Structured Quantum Hardness from Unstructured Assumptions

Xiao Liang[1*], Takashi Yamakawa[2], and Chak Fai Yung[3]

[1] The Chinese University of Hong Kong, Hong Kong
xiaoliang@cuhk.edu.hk
[2] NTT Social Informatics Laboratories, Japan
takashi.yamakawa@ntt.com
[3] The Chinese University of Hong Kong, Hong Kong
cfyung2@cse.cuhk.edu.hk

Abstract. Whether useful structured hardness can arise from unstructured assumptions is a fundamental question in complexity and cryptography. The longstanding Bennett–Gill question asks whether $\mathsf{NP} \cap \mathsf{coNP}$ remains hard relative to a random oracle; [GIK+23, STOC'23] recently gave a conditional affirmative answer in the classical setting. We establish the quantum analogue: if $\mathsf{UP} \not\subseteq \mathsf{BQP}$, then $\mathsf{NP}^O \cap \mathsf{coNP}^O \not\subseteq \mathsf{BQP}^O$ with probability 1 over a random oracle O.

Beyond extending the classical result, this separation has a distinctly quantum significance: Shor's algorithms defeat canonical number-theoretic hardness candidates, while lattice assumptions and other leading post-quantum cryptographic assumptions do not directly supply hard languages with two-sided classical certification on every input. Our conditional oracle separation provides evidence that such structured quantum hardness can nevertheless arise from unstructured sources.

Complementing this result, we rule out fully black-box quantum reductions of worst-case quantum hardness in $\mathsf{NP}\cap\mathsf{coNP}$ to post-quantum injective one-way functions and indistinguishability obfuscation. This barrier excludes a broad class of black-box approaches to deriving structured quantum hardness from generic cryptography primitives, providing partial evidence that the non-black-box techniques in our first result are essential.

Keywords: Quantum Complexity · Random Oracle · Non-Interactive Zero Knowledge · Pseudorandom Hash functions · Black-Box Impossibility

* The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No.: CUHK 24208025].

Table of Contents

Table of Contents . . . iii
1 Introduction . . . 1
1.1 Our Results . . . 2
1.2 Related Work . . . 4
2 Technical Overview . . . 5
2.1 Building NIZKs with an Unbounded Prover: the Classical Starting Point . . . 6
2.2 Strong δ-Dense PRHFs and Their Application to NIZK . . . 8
2.3 Completing the Separation . . . 10
2.4 A Random Oracle is a Strong PRHF . . . 10
2.5 Quantum Black-Box Impossibility via a Universal Decision Oracle . . . 14
3 Preliminaries . . . 16
3.1 Random Oracles . . . 17
3.2 Useful Lemmas . . . 17
3.3 Simulating Two Oracles using One . . . 20
3.4 Non-Interactive Zero-Knowledge Proof . . . 21
3.5 Compressed Partially-Fixed Oracle . . . 23
4 Strong δ-Dense PRHF: Definition and Construction . . . 24
4.1 Strong δ-Dense PRHF . . . 24
4.2 Random Oracle is a Strong $(1-\frac{1}{e})$-Dense PRHF . . . 25
4.3 Proof of Lem. 10 . . . 26
4.3.1 Hybrids H_0 and H_1 . . . 27
4.3.2 Hybrid H_2 . . . 28
4.3.3 Hybrid $H_{2.5}$ . . . 28
4.3.4 Hybrid H_3 . . . 29
4.3.5 Hybrid H_4 . . . 31
4.3.6 Hybrid H_5 . . . 32
4.3.7 Hybrid H_6 . . . 33
4.4 Proof of Claim 1: $H_0 \overset{s}{\approx} H_1$ . . . 34
4.4.1 Proof Strategy Overview . . . 34
4.4.2 Notation . . . 35
4.4.3 Technical Tools regarding Random Oracles . . . 35
4.4.4 Proving $H_0 \overset{s}{\approx} H_1$ . . . 39
4.5 Proof of $H_1 \overset{s}{\approx} H_2$ . . . 49
4.6 Proof of $H_2 \overset{c_q}{\approx} H_{2.5}$ . . . 54
5 NIZK in the URS-PRHF Model . . . 57
5.1 Construction . . . 58
5.2 Security Proof . . . 59
6 NIZK in QROM . . . 65
6.1 Structure of Our Construction . . . 65
6.2 Instantiating the PRHF . . . 66
6.3 Replacing the URS . . . 66
6.3.1 Completeness . . . 67
6.3.2 Soundness . . . 67

6.3.3 Zero Knowledge . . . 68
6.4 Combining the Two Oracles . . . 70
6.5 Final Construction . . . 70
7 Oracle Separation between $\mathsf{NP} \cap \mathsf{coNP}$ and BQP . . . 72
7.1 Technical Tool . . . 72
7.2 Post-Quantum injOWF Implies $\mathsf{UP} \not\subseteq \mathsf{BQP}$ . . . 75
7.3 The Oracle-Aided Language $\mathcal{L}^O$ . . . 76
7.4 Membership in $\mathsf{NP}^O \cap \mathsf{coNP}^O$ . . . 76
7.5 Non-Membership in BQP^O . . . 76
8 Black-Box Impossibility of Constructing Quantum-Hard $\mathsf{NP} \cap \mathsf{coNP}$ Problems . . . 81
8.1 Black-Box Framework and Technical Tools . . . 81
8.1.1 Oracle Access and the Black-Box Model . . . 81
8.1.2 The Universal Decision Oracle . . . 84
8.1.3 A Quantum Hybrid Bound for Coupled Oracles . . . 85
8.2 Warm-Up: Separation from injOWFs . . . 87
8.3 Separation from iO and injOWFs . . . 92
8.3.1 Oracle Construction and Resource Bounds . . . 92
8.3.2 One-Wayness in the Presence of the Augmented Oracle . . . 94
8.3.3 Obfuscation Security in the Presence of the Augmented Oracle . . . 96
8.3.4 Consequence for Efficient Black-Box Reductions . . . 103
9 Acknowledgments and AI Disclosure . . . 104
References . . . 105
A More Basic Preliminaries . . . 107
A.1 Probability Inequalities . . . 107
A.2 Statistical Distance and Properties . . . 108
A.3 Efficient Approximability of $\frac{1}{e}$ and $1 - \frac{1}{e}$ . . . 109
B Length-Tripling Random Oracles as Post-Quantum injOWFs . . . 110

1 Introduction

Understanding how the structure of a computational problem governs its complexity is a central theme in theoretical computer science. Additional structure may enable efficient algorithms, but it may also endow a problem with useful properties without making it tractable. Distinguishing these possibilities is essential to understanding which forms of computational hardness can exist and what they can accomplish. This tension is particularly pronounced in cryptography, where the algebraic structure underlying a hard problem often enables powerful applications, while simultaneously providing a foothold for algorithmic attacks. A fundamental question is therefore whether useful forms of structured hardness inherently require structured assumptions, or whether they can instead arise from generic, unstructured sources of computational difficulty.

A concrete setting for studying this question is the class $\mathsf{NP} \cap \mathsf{coNP}$, whose languages admit polynomial-size, efficiently verifiable certificates for both membership and nonmembership. This two-sided certification requirement provides a complexity-theoretic notion of structure. Whether such a requirement is compatible with intractability is captured by the classical conjecture $\mathsf{P} \neq \mathsf{NP} \cap \mathsf{coNP}$. However, compared with $\mathsf{P} \neq \mathsf{NP}$, this conjecture is supported by a much narrower collection of natural candidate problems. Among the most prominent are decision problems associated with integer factorization and discrete logarithms. Their presumed hardness has motivated extensive study and underlies important cryptographic constructions, but their considerable mathematical structure leaves open the possibility of unexpected algorithmic advances. For example, number-field-sieve algorithms achieve heuristic subexponential running times for integer factorization [BLP93] and for discrete logarithms in prime fields [Gor93].

A longstanding line of research asks whether hardness in $\mathsf{NP} \cap \mathsf{coNP}$ can instead be obtained from unstructured sources [BG81, BI87, Rud88, Tar89, BDV17, GIK+23]. Random oracles provide a natural framework for investigating this question. A uniformly random function supplies computational difficulty without imposing a particular algebraic structure; a separation relative to such an oracle can therefore provide evidence that hardness need not depend on special properties of an explicit mathematical construction. The seminal work of Bennett and Gill [BG81] established $\mathsf{P}^{\mathcal{O}} \neq \mathsf{NP}^{\mathcal{O}}$ with probability 1 over a random oracle $\mathcal{O}$, and asked whether $\mathsf{P}^{\mathcal{O}} \neq \mathsf{NP}^{\mathcal{O}} \cap \mathsf{coNP}^{\mathcal{O}}$ holds as well. They proved the latter separation when $\mathcal{O}$ is a random *permutation* (a structured source of hardness), but left the random-function case open.

The same distinction arises when constructing hard languages in $\mathsf{NP} \cap \mathsf{coNP}$ from cryptographic primitives. One-way permutations (OWPs) yield such languages through black-box constructions. Injective one-way functions (inj-OWFs), by contrast, are viewed as an unstructured source of hardness in this context: a sufficiently length-expanding random function is both injective and one-way except with negligible probability [GIK+23]. Here, "unstructured" does not mean the absence of every mathematical property, but rather the absence of additional constraints that cannot be obtained from a random function. In this regard,, results of Blum and Impagliazzo [BI87] and Rudich [Rud88] rule out black-box constructions of hard languages in $\mathsf{NP} \cap \mathsf{coNP}$ from OWFs; Bitansky, Degwekar, and Vaikuntanathan [BDV17] strengthened this barrier to inj-OWFs, even when additionally allowing black-box access to *indistinguishability obfuscation* (iO). These results expose a substantial obstacle to deriving two-sided certification together with hardness from generic cryptographic primitives.

Recently, [GIK+23] overcame this obstacle through a non-black-box approach. Assuming $\mathsf{UP} \not\subseteq \mathsf{RP}$ (an assumption implied by the existence of inj-OWFs), they established $\mathsf{NP}^{\mathcal{O}} \cap \mathsf{coNP}^{\mathcal{O}} \not\subseteq \mathsf{P}^{\mathcal{O}}$ with probability 1 over a random oracle $\mathcal{O}$. Their result gives a *conditional* affirmative answer to Bennett–Gill question.

Structured Quantum Hardness? In contrast, the picture in the quantum regime is much less understood. In fact, *this is not merely asking the quantum version of a classical question*: Shor's algorithms [Sho94] solve integer factorization and discrete logarithms in quantum polynomial time, removing two of the principal sources of evidence for classical hardness in $\mathsf{NP} \cap \mathsf{coNP}$. These algorithms sharply illustrate the underlying concern: the structure supporting useful certification and cryptographic applications can also enable an efficient quantum algorithm. Thus, the quantum setting asks whether two-sided classical certification can coexist with quantum hardness, *even after allowing the algorithmic capabilities that defeat canonical problems serving as evidence for the classical Bennett–Gill question.*

Moreover, the leading sources of conjectured *post-quantum* hardness makes the situation more fascinating: Lattice problems, including learning with errors (LWE), short integer solution (SIS), and the approximate shortest vector problem (GapSVP), are widely conjectured to resist quantum algorithms at appropriate parameters and underpin much of post-quantum cryptography [Pei15]. Nevertheless, these conjectures do not directly furnish hard languages in $\mathsf{NP} \cap \mathsf{coNP}$. The distinction here is subtle: the obstacle is not simply the absence of efficiently checkable no-certificates. Indeed, Aharonov and Regev [AR04] established two-sided classical certification for suitable gap versions of lattice problems, placing them in $\mathsf{Promise\text{-}NP} \cap \mathsf{Promise\text{-}coNP}$. These guarantees, however, apply only to inputs satisfying the promise; they do not provide certificates of the correct answer on *every* input. Likewise, standard formulations of SIS and LWE do not directly supply the required language with two-sided certification either. It is also worth mentioning that other approaches to post-quantum cryptography draw on problems in coding theory, multivariate polynomial systems, and isogenies of elliptic curves [CJL+16]. These approaches raise the same complexity-theoretic issue: conjectured hardness of a search task or a promise problem does not, by itself, provide a language with efficiently verifiable certificates for both answers on every input. Thus, the availability of post-quantum cryptographic candidates therefore does not by itself resolve the question considered here. This gives the quantum Bennett–Gill question a distinctive motivation beyond its classical version.

There is also an important distinction between assuming quantum hardness for an already structured primitive and deriving structured quantum hardness from an unstructured source. For example, a one-way permutation secure against quantum polynomial-time algorithms yields a language in $\mathsf{NP} \cap \mathsf{coNP}$ outside BQP: its unique inverse provides certificates for either value of an inverse bit, while an efficient quantum algorithm for deciding these bits would allow inversion, using error amplification. Here, however, the structure needed for two-sided certification is already supplied by the assumption: every output has exactly one preimage. Likewise, the random-permutation separation of [BBBV97] provides relativized evidence for quantum hardness in $\mathsf{NP} \cap \mathsf{coNP}$, but starts from an oracle with this same structural guarantee. Our aim is instead to obtain two-sided certification together with quantum hardness without assuming such permutation structure at the outset.

Question: *Can structured quantum hardness be derived from unstructured sources?*

1.1 Our Results

We obtain both positive and negative results concerning this question. We summarize them below.

Unstructured Quantum Hardness. Our first result gives a conditional affirmative answer to the quantum Bennett–Gill question. Specifically, assuming that UP contains a language outside BQP, we show that a random oracle almost surely yields a language with two-sided classical certification that remains hard even for quantum algorithms. Note that both the assumption $\mathsf{UP} \not\subseteq \mathsf{BQP}$ and a random oracle are regarded as sources of unstructured hardness. Thus, Thm. 1 provides evidence

toward an affirmative answer to the question posed above. (Here and throughout, $\mathsf{BQP}^{\mathcal{O}}$ denotes bounded-error quantum polynomial time with *quantum* query access to $\mathcal{O}$.)

Theorem 1. *If* $\mathsf{UP} \not\subseteq \mathsf{BQP}$*, then it holds that* $\mathsf{NP}^{\mathcal{O}} \cap \mathsf{coNP}^{\mathcal{O}} \not\subseteq \mathsf{BQP}^{\mathcal{O}}$ *with probability 1 over the choice of a random oracle* $\mathcal{O}$*.*

Importantly, the proof of Thm. 1 makes non-black-box use of a language $\mathcal{L} \in \mathsf{UP} \setminus \mathsf{BQP}$ to construct the desired quantum-hard language.[4] If this use of $\mathcal{L}$ could instead be made black-box, one could hope to use the random oracle itself to construct an oracle language $\mathcal{L}^{\mathcal{O}}$ in place of $\mathcal{L}$.[5] Such an approach, if carried out, would establish the conclusion of Thm. 1 without the additional assumption $\mathsf{UP} \not\subseteq \mathsf{BQP}$, thereby resolving the quantum Bennett–Gill question. However, replacing this non-black-box use with a black-box construction appears challenging. Our investigation of this obstacle leads to the impossibility result below, which provides partial evidence that the non-black-box nature of Thm. 1 seems essential.

Black-Box Impossibilities for Cryptographic Attempts. For lower bounds, an ideal result would be to rule out every construction of a quantum-hard language in $\mathsf{NP}^{\mathcal{O}} \cap \mathsf{coNP}^{\mathcal{O}}$ based solely on a random oracle. However, such a result would seem too ambitious at present. Instead, we consider a cryptographic analogue: can we establish an impossibility result when the quantum random oracle is replaced by a cryptographic primitive capturing a comparable form of unstructured hardness? Such a result would still provide useful insight into the limitations of black-box approaches.

Injective one-way functions are a natural candidate. As discussed earlier, inj-OWFs can be viewed as an unstructured source of hardness. Moreover, the analogous question has been studied in the classical setting: Bitansky, Degwekar, and Vaikuntanathan [BDV17] ruled out fully black-box constructions of hard languages in $\mathsf{NP} \cap \mathsf{coNP}$ from inj-OWFs, even with additional access to iO. We establish a corresponding impossibility result in the quantum setting.

Theorem 2. *There is no fully black-box*[6] *quantum reduction of any (even worst-case) quantum hard problem in* $\mathsf{NP} \cap \mathsf{coNP}$ *to post-quantum iO and post-quantum inj-OWFs. Consequently, there is no such reduction to the following post-quantum cryptographic primitives: public-key encryption, oblivious transfer, deniable encryption, and functional encryption.*

Strong PRHF and a Unified Reduction for QROM NIZK. An important technical ingredient for Thm. 1 is a non-interactive zero-knowledge (NIZK) proof for NP in the quantum random oracle model (QROM), without additional computational assumptions. As discussed in the classical setting [GIK+23], obtaining such a proof system is challenging: standard approaches either rely on structured assumptions beyond random oracles or provide soundness only against polynomial-time provers.[7] Constructing NIZK proofs for NP in the (Q)ROM without additional assumptions remains an important open problem. [GIK+23] bypasses this obstacle using an unbounded honest prover. We obtain in Thm. 3 its post-quantum analogue. However, as explained below, our construction relies on new technical ideas rather than a direct quantization of their techniques.

[4] As will become clear in the following sections, non-black-box use here refers to generating a *non-interactive zero-knowledge proof* for a statement involving the definition of $\mathcal{L}$.

[5] A sufficiently length-expanding restriction of a random oracle is injective and post-quantum one-way with overwhelming probability (see Lem. 41). This yields an oracle language in $\mathsf{UP}^{\mathcal{O}}$ outside $\mathsf{BQP}^{\mathcal{O}}$, rather than requiring the separate assumption $\mathsf{UP} \not\subseteq \mathsf{BQP}$.

[6] Strictly speaking, what we rule out is slightly more general than the traditional definition of fully black-box constructions, due to the non-black-box use of obfuscating circuits with injOWF-gates. This is similar to [BDV17].

[7] A system with soundness only against computationally bounded provers is called a NIZK *argument*, rather than a NIZK proof.

Theorem 3. *There exists an (unbounded-prover) post-quantum NIZK proof for* NP *in the quantum random oracle model.*

Strong δ-Dense PRHFs. The approach underlying Thm. 3 constitutes one of our main technical contributions. [GIK+23]'s results are obtained by introducing a cryptographic primitive called a *δ-dense pseudorandom hash function* (PRHF). Their approach has two steps: (i) constructing the desired NIZK from a PRHF, and (ii) showing that a random oracle satisfies PRHF security. There is, however, a subtle mismatch between these components. The step-(i) reduction from NIZK to PRHF requires a non-uniform PRHFs secure against "function-dependent" adversaries. In contrast, their step-(ii) proof of a random oracle's PRHF properties provides only a uniform guarantee. Consequently, directly substituting the random oracle for the PRHF in the NIZK construction can introduce inefficient "oracle-dependent" preprocessing that render the security reduction inefficient. Therefore, the two steps do not compose directly. To address this issue, [GIK+23] provides a separate security proof for the NIZK construction in the random oracle model, in addition to its analysis in the PRHF setting. Indeed, [GIK+23, Remark 7.6] leaves open whether a random oracle satisfies PRHF security against oracle-dependent adversaries.

Rather than attacking this open question, we take a different approach. Our starting point is a change in perspective: instead of treating the mismatch solely as a limitation of existing techniques for handling oracle-dependent adversaries, we view it as an issue with the interface provided by the PRHF definition. This suggests seeking a strengthened notion that both admits a random-oracle instantiation and supports a NIZK security reduction without the problematic oracle-dependent preprocessing. Following this perspective, we introduce *strong δ-dense PRHFs*. We show that a random oracle is a strong PRHF and that a strong PRHF suffices to construct the desired NIZK through a unified security reduction, independently of how the strong PRHF is instantiated. Moreover, we establish both components in the quantum setting, yielding Thm. 3 and demonstrating the robustness of the strengthened PRHF notion. More details are provided in Sec. 2. We believe that the strong PRHF is a cleaner and more natural primitive that will find further applications in cryptographic constructions.

1.2 Related Work

It is also worth noting that there do exist other candidates for quantum hardness in $\mathsf{NP} \cap \mathsf{coNP}$, beyond the number-theoretic problems mentioned earlier. Parity games, simple stochastic games, and related graph-game problems are important representatives [Con92, CJK+17]. These problems have motivated substantial algorithmic research; notably, parity games admit a quasipolynomial-time algorithm [CJK+17]. Nevertheless, no polynomial-time classical or quantum algorithms are known for these general game problems. Their unresolved complexity supplies candidates for hardness, but a different kind of evidence from the number-theoretic assumptions *traditionally used in cryptography*. In particular, proposing that one of these games is quantum-hard does not explain whether such hardness follows from a generic post-quantum cryptographic assumption. The evidence for quantum hardness in $\mathsf{NP} \cap \mathsf{coNP}$ is consequently narrower than the classical picture might initially suggest.

Another route to quantum hardness in $\mathsf{NP} \cap \mathsf{coNP}$ proceeds through classical derandomization. Under suitable circuit-hardness assumptions, $\mathsf{AM} = \mathsf{NP}$ and hence $\mathsf{AM} \cap \mathsf{coAM} = \mathsf{NP} \cap \mathsf{coNP}$ [MV05]. Since $\mathsf{SZK} \subseteq \mathsf{AM} \cap \mathsf{coAM}$ [AT03], any language in $\mathsf{SZK} \setminus \mathsf{BQP}$, together with such a derandomization assumption, would yield a language in $(\mathsf{NP} \cap \mathsf{coNP}) \setminus \mathsf{BQP}$. Graph non-isomorphism is a natural candidate. Other candidate languages in SZK include the decision versions of tensor isomorphism and code equivalence over finite fields, problems whose search variants have been studied for post-quantum cryptography purpose [GGPS23, DMR11]. Combined with classical derandomization,

these assumptions therefore provide another route to quantum hardness in $\mathsf{NP} \cap \mathsf{coNP}$. This route, however, starts from specific algebraic problems already admitting statistical zero-knowledge proofs and invokes an additional derandomization assumption. Our focus is instead on deriving two-sided classical certification together with quantum hardness from unstructured sources.

2 Technical Overview

Our starting point is the work of [GIK+23]: if $\mathsf{UP} \not\subseteq \mathsf{RP}$, then $\mathsf{NP}^O \cap \mathsf{coNP}^O \not\subseteq \mathsf{P}^O$ with probability 1 over a random oracle O. We first recall the main ingredients of their construction, emphasizing the aspects relevant to extending their approach to the quantum setting.

From a hard promise problem to a hard language. Fix a language $\mathcal{L}_{\mathsf{unique}} \in \mathsf{UP} \setminus \mathsf{RP}$, and let R_{unique} be its polynomial-time witness relation. Thus, every $x \in \mathcal{L}_{\mathsf{unique}}$ has a unique polynomial-length witness w_x, whereas every $x \notin \mathcal{L}_{\mathsf{unique}}$ has no witness. For simplicity, we use a fixed polynomial witness length, padding witnesses uniquely if necessary. Consider the language

$$\mathcal{L}_{\mathsf{almost}} := \big\{(x,i) \,\big|\, \exists w :\ R_{\mathsf{unique}}(x,w) = 1 \wedge w_i = 1\big\}.$$

Clearly, $\mathcal{L}_{\mathsf{almost}} \in \mathsf{NP}$. Moreover, a polynomial-time decider for $\mathcal{L}_{\mathsf{almost}}$ would recover the unique witness bit by bit whenever one exists; checking the recovered witness would then decide $\mathcal{L}_{\mathsf{unique}}$. Hence, $\mathcal{L}_{\mathsf{almost}} \notin \mathsf{P}$.

The difficulty is establishing membership in coNP. For a NO-instance (x,i), there are two cases:

- *Good case:* $x \in \mathcal{L}_{\mathsf{unique}}$. The unique witness w_x certifies nonmembership by satisfying $R_{\mathsf{unique}}(x,w_x) = 1$ and $(w_x)_i = 0$. Uniqueness rules out a different witness whose ith bit is 1.
- *Problematic case:* $x \notin \mathcal{L}_{\mathsf{unique}}$. No witness exists, and there need not be an efficiently verifiable certificate of this fact.

Thus, the witness-bit problem has both YES- and NO-certificates under the promise $x \in \mathcal{L}_{\mathsf{unique}}$, but extending this guarantee to all inputs is nontrivial.

The idea of [GIK+23] is to enforce this promise by appending an efficiently verifiable proof π of $x \in \mathcal{L}_{\mathsf{unique}}$. Given a deterministic polynomial-time verifier V, define

$$\mathcal{L}_{\mathsf{proof},V} := \big\{(x,i,\pi) \,\big|\, (V(x,\pi) = 1) \wedge (\exists w \ s.t.\ R_{\mathsf{unique}}(x,w) = 1 \wedge w_i = 1)\big\}.$$

Here, V is fixed for the language, and proofs have polynomially bounded length. For intuition, suppose that the proof system has perfect completeness and perfect soundness. An NP certificate is simply w_x. To certify nonmembership, either the verifier rejects π, which can be checked directly, or w_x certifies that its ith bit is 0. Soundness ensures that an accepting proof cannot occur in the problematic case. Consequently, $\mathcal{L}_{\mathsf{proof},V} \in \mathsf{NP} \cap \mathsf{coNP}$.

Preserving hardness is subtler. If π simply reveals w_x, then the construction satisfies completeness and soundness but yields an easy language. The proof must therefore enforce the promise without revealing the witness. This motivates using a *non-interactive zero-knowledge* (NIZK) proof: its efficient simulator generates proofs without knowing w_x.

