probability $\ge 1-2^{-n/10}$ over the choice of oracle $\mathsf{SG}_n, \mathsf{Mix}_n, \mathsf{Expand}_n$, the following holds. Let $\mathcal{A}^{(\cdot),\mathsf{PSPACE}}$ be any oracle circuit of size at most $q(n)$ (and thus also making at most $q(n)$ queries to its oracle). Then, for all sufficiently large n,*

$$\left|\Pr\left[\mathcal{A}^{\mathsf{SG}_n,\mathsf{Expand}_n,\mathsf{Mix}_n,\mathsf{PSPACE}}\to 1\right]-\Pr\left[\mathcal{A}^{\mathsf{Sim}_{\mathrm{eff}},\mathsf{PSPACE}}\to 1\right]\right|\le\frac{1}{p(n)}$$

The simulator runs in time $\mathsf{poly}(n, p(n), q(n))$. Again, as $\mathcal{A}^{(\cdot),\mathsf{PSPACE}}$ is an oracle circuit, it implicitly gets advice about the oracle.

Proof. Applying Corollary 5.4, our simulator will implement $\widetilde{\mathsf{SG}}_n, \widetilde{\mathsf{Expand}}_n, \widetilde{\mathsf{Mix}}_n$ for a freshly random choice of random oracle, random states, and random unitaries. Note that each of $\widetilde{\mathsf{SG}}_n, \widetilde{\mathsf{Expand}}_n$,

$\widetilde{\mathsf{Mix}}_n$ can be implemented efficiently by making polynomially many queries to fresh random oracles and unitaries. Furthermore, fresh classical random oracles and unitaries can be simulated efficiently using compressed oracles [Zha19] and path recording [MH25, GY26, FLM+26] up to error 2^{-n} in time $\mathsf{poly}(n, p(n), q(n))$. The total error is at most $1/(2p(n)) + 2^{-n} \leq 1/p(n)$ for sufficiently large n. □

5.3 Showing $\mathsf{BQP}^{\mathcal{O},\mathsf{PSPACE}} = \mathsf{QCMA}^{\mathcal{O},\mathsf{PSPACE}}$

Theorem 5.6. *With probability* 1 *over* $\mathcal{O} \leftarrow \mathcal{D}$, $\mathsf{BQP}^{\mathcal{O},\mathsf{PSPACE}} = \mathsf{QCMA}^{\mathcal{O},\mathsf{PSPACE}}$.

Proof. Let Π be any language in $\mathsf{QCMA}^{\mathcal{O},\mathsf{PSPACE}}$ with verifier $\mathsf{Ver}^{\mathcal{O},\mathsf{PSPACE}}$. For $n \in \mathbb{N}$, let $q(n)$ be a polynomial upper bound on the runtime of $\mathsf{Ver}^{\mathcal{O},\mathsf{PSPACE}}(x, w)$ and on its circuit size after hardcoding x, w, for $x \in \{0,1\}^n, w \in \{0,1\}^{\mathsf{poly}(n)}$. This is thus also a bound on the number of queries made as well as the length of the witness. We will describe a QPT algorithm $\mathcal{A}^{\mathcal{O},\mathsf{PSPACE}}$ which decides Π as follows:

1. On input $x \in \{0,1\}^n$, let $d = \lceil c \log n \rceil$ for some appropriate constant c.
2. Construct a new verifier $\widetilde{\mathsf{Ver}}^{\mathsf{PSPACE}}$ as follows
   (a) For $\ell \in [1, d]$, learn $(\mathsf{SG}_\ell, \mathsf{Expand}_\ell, \mathsf{Mix}_\ell)$ using Theorem 3.13 with error parameter $\frac{1}{3q(n)^3}$ and failure probability $2^{-d}/3$ for each component. This takes $2^{O(d)}\mathsf{poly}(q(n), d) = \mathsf{poly}(n)$ time, and all three estimates are accurate with probability at least $1 - 2^{-d}$. Define the corresponding oracles as $\widetilde{\mathcal{O}}_\ell = (\widetilde{\mathsf{SG}}_\ell, \widetilde{\mathsf{Expand}}_\ell, \widetilde{\mathsf{Mix}}_\ell)$.
   (b) For $\ell \in [d+1, q(|x|)]$, let $\mathsf{Sim}_{\mathrm{eff},\ell}$ be as in Corollary 5.5 with $n = \ell$, circuit-size bound $Q_n = \max(q(n), 2^{d/32})$, and accuracy parameter $P_n = \max(q(n)^2, 2^{d/32})$. Set $\widetilde{\mathcal{O}}_\ell = \mathsf{Sim}_{\mathrm{eff},\ell}$. Since P_n and Q_n are both bounded above by polynomials in n, $\mathsf{Sim}_{\mathrm{eff},\ell}$ can be implemented efficiently.
   (c) $\widetilde{\mathsf{Ver}}^{\mathsf{PSPACE}}$ will then run $\mathsf{Ver}^{\{\widetilde{\mathcal{O}}_\ell\}_{\ell \in [1, q(|x|)]},\mathsf{PSPACE}}$
3. Finally, output 1 if and only if there exists some witness w such that $\Pr\left[\widetilde{\mathsf{Ver}}^{\mathsf{PSPACE}}(x, w) = 1\right] \geq 0.5$, This can be computed using the PSPACE oracle since $\widetilde{\mathsf{Ver}}^{\mathsf{PSPACE}}$ is a polynomial time PSPACE oracle algorithm.

To see that this succeeds with high probability over the oracle, it is sufficient to show that with high probability over the oracle and the randomness of $\mathcal{A}$, $\widetilde{\mathsf{Ver}}^{\mathsf{PSPACE}}$ agrees with $\mathsf{Ver}^{\mathcal{O},\mathsf{PSPACE}}$ on all $x \in \{0,1\}^n$ and witnesses $w \in \{0,1\}^{q(|x|)}$. In particular, we will hybrid over all $\ell \in [q(n)]$. Define $\mathcal{O}^{(\ell)} = (\mathcal{O}_1, \ldots, \mathcal{O}_\ell, \widetilde{\mathcal{O}}_{\ell+1}, \ldots, \widetilde{\mathcal{O}}_{q(|x|)})$.

The following bounds hold simultaneously for all $x \in \{0,1\}^n$ and witnesses w.

For all $\ell \in [1, d]$ and all $\mathcal{O}$ in the support of $\mathcal{D}$, with probability at least $1 - 2^{-d}$ over the coins of $\mathcal{A}$, for all $x \in \{0,1\}^n, w \in \{0,1\}^{q(n)}$,

$$\left|\Pr\left[\mathsf{Ver}^{\mathcal{O}^{(\ell)},\mathsf{PSPACE}}(x, w) = 1\right] - \Pr\left[\mathsf{Ver}^{\mathcal{O}^{(\ell-1)},\mathsf{PSPACE}}(x, w) = 1\right]\right| \leq \frac{1}{q(|x|)^2}$$

For all $\ell \in [d+1, q(n)]$, as Corollary 5.5 relativizes we can apply it individually to each index. By Corollary 5.5 we have that for all sufficiently large n, with probability greater than $1 - 2^{-d/10}$, for all $x \in \{0,1\}^n, w \in \{0,1\}^{q(n)}$

$$\left|\Pr\left[\mathsf{Ver}^{\mathcal{O}^{(\ell)},\mathsf{PSPACE}}(x,w) \to 1\right] - \Pr\left[\mathsf{Ver}^{\mathcal{O}^{(\ell-1)},\mathsf{PSPACE}}(x,w) \to 1\right]\right| \leq \frac{1}{q(n)^2}$$

Note that in particular since the bound holds for all circuits simultaneously, it holds for the circuits which hardcode x and w for all possible values of x and w simultaneously.

And so by repeating all these hybrids, we get that for all sufficiently large n, with probability $\geq 1 - d \cdot 2^{-d} - q(n) \cdot \left(2^{-d/10}\right)$

$$\left|\Pr\left[\mathsf{Ver}^{\mathcal{O}^{(q(n))},\mathsf{PSPACE}}(x,w) \to 1\right] - \Pr\left[\mathsf{Ver}^{\mathcal{O}^{(0)},\mathsf{PSPACE}}(x,w) \to 1\right]\right| \leq q(n) \cdot \frac{1}{q(n)^2} \leq 0.01$$

By taking c from $d = \lceil c \log(n) \rceil$ sufficiently large, the probability over $\mathcal{O}$ that one of the hybrids with $\ell > d$ fails for some x, w is at most $q(n)2^{-d/10} \leq 1/n^2$. For every other oracle, tomography succeeds with probability at least $1 - d2^{-d} \geq 2/3$, and whenever it succeeds the algorithm correctly decides all inputs of length n.

Define BAD_n to be the event over $\mathcal{O}$ that there exists an $x \in \{0,1\}^n$ on which $\mathcal{A}^{\mathcal{O},\mathsf{PSPACE}}$ gives the wrong answer with probability greater than $1/3$. We have just shown that

$$\Pr[BAD_n] \leq \frac{1}{n^2}$$

And so by Borel-Cantelli, the probability that BAD_n occurs infinitely often is 0. Thus, $\mathcal{A}^{\mathcal{O},\mathsf{PSPACE}}$ correctly decides Π with probability at least $2/3$ on all sufficiently large n, and so by hardcoding whether $x \in \Pi$ for all $x \in \{0,1\}^{n'}$ for n' below this threshold, we get an algorithm which correctly decides Π on all inputs with probability at least $2/3$. Thus, $\Pi \in \mathsf{BQP}^{\mathcal{O},\mathsf{PSPACE}}$.

Since the number of verifiers is countable, taking the union bound gives that with probability 1 over the choice of oracle, $\mathsf{QCMA}^{\mathcal{O},\mathsf{PSPACE}} \subseteq \mathsf{BQP}^{\mathcal{O},\mathsf{PSPACE}}$. □

5.4 Building oblivious state preparation

We construct a claw state generator as in Definition 3.29 relative to our oracle.

Lemma 5.7 (Decoding bound [Nay99, Theorem 2.4])**.** *Let Z be uniform on $[M]$, and let $\{\rho_z\}_{z\in[M]}$ be density operators on a d-dimensional register. For any quantum decoding algorithm Dec,*

$$\Pr[\mathsf{Dec}(\rho_Z) = Z] \leq \frac{d}{M}.$$

The same bound holds if the decoder receives $\rho_Z \otimes \sigma$ for any fixed auxiliary state σ. In particular, a uniformly random $2n$-bit string encoded into q qubits can be recovered with probability at most 2^{q-2n}; for $q = n$, this is 2^{-n}.

Lemma 5.8 (Claw hardness)**.** *Let $\mathcal{A}$ be any quantum algorithm, and let $m, \ell \geq 1$ be integers. Consider the following game:*

1. *Sample $a_1, \ldots, a_\ell, b_1, \ldots, b_\ell \leftarrow \{0,1\}^m$ independently and uniformly at random.*

2. *For each $i \in [\ell]$, define the normalized m-qubit claw state*

$$|\chi_i\rangle = \frac{|a_i\rangle + |b_i\rangle}{\sqrt{2 + 2\mathbf{1}[a_i = b_i]}}.$$

3. *Run $\mathcal{A}$ on $\bigotimes_{i=1}^{\ell} |\chi_i\rangle$ and output a pair (a', b').*

4. *Output 1 if $(a', b') = (a_i, b_i)$ for some $i \in [\ell]$.*

Then the probability that $\mathcal{A}$ wins this game is at most $\ell \cdot 2^{-m}$. The same bound holds if $\mathcal{A}$ also receives any fixed auxiliary state independent of all sampled pairs.

Proof. For each $i \in [\ell]$, let E_i be the event that $\mathcal{A}$ outputs (a_i, b_i). Fix i and condition on arbitrary values of all pairs (a_j, b_j) with $j \neq i$. The pair (a_i, b_i) remains uniform on $\{0,1\}^{2m}$, and all other claw states, together with any additional fixed auxiliary state, form a fixed auxiliary state in tensor product with $|\chi_i\rangle$. Thus, $\mathcal{A}$ is a decoder for (a_i, b_i) from the m-qubit encoding $|\chi_i\rangle$, together with independent auxiliary input. Applying Lemma 5.7 with $M = 2^{2m}$ and $d = 2^m$ gives

$$\Pr[E_i \,|\, (a_j, b_j)_{j \neq i}] \leq \frac{2^m}{2^{2m}} = 2^{-m}.$$

Averaging over the conditioned pairs yields the same bound on $\Pr[E_i]$. Finally, by the union bound,

$$\Pr[\mathcal{A} \text{ wins}] = \Pr\left[\bigcup_{i=1}^{\ell} E_i\right] \leq \sum_{i=1}^{\ell} \Pr[E_i] \leq \ell \cdot 2^{-m}.$$

□

We now construct the claw state generator. Here x, y index oracle inputs; the corresponding claw endpoints are $u_{x,y}, v_{x,y}$. This will be a two message protocol as follows

Construction 5.9.

1. The verifier begins by running SG_n, measuring the register containing $|x\rangle$ and tracing out the purification and flag registers, to produce a state $|x\rangle |\phi_x\rangle$. The verifier sends x to the prover.

2. The prover runs $\mathsf{Expand}_n |x\rangle |0\rangle$ and traces out the purification and flag registers to produce the state $|x\rangle |y\rangle |c_{x,y}\rangle$. The prover sends y to the verifier and outputs the claw-state $|c_{x,y}\rangle$.

3. The verifier runs $\mathsf{Mix}_n |x\rangle |y\rangle |\phi_x\rangle |0\rangle$ and traces out the excess registers to produce the state $|x\rangle |y\rangle |u_{x,y}\rangle |v_{x,y}\rangle$. It then outputs $|u_{x,y}\rangle |v_{x,y}\rangle$.