Why zero knowledge preserves hardness. A NIZK construction in the random oracle model is required to have a simulator, which can produce both a simulated proof and simulated oracle responses. Informally, a decider for $\mathcal{L}^O_{\mathsf{proof},V}$, applied to the simulated proof for each index i, recovers the witness bit by bit. Zero knowledge transfers the correctness of this recovery from genuine proofs to simulated ones. Checking the recovered witness gives a one-sided-error algorithm for $\mathcal{L}_{\mathsf{unique}}$: it always rejects NO-instances and accepts YES-instances with high probability.

Making this argument rigorous requires more than the usual simulation argument. The desired conclusion quantifies over random oracles and all polynomial-time oracle machines. The proof in [GIK+23] uses a Bennett–Gill lemma [BG81] to pass from positive-probability decidability to a suitable fixed oracle machine, to which zero knowledge can be applied. (We remark that extending this hardness argument to quantum algorithms is a separate issue from constructing a quantum-secure NIZK; we return to it in Sec. 2.3.)

There is also a soundness qualification. The construction provides *adaptive statistical soundness*: except with negligible probability over the oracle, no false statement of a given length has an accepting proof. Since these failure probabilities are summable, the Borel–Cantelli lemma implies that almost every oracle has only finitely many exceptional statement lengths. For each such oracle, these lengths can be handled separately when establishing membership in $\mathsf{NP}^O \cap \mathsf{coNP}^O$. This is why statistical *proofs*, rather than merely computationally sound arguments, are essential.

NIZK from unstructured assumptions. The remaining challenge is to construct the required NIZK without structured assumptions. Previously known NIZK proofs in the uniform reference string model relied on structured primitives, whereas standard random-oracle constructions such as Fiat–Shamir generally provide only arguments. Importantly, this application does not require an efficient honest prover: it requires only an efficient verifier and simulator. The main technical contribution of [GIK+23] is an unbounded-prover NIZK proof system for NP in the classical random-oracle model.

2.1 Building NIZKs with an Unbounded Prover: the Classical Starting Point

Z-tamperable hidden bits. [GIK+23] proceeds through the *Z-tamperable-hidden-bits* model. The prover receives a hidden string $r = (r_1, \ldots, r_m)$ of independent, possibly biased bits and may reveal selected positions to the verifier. A malicious prover may additionally change any 0 to a 1, but cannot change a 1 to a 0.

To construct NIZK in this model, [GIK+23] adapts the classical hidden-bits construction for graph Hamiltonicity. The hidden bits encode many sparse matrices, some of which contain an n-cycle as a submatrix and zeros elsewhere. For each such matrix, the prover shows that the cycle is contained in a permutation of the input graph, revealing only entries corresponding to nonedges. The remaining matrices are fully revealed and checked not to be of this form.

The new difficulty is that one-sided tampering lets a cheating prover destroy every cycle matrix by adding a 1, thereby making it eligible for disclosure and disposal. The construction counters this attack with a statistical check on the number of revealed matrices containing at least $n + 1$ ones. Destroying cycle matrices increases this count. Moreover, the matrices are sufficiently sparse that almost all matrices already containing more than n ones occupy more than n rows or columns, and hence cannot be hidden inside an $n \times n$ submatrix. Thus, the prover cannot offset the increase by withholding enough other matrices. Concentration bounds then yield statistical soundness.

Implementing the hidden bits. Let $H : \{0,1\}^n \to \{0,1\}^n$ have image density approximately δ, for an efficiently approximable constant $\delta \in (0,1)$. For a uniformly random public value y, define

$$r(y) = \begin{cases} 0, & y \in \mathsf{Image}(H) \\ 1, & y \notin \mathsf{Image}(H) \end{cases}.$$

An unbounded prover can determine $r(y)$ and, when $r(y) = 0$, sample a uniformly random preimage v of y. To reveal a zero, it provides v, which the verifier checks by evaluating $H(v) = y$. To reveal a one, it provides the symbol $\perp$, without proving nonmembership.

This implements precisely the allowed direction of tampering: a prover may withhold an existing preimage and claim a one, but cannot produce a preimage of a value outside the image. Furthermore, independent uniform values $y_1, \ldots, y_m$ induce independent hidden bits with bias approximately $1 - \delta$. Density and one-sided verifiability therefore allow the soundness of the hidden-bits protocol to carry over.

The role of pseudorandomness. Zero knowledge requires additional properties. In [GIK+23, Definition 4.6], a *δ-dense pseudorandom hash function* (PRHF) satisfies, in addition to image density, the following:

– *Output pseudorandomness:* $H(U_n)$ is computationally indistinguishable from U_n.

– *Preimage pseudorandomness:* the following distributions are computationally indistinguishable:

$$(v, H(v)), \quad v \leftarrow \{0,1\}^n, \qquad \text{and} \qquad (v, y), \quad y \leftarrow \mathsf{Image}(H),\ v \leftarrow H^{-1}(y).$$

Both properties are required against non-uniform polynomial-size adversaries. The second property is not automatic from the first: sampling a uniform input weights each output according to its number of preimages, whereas sampling a uniform image element does not.

These properties enable a particularly simple simulator in the uniform reference string (URS) model. First, run the hidden-bits simulator to obtain the revealed bits and the associated proof. Independently sample uniform $v_1, \ldots, v_m$ and set $y_i = H(v_i)$. For each revealed zero, provide v_i; for each revealed one, provide $\perp$. Thus, the simulator knows a preimage for every public value, including those presented as having no preimage.

Why is this indistinguishable from an honest execution? For a revealed zero, preimage pseudorandomness justifies replacing a uniform image element and its uniform preimage with $(H(v_i), v_i)$. For a one, the honest public value is uniform outside the image. Output pseudorandomness and preimage pseudorandomness together imply that this distribution is also indistinguishable from $H(U_n)$: the uniform distribution is a constant-weight mixture of the uniform image and nonimage distributions, and the uniform image distribution is already indistinguishable from $H(U_n)$. After these replacements, the public values are generated independently of the hidden bits, allowing the hidden-bits zero-knowledge property to complete the argument.

PRHF does not directly give a random-oracle NIZK. A length-preserving random function has image density approximately $1 - e^{-1}$ and satisfies the required pseudorandomness properties against polynomial-query adversaries; see [GIK+23, Appendix B]. One might therefore hope simply to instantiate the PRHF with a random oracle. However, the security guarantees needed by the above reduction do not match those established for the oracle.

The PRHF-to-NIZK proof uses non-uniform hybrid reductions to hardwire intermediate values that need not be efficiently computable, including hidden-bit transcripts and samples conditioned on image membership. For a fixed explicit function H, this is permitted by the PRHF definition. If H is replaced by a random oracle, however, the corresponding advice may depend on the sampled oracle itself. The reduction would therefore require preimage pseudorandomness against adversaries equipped with oracle-dependent advice, potentially obtained by unbounded preprocessing of the oracle. [GIK+23] does not establish this stronger guarantee. *The distinction is oracle-dependent advice, not merely whether an adversary is uniform or non-uniform.*

As explained in [GIK+23, Remark 7.6], this mismatch prevents a direct composition of the two results. Instead, the authors prove zero knowledge separately in the classical random-oracle model. Their simulator samples public values, programs preimages for the simulated zeros, and answers subsequent oracle queries by lazy sampling. A direct hybrid argument analyzes the resulting joint distribution without invoking PRHF security with oracle-dependent advice. This proves security

against classical polynomial-query adversaries, but does not by itself establish security for quantum superposition queries. Consequently, both this simulation step and the subsequent hardness argument must be addressed when extending the approach to the quantum setting.

2.2 Strong δ-Dense PRHFs and Their Application to NIZK

Our starting point is a change in perspective: rather than overcoming oracle-dependent non-uniformity by proving security against stronger adversaries, we redesign the interface provided by the PRHF definition. The original reduction introduces oracle-dependent auxiliary information when it replaces individual samples while fixing the remaining transcript. We instead formulate a security experiment that exposes the entire polynomial-size collection of samples from the outset. This leads to *strong δ-dense PRHFs*.

Our approach simultaneously removes the need for oracle-dependent advice and extends the primitive to the post-quantum setting. We establish two complementary results: strong post-quantum PRHFs suffice for the NIZK construction through a uniform, black-box reduction, and a quantum random oracle instantiates this strengthened notion. Consequently, the random-oracle instantiation follows from the same abstract reduction, without a separate random-oracle-specific proof of the zero-knowledge property.

Strong preimage pseudorandomness. Our new notion of strong δ-dense PRHFs retains the density and pseudorandomness requirements: the image of H occupies approximately a δ fraction of its codomain, and $H(v)$ for uniform v is pseudorandom against quantum adversaries with oracle access to H. The principal change is the third requirement, *strong preimage pseudorandomness*.

Let $H : \{0,1\}^n \to \{0,1\}^{\ell(n)}$, let $p(n)$ be a polynomial, and set $\gamma = 1 - \delta$. Consider two distributions that both expose

$$T \coloneqq \left(r = (r_1, \ldots, r_{p(n)}),\ \{y_i\}_{i\in[p(n)]},\ \{v_i\}_{i\in\mathcal{I}_0}\right),$$

where $\mathcal{I}_0 \coloneqq \{i \mid i \in [p(n)] \ s.t.\ r_i = 0\}$. In both distributions, the bits r_i are sampled independently with probability γ of being one. The distributions differ in how they generate the remaining samples:

- **Backward sampling ($\mathcal{D}_0^H$).** If $r_i = 0$, sample y_i uniformly from $\mathsf{Image}(H)$ and then sample v_i uniformly from $\mathsf{PreImage}(y_i)$. If $r_i = 1$, sample y_i uniformly from $\{0,1\}^{\ell(n)} \setminus \mathsf{Image}(H)$.[8] This distribution need not be efficiently sampleable.
- **Forward sampling ($\mathcal{D}_1^H$).** Independently sample $v_i \xleftarrow{\$} \{0,1\}^n$ and set $y_i = H(v_i)$ for every i.

Let $\mathcal{H}_n$ be the family of H with security parameter n. Strong preimage pseudorandomness requires

$$\left|\Pr\left[\mathcal{A}^{|H\rangle}(T) = 1 \ :\ \begin{array}{l} H \leftarrow \mathcal{H}_n, \\ T \leftarrow \mathcal{D}_0^H \end{array}\right] - \Pr\left[\mathcal{A}^{|H\rangle}(T) = 1 \ :\ \begin{array}{l} H \leftarrow \mathcal{H}_n, \\ T \leftarrow \mathcal{D}_1^H \end{array}\right]\right| \leq \mathsf{negl}(n),$$

where $|H\rangle$ indicates that $\mathcal{A}$ has quantum access to H as an oracle.[9] We require security against polynomial-query quantum adversaries $\mathcal{A}$, allowing oracle-independent auxiliary input. Their computation between oracle queries need not be efficient; this distinction is essential because our NIZK application invokes an unbounded prover.

[8] If the set $\{0,1\}^{\ell(n)} \setminus \mathsf{Image}(H)$ is empty, we simply set y_i to a fixed value (e.g., $0^{\ell(n)}$ or $\perp$). We adopt this convention whenever sampling from an empty set. This does not affect any of our results, since the empty-set case occurs with only negligible probability throughout the paper.

[9] Here, we model H as an oracle because it is ultimately instantiated as a random oracle in our main theorem. Alternatively, H may be treated as a plain-model primitive that could potentially be instantiated under algebraic assumptions in future work. In that case, the description of H should be provided to $\mathcal{A}$ in the security definition.

The property replaces the entire collection at once, even though the adversary sees all hidden bits, receives preimages at every zero position, and can query H in superposition. In particular, it accommodates subsequent computations depending on the full hidden-bit string without fixing their outputs as oracle-dependent advice.

From strong PRHFs to NIZK with URS. We retain the construction of [GIK+23], but give a different zero-knowledge reduction. Recall that a Z-tamperable hidden-bits (ZHB) proof allows the prover to reveal selected hidden bits and to change a zero into a one, but not conversely. To implement these bits, parse the uniform reference string as $y_1, \ldots, y_{p_{\mathsf{hbs}}(n)}$ and define

$$r_i = \begin{cases} 0 & \text{if } y_i \in \mathsf{Image}(H), \\ 1 & \text{otherwise.} \end{cases}$$

The unbounded prover computes r and runs $\mathsf{ZHB.P}_\gamma(r, x)$, obtaining an index set I and a ZHB proof. It opens a zero by providing a uniformly sampled preimage of y_i, and opens a one by providing $\perp$. The verifier checks the supplied preimages and invokes the ZHB verifier on the resulting bits. A prover can always replace a preimage by $\perp$, but cannot provide a preimage of a point outside the image. This realizes exactly the permitted zero-to-one tampering. Completeness and soundness then follow from the corresponding ZHB properties and the density guarantee.

For zero knowledge, first rewrite the real experiment by sampling the hidden bits before the public values. Conditioned on H, the probability that $r_i = 1$ is $1 - |\mathsf{Image}(H)|/2^{\ell(n)}$. The density guarantee allows us to replace this probability by $1 - \delta$ at negligible statistical cost. The resulting samples are distributed as $\mathcal{D}_0$ (i.e., the **Backward sampling** defined above).

We now apply strong preimage pseudorandomness *once*, to replace $\mathcal{D}_0$ by $\mathcal{D}_1$. The reduction receives the complete challenge transcript, runs $\mathsf{ZHB.P}_\gamma(r, x)$, retains the openings selected by its output, and passes the resulting URS transcript to the distinguisher. Although the ZHB prover may be computationally unbounded, it makes no queries to H; hence the reduction remains polynomial-query-bounded. Unlike the coordinate-by-coordinate argument of [GIK+23], this reduction neither fixes the ZHB proof nor supplies the remaining samples as oracle-dependent advice.

After the replacement, all pairs $(v_i, H(v_i))$ are generated independently of r. The final transcript therefore depends on the ZHB execution only through its revealed bits and proof. Statistical zero knowledge of the ZHB system now allows us to replace these by the output of $\mathsf{ZHB.Sim}_\gamma(x)$. This gives an efficient simulator: generate the simulated ZHB transcript, sample all pairs forward, and reveal the preimages selected by its zero bits.

The reduction is genuinely black-box in H, independently of its implementation. In Sec. 2.4, we establish that a random oracle is a strong $(1 - e^{-1})$-dense post-quantum PRHF, which represents our another technical contribution. Together, these results yield a URS-model NIZK relative to a quantum random oracle.

Eliminating the URS. To obtain the eventual QROM-only NIZK, we need to remove the URS. Our strategy is to derive the URS from an independent oracle O_{URS}, while using O_{PRHF} for the PRHF. The simulator must answer superposition queries consistently with the simulated URS and its correlations with remaining entries of the oracle(s); and classical lazy sampling does not suffice in the quantum setting. We formalize a *compressed partially-fixed oracle*, $\mathrm{Com}(H)$, which perfectly emulates a random oracle conditioned on a prescribed partial function H. It stores the fixed entries explicitly and compresses the remaining entries using the compressed-oracle technique [Zha19]. Our simulator runs the URS simulator using a persistent compressed PRHF oracle, then initializes the URS oracle conditioned on the resulting URS. Retaining both oracle states preserves

the required correlations throughout subsequent quantum interaction, with polynomial overhead. This step implements the simulated experiment efficiently.

Combining the oracles. Finally, we combine the two independent oracles into one by domain separation: $O_{\mathsf{URS}}(z) = O(0\|z)$ and $O_{\mathsf{PRHF}}(z) = O(1\|z)$. A quantum adversary may query a superposition of the two prefixes, so separate oracle access does not immediately give the required combined interface. We simulate each combined query exactly using four separate queries: compute each oracle's answer into an ancilla, conditionally XOR it into the answer register, and uncompute. This works even when the prefix is entangled with the adversary's workspace. Thus, the two-oracle security argument transfers to a single quantum random oracle, completing the construction.

2.3 Completing the Separation

With our QROM NIZK in hand, we complete the separation by adapting the argument of [GIK+23]. We first instantiate the verifier in $\mathcal{L}^{\mathcal{O}}_{\mathsf{proof},V}$ with our NIZK verifier. The certificate argument remains classical: adaptive statistical soundness, together with the Borel–Cantelli lemma, ensures that $\mathcal{L}^{\mathcal{O}}_{\mathsf{proof},V} \in \mathsf{NP}^{\mathcal{O}} \cap \mathsf{coNP}^{\mathcal{O}}$ for almost every oracle. The remaining task is to rule out quantum polynomial-time deciders. Suppose a fixed quantum oracle machine decides this language with bounded error on every input for at least a 2/3 fraction of oracles. Given an accepting proof for $x \in \mathcal{L}_{\mathsf{unique}}$, we can use this machine to recover the unique witness bit by bit. Replacing the genuine proof and oracle access by our efficient NIZK simulator preserves each bit's recovery probability by quantum zero knowledge. Repeating with independent simulator instances and taking majority votes therefore recovers the entire witness with high probability. Checking the recovered witness ensures that NO-instances are always rejected, yielding an oracle-free BQP algorithm for $\mathcal{L}_{\mathsf{unique}}$. This contradicts $\mathcal{L}_{\mathsf{unique}} \notin \mathsf{BQP}$.

To pass from this high-probability bound to an almost-sure separation, we give a quantum formulation of [BG81, Lemma 1]. Its structural requirements hold here: bounded witness enumeration gives a *total*[10] oracle decider, and the verifier's input-length-separated oracle access ensures that changing finitely many oracle bits affects only finitely many language instances. The standard finite-prefix hardwiring and finite-input correction argument from [BG81] thus applies to quantum machines as well, showing that each fixed machine decides the language only on a measure-zero set of oracles. A countable union over machines completes the proof that, assuming $\mathsf{UP} \not\subseteq \mathsf{BQP}$, we have $\mathsf{NP}^{\mathcal{O}} \cap \mathsf{coNP}^{\mathcal{O}} \not\subseteq \mathsf{BQP}^{\mathcal{O}}$ with probability 1 over $\mathcal{O}$.

2.4 A Random Oracle is a Strong PRHF

We show that a uniformly random function $H : \{0,1\}^n \to \{0,1\}^n$ is a strong post-quantum δ-dense PRHF for $\delta = 1 - e^{-1}$. The δ-density property follows from [GIK+23], while image pseudorandomness against quantum queries follows from a standard one-way-to-hiding (O2H) argument [AHU19]. We focus on the new strong preimage pseudorandomness (recall the definition on Page 8): for every polynomial $p(n)$, backward sampling $\mathcal{D}_0$ and forward sampling $\mathcal{D}_1$ produce indistinguishable collections

$$T \coloneqq \left(r = (r_1, \ldots, r_{p(n)}),\ \{y_i\}_{i\in[p(n)]},\ \{v_i\}_{i\in\mathcal{I}_0}\right).$$

even when the distinguisher makes polynomially-many *quantum* queries to H.

Recall that backward sampling chooses y_i uniformly from $\mathsf{Image}(H)$ and then a uniform preimage v_i when $r_i = 0$, and chooses y_i outside the image when $r_i = 1$. Forward sampling instead chooses a uniform v_i and sets $y_i = H(v_i)$ at every position, revealing v_i only when $r_i = 0$. We fix r throughout

[10] An oracle machine is total if it halts on every input for every oracle. No polynomial-time bound is required.

the overview, and let $\mathcal{I}_0$ and $\mathcal{I}_1$ denote the sets of indices where $r_i = 0$ and $r_i = 1$, respectively; (note that $\mathcal{I}_0 \cup \mathcal{I}_1 = [p(n)]$). All bounds are uniform over its choice and therefore also hold after averaging over r. We suppress negligible collision and exceptional-size events, which our formal proof in Sec. 4.2 handles explicitly.

The difficulty is that T and the oracle available to the adversary depend on the same random function. Replacing one sample while fixing the others risks introducing oracle-dependent advice. We instead transform the entire collection and the queried oracle jointly. This is not a direct application of quantum tools, e.g., the O2H lemma [AHU19]. Although such tools can bound the effect of changing an oracle on a set of inputs, its usefulness depends on bounding the adversary's probability of querying that set. Here, the information in T is correlated with the locations we wish to change. Non-trivial ideas and efforts are required to bound the query-hitting probability.

We build a sequence of hybrids, where we gradually switch from the backward-sampling distribution to the forward-sampling distribution. In the following, we provide a high-level overview of the main steps in the proof, while deferring the technical details to Sec. 4.2.

Generating outputs outside the image by modifying the oracle. We start from the backward-sampling distribution $\mathcal{D}_0$. We first change how the outputs at positions $i \in \mathcal{I}_1$ are generated. Rather than choosing y_i outside the image, we sample $v_i \xleftarrow{\$} \{0,1\}^n$, set $y_i = H(v_i)$, and keep v_i hidden. Then, we modify every entry of the oracle H whose value is one of these selected outputs, replacing its value with fresh randomness outside the selected outputs. Consequently, these y_i's are absent from the image of the modified oracle, as required by backward sampling. To distinguish it from the orignial oracle H, we denote this modified oracle as H^*; also note that H^* is now the oracle supplied to the adversary.

To see why this modification is safe, notice that the number of outputs missing from a random function's image is sharply concentrated. Thus, choosing a random function and then uniform missing outputs is statistically close to choosing the outputs first and then a random function avoiding them. Meanwhile, polynomially many forward evaluations produce almost uniform distinct outputs. Conditional on the locations being modified, the other entries already avoid those outputs; resampling the modified entries yields the corresponding distribution of an oracle avoiding them.

We must also account for the (v_i, y_i) pairs for each $i \in \mathcal{I}_0$: these pairs were sampled backward from the original oracle, but the adversary now receives the modified oracle. Their sampling distribution need not coincide with backward sampling from that modified oracle, since the modification can change both its image and its preimage sets. We show that this discrepancy has negligible statistical cost. Only a small number of image elements and preimage sets are affected, and a uniformly selected image element encounters these changes with negligible probability. This establishes closeness of the joint distributions of T and the modified oracle, rather than merely closeness of the selected outputs. After this step, every y_i has a designated input v_i in the original oracle, but those with $i \in \mathcal{I}_1$ remain hidden.

Canceling the global bias introduced by backward sampling. Next, for all x satysfying $H(x) \notin \{y_i\}_{i\in[p(n)]}$, we modify the value of $H^*(x)$ to an independent uniform value outside $\{y_i\}_{i\in[p(n)]}$. That is, we resample entries of the constructed oracle H^*, but determine their locations using the original oracle H. A tempting argument is that, after fixing the selected outputs and all their preimage sets under H, the remaining entries of H are independent and uniform outside those outputs. Unfortunately, backward sampling prevents this argument from being correct.

Indeed, conditional on the original oracle being $H = h$, the probability of sampling any specified pairs $\{(y_i, v_i)\}_{i\in\mathcal{I}_0}$ consistent with h is

$$|\mathsf{Image}(h)|^{-|\mathcal{I}_0|} \cdot \prod_{i\in\mathcal{I}_0} \frac{1}{|\mathsf{PreImage}^h(y_i)|}.$$

Here the image and preimage sets are those of the original oracle $H = h$, before any modification. Now, consider the distribution of the remaining entries of h after we fix the selected outputs and their complete preimage sets under h: the preimage-size factors $\prod_{i\in\mathcal{I}_0} \frac{1}{|\mathsf{PreImage}^h(y_i)|}$ are fixed, but the factor $|\mathsf{Image}(h)|^{-|\mathcal{I}_0|}$ can still vary across completions of the remaining oracle entries. These completions therefore need not have equal conditional probabilities: backward sampling introduces a global bias into the unexposed part of the original table.

We cancel this bias by temporarily changing the distribution of the original oracle. We introduce a new oracle $\widetilde{H}$ that plays the same role as H, but have the following probability density (and all subsequent sampling and oracle-construction steps remain unchanged, using $\widetilde{H}$ in place of H):

$$\Pr\Big[\widetilde{H} = h\Big] = \frac{|\mathsf{Image}(h)|^{|\mathcal{I}_0|}}{\sum_{f:\{0,1\}^n\to\{0,1\}^n} |\mathsf{Image}(f)|^{|\mathcal{I}_0|}}.$$

The additional weight cancels the image-size factor from backward sampling exactly. Once the selected outputs, their designated inputs, and all entries mapping to them are fixed, the factors arising from choosing uniform preimages are also fixed. The remaining entries are now independent uniform values outside the selected outputs, so they can be replaced by fresh randomness.

We show that this change of distribution has negligible statistical cost. The image size is sharply concentrated around $(1 - e^{-1})2^n$, and its polynomial power varies by only a negligible relative amount on typical tables. Notice that conditioned on

$$\{(v_i, y_i)\}_{i\in[p(n)]} \quad \text{and} \quad \mathsf{PreImage}^{\widetilde{H}}(\{y_i\}_{i\in[p(n)]}),$$

the entries $\widetilde{H}(x)$ at inputs $x \notin \mathsf{PreImage}^{\widetilde{H}}(\{y_i\}_{i\in[p(n)]})$ are then independent and uniform over $\{0,1\}^n \setminus \{y_i : i \in [p(n)]\}$. At these inputs, the constructed oracle H^* still agrees with $\widetilde{H}$. Replacing these entries of H^* by fresh independent values from the same distribution therefore leaves the joint distribution of T and H^* unchanged. We can then return to the original uniform oracle H at negligible statistical cost. That is, the $\widetilde{H}$ is only an intermediate tool helping us to analyze the statistical closeness. In the remainder of the overview, H again denotes this uniformly sampled original oracle, while H^* denotes the constructed oracle supplied to the adversary.