Claim 5.10. *With probability 1 over $\mathcal{O} \leftarrow \mathcal{D}$, Construction 5.9 satisfies correctness.*

Proof. This is self-evident by construction. That is, the only way correctness can be violated is if there exists some pair $x \in \{0,1\}^n$, $y \in \{0,1\}^{4n}$ such that $u_{x,y} = v_{x,y}$. For a random function $f : \{0,1\}^n \times \{0,1\}^{4n} \to \{0,1\}^{12n}$, the probability that there exists a pair x, y such that $u_{x,y} = v_{x,y}$ is $\leq \frac{2^{5n}}{2^{6n}} = 2^{-n}$ by the union bound. But since

$$\sum_{n=1}^{\infty} 2^{-n} < \infty$$

by Borel-Cantelli the probability this happens infinitely often is 0. Thus, with probability 1 over $\mathcal{O} \leftarrow \mathcal{D}$, correctness is satisfied for all sufficiently large n. □

Claim 5.11. *With probability* 1 *over* $\mathcal{O} \leftarrow \mathcal{D}$, *Construction 5.9 satisfies search security.*

Proof. Fix some polynomial $p(n), q(n)$. We can assume without loss of generality that any adversary breaking search security outputs $(y, u_{x,y}, v_{x,y})$ instead of $(y, v_{x,y}, u_{x,y})$. This loses at most a factor of 2 in the security by randomly reordering the strings.

Let $BAD^{(0)}_{n,p,q}$ be the event (over randomness the choice of oracle) that for some circuit $C^{\mathcal{O},\mathsf{PSPACE}}$ of size $\leq p(n)$ we have

$$\Pr_{x\leftarrow\{0,1\}^n}[C^{\mathcal{O},\mathsf{PSPACE}}(x) = (y, u_{x,y}, v_{x,y}) \text{ for some } y \in \{0,1\}^{4n}] \geq \frac{1}{q(n)}$$

To prove the theorem, it suffices to show that for all polynomials p, q,

$$\Pr_{\mathcal{O}\leftarrow\mathcal{D}}[BAD^{(0)}_{n,p,q} \text{ occurs infinitely often}] = 0$$

We will first show that $\Pr\left[BAD^{(0)}_{n,p,q}\right] = 2^{-\Omega(n)}$.

Let $\widetilde{p}(n)$ be a polynomial upper bound on the size of the entire game. Let $\mathsf{Sim}_{\mathsf{aux}}$ and $L(n)$ be as in Theorem 5.2 with accuracy parameter $8q(n)$ and circuit-size bound $\widetilde{p}(n)$. Define a new event $BAD^{(1)}_{n,p,q}$ to be the event that for some circuit $C^{\mathcal{O},\mathsf{PSPACE}}$ of size $\leq p(n)$, we have

$$\Pr_{x\leftarrow\{0,1\}^n,\widetilde{\mathrm{RO}_n}}[C^{\mathsf{Sim}_{\mathsf{aux}}(\mathsf{aux}_{\widetilde{\mathrm{RO}_n}}),\mathsf{PSPACE}}(x) = (y, \widetilde{\mathrm{RO}_n}(x,y)) \text{ for some } y] \geq \frac{1}{2q(n)}$$

Applying Corollary 5.4 with error $\frac{1}{4q(n)}$ and circuit-size bound $\widetilde{p}(n)$ to the entire game (including the sampling of x using SG_n and final computation of $\mathrm{RO}_n(x,y)$ on input (x,y) using $|\phi_x\rangle$) gives that if $BAD^{(0)}_{n,p,q}$ holds and the event in Corollary 5.4 holds, we have that

$$\Pr_{x\leftarrow\{0,1\}^n,\widetilde{\mathsf{SG}_n},\widetilde{\mathsf{Expand}_n},\widetilde{\mathsf{Mix}_n}}\left[C^{\widetilde{\mathsf{SG}_n},\widetilde{\mathsf{Expand}_n},\widetilde{\mathsf{Mix}_n},\mathsf{PSPACE}}(x) = (y, \widetilde{\mathrm{RO}_n}(x,y)) \text{ for some } y\right] \geq \frac{3}{4q(n)} \tag{7}$$

We then use the hybrids in the proof of Theorem 5.2, keeping $\widetilde{\mathrm{RO}_n}$ fixed and testing whether C outputs $(y, \widetilde{\mathrm{RO}_n}(x,y))$. Since C is independent of the fresh random unitaries and states, Hyb_0 and Hyb_1 have the same distribution after averaging. The remaining hybrids change the probability of success by at most $1/(8q(n))$, since the final test makes no oracle queries. In particular, if Equation (7) holds then

$$\Pr_{x\leftarrow\{0,1\}^n,\widetilde{\mathrm{RO}_n}}[C^{\mathsf{Sim}_{\mathsf{aux}}(\mathsf{aux}_{\widetilde{\mathrm{RO}_n}}),\mathsf{PSPACE}}(x) = (y, \widetilde{\mathrm{RO}_n}(x,y)) \text{ for some } y] \geq \frac{3}{4q(n)} - \frac{1}{8q(n)} \geq \frac{1}{2q(n)}$$

and so $BAD^{(1)}_{n,p,q}$ also holds. And so by the union bound,

$$\Pr\left[BAD^{(0)}_{n,p,q}\right] \leq \Pr\left[BAD^{(1)}_{n,p,q}\right] + 2^{-\Omega(n)}$$

Let $c(\mathcal{O})$ describe a circuit witnessing $BAD^{(1)}_{n,p,q}$ when one exists, and any circuit otherwise. Below, $\mathsf{Eval}(c(\mathcal{O}), x)$ runs the described circuit on x. Write

$$p_{BAD} = \Pr_{\mathcal{O}\leftarrow\mathcal{D},x,\widetilde{\mathrm{RO}_n}}[\mathsf{Eval}^{\mathsf{Sim}_{\mathsf{aux}}(\mathsf{aux}_{\widetilde{\mathrm{RO}_n}}),\mathsf{PSPACE}}(c(\mathcal{O}), x) = (y, \widetilde{\mathrm{RO}_n}(x,y)) \text{ for some } y]$$

Observe that we can write the left hand side in terms of $\Pr\left[BAD^{(1)}_{n,p,q}\right]$ as follows:

$$\begin{aligned}
&\Pr_{\mathcal{O}\leftarrow\mathcal{D},x,\widetilde{\mathrm{RO}_n}}[\mathsf{Eval}^{\mathsf{Sim}_{\mathsf{aux}}(\mathsf{aux}_{\widetilde{\mathrm{RO}_n}}),\mathsf{PSPACE}}(c(\mathcal{O}),x)=(y,\widetilde{\mathrm{RO}_n}(x,y)) \text{ for some } y]\\
\geq &\Pr_{\mathcal{O}\leftarrow\mathcal{D},x,\widetilde{\mathrm{RO}_n}}[\mathsf{Eval}^{\mathsf{Sim}_{\mathsf{aux}}(\mathsf{aux}_{\widetilde{\mathrm{RO}_n}}),\mathsf{PSPACE}}(c(\mathcal{O}),x)=(y,\widetilde{\mathrm{RO}_n}(x,y)) \text{ for some } y | BAD^{(1)}_{n,p,q}]\\
&\cdot\Pr\left[BAD^{(1)}_{n,p,q}\right]\\
\geq &\frac{\Pr\left[BAD^{(1)}_{n,p,q}\right]}{2q(n)}
\end{aligned}$$

And so it suffices to bound p_{BAD}. Write $L=L(n)$ and consider $J_x=(y_{x,1},\ldots,y_{x,L})$. Let $DIST_x$ be the event that these are all distinct. First, $\Pr[x\in K^*]\leq L2^{-n}$ and $\Pr[\neg DIST_x]\leq L^2 2^{-4n}$. Condition on neither event occurring, and fix x, K, all sampled labels, and $c(\mathcal{O})$.