Making the additional preimage locations hidden. Having refreshed the entries at inputs x satisfying $H(x) \notin \{y_i : i \in [p(n)]\}$, we next consider the preimages of $y_1, \ldots, y_{p(n)}$ under H. For each i, the input v_i is one preimage of y_i, but y_i may also have other preimages. We now modify H^* at all such preimages except $v_1, \ldots, v_{p(n)}$; that is, at inputs in

$$S = \mathsf{PreImage}^H(\{y_i\}_{i\in[p(n)]}) \setminus \{v_1, \ldots, v_{p(n)}\}. \tag{1}$$

Previously, for an input x in this set with $H(x) = y_i$, $H^*(x)$ was y_i for $i \in \mathcal{I}_0$, and an independent uniform value in $\{0,1\}^n \setminus \{y_j : j \in \mathcal{I}_1\}$ for $i \in \mathcal{I}_1$. We now replace $H^*(x)$ in both cases by an independent uniform value in $\{0,1\}^n \setminus \{y_j : j \in [p(n)]\}$. We leave the entries at $v_1, \ldots, v_{p(n)}$ unchanged: $H^*(v_i) = y_i$ for $i \in \mathcal{I}_0$, while $H^*(v_i)$ remains a previously sampled uniform value in $\{0,1\}^n \setminus \{y_j : j \in \mathcal{I}_1\}$ for $i \in \mathcal{I}_1$.

Thus, the versions of H^* before and after this modification can differ only at inputs in the set S defined in Eq. (1). We plan to use O2H to argue that the adversary cannot distinguish these two versions. To do so, we need to bound the probability that a query hits S.

Let $\mathcal{B}$ be the *good* event that (i) the selected outputs $\{y_i\}_{i\in[p(n)]}$ are distinct, (ii) every output of H has at most n preimages, and (iii) $|\mathsf{Image}(H)| \geq 2^{n-1}$. It follows from standard collision bounds and random-function concentration bounds that $\Pr[\mathcal{B}] = 1 - \mathsf{negl}(n)$. Conditioned on event $\mathcal{B}$, it hold that $|S| \leq p(n)n$. But this size bound alone is insufficient for O2H: a small set can still be easy to query if the adversary's input reveals its location. We instead prove that S is hidden in the particular experiment used to bound the query-hitting probability.

Consider the new experiment with the modified H^*. For analysis, condition on the good event $\mathcal{B}$ and all pairs $\{(v_i, y_i)\}_{i\in[p(n)]}$, including the inputs with $i \in \mathcal{I}_1$ that are not revealed to the adversary. Once these pairs are fixed, the H^* depends only on fresh randomness, whereas S depends on the original H. The adversary's interaction with H^* is therefore conditionally independent of S.

Moreover, the original sampling procedure is invariant under permutations of the domain fixing all designated inputs $\{v_i\}_{i\in[p(n)]}$. Conditional on the designated pairs, the good event $\mathcal{B}$, and $|S| = s$, the set S is consequently a uniform s-element subset of the remaining domain. Hence a measured query in this experiment hits S with probability at most

$$\Pr[\text{measured query lies in } S] \leq \frac{p(n)n}{2^n - p(n)}.$$

O2H now bounds the distinguishing advantage of a q-query adversary by

$$2q\sqrt{\frac{p(n)n}{2^n - p(n)}} + \mathsf{negl}(n) = \mathsf{negl}(n).$$

The order of these arguments is essential. We first change the joint distribution to obtain conditional independence, then use symmetry to bound the query-hitting probability, and only then apply O2H. We do not infer that a hidden-set hitting probability is preserved merely because two distributions of the adversary's visible information are close. Also, conditioning on the designated pairs is only an analytical tool: no reduction receives them as oracle-dependent advice.

Identifying backward and forward sampling after resampling. The constructed oracle H^* now depends only on the designated pairs and fresh randomness. It no longer exposes information about additional preimages in the original table.

At this point, we change the sampling only for $i \in \mathcal{I}_0$: instead of choosing y_i uniformly from $\mathsf{Image}(H)$ and then a uniform preimage v_i, we choose v_i uniformly from $\{0,1\}^n$ and set $y_i = H(v_i)$. The positions with $i \in \mathcal{I}_1$ already use forward sampling and remain unchanged. Symmetry shows that these two ways of generating the designated pairs become statistically interchangeable after the preceding oracle modifications. Conditional on distinct designated inputs and distinct selected outputs, permutations of the domain and range act transitively on the possible tuples $\{(v_i, y_i)\}_{i\in[p(n)]}$. Both the mixed sampling procedure before this change and the all-forward procedure after it are invariant under these permutations; thus, they induce the same conditional distribution on the tuples. Given such a tuple, the constructed oracle H^* supplied to the adversary is generated by the same rule in both experiments. Consequently, the two experiments have the same conditional joint distribution of T and H^*. The excluded collision events contribute only negligible statistical error.

This argument does not identify the original joint distributions of T and H. Rather, it establishes statistical closeness of the joint distributions of T and H^* after the preceding resampling steps have removed the dependence of H^* on the original preimage sets beyond the information specified by $\{(v_i, y_i)\}_{i\in[p(n)]}$.

Restoring the original random oracle. Now that every pair is generated by forward sampling, we can undo the remaining oracle modifications using quantum collision-finding and preimage-search bounds, together with O2H.

First, restore the entries at the undisclosed inputs mapping to selected outputs. Querying such an input produces a second input with the same original output as a designated input, and hence a collision in the original random function. The reduction generates the designated pairs with polynomially many queries and simulates the relevant intermediate oracle by explicitly reprogramming the designated inputs with $r_i = 1$. Quantum collision-finding bounds make the resulting query-hitting probability negligible.

Finally, restore the entries at those hidden designated inputs. Hitting one of them entails finding a preimage of one of the forward-sampled outputs with $r_i = 1$. A quantum multi-target preimage-search bound applies to this distribution. The pairs revealed at positions $r_i = 0$ are generated by additional oracle queries within the same reduction, rather than supplied as oracle-dependent advice. O2H then shows that restoring these last entries is indistinguishable. The resulting experiment is exactly $\mathcal{D}_1$ with access to the unmodified random oracle.

We remark that the main work of this proof is not the final invocation of O2H or of quantum collision and preimage bounds. It is the distributional preparation that makes these quantum tools useful: canceling the global bias of image sampling, refreshing the unexposed oracle entries, and proving that the remaining disagreement locations are hidden. This allows the proof to handle the entire collection jointly, with bounds depending only on the number of oracle queries and not on the adversary's intervening computation.

2.5 Quantum Black-Box Impossibility via a Universal Decision Oracle

We construct an oracle world in which post-quantum injOWFs and iO exist, but every $\mathsf{NP} \cap \mathsf{coNP}$ problem produced by a fully black-box construction is easy. Besides the *source oracle*, which implements the two primitives, this world contains a *decision oracle*: an additional interface that solves the decision problems arising from such constructions. Both the adversary and the security reduction may query this interface, including in superposition. The challenge is therefore to make it powerful enough to defeat every proposed construction of $\mathsf{NP} \cap \mathsf{coNP}$ languages without enabling quantum attacks on the source primitives, i.e., injOWFs and iO.

A powerful decision oracle with short certificates. An $\mathsf{NP} \cap \mathsf{coNP}$ problem has two verifiers: one certifies membership, and the other certifies nonmembership. For each input, exactly one verifier has an accepting witness. Following [BDV17], our oracle Decide takes such a verifier pair and an input, and returns the side with an accepting witness.

There is an essential restriction on the pairs it accepts. Call a source implementation *admissible* if it satisfies the primitives' functional requirements, such as injectivity and correct evaluation of obfuscated programs; it need not satisfy their security requirements. A verifier pair is valid only if, for every input and *every admissible source implementation*, exactly one side has an accepting witness. We call this property *universal complementarity*. Invalid pairs receive $\perp$. The universality matches the functional guarantee of a fully black-box construction: its output must define an $\mathsf{NP} \cap \mathsf{coNP}$ problem for every functionally correct implementation of its sources. Thus, Decide solves the constructed problem, but need not solve arbitrary verifier pairs that happen to be complementary for the particular sampled oracle.

This restriction also protects the sources. Fix a valid decision query and one source oracle, called the *reference oracle*, against which we compare later modifications. If the verifiers make at most r source queries, an accepting witness under the reference oracle uses at most r oracle answers. Any modification preserving those answers preserves acceptance. Universal complementarity

prevents the opposite side from also accepting, so the decision bit cannot change. Each decision answer therefore has a small *critical set*, even though finding its witness may require unbounded computation. This stability property is inherited from [BDV17].

From local stability to quantum security. Quantum queries may place amplitude on many inputs simultaneously, and their amplitudes depend on earlier oracle interactions. Consequently, an unconditional bound on the probability that a fixed answer changes is insufficient. To resolve this difficulty, we establish a conditional-coupling lemma, which is similar to the quantum hybrid from [Unr14, ARU14]. For coupled oracles (F_1, F_2), suppose that

$$\Pr[F_1(u) \neq F_2(u) \mid F_2] \leq \gamma \qquad \text{for every reference realization } F_2 \text{ and input } u.$$

Then, for a T-query quantum algorithm whose initial state is independent of the sampled oracle pair, the expected trace distance between its final states under F_1 and F_2 is at most $2T\sqrt{\gamma}$. Conditioning on the entire reference oracle accommodates the dependence of its query states on that oracle. We apply this bound to the *combined source and decision interfaces*; critical sets control disagreement at decision queries.

As a warm-up, let $f : \{0,1\}^n \to \{0,1\}^{n+1}$ be a random injection. Sample y uniformly outside its image and plant y at an independent uniform input x, obtaining g. The joint distribution of (g, x, y) is exactly that obtained by sampling a uniform injection g and an independent uniform input x, and setting $y = g(x)$. In contrast, under the reference function f, the same challenge has no preimage. Conditioned on f and y, a fixed decision answer changes only if x hits its critical set, with probability at most $r/2^n$ for verifiers making at most r source queries. Checking an inverter's output with one additional query therefore gives

$$\Pr[\text{invert}] \leq 2(T+1)\sqrt{r/2^n}.$$

Thus, the unbounded decision interface does not destroy one-wayness, even when queried coherently.

Obfuscation with adaptive quantum side information. Following [BDV17], we implement obfuscation by an injective encoding table mapping a circuit and randomness (C, r) to a handle $\widehat{C}$. An evaluation oracle executes the circuit encoded by that handle; the obfuscated program is a wrapper calling this interface. We must also specify evaluation on handles outside the table's image. A *backup map* φ assigns a circuit description $\varphi(\widehat{C})$ to each (possibly malformed) handle $\widehat{C}$. If $\widehat{C}$ is outside the image, the evaluation oracle answers a query $(\widehat{C}, z)$ according to $\varphi(\widehat{C})$; otherwise, it evaluates the circuit encoded by the table. The admissible oracle family allows every such backup map. This freedom lets us move a challenge handle out of the encoding table while preserving its visible functionality.

The principal difficulty is adaptive challenge selection: the adversary chooses C_0, C_1 after oracle access and retains a quantum state correlated with the encoding table. Reprogramming only the eventual challenge entry does not establish the required independence. We use the row-reprogramming architecture of [BDV17], analyzing the full joint distribution of the oracle, challenge, and residual quantum state.

First, we sample the uniform tape $r \in \{0,1\}^n$ that will be used to generate the challenge handle. For the pre-challenge execution only, we replace the encoding table on the entire row $\{(C, r) : C\}$ by an ordered tuple D of distinct fresh handles outside the original table's image. The challenge and post-challenge oracle still use the original table. The coupling hides this modification: each fixed source query changes with probability $O(2^{-n})$, and critical sets give an $O(q2^{-n})$ bound for decision queries whose verifiers make at most q source queries. Conditioned on r, D, and the modified pre-challenge oracle, the adversary's classical outputs and residual quantum state are independent of

the erased original row. We can therefore sample the challenge handle $\widehat{C}$ uniformly outside the original table's image and D, and reprogram the post-challenge table to map (C_b, r) to $\widehat{C}$. This replacement preserves *exactly* the joint distribution of the challenge bit, pre-challenge output state, challenge handle, and post-challenge oracle.

Next, we restore only the original pre-challenge oracle, retaining both the fresh challenge handle and the single-entry post-challenge reprogramming. Closeness of averaged states would not suffice here, because the subsequent computation depends on the reprogramming randomness. Instead, our coupling bound controls the *expected trace distance of paired executions*. More precisely, for each fixed original oracle, let $\rho_{r,D}$ and ρ_{ref} be the complete pre-challenge output states under the modified and original oracles, respectively. The bound controls $\mathbb{E}_{r,D}[\mathrm{TD}(\rho_{r,D}, \rho_{\mathrm{ref}})]$, rather than merely the distance between the averaged states. For each fixed (r, D, b), challenge generation and the post-challenge execution form the same channel in the two experiments, so contractivity applies before averaging. With the original pre-challenge oracle restored, the pre-challenge output state is independent of r, and the selected circuits again satisfy $C_0^f \equiv C_1^f$.

Finally, we undo the single-entry post-challenge reprogramming, restoring the original table value at (C_b, r), and set $\varphi(\widehat{C}) = C_0$. The fresh handle $\widehat{C}$ is now outside the table's image, so it is evaluated through the backup map. Evaluation at this publicly known handle is unchanged because $C_b^f \equiv C_0^f$. The source interfaces can differ only at the encoding query (C_b, r) and at evaluation queries using its original handle. These locations are hidden by the uniform tape r, allowing another application of critical sets and the coupling bound. The final experiment is independent of the challenge bit. Throughout these comparisons, validity is tested against the same family containing all backup maps, so every modification remains admissible and the stability argument continues to apply.

Preserving one-wayness and concluding. To preserve one-wayness after adding obfuscation, we fix the encoding and backup tables and coherently simulate their interfaces using the injOWF-only oracle. One subtlety is that fixing these tables can make a previously invalid verifier pair complementary. The simulation must therefore retain the *original* universal-validity test before translating decision queries. This predicate is independent of the sampled function, so its finite truth table can be hardwired in the query-bounded simulation.

Our bounds restrict evaluation lengths and verifier resources in addition to external query counts. Every efficient construction and reduction meets these restrictions for some polynomial bound. The two source primitives thus remain quantum-secure, whereas one decision query solves the constructed language exactly. A fully black-box security reduction would turn this solver into an attack on a source primitive, giving the contradiction.

3 Preliminaries

Notation. Let PPT denotes for a probabilistic polynomial time, and QPT denotes for quantum polynomial time. We say that a function $f(n)$ is *negligible* in n if $f(n) = n^{-\omega(1)}$, and we denote this by $f(n) = \mathsf{negl}(n)$. We say that a function $g(n)$ is *polynomial* in n if $g(n) = p(n)$ for some fixed polynomial p, and we denote this by $g(n) = \mathsf{poly}(n)$.

When we consider a non-uniform QPT adversary, we often specify it by a sequence of polynomial-size quantum circuits with quantum advice $\{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$. In an execution with the security parameter λ, $\mathcal{A}$ runs $\mathcal{A}_\lambda$ taking ρ_λ as the advice. For simplicity, we often omit the index λ and just write $\mathcal{A}_\rho$ to mean a non-uniform QPT algorithm specified by $\{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$.

If R is a random variable (or distribution), then $r \leftarrow R$ denotes sampling r according to R. If T is a finite set, then $i \xleftarrow{\$} T$ denotes sampling i uniformly at random from T. Throughout this work,

we take the convention that once we need to sample from an empty set, the output is defined to be some special symbol (e.g., $\bot$ or 0^n).

For $t \in \mathbb{N}$, we use $[t]$ to denote the set $[t] := \{1, \ldots, t\}$.

For $x \in \{0,1\}^*$, we write $|x|$ for its bit length. For $i \in \mathbb{N}$, we write $\langle i \rangle$ for the standard binary representation of i without leading zeros. For $i \in \{0, \ldots, 2^\ell - 1\}$,we write $\langle i \rangle_\ell$ denotes its ℓ-bit binary representation, padded with leading zeros when necessary.

We collect more basic preliminaries in Sec. A.

3.1 Random Oracles

Let $\mathcal{O}$ denote the distribution over functions $O : \{0,1\}^* \to \{0,1\}$ obtained by independently sampling $O(x) \xleftarrow{\$} \{0,1\}$ for every $x \in \{0,1\}^*$. We refer to $O \leftarrow \mathcal{O}$ as a *random oracle*. Thus, $\mathcal{O}$ denotes the distribution of random oracles, whereas O denotes a particular oracle sampled from this distribution.

As in [GIK+23], a single random oracle $O : \{0,1\}^* \to \{0,1\}$ can be viewed as simultaneously providing independent random functions of arbitrary polynomially bounded input and output lengths. The following lemma justifies this interpretation by setting $m(n) = n$.

Lemma 1 ([GIK+23, Lemma 3.7]). *For any function $m : \mathbb{N} \to \mathbb{N}$, a random oracle $O \leftarrow \mathcal{O}$ can be interpreted as a countably infinite family of mutually independent random functions*

$$\{O_{n,m} : \{0,1\}^n \to \{0,1\}^{m(n)}\}_{n \in \mathbb{N}}.$$

In particular, for every $n \in \mathbb{N}$, we write $\mathcal{O}_n$ as the uniform distribution over all functions from $\{0,1\}^n$ to $\{0,1\}^n$, where each instance is denoted by O_n.

Throughout this paper, we will often need to sample oracles with proper input and output lengths as $O_{n,m} \leftarrow \mathcal{O}_{n,m}$. When the input and output lengths are clear from the context, we will simply write $O \leftarrow \mathcal{O}$, omitting the subscripts n and m.

Oracle access. For an oracle algorithm $\mathcal{A}$, we write $\mathcal{A}^{|O\rangle}$ to indicate that $\mathcal{A}$ is given quantum oracle access to O. The quantum access to $O_{n,m}$ is implemented through the standard unitary

$$U_{O_{n,m}} : |x\rangle\,|y\rangle \longmapsto |x\rangle\,|y \oplus O_{n,m}(x)\rangle\,,$$

for $x \in \{0,1\}^n$ and $y \in \{0,1\}^{m(n)}$. We refer to one application of $U_{O_{n,m}}$ as one *quantum query* to $O_{n,m}$. Classical oracle access corresponds to querying this unitary on computational-basis states.

When we write

$$\Pr_{O \leftarrow \mathcal{O}}\left[b = 1 : b \leftarrow \mathcal{A}^{|O\rangle}(x)\right],$$

the probability is taken over the sampling of $O \leftarrow \mathcal{O}$ as well as all internal randomness and measurement outcomes of $\mathcal{A}$.

3.2 Useful Lemmas

Lemma 2 ([YZ21, Corollary 4.13]). *For any $n, m, q \in \mathbb{N}$, any fixed $y \in \{0,1\}^n$, and any quantum algorithm $\mathcal{A}$ making at most q quantum queries to a random oracle $O_n : \{0,1\}^m \to \{0,1\}^n$, we have*

$$\Pr_{O \leftarrow \mathcal{O}}\left[O_n(x) = y \;:\; x \leftarrow \mathcal{A}^{|O\rangle}(y)\right] \leq \frac{(2q+1)^2}{2^n}.$$

Taking the probability over $y \xleftarrow{\$} \{0,1\}^n$ in the above lemma, we have the following corollary.

Corollary 1. *For any $n, m, q \in \mathbb{N}$ and any quantum algorithm $\mathcal{A}$ making at most q quantum queries to a random oracle $O_n : \{0,1\}^m \to \{0,1\}^n$, we have*

$$\Pr_{O \leftarrow \mathcal{O}, y \xleftarrow{\$} \{0,1\}^n} \left[O_n(x) = y \ : x \leftarrow \mathcal{A}^{|O\rangle}(y) \right] \leq \frac{(2q+1)^2}{2^n}.$$

Lemma 3 (Preimage Resistance). *For any $n, m, q \in \mathbb{N}$ and any quantum algorithm $\mathcal{A}$ making at most q quantum queries to a random oracle $O_n : \{0,1\}^n \to \{0,1\}^n$, we have*

$$\Pr_{O \leftarrow \mathcal{O}, x' \xleftarrow{\$} \{0,1\}^n} \left[O_n(x) = O_n(x') \ : x \leftarrow \mathcal{A}^{|O\rangle}(O_n(x')) \right] \leq \frac{2n(2q+1)^2 + 1}{2^n}.$$

Proof. We first show that, over the random choice of $O \leftarrow \mathcal{O}$, the following event occurs with probability at most 2^{-n}:

– E: $\max_{y \in \{0,1\}^n} \left| \mathsf{PreImage}_{O_n}(y) \right| > 2n$

Then,

$$\begin{aligned}
\Pr_{O \leftarrow \mathcal{O}}[E] &\leq \sum_{y \in \{0,1\}^n} \sum_{\substack{S \subseteq \{0,1\}^n \\ |S| = 2n+1}} \Pr_{O \leftarrow \mathcal{O}}[O_n(z) = y \text{ for every } z \in S] && \text{(Union Bound)} \\
&= \sum_{y \in \{0,1\}^n} \sum_{\substack{S \subseteq \{0,1\}^n \\ |S| = 2n+1}} \prod_{z \in S} \frac{1}{2^n} \\
&= 2^n \binom{2^n}{2n+1} 2^{-n(2n+1)} \\
&\leq 2^n \frac{(2^n)^{2n+1}}{(2n+1)!} 2^{-n(2n+1)} = \frac{2^n}{(2n+1)!} \\
&= \frac{2^n}{\prod_{i=1}^{n} (2i)(2i+1)} \leq \frac{2^n}{4^n} = \frac{1}{2^n}.
\end{aligned}$$

We then observe that, for each fixed oracle O and each $y \in \{0,1\}^n$,

$$\Pr_{x' \xleftarrow{\$} \{0,1\}^n} \left[O_n(x') = y \right] = \frac{|\{z \in \{0,1\}^n : O_n(z) = y\}|}{2^n}. \tag{2}$$

We can now bound the success probability using Eq. (2):

$$\begin{aligned}
&\Pr_{O \leftarrow \mathcal{O},\ x' \xleftarrow{\$} \{0,1\}^n} \left[O_n(x) = O_n(x') \ : \ x \leftarrow \mathcal{A}^{|O\rangle}(O_n(x')) \right] \\
&= \mathbb{E}_{O \leftarrow \mathcal{O}} \left[\sum_{y \in \{0,1\}^n} \Pr_{x' \xleftarrow{\$} \{0,1\}^n} \left[O_n(x') = y \right] \Pr \left[O_n(x) = y \ : \ x \leftarrow \mathcal{A}^{|O\rangle}(y) \right] \right] \\
&\leq \frac{2n}{2^n} \mathbb{E}_{O \leftarrow \mathcal{O}} \left[\sum_{y \in \{0,1\}^n} \Pr \left[O_n(x) = y \ : \ x \leftarrow \mathcal{A}^{|O\rangle}(y) \right] \right] + \Pr_{O \leftarrow \mathcal{O}}[E] \quad \text{(Eq. (2)+ conditional on } \neg E)
\end{aligned}$$

$$
\begin{aligned}
&= 2n \Pr_{O\leftarrow\mathcal{O},\, y\xleftarrow{\$}\{0,1\}^n}\left[O_n(x)=y \;:\; x\leftarrow\mathcal{A}^{|O\rangle}(y)\right] + \Pr_{O\leftarrow\mathcal{O}}[E] \\
&\le \frac{2n(2q+1)^2}{2^n} + \Pr_{O\leftarrow\mathcal{O}}[E] && \text{(Corollary 1)} \\
&\le \frac{2n(2q+1)^2+1}{2^n}. && (\Pr_{O\leftarrow\mathcal{O}}[E]\le 2^{-n})
\end{aligned}
$$

This completes the proof of Lem. 3. □

Corollary 2 (Multi-Target Preimage Resistance). *For any $n, m, k, q \in \mathbb{N}$ and any quantum algorithm $\mathcal{A}$ making at most q quantum queries to a random oracle $O_n : \{0,1\}^n \to \{0,1\}^n$, we have*

$$
\Pr\left[\exists i\in[k],\; O_n(x)=O_n(x_i) \;:\; \begin{array}{l} O\leftarrow\mathcal{O},\quad x_1,\ldots,x_k\xleftarrow{\$}\{0,1\}^n, \\ x\leftarrow\mathcal{A}^{|O\rangle}(O_n(x_1),\ldots,O_n(x_k)) \end{array}\right] \le k\cdot\frac{2n\big(2(q+k)-1\big)^2+1}{2^n}.
$$

Proof. The main idea is to reduce the multi-target preimage search problem to a single-target preimage search problem. Assume that there exist a quantum algorithm $\mathcal{A}$ that can find a preimage of one of the k targets with high probability ,i.e..

$$
\Pr\left[\exists i\in[k],\; O_n(x)=O_n(x_i) \;:\; \begin{array}{l} O\leftarrow\mathcal{O},\quad x_1,\ldots,x_k\xleftarrow{\$}\{0,1\}^n, \\ x\leftarrow\mathcal{A}^{|O\rangle}(O_n(x_1),\ldots,O_n(x_k)) \end{array}\right] > k\cdot\frac{2n\big(2(q+k)-1\big)^2+1}{2^n}.
$$

We can construct a new quantum algorithm $\mathcal{B}$ making at most $q+k-1$ quantum queries to the oracle O_n that breaks Lem. 3 as follows:

Algorithm 3.1: Reduction $\mathcal{B}$

$\mathcal{B}(y^*)$

1. Sample index $j\xleftarrow{\$}[k]$.
2. Set $y_j = y^*$, sample $x_i\xleftarrow{\$}\{0,1\}^n$ and set $y_i = O_n(x_i)$ for all $i\in[k]\setminus\{j\}$.
3. Output $\mathcal{A}^{|O\rangle}(y_1,\ldots,y_k)$

Define the index set $\mathcal{I}(x) = \{i : O_n(x) = y_i\}$ which corresponds to the set of indices of the targets that x is a preimage of. Notice that.

$$
\Pr[\mathcal{A}\text{ succeeds}] \le \Pr[\mathcal{I}(x)\neq\emptyset]
$$

We have that since j is sampled uniformly at random from $[k]$ and $\mathcal{B}$ succeeds if and only if $j\in\mathcal{I}(x)$, we have

$$
\begin{aligned}
\Pr[\mathcal{B}\text{ succeeds}] &\ge \Pr[j\in\mathcal{I}(x)|\mathcal{I}(x)\neq\emptyset]\Pr[\mathcal{I}(x)\neq\emptyset] \\
&\ge \frac{1}{k}\cdot\Pr[\mathcal{A}\text{ succeeds}] \\
&> \frac{2n\big(2(q+k-1)+1\big)^2+1}{2^n} \\
&> \frac{2n\big(2(q+k)-1\big)^2+1}{2^n},
\end{aligned}
$$

which contradicts Lem. 3.