Since $\widetilde{\mathrm{RO}_n}$ is fresh and independent of $\mathcal{O}$, the pairs $\widetilde{\mathrm{RO}_n}(x,y_{x,i})$ remain independent and uniform on $\{0,1\}^{12n}$. The only advice depending on $\widetilde{RO_n}(x,\cdot)$ consists of the corresponding L claw states. We may therefore view the entire simulated computation as a decoder receiving these claw states and independent auxiliary input. Applying Lemma 5.8 with $m=6n$ and $\ell=L$, and discarding the decoder's output label y, bounds the probability of recovering a pair at one of the sampled points by $L2^{-6n}$.

At an unsampled point, $\widetilde{\mathrm{RO}_n}(x,y)$ is independent of all the advice, so guessing it correctly has probability at most 2^{-12n}, even when y is chosen as a function of the simulated computation. Averaging over the fixed data, we obtain

$$p_{BAD}\leq L2^{-n}+L^2 2^{-4n}+L2^{-6n}+2^{-12n}\leq 2L2^{-n}+L^2 2^{-4n}.$$

Thus,

$$\Pr\left[BAD^{(1)}_{n,p,q}\right]\leq 6q(n)\cdot L\cdot 2^{-n}\leq 2^{-n/2}$$

And so with probability $1-2^{-\Omega(n)}$, $BAD^{(0)}_{n,p,q}$ does not occur.

Thus, for all polynomials p,q,

$$\sum_{n=1}^{\infty}\Pr\left[BAD^{(0)}_{n,p,q}\right]<\infty$$

and so by Borel-Cantelli, for all p,q,

$$\Pr_{\mathcal{O}\leftarrow\mathcal{D}}[BAD^{(0)}_{n,p,q} \text{ occurs infinitely often}]=0$$

□

By Theorem 3.30 and the remark following it, we obtain qv-OSP relative to $\mathcal{O},\mathsf{PSPACE}$.

6 AI Disclosure

The proofs of all theorems were developed in discussion with various versions of ChatGPT. The description of the oracles and most of the key ideas for the proofs were human originated. The primary exception to this is the technique for removing the Forrelated oracle in the non-uniform setting by considering the Turing machine whose success probability approximates the probability of the existence of a non-uniform attacker. This corresponds to the first half of the proof of Theorem 4.12. LLMs were also used to generate the proofs in Appendix B.

References

[AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 333–342, San Jose, CA, USA, June 6–8, 2011. ACM Press.

[AA15] Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th Annual ACM Symposium on Theory of Computing*, pages 307–316, Portland, OR, USA, June 14–17, 2015. ACM Press.

[Aar10] Scott Aaronson. BQP and the polynomial hierarchy. In Leonard J. Schulman, editor, *42nd Annual ACM Symposium on Theory of Computing*, pages 141–150, Cambridge, MA, USA, June 5–8, 2010. ACM Press.

[Aar21] Scott Aaronson. Open problems related to quantum query complexity. *ACM Transactions on Quantum Computing*, 2(4):14:1–14:9, 2021.

[AB17] Barış Aydınlıoğlu and Eric Bach. Affine relativization: Unifying the algebrization and relativization barriers. Electronic Colloquium on Computational Complexity, Report TR16-040, Revision 3, 2017.

[AG25] Prabhanjan Ananth and Eli Goldin. Less is more: On copy complexity in quantum cryptography. Cryptology ePrint Archive, Report 2025/1855, 2025.

[AGKZ20] Ryan Amos, Marios Georgiou, Aggelos Kiayias, and Mark Zhandry. One-shot signatures and applications to hybrid quantum/classical authentication. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd Annual ACM Symposium on Theory of Computing*, pages 255–268, Chicago, IL, USA, June 22–26, 2020. ACM Press.

[AGL24] Prabhanjan Ananth, Aditya Gulati, and Yao-Ting Lin. Cryptography in the common Haar state model: Feasibility results and separations. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024: 22nd Theory of Cryptography Conference, Part II*, volume 15365 of *Lecture Notes in Computer Science*, pages 94–125, Milan, Italy, December 2–6, 2024. Springer, Cham, Switzerland.

[AIK22] Scott Aaronson, DeVon Ingram, and William Kretschmer. The acrobatics of bqp. In *Proceedings of the 37th Computational Complexity Conference*, CCC '22, Dagstuhl, DEU, 2022. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[AW08] Scott Aaronson and Avi Wigderson. Algebrization: a new barrier in complexity theory. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 731–740, Victoria, BC, Canada, May 17–20, 2008. ACM Press.

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[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 Annual Symposium on Foundations of Computer Science*, pages 320–331, Paris, France, October 7–9, 2018. IEEE Computer Society Press.

[BHN+26] Adam Bouland, Andrew Huang, Anand Natarajan, Itay Shalit, and Avishay Tal. $\mathsf{BQP} \subseteq \mathsf{IP}$ does not relativize, 2026.

[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(2126):459–472, 2011. arXiv:1005.1407.

[BK25] James Bartusek and Dakshita Khurana. On the power of oblivious state preparation. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology – CRYPTO 2025, Part II*, volume 16001 of *Lecture Notes in Computer Science*, pages 575–607, Santa Barbara, CA, USA, August 17–21, 2025. Springer, Cham, Switzerland.

[BMM+25] Amit Behera, Giulio Malavolta, Tomoyuki Morimae, Tamer Mour, and Takashi Yamakawa. A new world in the depths of microcrypt: Separating OWSGs and quantum money from QEFID. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025, Part VII*, volume 15607 of *Lecture Notes in Computer Science*, pages 23–52, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland.

[CCD+03] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *35th Annual ACM Symposium on Theory of Computing*, pages 59–68, San Diego, CA, USA, June 9–11, 2003. ACM Press.

[CCG+94] Richard Chang, Benny Chor, Oded Goldreich, Juris Hartmanis, Johan Håstad, Desh Ranjan, and Pankaj Rohatgi. The random oracle hypothesis is false. *Journal of Computer and System Sciences*, 49(1):24–39, 1994.

[CCS25] Boyang Chen, Andrea Coladangelo, and Or Sattath. The power of a single haar random state: Constructing and separating quantum pseudorandomness. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025, Part VII*, volume 15607 of *Lecture Notes in Computer Science*, pages 108–137, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland.

[CDGS18] Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John P. Steinberger. Random oracles and non-uniformity. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 227–258, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Cham, Switzerland.

[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, *Advances in Cryptology – CRYPTO 2024, Part VII*, volume 14926 of *Lecture Notes in Computer Science*, pages 215–248, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[ÇGR26] Alper Çakan, Vipul Goyal, and Justin Raizes. How to delete without a trace: Certified deniability in a quantum world. In *Advances in Cryptology – CRYPTO 2026: 46th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17–20, 2026, Proceedings, Part V*, page 183–214, Berlin, Heidelberg, 2026. Springer-Verlag.

[CWX26] Boyang Chen, Yiming Wang, and Ziyi Xie. The breakdown of classical minicrypt equivalences in the quantum-computation classical-communication model, 2026.

[FLM+26] Ben Foxman, Alex Lombardi, Fermi Ma, Barak Nehoran, and John Wright. Quantum lazy sampling and path recording for any group, 2026.