□

Lemma 4 ([Zha15, Theorem 3.1]). *For any $n, m, q \in \mathbb{N}$ and any quantum algorithm $\mathcal{A}$ making at most q quantum queries to a random oracle $O_n : \{0,1\}^m \to \{0,1\}^n$, we have*

$$\Pr_{O \leftarrow \mathcal{O}}\left[O_n(x) = O_n(x') \ \wedge \ x \neq x' : (x, x') \leftarrow \mathcal{A}^{|O\rangle}\right] \leq O(\frac{q^3}{2^n}).$$

The following lemma, usually referred to as the one-way-to-hiding lemma, serves as an important tool in the proof of Lem. 10.

Lemma 5 ([AHU19, Theorem 3]). *Let S be a random subset of X. Let $G, H : X \to Y$ be arbitrary random functions satisfying $G(x) = H(x)$ for all $x \notin S$. Let z be an arbitrary, possibly random string. (S, G, H, and z may have arbitrary joint distribution.)*

Let $\mathcal{A}$ be a quantum oracle algorithm making at most q quantum queries to its oracle. Define $\mathcal{B}^H$ to be the oracle algorithm that, on input z, proceeds as follows:

1. *sample $i \xleftarrow{\$} [q]$;*
2. *run $\mathcal{A}^H(z)$ until immediately before its i-th query;*
3. *measure all query-input registers in the computational basis and output the set T of measurement outcomes.*

Define

$$P_{\text{left}} := \Pr\left[b = 1 : b \leftarrow \mathcal{A}^H(z)\right],$$

$$P_{\text{right}} := \Pr\left[b = 1 : b \leftarrow \mathcal{A}^G(z)\right],$$

and

$$P_{\text{guess}} := \Pr\left[S \cap T \neq \varnothing : T \leftarrow \mathcal{B}^H(z)\right].$$

Here, the probabilities are taken over the joint distribution of (S, G, H, z), the internal randomness and measurement outcomes of the corresponding algorithms.

Then, for any joint distribution of G, H, S, z, it holds that

$$|P_{\text{left}} - P_{\text{right}}| \leq 2q\sqrt{P_{\text{guess}}}.$$

3.3 Simulating Two Oracles using One

In this subsection, we recall that joint quantum queries to two oracles can be simulated exactly using separate access, with constant query overhead and linear additional gate cost per query. Consequently, security with respect to two independent quantum-accessible random oracles is preserved under the following implementation

$$O_0(x) = O(0\|x), \qquad O_1(x) = O(1\|x),$$

where O is a single quantum-accessible random oracle.

Separate and joint oracle access. Let $O_0, O_1 : \{0,1\}^n \to \{0,1\}^n$ be functions. Separate quantum access to these functions provides the query unitaries

$$U_{O_b} : |x\rangle\,|y\rangle \longmapsto |x\rangle\,|y \oplus O_b(x)\rangle \qquad (\forall b \in \{0,1\}).$$

Each oracle call applies one of these two unitaries, while the algorithm may perform arbitrary quantum operations between calls. In particular, its registers may be arbitrarily entangled.

Joint access additionally allows the oracle index to be queried in superposition through the unitary

$$U_{O_0,O_1} : |b\rangle\ |x\rangle\ |y\rangle \longmapsto |b\rangle\ |x\rangle\ |y \oplus O_b(x)\rangle\,.$$

The following lemma shows that this joint interface can be implemented using separate access.

Lemma 6 (Simulation of the Joint Oracle). *Given separate access to O_0 and O_1, the unitary U_{O_0,O_1} can be implemented exactly using two queries to each oracle and $O(n)$ additional gates. The simulation uses an n-qubit ancilla initialized and returned to $|0^n\rangle$.*

Proof. Introduce an n-qubit ancilla register t, initialized to $|0^n\rangle$. On a computational-basis input $|b\rangle\ |x\rangle\ |y\rangle\ |0^n\rangle$, perform the following steps.

1. Apply U_{O_0} to (x,t) to compute $O_0(x)$ into t, obtaining

$$|b\rangle\ |x\rangle\ |y\rangle\ |O_0(x)\rangle\,.$$

2. XOR t into y conditioned on $b=0$, obtaining

$$|b\rangle\ |x\rangle\ |y \oplus (1-b)O_0(x)\rangle\ |O_0(x)\rangle\,.$$

3. Apply U_{O_0} again to (x,t) to uncompute t. Since U_{O_0} is an XOR-query unitary, the resulting state is

$$|b\rangle\ |x\rangle\ |y \oplus (1-b)O_0(x)\rangle\ |0^n\rangle\,.$$

4. Repeat the above procedure using O_1, with the XOR conditioned on $b=1$. This gives

$$|b\rangle\ |x\rangle\ |y \oplus (1-b)O_0(x) \oplus bO_1(x)\rangle\ |0^n\rangle\,.$$

   Since

$$y \oplus (1-b)O_0(x) \oplus bO_1(x) = y \oplus O_b(x),$$

the procedure implements U_{O_0,O_1} and returns the ancilla to $|0^n\rangle$. By linearity, the simulation is exact for arbitrary inputs, including those entangled with other registers.

The procedure makes two queries to each oracle. Each controlled-XOR operation requires $O(n)$ gates, yielding the claimed gate complexity. □

3.4 Non-Interactive Zero-Knowledge Proof

We begin by recalling the definitions necessary for formalizing NIZK proof systems in our setting.

Definition 1 (Polynomial-Query-Bounded Quantum Oracle Algorithm). *We say that a quantum oracle algorithm $\mathcal{A}^{(\cdot)}$ is* polynomial-query-bounded *if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that, for every input $x \in \{0,1\}^*$ and every oracle H, the execution of $\mathcal{A}^{|H\rangle}$ makes at most $p(|x|)$ quantum queries to H.*

Definition 2 (Efficiently-Approximable Constant). *We say that a constant (real number) $\gamma \in (0,1)$ is* Efficiently-Approximable *if there exists a polynomial-time deterministic algorithm Approx_γ such that for all $n \in \mathbb{N}$,*

- *$\mathsf{Approx}_\gamma(1^n)$ outputs a rational number $\widehat{\gamma} = a/b$, where a, b are integers in binary representation.*
- *For all $n \in \mathbb{N}$, $|\gamma - \mathsf{Approx}_\gamma(1^n)| \leq \frac{1}{2^n}$.*

Then, we recall the definition of NIZK proof systems in the Z-Tamperable-Hidden-Bits model, which is a variant of the standard NIZK proof system that allows for tampering with hidden bits.

Definition 3 (NIZK Proof in the ZHB Model [GIK+23, Definition 4.2]). *An (unbounded-prover) non-interactive zero-knowledge (NIZK) proof system in the* Z-Tamperable-Hidden-Bits(ZHB) *model for a language L is a family of algorithms $\{\Pi_\gamma = (\mathsf{Gen}_\gamma, \mathsf{P}_\gamma, \mathsf{V}_\gamma)\}$ parameterized by* Efficiently-Approximable *constants $\gamma \in (0,1)$, where:*

- $\mathsf{Gen}_\gamma(1^n)$ *is a PPT hidden bit string generator that given an input length n outputs a hidden bit string $r \in \{0,1\}^{p_{\gamma,\mathsf{hbs}}(n)}$ for some (fixed) polynomial $p_{\gamma,\mathsf{hbs}}$ where each bit r_i of r is sampled independently from a Bernoulli distribution with $\Pr[r_i = 1] = \widehat{\gamma}$ where $\widehat{\gamma} = \mathsf{Approx}_\gamma(1^n)$.*
- $\mathsf{P}_\gamma(r, x)$ *is an (unbounded, randomized) honest prover algorithm that takes as input a hidden bit string $r \in \{0,1\}^{p_{\gamma,\mathsf{hbs}}(|x|)}$ and $x \in L$, and outputs a set of indices $I \subseteq [p_{\gamma,\mathsf{hbs}}(|x|)]$ and a proof π.*
- $\mathsf{V}_\gamma(x, I, \{r_i\}_{i\in I}, \pi)$ *is a polynomial-time deterministic verifier that takes as input x, an index set $I \subseteq [p_{\gamma,\mathsf{hbs}}(|x|)]$, the bits $\{r_i\}_{i\in I}$ of the hidden bit string r corresponding to I, and a proof π, and outputs either 1 (accept) or 0 (reject).*

We require Π to satisfy the following completeness, soundness, and zero knowledge requirements:

- ***Perfect Completeness:*** *For all Efficiently-Approximable constants $\gamma \in (0,1)$ and all $x \in L$,*

$$\Pr\left[\mathsf{V}_\gamma(x, I, \{r_i\}_{i\in I}, \pi) = 1 : r \leftarrow \mathsf{Gen}_\gamma(1^{|x|}), (I, \pi) \leftarrow \mathsf{P}_\gamma(r, x)\right] = 1$$

  where the probability is over the randomness of Gen_γ and P_γ.
- ***(Adaptive) Statistical Soundness:*** *For all Efficiently-Approximable constants $\gamma \in (0,1)$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that,*

$$\Pr_{r\leftarrow\mathsf{Gen}_\gamma(1^n)}[\exists(x, I, f, \pi) \text{ such that } x \notin L, |x| = n, \mathsf{V}_\gamma(x, I, \{\widetilde{r_i} = \mathsf{ZeroFlip}(r_i, f_i)\}_{i\in I}, \pi) = 1] \leq \mathsf{negl}(|x|)$$

  where $\mathsf{ZeroFlip}(r_i, f_i) = (r_i \oplus f_i$ if $r_i = 0$; r_i if $r_i = 1)$.
- ***Statistical Zero Knowledge:*** *For all efficiently-approximable constants $\gamma \in (0,1)$ (as per Def. 2), there exists a PPT simulator Sim_γ , for all adversaries A, there exists a negligible function $\mathsf{negl}(\cdot)$ such that all $x \in L$,*

$$\begin{aligned}&|\Pr\left[A(x, I, \{r_i\}_{i\in I}, \pi) = 1 : r \leftarrow \mathsf{Gen}_\gamma(1^{|x|}), (I, \pi) \leftarrow \mathsf{P}_\gamma(r, x)\right]\\ &- \Pr[A(x, I^*, \{r_i^*\}_{i\in I^*}, \pi^*) = 1 : (I^*, \{r_i^*\}_{i\in I^*}, \pi^*) \leftarrow \mathsf{Sim}_\gamma(x)]| \leq \mathsf{negl}(|x|)\end{aligned}$$

  where the probability is over the randomness of Gen_γ, P_γ, Sim_γ, and A.

Definition 4 (NIZK Proof in the QROM). *An (unbounded-prover) non-interactive zero knowledge (NIZK) proof system in the quantum random oracle model (QROM) for a language $\mathcal{L}$ is a tuple of algorithms $\Pi = (P^{(\cdot)}, V^{(\cdot)})$ with access to a random oracle $\mathcal{O}$.*

- *$P^O(x)$ is an (unbounded, randomized) honest prover algorithm that is given oracle access to a random oracle $\mathcal{O}$, takes as input $x \in \mathcal{L}$, and outputs a proof π.*
- *$V^O(x, \pi)$ is a polynomial-time deterministic verifier that is given oracle access to a random oracle $\mathcal{O}$, takes as input x and a proof π, and outputs either 1 (accept) or 0 (reject).*

We require Π to satisfy the following completeness, soundness, and zero knowledge requirements:

- ***Perfect Completeness:*** *For all $x \in \mathcal{L}$, it holds that*

$$\Pr_{O \leftarrow \mathcal{O}}\left[V^O(x,\pi) = 1 : \pi \leftarrow P^O(x)\right] = 1.$$

- ***(Adaptive) Statistical Soundness:*** *It holds that*

$$\Pr_{O \leftarrow \mathcal{O}}[\exists (x,\pi) \text{ such that } x \notin \mathcal{L}, |x| = n, \text{ and } V^O(x,\pi) = 1] \leq \mathsf{negl}(|x|).$$

- ***Zero Knowledge:*** *There exists a stateful QPT simulator* $\mathrm{Sim} = (\mathrm{SimProof}, \mathrm{SimRO})$ *such that for all $x \in \mathcal{L}$ andall polynomial-query-bounded quantum oracle algorithm $\mathcal{A}^{(\cdot)}$ (Def. 1),*

$$\Big|\Pr\left[\mathcal{A}^{|O\rangle}(x,\pi) = 1 : \pi \leftarrow P^O(x)\right] - \Pr\left[\mathcal{A}^{|\mathrm{SimRO}(\mathrm{st},\cdot)\rangle}(x,\pi^*) = 1 : (\mathrm{st},\pi^*) \leftarrow \mathrm{SimProof}(x)\right]\Big| \leq \mathsf{negl}(|x|).$$

3.5 Compressed Partially-Fixed Oracle

We introduced compressed partially-fixed oracles to simulate a random oracle conditioned on a prescribed classical table. This formulation adapts the compressed-oracle technique of [Zha19].

Our zero-knowledge simulator initializes the oracle with a prescribed classical table and retains its private state throughout subsequent quantum interaction. We therefore spell out this fixed-table specialization and its efficient sparse implementation.

We first formalize the conditional oracle distribution.

Definition 5 (Conditional Distribution of a Random Oracle). *Let $n, m \in \mathbb{N}$ be parameters. A subset $Q \subseteq \{0,1\}^n \times \{0,1\}^m$ is a* partial function *if $\forall (x,y),(x,y') \in Q$, it holds that $y = y'$. We write*

$$\mathrm{dom}(Q) := \{x : \exists y \text{ such that } (x,y) \in Q\}$$

and write $Q(x) = y$ when $(x,y) \in Q$.

Given a partial function Q, the conditional distribution of random oracle*, denoted as $O|_Q$, is the uniform distribution over all functions $F : \{0,1\}^n \to \{0,1\}^m$ consistent with Q. Equivalently,*

$$O|_Q(x) := \begin{cases} Q(x) & x \in \mathrm{dom}(Q), \\ y \xleftarrow{\$} \{0,1\}^m & x \in \{0,1\}^n \setminus \mathrm{dom}(Q) \end{cases}.$$

In the above definition, the unspecified values are sampled independently once and then retained, so repeated queries to the same input return the same value. We next describe an efficient stateful simulation of quantum access to this distribution.

For a stateful simulator $\mathcal{C} \leftarrow \mathrm{Com}(H)$, access to $|\mathcal{C}\rangle$ means coherent access to its query procedure, with its private state retained between queries. We also write $|\mathrm{Com}(H)\rangle$ when initialization is implicit; all queries then use the same initialized simulator. Thus, $\mathcal{A}^{|\mathrm{Com}(H)\rangle}$ denotes $\mathcal{A}$ interacting coherently with a single initialized simulator whose private state is retained and updated between queries.

The following lemma shows that quantum access to $\mathcal{O}|_H$ can be simulated exactly by a stateful oracle, with time and space polynomial in $n, m, |H|$, and the number of queries.

Lemma 7 (Compressed Partially-Fixed Oracle). *Let $n, m \in \mathbb{N}$ be parameters. For a given partial function $H \subseteq \{0,1\}^n \times \{0,1\}^m$, there exists a (stateful) oracle* $\mathsf{Com}(\mathsf{H})$ *called the* compressed partially-fixed oracle *satisfying the following properties:*

– **Perfect Indistinguishability:** *For any (potentially unbounded) quantum algorithm* $\mathcal{A}$*,*

$$\left|\Pr\left[\mathcal{A}^{|O|_H\rangle}=1\right]-\Pr\left[\mathcal{A}^{|\mathsf{Com}(H)\rangle}=1\right]\right|=0,$$

where the first probability includes sampling $O\leftarrow\mathcal{O}|_H$*, and both include the internal randomness and measurement outcomes of the interacting algorithms.*

– **Efficient constructability:** *Initializing* $\mathrm{Com}(H)$ *and answering* q *queries require time and space polynomial in* $n, m, |H|$*, and* q*.*

Proof. We derive the lemma from Zhandry's compressed-oracle simulation [Zha19]. Let $S=\mathrm{dom}(H)$, and let $F:\{0,1\}^n\to\{0,1\}^m$ be a uniformly random function. Define

$$G_{H,F}(x)=\begin{cases}H(x), & x\in S,\\ F(x), & x\notin S.\end{cases}$$

Then $G_{H,F}$ is distributed uniformly among the functions extending H.

Under the standard XOR-query convention, a quantum query to $G_{H,F}$ can be implemented using two queries to F. Specifically, first compute $F(x)$ into a fresh scratch register. Then XOR $H(x)$ into the response register when $x\in S$, and XOR the scratch register into the response register otherwise. Finally, use a second query to F to uncompute the scratch register. All computations involving H are performed coherently, with H hardwired. The resulting transformation is exactly

$$|x,z,0^m\rangle\longmapsto|x,z\oplus G_{H,F}(x),0^m\rangle\,.$$

Define $\mathsf{Com}(H)$ by replacing access to F in this implementation with access to Zhandry's ordinary compressed oracle. For any quantum algorithm $\mathcal{A}$, its composition with the above implementation is itself a quantum oracle algorithm. Zhandry's perfect simulation guarantee therefore implies that replacing the uniformly random oracle F by its compressed implementation preserves $\mathcal{A}$'s acceptance probability exactly. Since $G_{H,F}$ is distributed as $\mathcal{O}|_H$, the claimed equality follows. □

4 Strong δ-Dense PRHF: Definition and Construction

4.1 Strong δ-Dense PRHF

Our definition of the strong δ-dense PRHF considers polynomially many correlated samples and only requires strong pre-image pseudorandomness against non-unifrom QPT quantum adversaries.

In the zero-knowledge proof of [GIK+23], single-instance pre-image pseudorandomness is lifted to polynomially many samples through a coordinate-by-coordinate hybrid argument. The reduction for each adjacent pair non-uniformly fixes the hidden-bit string, the corresponding ZHB proof, and all samples outside the distinguished coordinate. Because this auxiliary information may depend on H, instantiating H with a random oracle would require security against adversaries whose advice depends on the sampled oracle.

Formally, we introduce our notion of a Strong PRHF as follows.

Definition 6 (Strong δ-Dense PRHF). *Let* $\delta\in(0,1)$ *be a* Efficiently-Approximable *constant as per Def. 2 and* $n\in\mathbb{N}$*. Let* $\ell:\mathbb{N}\to\mathbb{N}$ *be a polynomial, and let* $\{\mathcal{H}_n\}_{n\in\mathbb{N}}$ *be an ensemble of distributions over functions* $H:\{0,1\}^n\to\{0,1\}^{\ell(n)}$*. We call this ensemble a strong* δ*-dense post-quantum PRHF if the following conditions hold:*

Box 1: Distributions D_0 and D_1

Distribution $\mathcal{D}_0(n, H_n, p(n), \mathbf{r})$:

1. According to $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, define the index sets:

$$\mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\}$$
$$\mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \quad (= [p(n)] \setminus \mathcal{I}_0)\,.$$

2. Define preimage and image sets:

$$\mathsf{Image} := \{y \in \{0,1\}^{\ell(n)} \mid \exists z \in \{0,1\}^n : H(z) = y\}$$
$$\mathsf{PreImage}(y) := \{z \in \{0,1\}^n \mid H(z) = y\}.$$

3. For all $i \in [p(n)]$:
   (a) If $r_i = 0$, then sample $y_i \xleftarrow{\$} \mathsf{Image}$ and $v_i \xleftarrow{\$} \mathsf{PreImage}(y_i)$;
   (b) If $r_i = 1$, then sample $y_i \xleftarrow{\$} \{0,1\}^{\ell}(n) \setminus \mathsf{Image}$.

4. Output:

$$T = \left(\{r_i\}_{i\in[p(n)]},\ \{y_i\}_{i\in[p(n)]},\ \{v_i\}_{i\in\mathcal{I}_0}\right).$$

Distribution $\mathcal{D}_1(n, H_n, p(n), \mathbf{r})$:

1. According to $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, define the index sets:

$$\mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\}$$
$$\mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \quad (= [p(n)] \setminus \mathcal{I}_0)\,.$$

2. For all $i \in [p(n)]$, sample $v_i \xleftarrow{\$} \{0,1\}^n$ and define $y_i := H(v_i)$.

3. Output:

$$T = \left(\{r_i\}_{i\in[p(n)]},\ \{y_i\}_{i\in[p(n)]},\ \{v_i\}_{i\in\mathcal{I}_0}\right).$$

1. *(δ-Density) For any function $H_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}$, let p_{H_n} denote the probability that a random $y \xleftarrow{\$} \{0,1\}^{\ell(n)}$ is in the range of H_n. It holds that*

$$\Pr\left[p_{H_n} \in [\delta - \mathsf{negl}(n), \delta + \mathsf{negl}(n)] \ :\ H_n \leftarrow \mathcal{H}_n\right] \geq 1 - \mathsf{negl}(n),$$

2. *(Pseudorandomness) For all non-uniform QPT adversary $\mathcal{A}$, it holds that*

$$\left|\Pr\left[\mathcal{A}(1^n, H_n, H_n(x)) = 1 \ :\ \begin{array}{c} x \xleftarrow{\$} \{0,1\}^n, \\ H_n \leftarrow \mathcal{H}_n \end{array}\right] - \Pr\left[\mathcal{A}(1^n, H_n, y) = 1 \ :\ \begin{array}{c} y \xleftarrow{\$} \{0,1\}^{\ell(n)}, \\ H_n \leftarrow \mathcal{H}_n \end{array}\right]\right| \leq \mathsf{negl}(n).$$

3. *(Strong Pre-Image Pseudorandomness) For any non-uniform QPT adversary $\mathcal{A}$, any polynomial $p(n)$, any bit-vector $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, it holds that*

$$\left|\Pr\left[\mathcal{A}(1^n, H_n, T) = 1 \ :\ \begin{array}{l} H_n \leftarrow \mathcal{H}_n; \\ T \leftarrow \mathcal{D}_0 \end{array}\right] - \Pr\left[\mathcal{A}(1^n, H_n, T) = 1 \ :\ \begin{array}{l} H_n \leftarrow \mathcal{H}_n; \\ T \leftarrow \mathcal{D}_1 \end{array}\right]\right| \leq \mathsf{negl}(n),$$

where $\mathcal{D}_0$ and $\mathcal{D}_1$ are the $\mathcal{D}_0(n, H_n, p(n), \mathbf{r})$ and $\mathcal{D}_1(n, H_n, p(n), \mathbf{r})$, respectively, defined in Box 1.

4.2 Random Oracle is a Strong $(1 - \frac{1}{e})$-Dense PRHF

In this subsection, we show that a random oracle is a strong $(1 - \frac{1}{e})$-dense PRHF as per Def. 6.

Remark 1 (Random Oracles as δ-dense PRHFs). We first need to clarify an important modeling subtlety: in Def. 6, the QPT adversary $\mathcal{A}$ receives the full description of H as input. However, this is impossible when we talk about an oracle construction, since the QPT adversary cannot receive the full description of a random oracle. Thus, when we say "a random oracle is a strong $(1 - \frac{1}{e})$-dense PRHF," it actually means a variant of as per Def. 6 where the QPT adversary is modeled as a QPT oracle machine and receives *quantum* oracle access to H (which we will instantiate as a random oracle) instead of the full description of H. We take this modeling convention throughout the paper, whenever we instantiate the strong δ-dense PRHF with a random oracle.

The first two requirements of our strengthened δ-dense PRHF notion—δ-density and post-quantum pseudorandomness—follow from established analyses of random oracles in [GIK+23] and [HXY19], respectively. For completeness, we list them as Lem. 8 and Lem. 9 below.

Lemma 8 ([GIK+23, Lemma B.3]). *Fixed any $n \in \mathbb{N}$, let $\mathcal{O}_n : \{0,1\}^n \to \{0,1\}^n$ be a random oracle. There exists negligible function $\mathsf{negl}(\cdot)$ such that*

$$\Pr_{O_n \leftarrow \mathcal{O}_n}\left[p_{O_n} \in \left[\left(1-\frac{1}{e}\right) - \mathsf{negl}(n), \left(1-\frac{1}{e}\right) + \mathsf{negl}(n)\right]\right] \geq 1 - \mathsf{negl}(n),$$

where

$$p_{O_n} \coloneqq \Pr_{y \xleftarrow{\$} \{0,1\}^n}\left[\exists x \in \{0,1\}^n : \ O_n(x) = y\right].$$

Lemma 9 ([HXY19, Theorem 2]). *For all QPT oracle-aided adversary $\mathcal{A}$, it holds that*

$$\left|\Pr\left[\mathcal{A}^{|O_n\rangle}(1^n, O_n(x)) = 1 \ : \ \begin{array}{l} x \xleftarrow{\$} \{0,1\}^n, \\ O_n \leftarrow \mathcal{O}_n \end{array}\right] - \Pr\left[\mathcal{A}^{|O_n\rangle}(1^n, y) = 1 \ : \ \begin{array}{l} y \xleftarrow{\$} \{0,1\}^n, \\ O_n \leftarrow \mathcal{O}_n \end{array}\right]\right| \leq \mathsf{negl}(n).$$

Explanation of Lem. 9. The result of [HXY19, Theorem 2] is actually more general: it considers a salted random oracle $O\colon [K] \times [N] \to [M]$ for some $K, N, M \in \mathbb{N}$ and permits the adversary to receive an S-bit classical advice string that may depend on O. To obtain the lemma above, set $K = 1$, $N = M = 2^n$, use no auxiliary advice, and let $T = q(n)$ be the adversary's polynomial query bound. The resulting distinguishing advantage is at most

$$\widetilde{O}\left(\left(\frac{q(n)^4}{2^n}\right)^{1/6}\right) = \widetilde{O}\left(\frac{q(n)^{2/3}}{2^{n/6}}\right),$$

which is negligible in n. Identifying $[2^n]$ with $\{0,1\}^n$ through binary representation and omitting the trivial salt coordinate yields precisely the statement above.

Strong Pre-Image Pseudo-randomness. To establish strong pre-image pseudo-randomness, we depart from the proof of [GIK+23]. We state the lemma below and prove it in Sec. 4.3.

Lemma 10. *For any non-uniform QPT oracle-aided adversary $\mathcal{A}$, any polynomial $p(n)$, and any bit-vector $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, it holds that*

$$\left|\Pr\left[\mathcal{A}^{|O_n\rangle}(1^n, T) = 1 \ : \ \begin{array}{l} O_n \leftarrow \mathcal{O}_n; \\ T \leftarrow \mathcal{D}_0 \end{array}\right] - \Pr\left[\mathcal{A}^{|O_n\rangle}(1^n, T) = 1 \ : \ \begin{array}{l} O_n \leftarrow \mathcal{O}_n; \\ T \leftarrow \mathcal{D}_1 \end{array}\right]\right| \leq \mathsf{negl}(n),$$

where $\mathcal{D}_0$ and $\mathcal{D}_1$ are the $\mathcal{D}_0(n, O_n, p(n), \mathbf{r})$ and $\mathcal{D}_1(n, O_n, p(n), \mathbf{r})$, respectively, defined in Box 1.

4.3 Proof of Lem. 10

We proceed to prove Lem. 10. Our proof actually works for any (potentially unbounded) $\mathcal{A}$ as long as she makes only a polynomial number of quantum oracle queries.

Notation. We will prove the lemma via a sequence of hybrids $H_0, H_1, \ldots, H_6$. Each hybrid H_i depends on the same parameters including the security parameter n, the oracle distribution $\mathcal{O}_n$, the polynomial $p(n)$, the bit-vector $\mathbf{r} = (r_1, \ldots, r_{p(n)})$. For succinctness, we suppress them from the notation and simply write H_i, instead of $H_i(n, \mathcal{O}_n, p(n), \mathbf{r})$.

Throughout the proof, we use the convention that once we need to sample from an empty set, the output is defined to be some special symbol (e.g., $\perp$ or 0^n).

It is straightforward from the description that the output of H_0 (Expr. 1) is identically distributed to that of $\mathcal{A}^{|O_n\rangle}(1^n, T)$ when $O_n \leftarrow \mathcal{O}_n$ and $T \leftarrow \mathcal{D}_0(n, O_n, p(n), \mathbf{r})$. Similarly, the output of H_6 (Expr. 8) is identically distributed to that of $\mathcal{A}^{|O_n\rangle}(1^n, T)$ when $O_n \leftarrow \mathcal{O}_n$ and $T \leftarrow \mathcal{D}_1(n, O_n, p(n), \mathbf{r})$. Therefore, our goal is to show that the output distributions of H_1 and H_6 are negligibly close.

For two hybrid experiments H_i and H_j, we write $H_i \overset{\text{i.d.}}{=\!=} H_j$ if $\Pr[H_i = 1] = \Pr[H_j = 1]$ for every adversary $\mathcal{A}$, $H_i \overset{\text{s}}{\approx} H_j$ if $|\Pr[H_i = 1] - \Pr[H_j = 1]|$ is negligible for all possibly inefficient adversaries $\mathcal{A}$, and $H_i \overset{\text{cq}}{\approx} H_j$ if $|\Pr[H_i = 1] - \Pr[H_j = 1]|$ is negligible for every quantum adversary $\mathcal{A}$ making polynomially many oracle queries.

4.3.1 Hybrids H_0 and H_1

Hybrid H_0: This is the game corresponding to $\mathcal{D}_0$. We formally describe it in Expr. 1.

Experiment 1: Hybrid H_0

1. According to the input $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, define $\begin{cases} \mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \end{cases}$.
2. Sample $O_n \leftarrow \mathcal{O}_n$;
3. Define $\begin{cases} \mathsf{Image}^{O_n} := \{y \in \{0,1\}^n : \exists z \in \{0,1\}^n \text{ such that } O_n(z) = y\} \\ \mathsf{PreImage}^{O_n}(y) := \{z \in \{0,1\}^n : O_n(z) = y\} \end{cases}$
4. $\forall i \in [p(n)]$, sample $\{y_i\}_{i\in[p(n)]}$ and $\{v_i\}_{i\in\mathcal{I}_0}$ as follows:
   - If $r_i = 0$, then set $y_i \overset{\$}{\leftarrow} \mathsf{Image}^{O_n}, v_i \overset{\$}{\leftarrow} \mathsf{PreImage}^{O_n}(y_i)$,
   - If $r_i = 1$, then set $y_i \overset{\$}{\leftarrow} \{0,1\}^n \setminus \mathsf{Image}^{O_n}$.
5. Invoke the adversary $\mathcal{A}^{|O_n\rangle}(\{r_i\}_{i\in[p(n)]}, \{y_i\}_{i\in[p(n)]}, \{v_i\}_{i\in\mathcal{I}_0})$

Hybrid H_1: We present H_1 in Expr. 2 and highlight its difference from H_0 in red color. In this hybrid, we change the sampling method for y_i when $r_i = 1$ and redirect the points to form a new oracle $\mathcal{O}_n^*$.

Experiment 2: Hybrid H_1

1. According to the input $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, define $\begin{cases} \mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \end{cases}$.
2. Sample $O_n \leftarrow \mathcal{O}_n$;
3. Define $\begin{cases} \mathsf{Image}^{O_n} := \{y \in \{0,1\}^n : \exists z \in \{0,1\}^n \text{ such that } O_n(z) = y\} \\ \mathsf{PreImage}^{O_n}(y) := \{z \in \{0,1\}^n : O_n(z) = y\} \end{cases}$
4. Define $\begin{cases} \mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \end{cases}$
5. $\forall i \in [p(n)]$, sample $\{y_i\}_{i\in[p(n)]}$ and $\{v_i\}_{i\in\mathcal{I}_0}$ as follows:

– If $r_i = 0$, then set $y_i \xleftarrow{\$} \mathsf{Image}^{O_n}, v_i \xleftarrow{\$} \mathsf{PreImage}^{O_n}(y_i)$,

– If $r_i = 1$, then set $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$

6. $O_n^*(z) \coloneqq \begin{cases} \xleftarrow{\$} \{0,1\}^n \setminus \{y_i\}_{i\in\mathcal{I}_1} & \text{if } z \in \bigcup_{i\in\mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i) \\ O_n(z) & \text{otherwise} \end{cases}$

7. Invoke the adversary $\mathcal{A}^{|O_n^*\rangle}(\{r_i\}_{i\in[p(n)]}, \{y_i\}_{i\in[p(n)]}, \{v_i\}_{i\in\mathcal{I}_0})$

Claim 1. *It holds that* $H_0 \overset{s}{\approx} H_1$.

The proof of Claim 1 is one of the most technically involved parts. We present it to Sec. 4.4.

4.3.2 Hybrid H_2

We present H_2 in Expr. 3 and highlight its difference from H_1 in red color. In this hybrid, we sample a fresh oracle O_n^{fresh} while preserving all the points in $\bigcup_{i\in\mathcal{I}_0\cup\mathcal{I}_1} \mathsf{PreImage}(y_i)$ and redirect those "bad" points (some x not in O_n^{fresh} while preserving all the points in $\bigcup_{i\in\mathcal{I}_0\cup\mathcal{I}_1} \mathsf{PreImage}(y_i)$ but $x \mapsto y_i$ in $\mathcal{O}^{\mathsf{fresh}}$) to form a new oracle $\mathcal{O}_n^*$.

Experiment 3: Hybrid H_2

1. According to the input $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, define $\begin{cases} \mathcal{I}_0 \coloneqq \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 \coloneqq \{i \in [p(n)] \mid r_i = 1\} \end{cases}$.
2. Sample $O_n \leftarrow \mathcal{O}_n$
3. Define $\begin{cases} \mathsf{Image}^{O_n} \coloneqq \{y \in \{0,1\}^n : \exists z \in \{0,1\}^n \text{ such that } O_n(z) = y\} \\ \mathsf{PreImage}^{O_n}(y) \coloneqq \{z \in \{0,1\}^n : O_n(z) = y\} \end{cases}$
4. Define $\begin{cases} \mathcal{I}_0 \coloneqq \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 \coloneqq \{i \in [p(n)] \mid r_i = 1\} \end{cases}$
5. $\forall i \in [p(n)]$:

   – If $r_i = 0$, then set $y_i \xleftarrow{\$} \mathsf{Image}^{O_n}, v_i \xleftarrow{\$} \mathsf{PreImage}^{O_n}(y_i)$

   – If $r_i = 1$, then set $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$

6. $O_n^*(z) \coloneqq \begin{cases} \xleftarrow{\$} \{0,1\}^n \setminus \{y_i\}_{i\in\mathcal{I}_1} & \text{if } z \in \bigcup_{i\in\mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i) \\ O_n(z) & \text{if } z \in \bigcup_{i\in\mathcal{I}_0} \mathsf{PreImage}^{O_n}(y_i) \\ \xleftarrow{\$} \{0,1\}^n \setminus \{y_i\}_{i\in\mathcal{I}_0\cup\mathcal{I}_1} & \text{otherwise} \end{cases}$
7. Invoke the adversary $\mathcal{A}^{|O_n^*\rangle}(\{r_i\}_{i\in[p(n)]}, \{y_i\}_{i\in[p(n)]}, \{v_i\}_{i\in\mathcal{I}_0})$

Claim 2. *It holds that* $H_1 \overset{s}{\approx} H_2$.

The proof of Claim 2 is another technically involved part of the proof. We present it in Sec. 4.5.

4.3.3 Hybrid $H_{2.5}$

We introduce an intermediate hybrid $H_{2.5}$. The only difference is that $H_{2.5}$ handles the $i \in \mathcal{I}_0$ case in the same manner as H_2 (i.e., y_i uniformly from the image of the random oracle, and then

sample δ_i uniformly from its preimage), but everything else is the same as in H_3. We present the full description of $H_{2.5}$ in Expr. 4, highlighting its differences from H_2 in red color.

Experiment 4: Hybrid $H_{2.5}$

1. According to the input $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, define $\begin{cases} \mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \end{cases}$.
2. Sample $O_n \leftarrow \mathcal{O}_n$
3. Define $\begin{cases} \mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \end{cases}$
4. $\forall i \in [p(n)]$:
   - If $r_i = 0$, then sample $y_i \xleftarrow{\$} \mathsf{Image}^{\mathcal{O}_n}$ and $\delta_i \xleftarrow{\$} \mathsf{PreImage}^{\mathcal{O}_n}(y_i)$, and define $v_i := \delta_i$.
   - If $r_i = 1$, then set $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$
5. $O_n^*(z) := \begin{cases} \xleftarrow{\$} \{0,1\}^n \setminus \{y_i\}_{i \in \mathcal{I}_1} & \text{if } z \in \bigcup_{i \in \mathcal{I}_1} \{\delta_i\} \\ O_n(z) & \text{if } z \in \bigcup_{i \in \mathcal{I}_0} \{\delta_i\} \\ \xleftarrow{\$} \{0,1\}^n \setminus \{y_i\}_{i \in \mathcal{I}_0 \cup \mathcal{I}_1} & \text{otherwise} \end{cases}$
6. Invoke the adversary $\mathcal{A}^{|O_n^*\rangle}(\{r_i\}_{i \in [p(n)]}, \{y_i\}_{i \in [p(n)]}, \{v_i\}_{i \in \mathcal{I}_0})$

Claim 3. *It holds that* $H_2 \overset{c_q}{\approx} H_{2.5}$.

The proof of Claim 3 is another technically involved part of the proof. We present it in Sec. 4.6.

4.3.4 Hybrid H_3

We present H_3 in Expr. 5 and highlight its difference from H_2 in red color. In this hybrid, we try to redirect less points and show the indistinguishability of the two oracles towards polynomial-query bound quantum adversary through one-way to hiding lemma.

Experiment 5: Hybrid H_3

1. According to the input $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, define $\begin{cases} \mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \end{cases}$.
2. Sample $O_n \leftarrow \mathcal{O}_n$
3. $\forall i \in [p(n)]$:
   - If $r_i = 0$, then sample $\delta_i \xleftarrow{\$} \{0,1\}^n$ and define $y_i := O_n(\delta_i)$ and $v_i := \delta_i$.
   - If $r_i = 1$, then set $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$
4. $O_n^*(z) := \begin{cases} \xleftarrow{\$} \{0,1\}^n \setminus \{y_i\}_{i \in \mathcal{I}_1} & \text{if } z \in \bigcup_{i \in \mathcal{I}_1} \{\delta_i\} \\ O_n(z) & \text{if } z \in \bigcup_{i \in \mathcal{I}_0} \{\delta_i\} \\ \xleftarrow{\$} \{0,1\}^n \setminus \{y_i\}_{i \in \mathcal{I}_0 \cup \mathcal{I}_1} & \text{otherwise} \end{cases}$
5. Invoke the adversary $\mathcal{A}^{|O_n^*\rangle}(\{r_i\}_{i \in [p(n)]}, \{y_i\}_{i \in [p(n)]}, \{v_i\}_{i \in \mathcal{I}_0})$

Claim 4. *It holds that* $H_{2.5} \overset{s}{\approx} H_3$.

Proof. Let $\mathbf{r} = (r_1, \ldots, r_{p(n)})$ be the corresponding input vector.

Define the following event:

– **Event** E: for every distinct $i, j \in [p(n)]$, it holds that $y_i \neq y_j$.

Since $O_n(\delta_i) = y_i$, event E also implies that all the δ_i's are distinct.

Conditional on the event E, the tuple $\{(\delta_i, y_i)\}_{i\in[p(n)]}$ has exactly the same distribution in both games: it is uniform over pairs of ordered lists with distinct entries within each list. To see this, both sampling procedures are invariant under arbitrary, independent relabelings of the domain and codomain. Those relabelings can map any such collision-free tuple to any other.

Furthermore, it holds that once this tuple $\{(\delta_i, y_i)\}_{i\in[p(n)]}$ is fixed, the construction of O_n^* is identical in both hybrids: the only retained original-oracle values are

$$O_n(\delta_i) = y_i \qquad (i \in \mathcal{I}_0),$$

and all other entries are freshly sampled according to the same rules described in Expr. 4.

Thus, the adversary's views conditioned on E are identical.

Next, we proceed to bound $\Pr[\neg E]$ in both hybrids.

In $H_{2.5}$, consider any pair $\{y_i, y_j\}$: if the index i belongs to $\mathcal{I}_0$, conditional on O_n, then y_i is uniform over $\mathsf{Image}(O_n)$ and is independent of y_j. Hence, for such an $i \in \mathcal{I}_0$, it holds that

$$\begin{aligned}
\Pr_{H_{2.5}}[y_i = y_j] &= \mathop{\mathbb{E}}_{H_{2.5}}\left[\frac{1}{|\mathsf{Image}(O_n)|}\right] \\
&\leq 2^{-n/2} + \Pr_{H_{2.5}}[|\mathsf{Image}(O_n)| \leq 2^{n/2}] \\
&= 2^{-n/2} + \mathsf{negl}(n). && \text{(Lem. 11)}
\end{aligned}$$

Otherwise (also in $H_{2.5}$), both i and j belong to $\mathcal{I}_1$, it holds that

$$\Pr[y_i = y_j] = 2^{-n} + (1 - 2^{-n})2^{-n} \leq 2^{1-n} \leq 2^{-n/2}$$

for sufficiently large n.

In H_3, all δ_i's are sampled independently and uniformly from $\{0,1\}^n$. Consequently, this bound $\Pr[y_i = y_j] \leq 2^{-n/2}$ holds for all pairs $\{y_i, y_j\}_{i,j\in[p(n)]}$ in H_3 as well.

Therefore, for either $b \in \{2.5, 3\}$, a union bound gives

$$\begin{aligned}
\Pr_{H_b}[\neg E] &= \Pr_{H_b}[\exists\, 1 \leq i < j \leq p(n) :\ y_i = y_j] \\
&\leq \sum_{1\leq i<j\leq p(n)} \Pr_{H_b}[y_i = y_j] \\
&\leq \binom{p(n)}{2}\left(2^{-n/2} + \mathsf{negl}(n)\right) = \mathsf{negl}(n).
\end{aligned}$$

We then have

$$\begin{aligned}
\Delta(\mathsf{distr}(V_{2.5}), \mathsf{distr}(V_3)) &\leq \Pr_{H_{2.5}}[\neg E] + \Pr_{H_3}[\neg E] \\
&\leq 2\binom{p(n)}{2}\left(2^{-n/2} + \mathsf{negl}(n)\right) = \mathsf{negl}(n),
\end{aligned}$$

where V_b denotes the adversary's view in hybrid H_b. The negligible bound above is uniform over $\boldsymbol{\rho}$.

This completes the proof of Claim 4.

□

4.3.5 Hybrid H_4

We present H_4 in Expr. 6 and highlight its difference from H_3 in red color.

Experiment 6: Hybrid H_4

1. According to the input $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, define $\begin{cases} \mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \end{cases}$.
2. Sample $O_n \leftarrow \mathcal{O}_n$
3. Define $\begin{cases} \mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \end{cases}$
4. $\forall i \in [p(n)]$:
   - If $r_i = 0$, then sample $\delta_i \xleftarrow{\$} \{0,1\}^n$ and define $y_i := O_n(\delta_i)$ and $v_i := \delta_i$.
   - If $r_i = 1$, then set $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$
5. $O_n^*(z) := \begin{cases} \xleftarrow{\$} \{0,1\}^n \setminus \{y_i\}_{i \in \mathcal{I}_1} & \text{if } z \in \bigcup_{i \in \mathcal{I}_1} \{\delta_i\} \\ \xleftarrow{\$} \{0,1\}^n \setminus \{y_i\}_{i \in \mathcal{I}_0 \cup \mathcal{I}_1} & \text{if } z \in \bigcup_{i \in \mathcal{I}_0 \cup \mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i) \setminus \{\delta_i\}_{i \in \mathcal{I}_0 \cup \mathcal{I}_1} \\ O_n(z) & \text{otherwise} \end{cases}$
6. Invoke the adversary $\mathcal{A}^{|O_n^*\rangle}(\{r_i\}_{i \in [p(n)]}, \{y_i\}_{i \in [p(n)]}, \{v_i\}_{i \in \mathcal{I}_0})$

Remark 2. Note that the conditions in each piecewise oracle definition are checked sequentially, from top to bottom. Each condition is considered only if all preceding conditions are false; the first satisfied condition determines the applicable branch. The "otherwise" branch applies only if every preceding condition is false. This convention applies to both H_3 and H_4. In particular, if $\delta_i = \delta_j$ for some $i \in \mathcal{I}_0$ and $j \in \mathcal{I}_1$, their common input is handled by the first branch in both hybrids. Thus, overlapping conditions are resolved consistently, including when designated inputs collide.

Claim 5. *It holds that* $H_3 \overset{i.d.}{\equiv} H_4$.

Proof. First, notice that the joint distribution of the following variables is identical in both hybrids:

$$\{(\delta_i, y_i)\}_{i \in [p(n)]} \tag{3}$$

Then, to show $H_3 \overset{\text{i.d.}}{\equiv} H_4$, it suffices to show that conditioned on these variables, the distribution of the oracle table of O_n^* is also identical in both hybrids.

Toward that, we consider the distribution of $O_n^*(z)$ for any $z \in \{0,1\}^n$ conditioned on the above variables. We have the following cases:

1. For $z \in \bigcup_{i \in \mathcal{I}_0} \{\delta_i\}$, both of the oracle return the same value y_i
2. For $z \in \bigcup_{i \in \mathcal{I}_1} \{\delta_i\}$, both of the oracle return a uniformly random value in $\{0,1\}^n \setminus \{y_i\}_{i \in \mathcal{I}_1}$
3. For the remaining z, conditioning on Expression (3) leaves $O_n(z)$ independent and uniform over $\{0,1\}^n$. In H_3, $O_n^*(z)$ is sampled uniformly from $\{0,1\}^n \setminus \{y_i\}_{i \in \mathcal{I}_0 \cup \mathcal{I}_1}$. In H_4, $O_n(z)$ is retained if it lies outside $\{y_i\}_{i \in \mathcal{I}_0 \cup \mathcal{I}_1}$; otherwise, it is resampled uniformly from $\{0,1\}^n \setminus \{y_i\}_{i \in \mathcal{I}_0 \cup \mathcal{I}_1}$. Thus, the remaining entries are independently uniform over $\{0,1\}^n \setminus \{y_i\}_{i \in \mathcal{I}_0 \cup \mathcal{I}_1}$ in both hybrids.

Thus, the two final oracles are identically distributed and we can conclude that $H_3 \overset{\text{i.d.}}{\equiv} H_4$.

This completes the proof of Claim 5.

□

4.3.6 Hybrid H_5

We present H_5 in Expr. 7 and highlight its difference from H_4 in red color.

Experiment 7: Hybrid H_5

1. According to the input $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, define $\begin{cases} \mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \end{cases}$.
2. Sample $O_n \leftarrow \mathcal{O}_n$
3. Define $\begin{cases} \mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \end{cases}$
4. $\forall i \in [p(n)]$:
   - If $r_i = 0$, then sample $\delta_i \xleftarrow{\$} \{0,1\}^n$ and define $y_i := O_n(\delta_i)$ and $v_i := \delta_i$.
   - If $r_i = 1$, then set $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$
5. $O_n^*(z) := \begin{cases} \xleftarrow{\$} \{0,1\}^n \setminus \{y_i\}_{i\in\mathcal{I}_1} & \text{if } z \in \bigcup_{i\in\mathcal{I}_1}\{\delta_i\} \\ O_n(z) & \text{if } z \in \bigcup_{i\in\mathcal{I}_0\cup\mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i) \setminus \{\delta_i\}_{i\in\mathcal{I}_0\cup\mathcal{I}_1} \\ O_n(z) & \text{otherwise} \end{cases}$
6. Invoke the adversary $\mathcal{A}^{|O_n^*\rangle}(\{r_i\}_{i\in[p(n)]}, \{y_i\}_{i\in[p(n)]}, \{v_i\}_{i\in\mathcal{I}_0})$

Claim 6. *It holds that $H_4 \overset{c_q}{\approx} H_5$, where $\overset{c_q}{\approx}$ denotes indistinguishability against polynomial-query quantum adversary.*

Proof. We use one-way-to-hiding lemma to reduce indistinguishability to collision resistance.

Apply Lem. 5 with auxiliary input $z = (\{r_i\}_{i\in[p(n)]}, \{y_i\}_{i\in[p(n)]}, \{v_i\}_{i\in\mathcal{I}_0})$, G and H equal to the final oracle O_n^* in H_4 and H_5 respectively, and $S = \bigcup_{i\in\mathcal{I}_0\cup\mathcal{I}_1 \mathsf{PreImage}^{O_n}(y_i)} \setminus \{\delta_i\}_{i\in\mathcal{I}_0\cup\mathcal{I}_1}$. It is obvious that the two oracles agree outside S. Let $q(n)$ be the query count of the fixed adversary and let B be the measured-query extractor defined in Lem. 5. Then its distinguishing advantage is at most $2q(n)\sqrt{P_{\text{guess}}}$, where

$$P_{\text{guess}} = \Pr_{H_5}[T \cap S \neq \emptyset \ : \ T \leftarrow B^{|O_n^*\rangle}(z)]. \tag{4}$$

Then, to finish the proof of Claim 6, it suffices to show that P_{guess} defined in (4) is negligible for any $q(n)$-query adversary. We proceed to establish that below.