[For09] Lance Fortnow. A simple proof of Toda's theorem. *Theory of Computing*, 5:135–140, 2009.

[FS88] Lance Fortnow and Michael Sipser. Are there interactive protocols for co-NP languages? *Information Processing Letters*, 28(5):249–251, 1988.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 113–122, Victoria, BC, Canada, May 17–20, 2008. ACM Press.

[GMMY24] Eli Goldin, Tomoyuki Morimae, Saachi Mutreja, and Takashi Yamakawa. CountCrypt: Quantum cryptography between QCMA and PP. Cryptology ePrint Archive, Report 2024/1707, 2024.

[GS86] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *18th Annual ACM Symposium on Theory of Computing*, pages 59–68, Berkeley, CA, USA, May 28–30, 1986. ACM Press.

[GY26] Dmitry Grinko and Satoshi Yoshida. Quantum simulation of random unitaries from clebsch-gordan transforms, 2026.

[GZ25] Eli Goldin and Mark Zhandry. Translating between the common haar random state model and the unitary model. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology – CRYPTO 2025, Part II*, volume 16001 of *Lecture Notes in Computer Science*, pages 269–300, Santa Barbara, CA, USA, August 17–21, 2025. Springer, Cham, Switzerland.

[HKOT23] Jeongwan Haah, Robin Kothari, Ryan O'Donnell, and Ewin Tang. Query-optimal estimation of unitary channels in diamond distance. In *64th Annual Symposium on Foundations of Computer Science*, pages 363–390, Santa Cruz, CA, USA, November 6–9, 2023. IEEE Computer Society Press.

[HMOY25] Minki Hhan, Tomoyuki Morimae, Yasuaki Okinaka, and Takashi Yamakawa. Proofs of quantum memory. Cryptology ePrint Archive, Report 2025/1837, 2025.

[Hoz26] William M. Hoza. A technique for hardness amplification against ac0. *computational complexity*, 35(2):12, Sep 2026.

[HY25] Minki Hhan and Shogo Yamada. Pseudorandom function-like states from common haar unitary. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025: 23rd Theory of Cryptography Conference, Part III*, volume 16270 of *Lecture Notes in Computer Science*, pages 134–165, Aarhus, Denmark, December 1–5, 2025. Springer, Cham, Switzerland.

[IJKW10] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. Uniform direct product theorems: Simplified, optimized, and derandomized. *SIAM Journal on Computing*, 39(4):1637–1665, 2010.

[KLVY23] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Lisa Yang. Quantum advantage from any non-local game. In Barna Saha and Rocco A. Servedio, editors, *55th Annual ACM Symposium on Theory of Computing*, pages 1617–1628, Orlando, FL, USA, June 20–23, 2023. ACM Press.

[KPG13] Swastik Kopparty, Jason Perry, and Brian Garnett. Lecture 4: AC0 lower bounds and pseudorandomness. 2013.

[Kre21] William Kretschmer. Quantum Pseudorandomness and Classical Complexity. *Leibniz Int. Proc. Inf.*, 197:2:1–2:20, 2021.

[Mah18] Urmila Mahadev. Classical verification of quantum computations. In Mikkel Thorup, editor, *59th Annual Symposium on Foundations of Computer Science*, pages 259–267, Paris, France, October 7–9, 2018. IEEE Computer Society Press.

[Man26] Atul Mantri. Classical verification of quantum computation beyond the polynomial hierarchy, 2026.

[Mec19] Elizabeth S. Meckes. *The Random Matrix Theory of the Classical Compact Groups*. Cambridge Tracts in Mathematics. Cambridge University Press, 2019.

[MH25] Fermi Ma and Hsin-Yuan Huang. How to construct random unitaries. In Michal Koucký and Nikhil Bansal, editors, *57th Annual ACM Symposium on Theory of Computing*, pages 806–809, Prague, Czechia, June 23–27, 2025. ACM Press.

[MSY25] Tomoyuki Morimae, Yuki Shirakawa, and Takashi Yamakawa. Cryptographic characterization of quantum advantage. In Michal Koucký and Nikhil Bansal, editors, *57th Annual ACM Symposium on Theory of Computing*, pages 1863–1874, Prague, Czechia, June 23–27, 2025. ACM Press.

[MY23] Tomoyuki Morimae and Takashi Yamakawa. Proofs of quantumness from trapdoor permutations. In Yael Tauman Kalai, editor, *ITCS 2023: 14th Innovations in Theoretical Computer Science Conference*, volume 251, pages 87:1–87:14, Cambridge, MA, USA, January 10–13, 2023. Leibniz International Proceedings in Informatics (LIPIcs).

[Nay99] Ashwin Nayak. Optimal lower bounds for quantum automata and random access codes. In *40th Annual Symposium on Foundations of Computer Science*, pages 369–377, New York, NY, USA, October 17–19, 1999. IEEE Computer Society Press.

[Ope26] OpenAI. Polynomial-Time Unitary Synthesis from a Boolean Oracle. OpenAI Math Release preprint, October 2026.

[Raz87] A. A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR*, 41(4):333–338, April 1987.

[RT19] Ran Raz and Avishay Tal. Oracle separation of BQP and PH. In Moses Charikar and Edith Cohen, editors, *51st Annual ACM Symposium on Theory of Computing*, pages 13–23, Phoenix, AZ, USA, June 23–26, 2019. ACM Press.

[Sha90] Adi Shamir. IP=PSPACE. In *31st Annual Symposium on Foundations of Computer Science*, pages 11–15, St. Louis, MO, USA, October 22–24, 1990. IEEE Computer Society Press.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Santa Fe, NM, USA, November 20–22, 1994. IEEE Computer Society Press.

[Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997.

[Sto83] Larry J. Stockmeyer. The complexity of approximate counting (preliminary version). In *15th Annual ACM Symposium on Theory of Computing*, pages 118–126, Boston, MA, USA, April 25–27, 1983. ACM Press.

[SZ25] Omri Shmueli and Mark Zhandry. On one-shot signatures, quantum vs. classical binding, and obfuscating permutations. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology – CRYPTO 2025, Part II*, volume 16001 of *Lecture Notes in Computer Science*, pages 350–383, Santa Barbara, CA, USA, August 17–21, 2025. Springer, Cham, Switzerland.

[TWZ26] Ewin Tang, John Wright, and Mark Zhandry. Conjugate queries can help, 2026.

[TZ26] Kabir Tomer and Mark Zhandry. On the cryptographic foundations of interactive quantum advantage. In *STOC 2026*, 2026. arXiv:2510.05082.

[YZ22] Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. In *63rd Annual Symposium on Foundations of Computer Science*, pages 69–74, Denver, CO, USA, October 31 – November 3, 2022. IEEE Computer Society Press.

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 239–268, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Cham, Switzerland.

A From state preparation channels to unitaries

Let $\vec{\sigma} = \{\sigma_t\}_{t\in\mathbb{N}}$ be any fixed family of states, where σ_t is a state on t qubits. Fix a purification of each σ_t on $2t$ qubits, and independently sample $U_t \leftarrow \mathsf{Haar}(\{0,1\}^t)$. Write $S = \{S_{\sigma_t}^{U_t}\}_{t\in\mathbb{N}}$. Access to $\vec{\sigma}$ means access to a channel which, on input t, returns a fresh copy of σ_t. Queries to level t take at least t time, and we allow controlled queries to S.