Assume for contradiction that there is a polynomial $\mathsf{poly}(n)$ such that $P_{\mathsf{guess}} \geq \frac{1}{\mathsf{poly}(n)}$ for infinitely many n. We then construct an oracle-aided quantum algorithm **CollisionFinder** that can find a collision pair for oracle O_n with non-negligible probability as follows:

CollisionFinder$^{|O_n\rangle}$:

1. Repeat the steps to produce $z := (\{y_i\}_{i\in[p(n)]}, \{v_i\}_{i\in\mathcal{I}_0})$ as in hybrid H_5; In particular, record the values $\{\delta_i\}_{i\in\mathcal{I}_0\cup\mathcal{I}_1}$ for later use.
2. Invoke the oracle algorithm $T \leftarrow B^{O_n^*}(z)$ where the oracle O_n^* is defined in the same manner as in the hybrid H_5.
3. Check whether there exists a $t \in T$ such that $O_n(t) = y_i$ and $t \neq \delta_i$ for some $i \in \mathcal{I}_0 \cup \mathcal{I}_1$.
4. If such a pair exists, then output the corresponding pair (δ_i, t) as the collision pair and halt.

Although **CollisionFinder** is given oracle access only to O_n, it can coherently simulate O_n^* using recorded value in Step 1. It can be easily seen that whenever $T \cap S \neq \emptyset$, this **CollisionFinder** algorithm returns a collision for $|O_n\rangle$; hence the success probability of collision finding is at least P_{guess}, which is non-negligible. However, this contradicts Lem. 4.

This completes the proof of Claim 6.

□

4.3.7 Hybrid H_6

We present H_6 in Expr. 8 and highlight its difference from H_5 in red color.

Experiment 8: Hybrid H_6

1. According to the input $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, define $\begin{cases} \mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \end{cases}$.
2. Sample $O_n \leftarrow \mathcal{O}_n$
3. Define $\begin{cases} \mathcal{I}_0 := \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 := \{i \in [p(n)] \mid r_i = 1\} \end{cases}$
4. $\forall i \in [p(n)]$:
   - If $r_i = 0$, then sample $\delta_i \xleftarrow{\$} \{0,1\}^n$ and define $y_i := O_n(\delta_i)$ and $v_i := \delta_i$.
   - If $r_i = 1$, then set $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$
5. Delete the definition of O_n^*.
6. Invoke the adversary $\mathcal{A}^{|O_n\rangle}(\{r_i\}_{i\in[p(n)]}, \{y_i\}_{i\in[p(n)]}, \{v_i\}_{i\in\mathcal{I}_0})$

Claim 7. *It holds that* $H_5 \overset{c_q}{\approx} H_6$, *where* $\overset{c_q}{\approx}$ *denotes indistinguishability against polynomial-query quantum adversary.*

Proof. We use one-way-to-hiding lemma to reduce indistinguishability to preimage finding.

Apply Lem. 5 with auxiliary input $z = (\{y_i\}_{i\in[p(n)]}, \{v_i\}_{i\in\mathcal{I}_0})$, G and H equal to the final oracle in H_5 and H_6 respectively, and $S = \bigcup_{i\in\mathcal{I}_1}\{\delta_i\}$. It is obvious that the two oracles agree outside S. Let $q(n)$ be the query count of the fixed adversary and let B be the measured-query extractor defined in Lem. 5. Then its distinguishing advantage is at most $2q(n)\sqrt{P_{\text{guess}}}$, where

$$P_{\text{guess}} = \Pr_{H_6}[T \cap S \neq \emptyset \ : \ T \leftarrow B^{|O_n\rangle}(z)]. \tag{5}$$

Then, to finish the proof of Claim 7, it suffices to show that P_{guess} defined in (5) is negligible for any $q(n)$-query adversary. We proceed to establish that below.

Assume for contradiction that there is a polynomial $\mathsf{poly}(n)$ such that $P_{\mathsf{guess}} \geq \frac{1}{\mathsf{poly}(n)}$ for infinitely many n. We then construct an oracle-aided quantum algorithm **PreImageFinder** that, given challenge values $z_1, \ldots, z_{p(n)}$ sampled as in Corollary 2, finds a preimage of one of these values under O_n with non-negligible probability, as follows:

PreImageFinder$^{|O_n\rangle}(z_1, z_2, ..., z_{p(n)})$:

1. Repeat the steps to produce $z := (\{r_i\}_{i\in[p(n)]}, \{y_i\}_{i\in[p(n)]}, \{v_i\}_{i\in\mathcal{I}_0})$ as in hybrid H_5; except for the following change in step 4:

– If $r_i = 1$, set $y_i = z_i$

2. Invoke the oracle algorithm $T \leftarrow B^{|O_n\rangle}(z)$

3. Check whether there exists a $t \in T$ such that such that $O_n(t) = y_i$ for some $i \in \mathcal{I}_1$.

4. If such a pair exists, then output the corresponding point t as the targeted pre-image and halt.

The substitution $y_i = z_i$ for $i \in \mathcal{I}_1$ in Step 1 preserves the joint distribution of the oracle and the transcript π in H_6, since the challenge values in Corollary 2 are sampled in exactly the same manner as the corresponding y_i. Whenever $T \cap S \neq \emptyset$, **PreImageFinder** returns a point t satisfying $O_n(t) = y_i = z_i$ for some $i \in \mathcal{I}_1$. Hence, its success probability is at least P_{guess}, which is non-negligible. This contradicts Corollary 2.

This completes the proof of Claim 7.

□

4.4 Proof of Claim 1: $H_0 \overset{s}{\approx} H_1$

4.4.1 Proof Strategy Overview

We prove $H_0 \overset{s}{\approx} H_1$ by establishing the stronger statement that the joint distributions of the complete oracle table and the indexed inputs supplied to $\mathcal{A}$ are statistically close in the two experiments. We first introduce the notation used throughout the proof in Sec. 4.4.2. We then establish useful technical lemmas concerning random oracles and their sampling distributions in Sec. 4.4.3. Finally, in Sec. 4.4.4, we use these lemmas to construct the hybrid argument proving $H_0 \overset{s}{\approx} H_1$.

To analyze these joint distributions directly, we introduce the joint-output experiments $\widehat{H}_0$ and $\widehat{H}_1$, defined in Expr. 9 and 19, respectively. Similar as H_0 and H_1, these experiments have a input bit vector $\boldsymbol{r} = (r_i)_{i \in [p(n)]}$ and output the complete oracle table together with the inputs that would be supplied to $\mathcal{A}$, rather than invoking $\mathcal{A}$.

For convenience, we organize these inputs into two ordered tuples: T contains the labels y_i at indices with $r_i = 1$, and W contains the pairs (y_i, v_i) at indices with $r_i = 0$, both listed in increasing index order. These tuples and the associated sampling notation are defined formally in the following Sec. 4.4.2.

The joint-output formulation allows us to compare distributions without tracking the adversary's oracle queries. It also permits convenient changes in sampling order: conditional on the originally sampled oracle O_n and the fixed bit vector, the pairs comprising W are independent of one another, of T, and of the randomness used to modify the oracle. Thus, we may sample T and construct the reported oracle before sampling W, without changing their joint distribution. We then establish a chain of statistically close intermediate experiments,

$$\widehat{H}_0 \overset{s}{\approx} \cdots \overset{s}{\approx} \widehat{H}_1,$$

by gradually changing the sampling procedures and the reported oracle. Each transition incurs only negligible statistical distance, with bounds uniform over $\boldsymbol{r}$.

Finally, given $\boldsymbol{r}$ and the output of either joint-output experiment, we can reconstruct all indexed inputs and run $\mathcal{A}$ with the reported oracle. For each $b \in \{0, 1\}$, this common randomized transformation recovers H_b conditioned on $\boldsymbol{r}$. Since randomized post-processing cannot increase statistical distance (Lem. 38), the statistical closeness of $\widehat{H}_0$ and $\widehat{H}_1$ implies that of the corresponding conditional experiments. Averaging over the common distribution of $\boldsymbol{r}$ therefore yields $H_0 \overset{s}{\approx} H_1$.

4.4.2 Notation

Let $N = 2^n$. Since $p(n)$ is polynomially bounded, we restrict attention to sufficiently large n such that $p(n) < N$. Write

$$\mathbf{r} = (r_i)_{i\in[p(n)]}, \qquad \mathcal{I}_0 = \{i : r_i = 0\}, \qquad \mathcal{I}_1 = \{i : r_i = 1\}.$$

To simplify the presentation, define the ordered tuples

$$T = (y_i)_{i\in\mathcal{I}_1},\ W = ((y_i, v_i))_{i\in\mathcal{I}_0},$$

with indices in increasing order. Given $\mathbf{r}$, the pair (T, W) contains precisely the labels and witnesses supplied to $\mathcal{A}$.

For any function $f : \{0,1\}^n \to \{0,1\}^n$, use the image and preimage notation from the hybrids:

$$\mathsf{Image}^f = \{y \in \{0,1\}^n : \exists z \in \{0,1\}^n,\ f(z) = y\}, \qquad \mathsf{PreImage}^f(y) = \{z \in \{0,1\}^n : f(z) = y\}.$$

Let K_f denote the distribution of one positive label–witness pair:

$$K_f(y, v) = \begin{cases} \dfrac{1}{|\mathsf{Image}^f|\,|\mathsf{PreImage}^f(y)|}, & f(v) = y, \\ 0, & f(v) \neq y. \end{cases} \tag{6}$$

Thus, sampling from K_f means sampling y uniformly from Image^f and then sampling v uniformly from $\mathsf{PreImage}^f(y)$. The distribution $K_f^{\otimes a}$ describes all a independently sampled positive pairs for some integer a. When $a = 0$, it is the point mass on the empty tuple.

Finally, we recall the convention that once we need to sample from an empty set, the output is defined to be some special symbol (e.g., $\perp$ or 0^n).

4.4.3 Technical Tools regarding Random Oracles

In this subsection, we establish several properties of random oracles that will be used in the proof of $H_0 \overset{\mathrm{s}}{\approx} H_1$. These results allow us to bound the abort probabilities and justify the changes in sampling across the subsequent hybrids.

Lemma 11 (Image size of a random oracle). *Fix any $n \in \mathbb{N}$, let O_n be a random oracle from $\{0,1\}^n$ to $\{0,1\}^n$.*

$$\Pr_{O_n\leftarrow\mathcal{O}_n}[|\mathsf{Image}(O_n)| \le 2^{n/2}] \le 2^{n2^{n/2}-n2^{n-1}} = \mathsf{negl}(n).$$

Proof. Let $L = \lfloor 2^{n/2} \rfloor$. For any fixed L-element subset $A \subseteq \{0,1\}^n$, independence of the oracle outputs gives

$$\Pr_{O_n\leftarrow\mathcal{O}_n}[\mathsf{Image}(O_n) \subseteq A] = \left(\frac{L}{2^n}\right)^{2^n}.$$

Thus, by a union bound,

$$\begin{aligned}\Pr_{O_n\leftarrow\mathcal{O}_n}[|\mathsf{Image}(O_n)| \le 2^{n/2}] &= \Pr_{O_n\leftarrow\mathcal{O}_n}[\exists A \subseteq \{0,1\}^n : |A| = L,\ \mathsf{Image}(O_n) \subseteq A] \\ &\le \binom{2^n}{L}\left(\frac{L}{2^n}\right)^{2^n}\end{aligned}$$

$$\leq (2^n)^L \left(\frac{L}{2^n}\right)^{2^n}$$
$$\leq (2^n)^{2^{n/2}} \left(\frac{2^{n/2}}{2^n}\right)^{2^n}$$
$$= 2^{n2^{n/2} - n2^{n-1}}$$

□

Lemma 12 (Concentration of the Image Size). *For any $n \in \mathbb{N}$, let $O_n : \{0,1\}^n \to \{0,1\}^n$ be sampled from $\mathcal{O}_n$, and let*

$$\mu \coloneqq \mathbb{E}_{O_n \leftarrow \mathcal{O}_n}\left[|\mathsf{Image}^{O_n}|\right].$$

Then

$$\mu = 2^n \left(1 - \left(1 - \frac{1}{2^n}\right)^{2^n}\right) \geq \left(1 - \frac{1}{e}\right) 2^n \geq 2^{n-1}.$$

Moreover, for every $t \geq 0$,

$$\Pr_{O_n \leftarrow \mathcal{O}_n}\left[\left||\mathsf{Image}^{O_n}| - \mu\right| \geq t\right] \leq 2\exp\left(-\frac{2t^2}{2^n}\right).$$

In particular,

$$\Pr_{O_n \leftarrow \mathcal{O}_n}\left[\left||\mathsf{Image}^{O_n}| - \mu\right| \geq 2^{3n/4}\right] \leq 2\exp\left(-2 \cdot 2^{n/2}\right). \tag{7}$$

Proof. For each $y \in \{0,1\}^n$, we have

$$\Pr_{O_n \leftarrow \mathcal{O}_n}\left[y \notin \mathsf{Image}^{O_n}\right] = \left(1 - \frac{1}{2^n}\right)^{2^n}.$$

For each $y \in \{0,1\}^n$, let I_y be the indicator of the event $y \in \mathsf{Image}^{O_n}$. Since

$$|\mathsf{Image}^{O_n}| = \sum_{y \in \{0,1\}^n} I_y,$$

linearity of expectation gives

$$\mu = \sum_{y \in \{0,1\}^n} \mathbb{E}[I_y] = \sum_{y \in \{0,1\}^n} \Pr_{O_n \leftarrow \mathcal{O}_n}\left[y \in \mathsf{Image}^{O_n}\right].$$

For any fixed $y \in \{0,1\}^n$, the event $y \notin \mathsf{Image}^{O_n}$ occurs precisely when $O_n(x) \neq y$ for every $x \in \{0,1\}^n$. The oracle entries are independent and uniformly distributed, so

$$\Pr_{O_n \leftarrow \mathcal{O}_n}\left[y \notin \mathsf{Image}^{O_n}\right] = \prod_{x \in \{0,1\}^n} \Pr_{O_n \leftarrow \mathcal{O}_n}[O_n(x) \neq y] = \left(1 - \frac{1}{2^n}\right)^{2^n}.$$

Substituting this identity into the expression for μ yields

$$\mu = 2^n \left(1 - \left(1 - \frac{1}{2^n}\right)^{2^n}\right).$$

Finally, the inequality $1 - u \le e^{-u}$ for $u \ge 0$ implies

$$\left(1 - \frac{1}{2^n}\right)^{2^n} \le \left(e^{-2^{-n}}\right)^{2^n} = e^{-1}.$$

Consequently,

$$\mu \ge \left(1 - \frac{1}{e}\right) 2^n \ge 2^{n-1},$$

where the last inequality follows from $e > 2$

The image size is a function of the 2^n independent oracle entries $\{O_n(x)\}_{x \in \{0,1\}^n}$. Changing any one entry changes the image size by at most one. Thus, lem. 36, with each $c_i = 1$, gives

$$\Pr_{O_n \leftarrow \mathcal{O}_n}\left[\left||\mathsf{Image}^{O_n}| - \mu\right| \ge t\right] \le 2\exp\left(-\frac{2t^2}{2^n}\right).$$

Setting $t = 2^{3n/4}$ proves eq. (7). □

Lemma 13 (Preimage distribution for forward sampling). *Fix $n \in \mathbb{N}$. Sample $O_n \leftarrow \mathcal{O}_n$ and $\delta \xleftarrow{\$} \{0,1\}^n$ independently. Then*

$$|\mathsf{PreImage}^{O_n}(O_n(\delta))| \overset{i.d.}{=\!=} 1 + X, \qquad X \sim \mathsf{Binomial}(2^n - 1, 2^{-n}).$$

Proof. Condition on $\delta = z$ and $O_n(z) = y$. The input z itself belongs to $\mathsf{PreImage}^{O_n}(y)$. Each of the remaining $2^n - 1$ inputs maps to y independently with probability 2^{-n}. The claimed distribution follows, since it does not depend on the conditioned values z and y.

□

Corollary 3 (of Lem. 13, tail bound for preimage size). *Sample $O_n \leftarrow \mathcal{O}_n$ and $\delta \xleftarrow{\$} \{0,1\}^n$ independently. Then*

$$\Pr_{\delta \xleftarrow{\$} \{0,1\}^n,\, O_n \leftarrow \mathcal{O}_n}\left[|\mathsf{PreImage}^{O_n}(O_n(\delta))| > \log n\right] = \mathsf{negl}(n).$$

Proof. By Lem. 13, the preimage size is distributed as $1 + X$, where

$$X \sim \mathsf{Binomial}(2^n - 1, 2^{-n}), \qquad \mathbb{E}[X] = 1 - 2^{-n} < 1.$$

Let $t = \lfloor \log n \rfloor$. Since X is integer-valued, $1 + X > \log n$ is equivalent to $X \ge t$. For sufficiently large n, we have $t \ge 1$. Applying Corollary 6 with $\mu = 1$ gives

$$\begin{aligned}\Pr\left[|\mathsf{PreImage}^{O_n}(O_n(\delta))| > \log n\right] &= \Pr[X \ge t] \\ &\le e^{-1}\left(\frac{e}{t}\right)^t \\ &= \mathsf{negl}(n).\end{aligned}$$

The final bound is negligible because $t = \Theta(\log n)$ and $(e/t)^t = \exp(-\Theta(\log n \log\log n))$. □

Lemma 14 (Changing the sampling order). *Let $\mathcal{I}_1$ be any fixed subset of $[p(n)]$ where $p(n)$ is a polynomially bounded function of n. The games G_0 and G_1 defined in Box 2 are statistically close: $G_0 \overset{s}{\approx} G_1$.*

Box 2: Games G_0 and G_1: Changing the sampling order	
Game $G_0(\mathcal{I}_1)$: 1. Sample $O_n \leftarrow \mathcal{O}_n$. 2. $\forall i \in \mathcal{I}_1$, sample $y_i \xleftarrow{\$} \{0,1\}^n \setminus \mathsf{Image}^{O_n}$. (Recall our convention that y_i is set to a special symbol (e.g., 0^n) if $\{0,1\}^n \setminus \mathsf{Image}^{O_n} = \emptyset$.) 3. Set $T \coloneqq \{y_i\}_{i \in \mathcal{I}_1}$. 4. Output (O_n, T).	**Game $G_1(\mathcal{I}_1)$:** 1. $\forall i \in \mathcal{I}_1$, sample $y_i \xleftarrow{\$} \{0,1\}^n$. 2. Set $T \coloneqq \{y_i\}_{i \in \mathcal{I}_1}$. 3. Sample an oracle O_n where each $O_n(x)$ is independently sampled from $\{0,1\}^n \setminus T$. 4. Output (O_n, T).

Proof. Write $N \coloneqq 2^n$ and $m \coloneqq |\mathcal{I}_1| = \mathsf{poly}(n)$. The case $m = 0$ is immediate, so assume $1 \le m < N$, as holds for all sufficiently large n.

In this proof, instead of T, we use the ordered samples $y = (y_1, \ldots, y_m)$ in the output; replacing y by its underlying set T cannot increase statistical distance.

Let $P_0(\cdot,\cdot)$ and $P_1(\cdot,\cdot)$ denote the probability mass functions of the joint distribution of (O_n, y) in G_0 and G_1, respectively, and write $\Pr_{G_b}$ for probability in game G_b.

Define $K(O_n) \coloneqq N - |\mathsf{Image}^{O_n}|$. To make G_0 well-defined when $K(O_n) = 0$, set $y = (0^n, \ldots, 0^n)$ in that case; the analysis below shows that this convention affects only a negligible probability mass.

Define the bad event

$$E_{\mathsf{bad}} \coloneqq \Big(|K(O_n) - N/e| > N^{3/4}\Big) \;\vee\; \big(\exists\, 1 \le i < j \le m,\ s.t.\ y_i = y_j\big)\,, \tag{8}$$

and define the set of good output pairs by

$$\mathcal{G} \coloneqq \left\{ (O_n, y) : \begin{array}{c} y_i \notin \mathsf{Image}^{O_n} \text{ for every } i \in [m], \\ y_i \neq y_j \text{ for every } 1 \le i < j \le m, \\ |K(O_n) - N/e| \le N^{3/4} \end{array} \right\}.$$

For all sufficiently large n, in either game the output belongs to $\mathcal{G}$ exactly when E_{bad} does not occur.

For a uniform random oracle, the missing-image count satisfies

$$\mathbb{E}[K(O_n)] = N(1 - 1/N)^N = N/e + O(1)$$

For a uniformly sampled random oracle, the missing-image count $K(O_n) = N - |\mathsf{Image}^{O_n}|$ satisfies

$$\mathbb{E}[K(O_n)] = N\left(1 - \frac{1}{N}\right)^N = \frac{N}{e} + O(1).$$

Since $K(O_n)$ differs from the image size only by subtraction from N, lem. 12 gives the same concentration bound for $K(O_n)$ around its expectation. Moreover, for all sufficiently large N,

$$\left|\mathbb{E}[K(O_n)] - \frac{N}{e}\right| \le \frac{1}{2}N^{3/4}.$$

Therefore,

$$\begin{aligned} \Pr_{O_n \leftarrow \mathcal{O}_n}\left[\left|K(O_n) - \frac{N}{e}\right| > N^{3/4}\right] &\le \Pr_{O_n \leftarrow \mathcal{O}_n}\left[|K(O_n) - \mathbb{E}[K(O_n)]| > \frac{1}{2}N^{3/4}\right] \\ &\le 2e^{-\frac{1}{2}\sqrt{N}}. \end{aligned}$$

Otherwise, $K(O_n) = \Omega(N)$, and a union bound bounds the probability of a sample collision by $\binom{m}{2}/K(O_n) = O(m^2/N)$. For all sufficiently large n, the case $K(O_n) = 0$ is included in the deviation event. Hence, by Eq. (8),

$$\Pr_{G_0}[E_{\mathsf{bad}}] = O(e^{-\frac{1}{2}\sqrt{N}} + m^2/N) = \mathsf{negl}(n). \tag{9}$$

Fix $(O_n, y) \in \mathcal{G}$. Since the samples are distinct,

$$P_0(O_n, y) = N^{-N} K(O_n)^{-m}, \qquad P_1(O_n, y) = N^{-m}(N-m)^{-N}.$$

Set $a := N^{-N}(e/N)^m$. For every pair in $\mathcal{G}$, $K(O_n) = (N/e)(1 + O(N^{-1/4}))$, uniformly over these pairs. Consequently,

$$P_0(O_n, y) = a \cdot \exp\big(O(mN^{-1/4})\big), \qquad P_1(O_n, y) = a \cdot \exp\big(O(m^2/N)\big).$$

Since $m = \mathrm{poly}(n)$, there exists a negligible function $\varepsilon = \varepsilon(n)$ such that, uniformly over $(O_n, y) \in \mathcal{G}$,

$$|P_0(O_n, y) - a| \le \varepsilon a, \qquad |P_1(O_n, y) - a| \le \varepsilon a. \tag{10}$$

In particular,

$$(1-\varepsilon)|\mathcal{G}|a \le \sum_{(O_n,y)\in\mathcal{G}} P_0(O_n, y) \le 1.$$

Summing Eq. (10) therefore yields

$$\sum_{(O_n,y)\in\mathcal{G}} |P_0(O_n, y) - P_1(O_n, y)| \le 2\varepsilon|\mathcal{G}|a \le \frac{2\varepsilon}{1-\varepsilon} = \mathsf{negl}(n). \tag{11}$$

By Eq. (9) and (11),

$$\begin{aligned}\Pr_{G_1}[E_{\mathsf{bad}}] &\le \Pr_{G_0}[E_{\mathsf{bad}}] + \left|\Pr_{G_0}[(O_n, y) \in \mathcal{G}] - \Pr_{G_1}[(O_n, y) \in \mathcal{G}]\right| \\ &= \mathsf{negl}(n).\end{aligned}$$

Finally,

$$\Delta(P_0, P_1) \le \frac{1}{2}\left(\sum_{(O_n,y)\in\mathcal{G}} |P_0(O_n, y) - P_1(O_n, y)| + \Pr_{G_0}[E_{\mathsf{bad}}] + \Pr_{G_1}[E_{\mathsf{bad}}]\right) = \mathsf{negl}(n).$$

□

Lemma 15 (Resampling outside the preimage). *The games G_0 and G_1 defined in Box 3 are identically distributed: $G_0 \overset{i.d.}{=\!=} G_1$.*

Proof. Fix any attainable pair (S, T). Since $S = \bigcup_{i\in\mathcal{I}_1} \mathsf{PreImage}(y_i)$, every oracle value outside S must lie outside T. All remaining constraints defining (S, T) involve only the sampled inputs and the restriction $O_n|_S$. Consequently, conditional on (S, T), the values $\{O_n(x)\}_{x\in\overline{S}}$ are independent and uniformly distributed over $\{0,1\}^n \setminus T$.

The resampling step in G_1 therefore samples $O_n|_{\overline{S}}$ from exactly its conditional distribution in G_0, preserving the joint distribution of $(S, T, O_n|_{\overline{S}})$.

□

4.4.4 Proving $H_0 \overset{\mathrm{s}}{\approx} H_1$

Hybrid $\widehat{H}_0$: We present $\widehat{H}_0$, the joint-output experiment corresponding to H_0, in Expr. 9.

Box 3: Games G_0 and G_1: Resampling Outside the Preimage

Game G_0:

1. Sample $O_n \leftarrow \mathcal{O}_n$.
2. For each $i \in \mathcal{I}_1$, independently sample $\delta_i \xleftarrow{\$} \{0,1\}^n$ and set $y_i := O_n(\delta_i)$.
3. Set $T := \{y_i\}_{i\in\mathcal{I}_1}$.
4. Set $S := \bigcup_{i\in\mathcal{I}_1} \mathsf{PreImage}(y_i)$ and $\overline{S} := \{0,1\}^n \setminus S$.
5. Name the resulting function generated by O_n restricted to $\overline{S}$ by $O_n|_{\overline{S}}$.

Output $(S, T, O_n|_{\overline{S}})$.

Game G_1:

1. Run Steps 1–4 of G_0 to obtain $(O_n, S, \overline{S}, T)$.
2. For every $x \in \overline{S}$, independently resample
$$O_n(x) \xleftarrow{\$} \{0,1\}^n \setminus T.$$
3. Name the resulting function generated by O_n restricted to $\overline{S}$ by $O_n|_{\overline{S}}$.

Output $(S, T, O_n|_{\overline{S}})$.

Experiment 9: Hybrid $\widehat{H}_0(\{r_i\}_{i\in[p(n)]})$

1. Sample $O_n \leftarrow \mathcal{O}_n$;
2. For $i \in \mathcal{I}_1$:
   - Sample $y_i \xleftarrow{\$} \{0,1\}^n \setminus \mathsf{Image}^{O_n}$.
3. Set $T := \{y_i\}_{i\in\mathcal{I}_1}$
4. Sample $W \sim K_{O_n}^{\otimes|\mathcal{I}_0|}$
5. Output (O_n, T, W)

Hybrid $\widehat{H}_{0.1}$: We present $\widehat{H}_{0.1}$ in Expr. 10 and highlight its difference from $\widehat{H}_0$ in red color. In this hybrid, we first sample $y_i \xleftarrow{\$} \{0,1\}^n$ independently for each $i \in \mathcal{I}_1$ and set $T = \{y_i\}_{i\in\mathcal{I}_1}$. We then sample each $O_n(x)$ independently and uniformly from $\{0,1\}^n \setminus T$, ensuring that $T \cap \mathsf{Image}^{O_n} = \varnothing$. For $i \in \mathcal{I}_0$, the pairs (y_i, v_i) are sampled as before, namely $W \sim K_{O_n}^{\otimes|\mathcal{I}_0|}$.

Experiment 10: Hybrid $\widehat{H}_{0.1}(\{r_i\}_{i\in[p(n)]})$

1. For $i \in \mathcal{I}_1$:
   - Sample $y_i \xleftarrow{\$} \{0,1\}^n$
2. Set $T := \{y_i\}_{i\in\mathcal{I}_1}$
3. Sample O_n where each $x \in \{0,1\}^n$, $O_n(x)$ is sampled independently and uniformly from $\{0,1\}^n \setminus T$;
4. Sample $W \sim K_{O_n}^{\otimes|\mathcal{I}_0|}$
5. Output (O_n, T, W)

Claim 8. *It holds that $\widehat{H}_0 \overset{s}{\approx} \widehat{H}_{0.1}$.*

Proof. First, by Lem. 14, the joint distributions of the following variables are statistically indistinguishable in the two hybrids:
$$O_n, \qquad T = \{y_i\}_{i\in\mathcal{I}_1}. \tag{12}$$

Next, observe that, conditioned on the variables in Expression (12), both hybrids sample W from the same distribution $K_{O_n}^{\otimes|\mathcal{I}_0|}$. Thus, their full outputs (O_n, T, W) are obtained by applying the

same randomized transformation to their respective distributions of the variables in Expression (12). Since statistical distance cannot increase under such a transformation by Lem. 38, the full output distributions are also statistically indistinguishable.

This yields $\widehat{H}_0 \overset{s}{\approx} \widehat{H}_{0.1}$, completing the proof of Claim 8.

□

Hybrid $\widehat{H}_{0.2}$: We present $\widehat{H}_{0.2}$ in Expr. 11 and highlight its difference from $\widehat{H}_{0.1}$ in red color. In this hybrid, we rename the sampled function O_n to F. Accordingly, we sample $W \sim K_F^{\otimes|\mathcal{I}_0|}$ and output (F, T, W). This change is purely notational and preserves the output distribution.

Experiment 11: Hybrid $\widehat{H}_{0.2}(\{r_i\}_{i\in[p(n)]})$

1. For $i \in \mathcal{I}_1$:
   - Sample $y_i \xleftarrow{\$} \{0,1\}^n$
2. Set $T \coloneqq \{y_i\}_{i\in\mathcal{I}_1}$
3. Sample F where each $x \in \{0,1\}^n$, $F(x)$ is sampled independently and uniformly from $\{0,1\}^n \setminus T$;
4. Sample $W \sim K_F^{\otimes|\mathcal{I}_0|}$
5. Output (F, T, W)

Claim 9. *It holds that $\widehat{H}_{0.1} \overset{i.d.}{=\!=} \widehat{H}_{0.2}$.*

Proof. This hybrid is identical to $\widehat{H}_{0.1}$ except a notational change where we rename the sampled function O_n to F. Therefore, the two distributions are identical. □

Hybrid $\widehat{H}_{0.3}$: We present $\widehat{H}_{0.3}$ in Expr. 12 and highlight its difference from $\widehat{H}_{0.2}$ in red color. In this hybrid, we generate $\{y_i\}_{i\in\mathcal{I}_1}$ by first sampling $O_n \leftarrow \mathcal{O}_n$ and then independently sampling $\delta_i \xleftarrow{\$} \{0,1\}^n$ and setting $y_i = O_n(\delta_i)$ for each $i \in \mathcal{I}_1$. We set $T = \{y_i\}_{i\in\mathcal{I}_1}$ and introduce $S = \bigcup_{i\in\mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i)$, the set of all inputs that O_n maps into T.

Experiment 12: Hybrid $\widehat{H}_{0.3}(\{r_i\}_{i\in[p(n)]})$

1. Sample $O_n \leftarrow \mathcal{O}_n$;
2. For $i \in \mathcal{I}_1$:
   - Sample $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$
3. Set $T \coloneqq \{y_i\}_{i\in\mathcal{I}_1}$
4. Define $S \coloneqq \bigcup_{i\in\mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i)$, $\overline{S} \coloneqq \{0,1\}^n \setminus S$
5. Sample a function $F : \{0,1\}^n \to \{0,1\}^n$ where each entry independently:

$$F(x) = \begin{cases} \text{pick uniformly from } \{0,1\}^n \setminus T & \text{if } x \in S, \\ \text{pick uniformly from } \{0,1\}^n \setminus T & \text{if } x \in \overline{S}. \end{cases}$$

6. Sample $W \sim K_F^{\otimes|\mathcal{I}_0|}$
7. Output (F, T, W)

Claim 10. *It holds that* $\widehat{H}_{0.2} \overset{s}{\approx} \widehat{H}_{0.3}$.

Proof. Write $k = |\mathcal{I}_1|$. Let E denote the event that the inputs $\{\delta_i\}_{i\in\mathcal{I}_1}$ sampled in $\widehat{H}_{0.3}$ are pairwise distinct.

First, conditioned on E, the values $y_i = O_n(\delta_i)$ are independent and uniformly distributed over $\{0,1\}^n$. Thus, the distribution of T in $\widehat{H}_{0.3}$ conditioned on E is identical to its distribution in $\widehat{H}_{0.2}$.

Next, conditioned on T in $\widehat{H}_{0.2}$ and on T together with E in $\widehat{H}_{0.3}$, both hybrids sample every entry of F independently and uniformly from $\{0,1\}^n \setminus T$, and then sample $W \sim K_F^{\otimes|\mathcal{I}_0|}$. Consequently, we have $\widehat{H}_{0.2} \overset{\text{i.d.}}{\equiv} (\widehat{H}_{0.3} \mid E)$. Therefore,

$$\Delta(\widehat{H}_{0.2}, \widehat{H}_{0.3}) \le \Pr[\neg E] \le \Pr[\exists i \neq j \in \mathcal{I}_1 : \delta_i = \delta_j] \le \binom{|\mathcal{I}_1|}{2} 2^{-n} = \mathsf{negl}(n).$$

□

Hybrid $\widehat{H}_{0.4}$: We present $\widehat{H}_{0.4}$ in Expr. 13 and highlight its difference from $\widehat{H}_{0.3}$ in red color. In this hybrid, we set $F(x) = O_n(x)$ for every $x \in \{0,1\}^n \setminus S$ instead of resampling these values. Since retaining these values has the same distribution as resampling them, we have $\widehat{H}_{0.3} \overset{\text{i.d.}}{\equiv} \widehat{H}_{0.4}$.

Experiment 13: Hybrid $\widehat{H}_{0.4}(\{r_i\}_{i\in[p(n)]})$

1. Sample $O_n \leftarrow \mathcal{O}_n$;
2. For $i \in \mathcal{I}_1$:
   - Sample $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$
3. Set $T \coloneqq \{y_i\}_{i\in\mathcal{I}_1}$
4. Define $S \coloneqq \bigcup_{i\in\mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i)$, and $\overline{S} \coloneqq \{0,1\}^n \setminus S$
5. Sample a function $F : \{0,1\}^n \to \{0,1\}^n$ where each entry independently:

$$F(x) = \begin{cases} \text{pick uniformly from } \{0,1\}^n \setminus T & \text{if } x \in S, \\ O_n(x) & \text{if } x \in \overline{S}. \end{cases}$$

6. Sample $W \sim K_F^{\otimes|\mathcal{I}_0|}$
7. Output (F, T, W)

Claim 11. *It holds that* $\widehat{H}_{0.3} \overset{\text{i.d.}}{\equiv} \widehat{H}_{0.4}$.

Proof. First, by Lem. 15, the following variables have identical joint distributions in both hybrids:

$$S,\ T,\ F|_{\overline{S}}, \tag{13}$$

where $F|_{\overline{S}}$ denotes the restriction of F to inputs in $\overline{S}$. Here, $\widehat{H}_{0.3}$ independently resamples the oracle entries at inputs $x \in \overline{S}$, while $\widehat{H}_{0.4}$ retains their original values.

Next, conditioned on the variables in Expression (13), both hybrids sample each $F(x)$ for $x \in S$ independently and uniformly from $\{0,1\}^n \setminus T$. Thus, (S, T, F) has the same joint distribution in both hybrids.

Finally, both hybrids sample $W \sim K_F^{\otimes|\mathcal{I}_0|}$ using fresh randomness. Consequently, their full outputs (F, T, W) are identically distributed, establishing $\widehat{H}_{0.3} \overset{\text{i.d.}}{\equiv} \widehat{H}_{0.4}$.

□

Hybrid $\widehat{H}_{0.5}$: We present $\widehat{H}_{0.5}$ in Expr. 14 and highlight its difference from $\widehat{H}_{0.4}$ in red color. In this hybrid, for each $i \in \mathcal{I}_0$, we sample $y_i \xleftarrow{\$} \mathsf{Image}(F)$ and abort if $\mathsf{PreImage}^{F}(y_i) \neq \mathsf{PreImage}^{O_n}(y_i)$. Intuitively, an abort is unlikely because it can occur only when $y_i \in F(S)$, which is small compared with $\mathsf{Image}(F)$, giving $\widehat{H}_{0.4} \overset{s}{\approx} \widehat{H}_{0.5}$.

Experiment 14: Hybrid $\widehat{H}_{0.5}(\{r_i\}_{i\in[p(n)]})$

1. Sample $O_n \leftarrow \mathcal{O}_n$;
2. For $i \in \mathcal{I}_1$:
   - Sample $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$
3. Set $T := \{y_i\}_{i\in\mathcal{I}_1}$
4. Define $S := \bigcup_{i\in\mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i)$, $\overline{S} := \{0,1\}^n \setminus S$
5. Sample a function $F : \{0,1\}^n \to \{0,1\}^n$ where each entry independently:

$$F(x) = \begin{cases} \text{pick uniformly from } \{0,1\}^n \setminus T & \text{if } x \in S, \\ O_n(x) & \text{if } x \in \overline{S}. \end{cases}$$

6. Initialize $W = \{\}$.
7. For $i \in \mathcal{I}_0$:
   - Sample $y_i \xleftarrow{\$} \mathsf{Image}(F)$
   - Abort if $\mathsf{PreImage}^{F}(y_i) \neq \mathsf{PreImage}^{O_n}(y_i)$
   - Sample $v_i \xleftarrow{\$} \mathsf{PreImage}^{O_n}(y_i)$
   - $W = W \cup \{(y_i, v_i)\}$
8. Output (F, T, W)

Claim 12. *It holds that $\widehat{H}_{0.4} \overset{s}{\approx} \widehat{H}_{0.5}$.*

Proof. First, the following variables have the same joint distribution in both hybrids:

$$O_n,\ S,\ T,\ F. \tag{14}$$

Next, conditioned on the variables in Expression (14), whenever $\widehat{H}_{0.5}$ does not abort, both hybrids use the same preimage sets and therefore sample each v_i from the same distribution. Thus, it suffices to bound the probability that $\widehat{H}_{0.5}$ aborts.

Since F agrees with O_n outside S and never outputs a value in T, the preimages of a label $y \in \mathsf{Image}(F)$ can differ only if $y \in F(S)$. Thus, a union bound over the $|\mathcal{I}_0|$ label samples gives

$$\begin{aligned} \Pr[\mathsf{Abort} \mid O_n, S, T, F] &\leq \sum_{i\in\mathcal{I}_0} \Pr[y_i \in F(S) \mid O_n, S, T, F] \\ &= \sum_{i\in\mathcal{I}_0} \frac{|F(S)|}{|\mathsf{Image}(F)|} \\ &= \frac{|\mathcal{I}_0|\,|F(S)|}{|\mathsf{Image}(F)|} \end{aligned}$$

$$\leq \frac{|\mathcal{I}_0|\,|S|}{|\mathsf{Image}(F)|}.$$

Moreover,

$$\begin{aligned}
\mathbb{E}[|S|] &= \mathbb{E}\left[\left|\bigcup_{i\in\mathcal{I}_1} \mathsf{PreImage}^{O_n}(O_n(\delta_i))\right|\right] \\
&\leq \mathbb{E}\left[\sum_{i\in\mathcal{I}_1} |\mathsf{PreImage}^{O_n}(O_n(\delta_i))|\right] \\
&= \sum_{i\in\mathcal{I}_1} \mathbb{E}\left[|\mathsf{PreImage}^{O_n}(O_n(\delta_i))|\right] \\
&= \sum_{i\in\mathcal{I}_1} \left(1+\frac{2^n-1}{2^n}\right) && \text{(Lem. 13)} \\
&= |\mathcal{I}_1|\left(1+\frac{2^n-1}{2^n}\right) \\
&\leq 2|\mathcal{I}_1|.
\end{aligned}$$

Here each preimage class contains δ_i itself, and each other input has the same oracle output with probability 2^{-n}.

By Lem. 11,

$$\Pr\left[|\mathsf{Image}(O_n)| \leq 2^{n/2}\right] \leq 2^{n2^{n/2}-n2^{n-1}}.$$

Outside this exceptional event, F retains every original image value outside T, so

$$|\mathsf{Image}(F)| \geq |\mathsf{Image}(O_n)| - |T| > 2^{n/2} - |\mathcal{I}_1| \geq 2^{n/2-1},$$

where the last inequality holds for sufficiently large n, since $|\mathcal{I}_1|$ is polynomially bounded.

Combining these bounds and averaging over the randomness yields

$$\begin{aligned}
\Pr[\mathsf{Abort}] &\leq \Pr\left[|\mathsf{Image}(O_n)| \leq 2^{n/2}\right] + \Pr\left[\mathsf{Abort} \ \wedge \ |\mathsf{Image}(O_n)| > 2^{n/2}\right] \\
&\leq 2^{n2^{n/2}-n2^{n-1}} + \frac{|\mathcal{I}_0|\,\mathbb{E}[|S|]}{2^{n/2-1}} \\
&\leq 2^{n2^{n/2}-n2^{n-1}} + \frac{4|\mathcal{I}_0|\,|\mathcal{I}_1|}{2^{n/2}} \\
&= \mathsf{negl}(n).
\end{aligned}$$

We therefore establish $\widehat{H}_{0.4} \overset{\text{s}}{\approx} \widehat{H}_{0.5}$. □

Hybrid $\widehat{H}_{0.6}$: We present $\widehat{H}_{0.6}$ in Expr. 15 and highlight its difference from $\widehat{H}_{0.5}$ in red color. In this hybrid, we remove the abort check introduced in $\widehat{H}_{0.5}$. Since this check causes an abort only with negligible probability, removing it gives $\widehat{H}_{0.5} \overset{\text{s}}{\approx} \widehat{H}_{0.6}$.

Experiment 15: Hybrid $\widehat{H}_{0.6}(\{r_i\}_{i\in[p(n)]})$
1. Sample $O_n \leftarrow \mathcal{O}_n$;
2. For $i \in \mathcal{I}_1$:

– Sample $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$

3. Set $T := \{y_i\}_{i\in\mathcal{I}_1}$
4. Define $S := \bigcup_{i\in\mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i)$, $\overline{S} := \{0,1\}^n \setminus S$
5. Sample a function $F : \{0,1\}^n \to \{0,1\}^n$ where each entry independently:

$$F(x) = \begin{cases} \text{pick uniformly from } \{0,1\}^n \setminus T & \text{if } x \in S, \\ O_n(x) & \text{if } x \in \overline{S}. \end{cases}$$

6. Initialize $W = \{\}$.
7. For $i \in \mathcal{I}_0$:
   – Sample $y_i \xleftarrow{\$} \mathsf{Image}(F)$
   – ~~Abort if $\mathsf{PreImage}^{F}(y_i) \neq \mathsf{PreImage}^{O_n}(y_i)$~~
   – Sample $v_i \xleftarrow{\$} \mathsf{PreImage}^{O_n}(y_i)$
   – $W = W \cup \{(y_i, v_i)\}$
8. Output (F, T, W)

Claim 13. *It holds that* $\widehat{H}_{0.5} \overset{s}{\approx} \widehat{H}_{0.6}$.

Proof. The two hybrids differ only in the abort check removed in $\widehat{H}_{0.6}$. The argument for $\widehat{H}_{0.4} \overset{s}{\approx} \widehat{H}_{0.5}$ already shows that this check is triggered with probability at most $\mathsf{negl}(n)$, establishing $\widehat{H}_{0.5} \overset{s}{\approx} \widehat{H}_{0.6}$. □

Hybrid $\widehat{H}_{0.7}$: We present $\widehat{H}_{0.7}$ in Expr. 16 and highlight its difference from $\widehat{H}_{0.6}$ in red color. In this hybrid, we introduce abort checks to ensure that $|S|$ is polynomially bounded and $|\mathsf{Image}(O_n)|$ is exponentially large. These bounds let us control the effect of modifying O_n on S in the subsequent hybrids. Since the checks trigger only with negligible probability, we have $\widehat{H}_{0.6} \overset{s}{\approx} \widehat{H}_{0.7}$.

Experiment 16: Hybrid $\widehat{H}_{0.7}(\{r_i\}_{i\in[p(n)]})$

1. Sample $O_n \leftarrow \mathcal{O}_n$;
2. For $i \in \mathcal{I}_1$:
   – Sample $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$
3. Set $T := \{y_i\}_{i\in\mathcal{I}_1}$
4. Define $S := \bigcup_{i\in\mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i)$, $\overline{S} := \{0,1\}^n \setminus S$
5. Sample a function $F : \{0,1\}^n \to \{0,1\}^n$ where each entry independently:

$$F(x) = \begin{cases} \text{pick uniformly from } \{0,1\}^n \setminus T & \text{if } x \in S, \\ O_n(x) & \text{if } x \in \overline{S}. \end{cases}$$

6. Abort if $|S| \geq p(n)\log(n)$ or $|\mathsf{Image}(O_n)| \leq 2^{n/2}$
7. Initialize $W = \{\}$.
8. For $i \in \mathcal{I}_0$:

- Sample $y_i \xleftarrow{\$} \mathsf{Image}(F)$
- Sample $v_i \xleftarrow{\$} \mathsf{PreImage}^{O_n}(y_i)$
- $W = W \cup \{(y_i, v_i)\}$

9. Output (F, T, W)

Claim 14. *It holds that* $\widehat{H}_{0.6} \overset{s}{\approx} \widehat{H}_{0.7}$.

Proof. The two hybrids differ only in the two abort checks introduced in $\widehat{H}_{0.7}$. Thus, it suffices to bound the abort probability.

First, recall that S is the union of the preimage sets of the $k = |\mathcal{I}_1|$ sampled oracle outputs $y_i = O_n(\delta_i)$. Hence,

$$\begin{aligned} |S| &= \left| \bigcup_{i \in \mathcal{I}_1} \mathsf{PreImage}^{O_n}(O_n(\delta_i)) \right| \\ &\leq \sum_{i \in \mathcal{I}_1} |\mathsf{PreImage}^{O_n}(O_n(\delta_i))|. \end{aligned}$$

Since $|\mathcal{I}_1| \leq p(n)$, the event $|S| \geq p(n) \log n$ implies that at least one of these preimage sets has size at least $\log n$.

Next, Corollary 3 also bounds the probability that each such preimage set has size at least $\log n$ by $\mathsf{negl}(n)$: take $t = \lceil \log n \rceil - 1$ in that calculation. Thus, by a union bound and Lem. 11,

$$\begin{aligned} \Pr[\mathsf{Abort}] &\leq \Pr[|S| \geq p(n) \log n] + \Pr\left[|\mathsf{Image}(O_n)| \leq 2^{n/2}\right] \\ &\leq \sum_{i \in \mathcal{I}_1} \Pr\left[|\mathsf{PreImage}^{O_n}(O_n(\delta_i))| \geq \log n\right] + \Pr\left[|\mathsf{Image}(O_n)| \leq 2^{n/2}\right] \\ &\leq |\mathcal{I}_1| \cdot \mathsf{negl}(n) + 2^{n2^{n/2} - n2^{n-1}} && \text{(Corollary 3+Lem. 11)} \\ &= \mathsf{negl}(n). \end{aligned}$$

We therefore conclude that $\widehat{H}_{0.6} \overset{s}{\approx} \widehat{H}_{0.7}$. □

Hybrid $\widehat{H}_{0.8}$: We present $\widehat{H}_{0.8}$ in Expr. 17 and highlight its difference from $\widehat{H}_{0.7}$ in red color. In this hybrid, for each $i \in \mathcal{I}_0$, we sample y_i uniformly from $\mathsf{Image}(O_n)$ instead of $\mathsf{Image}(F)$. When the abort checks pass, the two images differ by only a negligible fraction of their elements, giving $\widehat{H}_{0.7} \overset{s}{\approx} \widehat{H}_{0.8}$.

Experiment 17: Hybrid $\widehat{H}_{0.8}(\{r_i\}_{i \in [p(n)]})$

1. Sample $O_n \leftarrow \mathcal{O}_n$;
2. For $i \in \mathcal{I}_1$:
   - Sample $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$
3. Set $T := \{y_i\}_{i \in \mathcal{I}_1}$
4. Define $S := \bigcup_{i \in \mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i)$, $\overline{S} := \{0,1\}^n \setminus S$

5. Sample a function $F : \{0,1\}^n \to \{0,1\}^n$ where each entry independently:

$$F(x) = \begin{cases} \text{pick uniformly from } \{0,1\}^n \setminus T & \text{if } x \in S, \\ O_n(x) & \text{if } x \in \overline{S}. \end{cases}$$

6. Abort if $|S| \geq p(n)\log(n)$ or $|\mathsf{Image}(O_n)| \leq 2^{n/2}$
7. Initialize $W = \{\}$.
8. For $i \in \mathcal{I}_0$:
   - Sample $y_i \xleftarrow{\$} \mathsf{Image}(O_n)$
   - Sample $v_i \xleftarrow{\$} \mathsf{PreImage}^{O_n}(y_i)$
   - $W = W \cup \{(y_i, v_i)\}$
9. Output (F, T, W)

Claim 15. *It holds that* $\widehat{H}_{0.7} \overset{s}{\approx} \widehat{H}_{0.8}$.

Proof. First, the two hybrids have identical initial sampling and abort checks. In particular, the following variables have the same joint distribution:

$$O_n,\ T,\ F. \tag{15}$$

If either initial abort check is triggered, both hybrids abort. Otherwise, condition on fixed values of the variables in Expression (15). Define the following event:

– **Event** E: for all $i \in \mathcal{I}_0$, it holds that $y_i \in \mathsf{Image}(F) \cap \mathsf{Image}(O_n)$.

Next, conditioned on these variables and on E, both hybrids sample each y_i independently and uniformly from $\mathsf{Image}(F) \cap \mathsf{Image}(O_n)$, and then sample each v_i uniformly from $\mathsf{PreImage}^{O_n}(y_i)$. Their conditional output distributions are therefore identical.

It remains to bound the probability that E fails under each hybrid. Notice that F and O_n differ only on S; then, since the difference in their range must be induced by the difference in their domain, we have

$$|\mathsf{Image}(F) \setminus \mathsf{Image}(O_n)| \leq |S| \leq p(n)\log n \tag{16}$$
$$|\mathsf{Image}(O_n) \setminus \mathsf{Image}(F)| \leq |S| \leq p(n)\log n \tag{17}$$

where the $|S| \leq p(n)\log n$ bound follows from the common abort check at Step 6 in $\widehat{H}_{0.7}$ and $\widehat{H}_{0.8}$.