Throughout this section, complexity classes refer to promise problems. An oracle for $\mathsf{QCMA}^{\vec{\sigma}}$ means an oracle for its complete verification promise problem: given a verifier and a runtime bound in unary, distinguish whether some classical witness is accepted with probability at least $2/3$, or every classical witness is accepted with probability at most $1/3$. Its behavior outside the promise is arbitrary.

Theorem A.1. *With probability* 1 *over* $\{U_t\}_{t\in\mathbb{N}}$,

$$\mathsf{QCMA}^{S} \subseteq \mathsf{BQP}^{S,\mathsf{QCMA}^{\vec{\sigma}}}.$$

Proof. Let Π be any promise problem in QCMA^S with verifier Ver^S. By amplification, assume that its completeness is at least $11/12$ and its soundness is at most $1/12$. Let $q(n) \geq n$ be a polynomial upper bound on its runtime, witness length, and circuit size after hardcoding x, w. In particular, it makes at most $q(n)$ queries, all to levels at most $q(n)$. We describe a QPT algorithm $\mathcal{A}^{S,\mathsf{QCMA}^{\vec{\sigma}}}$ as follows:

1. On input $x \in \{0,1\}^n$, let $d = \lceil c \log n \rceil$ for a sufficiently large constant c.
2. For each $t \in [1,d]$, apply Theorem 3.13 to $S_{\sigma_t}^{U_t}$ with error parameter $1/q(n)^3$ and failure probability 2^{-d}. This takes $2^{O(d)}\mathsf{poly}(n) = \mathsf{poly}(n)$ time. Let $\widetilde{S}_t$ be the resulting circuit.
3. Construct a new verifier $\widetilde{\mathsf{Ver}}^{\vec{\sigma}}$ which hardcodes x and the descriptions of the circuits $\widetilde{S}_t$. It runs $\mathsf{Ver}^S(x,w)$, replacing queries to levels $t \leq d$ by $\widetilde{S}_t$. For each $t > d$, replace queries by the stateful simulator from Lemma 3.22, with query bound $q(n)$ and error $1/q(n)^2$, using fresh copies of σ_t. Use a separate simulator for each level, retaining its state between queries.
4. Query the complete $\mathsf{QCMA}^{\vec{\sigma}}$ verification promise problem on $\widetilde{\mathsf{Ver}}^{\vec{\sigma}}$ and output its answer.

Each simulator uses $O(q(n)^4)$ copies, and there are at most $q(n)$ levels. Thus, $\widetilde{\mathsf{Ver}}^{\vec{\sigma}}$ has polynomial runtime and its description has polynomial length.

To show correctness, first leave the oracles at levels at most d unchanged and average over independent Haar unitaries at the larger levels. For $\ell \in [d, q(n)]$, define

$$a_\ell(x,w) = \mathop{\mathbb{E}}_{\{U'_t\}_{t>\ell}}\left[\Pr\left[\mathsf{Ver}^{S_{\sigma_1}^{U_1},\ldots,S_{\sigma_\ell}^{U_\ell},S_{\sigma_{\ell+1}}^{U'_{\ell+1}},\ldots,S_{\sigma_{q(n)}}^{U'_{q(n)}}}(x,w) = 1\right]\right].$$

In particular, $a_{q(n)}(x,w) = \Pr\left[\mathsf{Ver}^S(x,w) = 1\right]$. For fixed $U_1,\ldots,U_{\ell-1}$, the expectation of $a_\ell(x,w)$ over U_ℓ is $a_{\ell-1}(x,w)$. Each query to $S_{\sigma_\ell}^{U_\ell} = C_{U_\ell} S_{\sigma_\ell} C_{U_\ell}^\dagger$ uses at most two queries to U_ℓ or its inverse. Thus, by Theorem 3.16,

$$\Pr\left[|a_\ell(x,w) - a_{\ell-1}(x,w)| > \frac{1}{q(n)^2}\right] \leq 2\exp\left(-\frac{2^\ell - 2}{384 q(n)^6}\right).$$

The other oracle levels can be treated as fixed side information or independent randomness in this application. Taking a union bound over $\ell > d$, $x \in \{0,1\}^n$, and $w \in \{0,1\}^{q(n)}$, the probability that any of these bounds fails is at most

$$2^{n+q(n)+1} q(n) \exp\left(-\frac{2^{d+1}-2}{384 q(n)^6}\right) \leq \frac{1}{n^2}$$

for all sufficiently large n, by choosing c sufficiently large. Whenever none fails, simultaneously for all x, w,

$$\left|\Pr\left[\mathsf{Ver}^S(x,w)=1\right] - a_d(x,w)\right| \leq \frac{1}{q(n)}.$$

By Lemma 3.22, replacing the averaged oracles at levels above d by their simulators changes acceptance probabilities by at most $q(n)/q(n)^2 = 1/q(n)$. This follows by replacing one level at a time, keeping the other levels as side oracles. Whenever all tomography estimates are accurate, replacing the remaining oracles by $\widetilde{S}_t$ changes acceptance probabilities by at most $q(n)/q(n)^3 = 1/q(n)^2$. These bounds hold for all x, w, so

$$\left|\Pr\left[\mathsf{Ver}^S(x,w)=1\right] - \Pr\left[\widetilde{\mathsf{Ver}}^{\vec{\sigma}}(w)=1\right]\right| \leq \frac{2}{q(n)} + \frac{1}{q(n)^2} \leq \frac{1}{12}$$

for all sufficiently large n. Consequently, on a yes input some witness is accepted by $\widetilde{\mathsf{Ver}}^{\vec{\sigma}}$ with probability at least $5/6$, and on a no input every witness is accepted with probability at most $1/6$. The query to the $\mathsf{QCMA}^{\vec{\sigma}}$ oracle is therefore within its promise and gives the correct answer.

Except for a set of choices of $\{U_t\}_t$ of probability at most $1/n^2$, the algorithm is correct on every promised input of length n whenever tomography succeeds. Tomography succeeds with probability at least $1 - d2^{-d} \geq 5/6$ for all sufficiently large n. By Borel–Cantelli, with probability 1 over $\{U_t\}_t$, the algorithm correctly decides Π on all sufficiently large input lengths. Hardcoding the answers at the finitely many remaining lengths gives an algorithm deciding Π on all promised inputs. Finally, there are only countably many verifiers, so with probability 1 the containment holds for all of them simultaneously. □

Corollary A.2. *If* $\mathsf{BQP}^{\vec{\sigma}} = \mathsf{QCMA}^{\vec{\sigma}}$*, then with probability* 1 *over* $\{U_t\}_{t\in\mathbb{N}}$*,*

$$\mathsf{BQP}^S = \mathsf{QCMA}^S.$$

Proof. By Theorem A.1, it suffices to implement the $\mathsf{QCMA}^{\vec{\sigma}}$ verification oracle in BQP^S. By assumption, this promise problem has a $\mathsf{BQP}^{\vec{\sigma}}$ algorithm, which we amplify to have error at most $1/12$. Each copy of σ_t needed by this algorithm can be prepared by applying $S_{\sigma_t}^{U_t}$ to the all-zero state and tracing out the purification register. Thus, $\mathsf{QCMA}^S \subseteq \mathsf{BQP}^S$, and the reverse containment is immediate. □

B EV-OWPuzzs from OSP

B.1 Commitments and the implication from OSP

Definition B.1 (Commitments with a classical committer [BK25])**.** A statistically sum-binding, computationally hiding commitment with a classical committer and a quantum receiver consists of

four algorithms $(\mathsf{Com}, \mathsf{Rec}, \mathsf{Open}, \mathsf{Ver})$ with the following syntax. In the commit phase, the parties interact as follows:

$$(\mathsf{st}, \rho) \leftarrow \langle \mathsf{Com}(1^n, b), \mathsf{Rec}(1^n) \rangle, \tag{8}$$

where Com is PPT, Rec is QPT, and all communication is classical. The committer's output st is classical, whereas the receiver's output ρ may be quantum. In the opening phase, the PPT algorithm $\mathsf{Open}(\mathsf{st})$ outputs a single classical message (b, d), where b is the original input bit. The QPT algorithm $\mathsf{Ver}(\rho, b, d)$ accepts or rejects. We require the following properties.

- *Correctness.* For each $b \in \{0,1\}$, an honest opening to b is accepted with probability at least $1 - \mathsf{negl}(n)$.
- *Computational hiding.* The final views of every nonuniform QPT receiver interacting with $\mathsf{Com}(1^n, 0)$ and $\mathsf{Com}(1^n, 1)$ are computationally indistinguishable.
- *Statistical sum-binding.* Consider any computationally unbounded adversarial committer $\mathcal{C}$ that interacts with the honest receiver in the commit phase and retains a private state σ. After this phase, sample an independent uniform bit b', and let $\mathcal{C}$, given σ and b', output an opening string d. For every such $\mathcal{C}$, there is a negligible function $\eta_{\mathcal{C}}$ such that

$$\Pr\big[\mathsf{Ver}(\rho, b', d) = 1\big] \leq \frac{1}{2} + \eta_{\mathcal{C}}(n). \tag{9}$$

  If the committer does not provide an opening string, the receiver rejects.

Theorem B.2 (Bartusek–Khurana [BK25])**.** *If OSP exist, then commitments as in Definition B.1 exist.*

Theorem B.3. *If commitments as in Definition B.1 exist, then EV-OWPuzzs exist with perfect correctness, a QPT sampler, and a classical deterministic polynomial-time verifier.*

B.2 The transcript determines the committed bit

Run the honest commit phase with a uniform bit B. Let R be the committer's random tape for this phase, padded to a fixed polynomial length, and let T be the transcript. Write $W = (B, R) \in \{0,1\}^m$, where $m = m(n) = \mathsf{poly}(n)$, so that W is uniform. For $w = (b, r)$, define $\mathsf{Cons}(t, w) = 1$ if replaying $\mathsf{Com}(1^n, b)$ with tape r and the receiver messages in t reproduces all the committer messages in t and terminates after exactly the messages in t. Otherwise set $\mathsf{Cons}(t, w) = 0$. This is a classical deterministic polynomial-time test, including on malformed transcripts. For each fixed bit $b \in \{0,1\}$, define

$$\begin{aligned} &S_t \coloneqq \{w \in \{0,1\}^m : \mathsf{Cons}(t, w) = 1\}, \qquad S_{t,b} \coloneqq \{(\beta, r) \in S_t : \beta = b\}, \\ &p_t \coloneqq \Pr[B = 0 \mid T = t]. \end{aligned} \tag{10}$$

All conditional statements below concern transcripts t with $\Pr[T = t] > 0$.

Lemma B.4. *Conditioned on $T = t$, the tape W is uniform over S_t. Moreover, the receiver's conditional state ρ_t is independent of W. In particular, $p_t = |S_{t,0}|/|S_t|$.*

Proof. For a fixed tape $w \in S_t$, the probability of the transcript t is the probability that the receiver sends the receiver messages in t upon receiving the committer messages in t. This probability depends only on t, and is zero for $w \notin S_t$. Since W is uniform, Bayes' rule proves the first assertion. The receiver starts in a state independent of W and communicates with the committer only classically. Conditioned on its incoming and outgoing messages, its final state is therefore determined by t alone, proving the second assertion. □

Lemma B.5. *There is a negligible function ν such that*

$$\mathop{\mathbb{E}}_{T}[\min\{p_T, 1-p_T\}] \le \nu(n). \tag{11}$$

Consequently, $\mathbb{E}_T[p_T(1-p_T)] \le \nu(n)$.

Proof. For $b \in \{0,1\}$ with $S_{t,b} \neq \varnothing$, define $q_b(t)$ by the following experiment. Sample $w = (b, r)$ uniformly from $S_{t,b}$, replay the committer to reconstruct its final state st, and run $\mathsf{Open}(\mathsf{st})$ with fresh randomness. Let $q_b(t)$ be the probability that this opening to b is accepted against ρ_t. Set $q_b(t) = 0$ when $S_{t,b} = \varnothing$. By Lemma B.4, this experiment gives exactly the honest opening success probability conditioned on $T = t$ and $B = b$, whenever this event has positive probability. Thus correctness gives

$$\mathop{\mathbb{E}}_{T}[p_T q_0(T) + (1-p_T) q_1(T)] \ge 1 - \mathsf{negl}(n). \tag{12}$$

Consider the following unbounded committer $\mathcal{C}$ in the sum-binding experiment. In the commit phase, it samples a uniform bit and follows the honest committer, retaining t. Given the target bit $b' = b$ in the opening phase, it enumerates $S_{t,b}$. If this set is empty, it aborts; otherwise, it samples a uniform tape from this set, reconstructs st, and runs $\mathsf{Open}(\mathsf{st})$. Its commit phase has the honest transcript distribution, and the receiver's conditional state is ρ_t regardless of the tape originally used. Its opening success probability is therefore $\mathbb{E}_T[(q_0(T)+q_1(T))/2]$. By Eq. (9), for a negligible $\eta = \eta_{\mathcal{C}}$,

$$\mathop{\mathbb{E}}_{T}\left[\frac{q_0(T)+q_1(T)}{2}\right] \le \frac{1}{2} + \eta(n). \tag{13}$$

For every $p, q_0, q_1 \in [0,1]$,

$$\begin{aligned} pq_0 + (1-p)q_1 &= \frac{q_0+q_1}{2} + \left(p - \frac{1}{2}\right)(q_0 - q_1) \\ &\le \frac{q_0+q_1}{2} + \left|p - \frac{1}{2}\right|. \end{aligned} \tag{14}$$

Combining this inequality with Eqs. (12) and (13) yields

$$\mathop{\mathbb{E}}_{T}[\min\{p_T, 1-p_T\}] = \frac{1}{2} - \mathop{\mathbb{E}}_{T}\left[\left|p_T - \frac{1}{2}\right|\right] \le \mathsf{negl}(n) + \eta(n) = \mathsf{negl}(n). \tag{15}$$

The second claim follows from $p(1-p) \le \min\{p, 1-p\}$. □

B.3 Construction and proof of security

Let $L = 2^{\lceil \log_2(m+3) \rceil}$. For $0 \le \ell < L$, let $\mathcal{H}_\ell = \{h_A : A \in \{0,1\}^{\ell \times m}\}$, where $h_A(w) = Aw$ over $\mathbb{F}_2$. These families can be sampled and evaluated in classical polynomial time; $\mathcal{H}_0$ consists of the map with empty output. For distinct w, w', the vector $A(w - w')$ is uniform over $\{0,1\}^\ell$ when A is uniform, so

$$\Pr_{h \leftarrow \mathcal{H}_\ell}[h(w) = h(w')] = 2^{-\ell} \quad \text{for all distinct } w, w' \in \{0,1\}^m. \tag{16}$$

The sampler $\mathsf{Samp}_0(1^n)$ does the following.

1. Sample $W = (B, R) \leftarrow \{0,1\}^m$ and run the honest commit phase with input B and committer tape R, obtaining the transcript T.
2. Independently sample $\ell \leftarrow \{0, \dots, L-1\}$ and $h \leftarrow \mathcal{H}_\ell$ uniformly.
3. Output the puzzle $x = (1^n, T, \ell, h, h(W))$ and the answer W.

On input $x = (1^n, t, \ell, h, z)$ and an answer w', the verifier Ver_0 accepts exactly when

$$\mathsf{Cons}(t, w') = 1 \quad \text{and} \quad h(w') = z, \tag{17}$$

Malformed inputs are rejected. The sampler is QPT and the verifier is deterministic classical polynomial-time. Correctness is perfect because the actual tape always reproduces the actual transcript and passes the hash check.

Lemma B.6 (Weak security)**.** *For every nonuniform QPT algorithm $\mathcal{A}$ and all sufficiently large n,*

$$\Pr[\mathsf{Ver}_0(x, \mathcal{A}(x)) = 1 : (x, w) \leftarrow \mathsf{Samp}_0(1^n)] \le 1 - \frac{1}{64L(n)}. \tag{18}$$

Proof. Suppose a nonuniform QPT solver $\mathcal{A}$ violates Eq. (18) for infinitely many n, and fix such an n. Construct a nonuniform QPT algorithm $\mathcal{F}$ as follows. Given $1^n, t$, for each $\ell = L-1, L-2, \dots, 0$, run n independent trials in which $h \leftarrow \mathcal{H}_\ell$ and $z \leftarrow \{0,1\}^\ell$ are sampled uniformly and $\mathcal{A}$ is run on $(1^n, t, \ell, h, z)$. Return the first answer accepted by Ver_0, or $\perp$ if none is accepted. Each trial uses a fresh execution of $\mathcal{A}$. Whenever $\mathcal{F}$ returns a value other than $\perp$, that value lies in S_t.

For each possible honest transcript t, set

$$k_t := \lceil \log_2 |S_t| \rceil + 2. \tag{19}$$

Since $1 \le |S_t| \le 2^m$, this length occurs in the search and satisfies

$$k_t \le L - 1, \qquad 4|S_t| \le 2^{k_t} < 8|S_t|. \tag{20}$$

Let e_t be the failure probability of $\mathcal{A}$ on a genuine puzzle conditioned on $T = t$ and $\ell = k_t$. By Lemma B.4, such a puzzle uses a uniform $w \leftarrow S_t$ and target $h(w)$. Since the sampler chooses each length with probability $1/L$, the assumed solver failure probability gives

$$\mathop{\mathbb{E}}_T[e_T] < \frac{1}{64}. \tag{21}$$

Call t good if $e_t \le 1/4$. By Markov's inequality, $\Pr[T \text{ is not good}] \le 1/16$.

Success at the critical length. Fix a good transcript t. For any fixed $w \in S_t$, a random $h \leftarrow \mathcal{H}_{k_t}$ maps another tape in S_t to $h(w)$ with probability at most $(|S_t|-1)2^{-k_t} \le 1/4$, by the union bound. Hence, on a genuine puzzle, $\mathcal{A}$ succeeds and w has a unique hash value with probability at least $1 - e_t - 1/4 \ge 1/2$. For each fixed h, these tapes give distinct targets $h(w)$. Each such target has probability $1/|S_t|$ when sampling $w \leftarrow S_t$ and using $h(w)$, and probability 2^{-k_t} when sampling z uniformly. Restricting to these targets and averaging over h gives

$$\begin{aligned}\Pr[\text{a trial at length } k_t \text{ succeeds}] &\ge \frac{|S_t|}{2^{k_t}}\left(1 - e_t - \frac{1}{4}\right) \\ &\ge \frac{1}{8}\cdot\frac{1}{2} = \frac{1}{16}. \end{aligned} \tag{22}$$

Thus all n independent trials at this length fail with probability at most $(15/16)^n$.

Excluding tapes with the wrong bit. Condition on $T = t$ and $B = b$. To accept a tape with bit $1-b$, a trial must choose $z \in h(S_{t,1-b})$. For each fixed h, this set has at most $|S_{t,1-b}|$ elements, so a uniform length-ℓ target hits it with probability at most $|S_{t,1-b}|2^{-\ell}$. For the analysis, run all trials, including those $\mathcal{F}$ would skip after returning. Let Wrong denote the event that some trial with $\ell \ge k_t$ accepts a tape with bit $1-b$. The union bound over the n trials at each length gives

$$\begin{aligned}\Pr[\mathsf{Wrong} \mid T = t, B = b] &\le n|S_{t,1-b}| \sum_{\ell=k_t}^{L-1} 2^{-\ell} \\ &\le \frac{n|S_{t,1-b}|}{2|S_t|}, \end{aligned} \tag{23}$$

since $\sum_{\ell=k_t}^{L-1} 2^{-\ell} \le 2^{1-k_t}$ and $2^{k_t} \ge 4|S_t|$. By Lemma B.4, averaging over B gives $\Pr[\mathsf{Wrong} \mid T = t] \le \frac{n}{2}[p_t(1-p_t) + (1-p_t)p_t] = np_t(1-p_t)$.

For good t, some trial at length k_t succeeds except with probability $(15/16)^n$. Since $\mathcal{F}$ searches lengths in decreasing order, it then returns at length at least k_t. Unless Wrong occurs, this output has bit B. Combining these bounds with $\Pr[T \text{ is not good}] \le 1/16$ and Lemma B.5 gives

$$\begin{aligned}\Pr[\mathcal{F}(1^n, T) \text{ returns a tape whose bit is } B] &\ge 1 - \frac{1}{16} - \left(\frac{15}{16}\right)^n - n\,\mathop{\mathbb{E}}_{T}[p_T(1-p_T)] \\ &\ge \frac{15}{16} - \mathsf{negl}(n). \end{aligned} \tag{24}$$

A nonuniform QPT receiver can follow the honest commit phase and run $\mathcal{F}(1^n, T)$, outputting the bit of the returned tape or a uniform bit if $\mathcal{F}$ returns $\perp$. This contradicts computational hiding for infinitely many n. □

Proof of Theorem B.3. The construction above has perfect correctness, a QPT sampler, and a classical deterministic polynomial-time verifier. By Lemma B.6, its security gap $1/(64L(n))$ is inverse polynomial. Apply the security amplification theorem for EV-OWPuzzs of Chung, Goldin, and Gray [CGG24]. The resulting puzzle consists of polynomially many independent component puzzles and accepts exactly when every component answer is accepted. Its solving probability is negligible, and its correctness remains perfect. The sampler remains QPT and the verifier remains classical, deterministic, and polynomial-time. □