Moreover, the initial abort checks also ensure that

$$|\mathsf{Image}(O_n)| > 2^{n/2} \tag{18}$$
$$|\mathsf{Image}(F)| \geq |\mathsf{Image}(O_n)| - |S| \tag{19}$$
$$> 2^{n/2} - p(n)\log n \tag{20}$$

Thus, by a union bound,

$$\begin{aligned} \Pr_{\widehat{H}_{0.7}}[\neg E \mid O_n, T, F] &= \Pr_{\widehat{H}_{0.7}}[\exists i \in \mathcal{I}_0 :\ y_i \in \mathsf{Image}(F) \setminus \mathsf{Image}(O_n) \mid O_n, T, F] \\ &\leq \frac{|\mathcal{I}_0|\,|\mathsf{Image}(F) \setminus \mathsf{Image}(O_n)|}{|\mathsf{Image}(F)|} \end{aligned}$$

$$\leq \frac{|\mathcal{I}_0| p(n) \log n}{2^{n/2} - p(n) \log n}. \qquad \text{(Eq. (16) and Eq. (20))}$$

Similarly,

$$\begin{aligned}
\Pr_{\widehat{H}_{0.8}}[\neg E \mid O_n, T, F] &= \Pr_{\widehat{H}_{0.8}}[\exists i \in \mathcal{I}_0 : \ y_i \in \mathsf{Image}(O_n) \setminus \mathsf{Image}(F) \mid O_n, T, F] \\
&\leq \frac{|\mathcal{I}_0| \, |\mathsf{Image}(O_n) \setminus \mathsf{Image}(F)|}{|\mathsf{Image}(O_n)|} \\
&\leq \frac{|\mathcal{I}_0| p(n) \log n}{2^{n/2}}. \qquad \text{(Eq. (17) and Eq. (18))}
\end{aligned}$$

For each fixed nonaborting initial sample, the statistical distance is at most the sum of these two failure probabilities by Lem. 38. Averaging over the common initial randomness gives

$$\Delta(\widehat{H}_{0.7}, \widehat{H}_{0.8}) \leq \frac{|\mathcal{I}_0| p(n) \log n}{2^{n/2} - p(n) \log n} + \frac{|\mathcal{I}_0| p(n) \log n}{2^{n/2}} = \mathsf{negl}(n),$$

establishing $\widehat{H}_{0.7} \overset{s}{\approx} \widehat{H}_{0.8}$. □

Hybrid $\widehat{H}_{0.9}$: We present $\widehat{H}_{0.9}$ in Expr. 18 and highlight its difference from $\widehat{H}_{0.8}$ in red color.

Experiment 18: Hybrid $\widehat{H}_{0.9}(\{r_i\}_{i \in [p(n)]})$

1. Sample $O_n \leftarrow \mathcal{O}_n$;
2. For $i \in \mathcal{I}_1$:
   - Sample $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$
3. Set $T := \{y_i\}_{i \in \mathcal{I}_1}$
4. Define $S := \bigcup_{i \in \mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i)$, $\overline{S} := \{0,1\}^n \setminus S$
5. Sample a function $F : \{0,1\}^n \to \{0,1\}^n$ where each entry independently:

$$F(x) = \begin{cases} \text{pick uniformly from } \{0,1\}^n \setminus T & \text{if } x \in S, \\ O_n(x) & \text{if } x \in \overline{S}. \end{cases}$$

6. ~~Abort if $|S| \geq p(n) \log(n)$ or $|\mathsf{Image}(O_n)| \leq 2^{n/2}$~~
7. Initialize $W = \{\}$.
8. For $i \in \mathcal{I}_0$:
   - Sample $y_i \xleftarrow{\$} \mathsf{Image}(O_n)$
   - Sample $v_i \xleftarrow{\$} \mathsf{PreImage}^{O_n}(y_i)$
   - $W = W \cup \{(y_i, v_i)\}$
9. Output (F, T, W)

Claim 16. *It holds that $\widehat{H}_{0.8} \overset{s}{\approx} \widehat{H}_{0.9}$.*

Proof. The two hybrids differ only in the abort check removed in $\widehat{H}_{0.9}$. and the proof of $\widehat{H}_{0.6} \overset{s}{\approx} \widehat{H}_{0.7}$ already shows that this check triggers with negligible probability, establishing $\widehat{H}_{0.8} \overset{s}{\approx} \widehat{H}_{0.9}$. □

Hybrid $\widehat{H_1}$: We present $\widehat{H_1}$ in Expr. 19 and highlight its difference from $\widehat{H}_{0.9}$ in red color. In this hybrid, we express the sampling of the pairs (y_i, v_i) for $i \in \mathcal{I}_0$ as $W \sim K_O^{\otimes|\mathcal{I}_0|}$. This is purely a notational change and preserves the output distribution, giving $\widehat{H}_{0.9} \overset{\text{i.d.}}{\equiv} \widehat{H_1}$.

Experiment 19: Hybrid $\widehat{H_1}(\{r_i\}_{i\in[p(n)]})$

1. Sample $O_n \leftarrow \mathcal{O}_n$;
2. For $i \in \mathcal{I}_1$:
   - Sample $\delta_i \xleftarrow{\$} \{0,1\}^n, y_i = O_n(\delta_i)$
3. Set $T \coloneqq \{y_i\}_{i\in\mathcal{I}_1}$
4. Define $S \coloneqq \bigcup_{i\in\mathcal{I}_1} \mathsf{PreImage}^{O_n}(y_i)$, $\overline{S} \coloneqq \{0,1\}^n \setminus S$
5. Sample a function $F : \{0,1\}^n \to \{0,1\}^n$ where each entry independently:

$$F(x) = \begin{cases} \text{pick uniformly from } \{0,1\}^n \setminus T & \text{if } x \in S, \\ O_n(x) & \text{if } x \in \overline{S}. \end{cases}$$

6. Sample $W \sim K_{O_n}^{\otimes|\mathcal{I}_0|}$
7. Output (F, T, W)

Claim 17. *It holds that $\widehat{H}_{0.9} \overset{\text{i.d.}}{\equiv} \widehat{H_1}$.*

Proof. This is just a notational change,where we do not explicity write out the sampling of W in $\widehat{H}_{0.9}$ anymore. Therefore, the two distributions are identical. □

Completing the proof of $H_0 \overset{\mathbf{s}}{\approx} H_1$: By the previous hybrids indistinguishability claims and the triangle inequality, for every $\boldsymbol{r} = (r_i)_{i\in[p(n)]}$, we have

$$\Delta(\widehat{H}_0(\boldsymbol{r}), \widehat{H_1}(\boldsymbol{r})) = \mathsf{negl}(n),$$

These experiments only reorder independent sampling operations and replace the invocation of $\mathcal{A}$ by the output of its complete input and oracle table. Thus, applying the original adversary invocation to $\widehat{H_b}$ recovers H_b for $b \in \{0,1\}$. By Lem. 38, statistical distance cannot increase under this transformation. Then the above hybrids indistinguishability establishing $H_0 \overset{\mathrm{s}}{\approx} H_1$.

4.5 Proof of $H_1 \overset{\mathbf{s}}{\approx} H_2$

Proof Strategy Overview. We prove $H_1 \overset{\mathrm{s}}{\approx} H_2$ by establishing the stronger statement that the joint distributions of the complete constructed oracle O_n^* and the indexed inputs supplied to $\mathcal{A}$ are statistically close.

Recall that the two hybrids have the same initial sampling. For each $i \in \mathcal{I}_0$, they independently sample

$$y_i \xleftarrow{\$} \mathsf{Image}^{O_n}, \qquad v_i \xleftarrow{\$} \mathsf{PreImage}^{O_n}(y_i),$$

and, for each $i \in \mathcal{I}_1$, they independently sample

$$\delta_i \xleftarrow{\$} \{0,1\}^n, \qquad y_i \coloneqq O_n(\delta_i).$$

Their constructions of O_n^* differ only at inputs

$$z \notin \bigcup_{i\in[p(n)]} \mathsf{PreImage}^{O_n}(y_i).$$

At these inputs, H_1 retains $O_n(z)$, whereas H_2 independently samples a fresh uniform value from $\{0,1\}^n \setminus \{y_i : i \in [p(n)]\}$.

Intuitively, after fixing the sampled information and the preimage sets of the selected labels, the remaining oracle entries should already be uniform over the outputs other than the selected labels. However, sampling y_i uniformly from the image for each $i \in \mathcal{I}_0$ introduces the factor

$$\frac{1}{|\mathsf{Image}^{O_n}|^{|\mathcal{I}_0|}}.$$

The image size can depend on the remaining oracle entries, so these entries are not immediately independent and uniform.

To remove this dependence, we introduce modified experiments $\widetilde{H}_1$ and $\widetilde{H}_2$ in which each initial oracle table is sampled with probability proportional to $|\mathsf{Image}^{O_n}|^{|\mathcal{I}_0|}$. The two factors then cancel exactly, allowing us to prove $\widetilde{H}_1 \overset{\text{i.d.}}{\equiv} \widetilde{H}_2$. We next show that the modified oracle sampling is statistically close to uniform sampling, establishing $H_b \overset{\text{s}}{\approx} \widetilde{H}_b$ for each $b \in \{1,2\}$.

Throughout the proof, restrict attention to sufficiently large n such that $p(n) < 2^n$. We retain the notation

$$\mathcal{I}_0 = \{i \in [p(n)] : r_i = 0\}, \qquad \mathcal{I}_1 = \{i \in [p(n)] : r_i = 1\}.$$

The bit vector has the same distribution in both original hybrids and is independent of the initially sampled oracle. Thus, we may sample the bit vector first without changing either original experiment.

Hybrid $\widetilde{H}_1$. We present $\widetilde{H}_1$ in Expr. 20. This hybrid differs from H_1 only in the distribution of the initially sampled oracle. The modified sampling is used only for the proof and need not have an efficient implementation.

Experiment 20: Hybrid $\widetilde{H}_1$

1. According to the input $\mathbf{r} = (r_1, \ldots, r_{p(n)})$, define $\begin{cases} \mathcal{I}_0 \coloneqq \{i \in [p(n)] \mid r_i = 0\} \\ \mathcal{I}_1 \coloneqq \{i \in [p(n)] \mid r_i = 1\} \end{cases}$.
2. Sample $O_n \in \mathcal{O}_n$ according to

$$\Pr\left[O_n = f \mid \{r_i\}_{i\in[p(n)]}\right] = \frac{|\mathsf{Image}^{f}|^{|\mathcal{I}_0|}}{\sum_{g\in\mathcal{O}_n} |\mathsf{Image}^{g}|^{|\mathcal{I}_0|}} \qquad \text{for every } f \in \mathcal{O}_n.$$

3. For each $i \in \mathcal{I}_0$, independently sample

$$y_i \xleftarrow{\$} \mathsf{Image}^{O_n}, \qquad v_i \xleftarrow{\$} \mathsf{PreImage}^{O_n}(y_i).$$

4. For each $i \in \mathcal{I}_1$, independently sample

$$\delta_i \xleftarrow{\$} \{0,1\}^n, \qquad y_i \coloneqq O_n(\delta_i).$$

5. Construct O_n^* as in H_1. In particular, set

$$O_n^*(z) \coloneqq O_n(z) \qquad \text{if } z \notin \bigcup_{i\in[p(n)]} \mathsf{PreImage}^{O_n}(y_i).$$

All other branches of the oracle construction remain unchanged.

6. Invoke $\mathcal{A}$ with the same indexed inputs and oracle access as in H_1, and output its result.

Hybrid $\widetilde{H}_2$. We present $\widetilde{H}_2$ in Expr. 21. This hybrid uses the same modified initial sampling as $\widetilde{H}_1$, but resamples the oracle entries outside the selected preimage sets, as in H_2. Equivalently, it differs from H_2 only in the distribution of the initially sampled oracle.

Experiment 21: Hybrid $\widetilde{H}_2$

1. Run Steps 1–4 of $\widetilde{H}_1$ to obtain
$$O_n, \quad \{y_i\}_{i\in[p(n)]}, \quad \{v_i\}_{i\in\mathcal{I}_0}, \quad \{\delta_i\}_{i\in\mathcal{I}_1}.$$
2. Construct O_n^* as in H_2. In particular, for every input
$$z \notin \bigcup_{i\in[p(n)]} \mathsf{PreImage}^{O_n}(y_i),$$
independently sample
$$O_n^*(z) \xleftarrow{\$} \{0,1\}^n \setminus \{y_i : i \in [p(n)]\},$$
using fresh randomness. On all remaining inputs, use the same oracle construction as in $\widetilde{H}_1$.
3. Invoke $\mathcal{A}$ with the same indexed inputs and oracle access as in H_2, and output its result.

Claim 18. *It holds that $\widetilde{H}_1 \overset{i.d.}{\equiv} \widetilde{H}_2$.*

Proof. We prove that the complete constructed oracle O_n^* and the indexed inputs supplied to $\mathcal{A}$ have identical joint distributions in the two experiments.

First, the following variables have the same joint distribution in both experiments:
$$\{y_i\}_{i\in[p(n)]}, \qquad \{v_i\}_{i\in\mathcal{I}_0}, \qquad \{\delta_i\}_{i\in\mathcal{I}_1}, \qquad \left(\mathsf{PreImage}^{O_n}(y_i)\right)_{i\in[p(n)]}. \tag{21}$$
Condition on any attainable values of these variables.

Consider an oracle table f consistent with the fixed values in Expression (21). Conditional on $O_n = f$, the probability of sampling the specified labels, witnesses, and inputs is
$$2^{-n|\mathcal{I}_1|} \prod_{i\in\mathcal{I}_0} \frac{1}{|\mathsf{Image}^f|\,|\mathsf{PreImage}^f(y_i)|}.$$
Indeed, for each $i \in \mathcal{I}_1$, sampling the specified δ_i has probability 2^{-n}, after which y_i is determined by $f(\delta_i)$. For each $i \in \mathcal{I}_0$, the two denominator terms correspond to sampling y_i uniformly from the image and then v_i uniformly from its preimage set.

Multiplying by the modified probability of sampling the table f gives
$$\begin{aligned}
&\frac{|\mathsf{Image}^f|^{|\mathcal{I}_0|}}{\sum_{g\in\mathcal{O}_n} |\mathsf{Image}^g|^{|\mathcal{I}_0|}} \cdot 2^{-n|\mathcal{I}_1|} \prod_{i\in\mathcal{I}_0} \frac{1}{|\mathsf{Image}^f|\,|\mathsf{PreImage}^f(y_i)|} \\
&\qquad = \frac{2^{-n|\mathcal{I}_1|}}{\sum_{g\in\mathcal{O}_n} |\mathsf{Image}^g|^{|\mathcal{I}_0|}} \prod_{i\in\mathcal{I}_0} \frac{1}{|\mathsf{PreImage}^f(y_i)|}.
\end{aligned}$$

The factors involving $|\mathsf{Image}^f|$ cancel exactly. Moreover, the preimage sizes in the remaining expression are fixed by Expression (21). Thus, every consistent oracle table has the same conditional probability.

Next, the fixed preimage sets determine the oracle value at every input in

$$\bigcup_{i\in[p(n)]} \mathsf{PreImage}^{O_n}(y_i).$$

At any other input, the oracle value must lie in $\{0,1\}^n \setminus \{y_i : i \in [p(n)]\}$. Conversely, any assignment of the remaining oracle entries to values in this set preserves all the fixed information in Expression (21).

Consequently, conditional on the fixed information, the entries

$$\left\{ O_n(z) : z \notin \bigcup_{i\in[p(n)]} \mathsf{PreImage}^{O_n}(y_i) \right\}$$

are independent and uniformly distributed over $\{0,1\}^n \setminus \{y_i : i \in [p(n)]\}$. This conclusion does not require the labels y_i to be distinct.

Therefore, $\widetilde{H}_1$ retains entries that already have exactly the distribution from which $\widetilde{H}_2$ resamples. On the remaining inputs, both experiments use the same oracle construction, which depends only on the fixed information and fresh randomness. Their complete constructed oracle tables and indexed inputs therefore have identical conditional distributions.

Averaging over Expression (21) and then over the common distribution of $\{r_i\}_{i\in[p(n)]}$ preserves this identity. Applying the common adversary invocation establish $\widetilde{H}_1 \overset{\text{i.d.}}{\equiv} \widetilde{H}_2$.

Notice that the identity concerns the constructed oracle O_n^* supplied to $\mathcal{A}$. We do not additionally reveal the original table O_n. □

Claim 19. *For every $b \in \{1,2\}$, it holds that $H_b \overset{s}{\approx} \widetilde{H}_b$.*

Proof. Consider an arbitrary $b \in \{1,2\}$. The experiments H_b and $\widetilde{H}_b$ differ only in the distribution of the initially sampled oracle. Thus, it suffices to bound the statistical distance between uniform oracle sampling and the modified sampling.

Intuitively, the modified sampling assigns larger weights to oracle tables with larger images. However, almost all uniformly sampled tables have nearly the same image size, so their sampling weights are nearly equal.

Let

$$\mu \coloneqq \mathbb{E}_{O_n\leftarrow\mathcal{O}_n}\left[|\mathsf{Image}^{O_n}|\right].$$

By Lem. 12, we have $\mu \geq 2^{n-1}$ and

$$\Pr_{O_n\leftarrow\mathcal{O}_n}\left[\left||\mathsf{Image}^{O_n}| - \mu\right| > 2^{3n/4}\right] \leq 2e^{-2\cdot 2^{n/2}}. \tag{22}$$

We use this concentration bound to compare the weights of the typical tables and separately bound the total contribution of the exceptional tables to the normalization.

For the following calculation, define the rescaled weight of a table f by

$$w(f) \coloneqq \left(\frac{|\mathsf{Image}^f|}{\mu}\right)^{|\mathcal{I}_0|}.$$

Dividing all weights by the same number $\mu^{|\mathcal{I}_0|}$ does not change the modified sampling distribution.

First, consider a table f satisfying

$$\Big||\mathsf{Image}^f| - \mu\Big| \le 2^{3n/4}.$$

Since $\mu \ge 2^{n-1}$, we have

$$1 - 2^{1-n/4} \le \frac{|\mathsf{Image}^f|}{\mu} \le 1 + 2^{1-n/4}.$$

Consequently,

$$(1 - 2^{1-n/4})^{|\mathcal{I}_0|} \le w(f) \le (1 + 2^{1-n/4})^{|\mathcal{I}_0|}.$$

Using $|\mathcal{I}_0| \le p(n)$, together with $(1-x)^k \ge 1 - kx$ and $(1+x)^k \le e^{kx}$, gives

$$|w(f) - 1| = O\big(p(n)2^{-n/4}\big). \tag{23}$$

Here we use $p(n)2^{-n/4} = o(1)$. The bound is uniform over all such tables and all choices of the bit vector.

Next, for every oracle table, including those outside the preceding image-size bound, we have

$$0 \le w(f) = \left(\frac{|\mathsf{Image}^f|}{\mu}\right)^{|\mathcal{I}_0|} \le \left(\frac{2^n}{\mu}\right)^{|\mathcal{I}_0|} \le 2^{p(n)}.$$

where the last inequality uses $\mu \ge 2^{n-1}$. Thus, the exceptional tables have very small probability, and even their largest possible weights have negligible total contribution. More precisely, combining Eq. (22) and (23) gives

$$\begin{aligned}\mathbb{E}_{O_n \leftarrow \mathcal{O}_n}\big[|w(O_n) - 1|\big] &\le O\big(p(n)2^{-n/4}\big) + (2^{p(n)} + 1) \cdot 2e^{-2\cdot 2^{n/2}} \\ &= \mathsf{negl}(n).\end{aligned} \tag{24}$$

It remains to account for the normalization of the weights. Let $\varepsilon(n)$ be a negligible upper bound on Equation (24), uniform over the bit vector. Then by triangle inequality, we have

$$|\mathbb{E}_{O_n \leftarrow \mathcal{O}_n}[w(O_n)] - 1| \le \mathbb{E}_{O_n \leftarrow \mathcal{O}_n}[|w(O_n) - 1|] \le \varepsilon(n).$$

In particular, the average weight is also close to one.

Since $|\mathcal{O}_n| = 2^{n\cdot 2^n}$, the probability of a particular table f under the modified sampling is

$$\frac{w(f)}{\sum_{g \in \mathcal{O}_n} w(g)} = \frac{1}{2^{n\cdot 2^n}} \cdot \frac{w(f)}{\mathbb{E}_{O_n \leftarrow \mathcal{O}_n}[w(O_n)]}.$$

Therefore, the statistical distance between uniform and modified oracle sampling is

$$\begin{aligned}&\frac{1}{2}\sum_{f \in \mathcal{O}_n}\left|\frac{1}{2^{n\cdot 2^n}} - \frac{1}{2^{n\cdot 2^n}} \cdot \frac{w(f)}{\mathbb{E}_{O_n \leftarrow \mathcal{O}_n}[w(O_n)]}\right| \\ &\quad= \frac{\mathbb{E}_{O_n \leftarrow \mathcal{O}_n}\left[|w(O_n) - \mathbb{E}_{O_n \leftarrow \mathcal{O}_n}[w(O_n)]|\right]}{2\mathbb{E}_{O_n \leftarrow \mathcal{O}_n}[w(O_n)]} \\ &\quad\le \frac{\mathbb{E}_{O_n \leftarrow \mathcal{O}_n}[|w(O_n) - 1|] + |1 - \mathbb{E}_{O_n \leftarrow \mathcal{O}_n}[w(O_n)]|}{2\mathbb{E}_{O_n \leftarrow \mathcal{O}_n}[w(O_n)]}\end{aligned}$$

$$\leq \frac{\varepsilon(n)}{1-\varepsilon(n)} = \mathsf{negl}(n).$$

Finally, for each fixed bit vector and each $b \in \{1,2\}$, all subsequent steps of H_b and $\widetilde{H}_b$ apply the same randomized transformation to the initially sampled oracle. Since statistical distance cannot increase under such a transformation, the joint distributions of the constructed oracle and the indexed inputs are statistically close. The same holds after invoking $\mathcal{A}$.

The bound is uniform over $\{r_i\}_{i\in[p(n)]}$. Averaging over its common distribution therefore gives $H_b \overset{\mathrm{s}}{\approx} \widetilde{H}_b$ for every $b \in \{1,2\}$.

□

Completing the proof of $H_1 \overset{\mathbf{s}}{\approx} H_2$. By Claims 18 and 19, we have

$$H_1 \overset{\mathrm{s}}{\approx} \widetilde{H}_1 \overset{\text{i.d.}}{=\!=} \widetilde{H}_2 \overset{\mathrm{s}}{\approx} H_2.$$

The triangle inequality therefore gives

$$\Delta(H_1, H_2) \leq \Delta(H_1, \widetilde{H}_1) + \Delta(\widetilde{H}_1, \widetilde{H}_2) + \Delta(\widetilde{H}_2, H_2) = \mathsf{negl}(n),$$

establishing $H_1 \overset{\mathrm{s}}{\approx} H_2$.

4.6 Proof of $H_2 \overset{\mathbf{c_q}}{\approx} H_{2.5}$

Below, we establishes $H_2 \overset{\mathrm{c_q}}{\approx} H_{2.5}$.

Proof strategy overview. We couple the two experiments so that their constructed oracles agree outside a small set of inputs. We then show that, in $H_{2.5}$, the locations of these inputs remain hidden even given the complete constructed oracle and the sampled information. Consequently, measuring a query in an execution with the $H_{2.5}$ oracle hits this set with only negligible probability. By the one-way-to-hiding (O2H) bound, a non-negligible distinguishing advantage would contradict this query bound.

Fix a quantum polynomial-time adversary $\mathcal{A}$, and let $q = q(n) \geq 1$ be a polynomial upper bound on its number of oracle queries. Write $p = p(n)$. For each $i \in \mathcal{I}_0$, write $\delta_i = v_i$, and define

$$W \coloneqq \big((y_i, \delta_i)\big)_{i\in[p]}, \qquad D \coloneqq \{\delta_i : i \in [p]\}, \qquad Y \coloneqq \{y_i : i \in [p]\}.$$

We also write

$$Y_1 \coloneqq \{y_i : i \in \mathcal{I}_1\}.$$

The variable W contains all the indexed inputs supplied to $\mathcal{A}$, as well as the designated inputs δ_i for $i \in \mathcal{I}_1$. We condition on these additional inputs only for the analysis; the experiments do not reveal them to $\mathcal{A}$.

Restricting to the good event. For an oracle table f, define

$$K(f) \coloneqq \max_{y\in\{0,1\}^n} |\mathsf{PreImage}^f(y)|, \qquad M(f) \coloneqq |\mathsf{Image}^f|.$$

Consider the event

$$B \coloneqq \big[\text{all } y_i \text{ are distinct}\big] \cap \big[K(O_n) \leq n\big] \cap \big[M(O_n) \geq 2^{n-1}\big].$$

The distinctness bound established in the proof of Claim 4 applies here, since H_2 and $H_{2.5}$ have the same initial sampling. Combining this bound with Lem. 12 and the preimage-size bound obtained from Lem. 13 and Corollary 6 by a union bound over all 2^n inputs, we obtain

$$\Pr[\neg B] = \mathsf{negl}(n).$$

The event B concerns only the common initial sampling, so it has the same probability in both experiments.

For now, condition both experiments on B. Since $O_n(\delta_i) = y_i$ for every $i \in [p]$, distinctness of the labels implies distinctness of the designated inputs. Thus, $|D| = p$. Define the hidden set

$$S := \left(\bigcup_{i \in [p]} \mathsf{PreImage}^{O_n}(y_i) \right) \setminus D.$$

On B, we have

$$|S| \leq pn. \tag{25}$$

Coupling the constructed oracles. Let $O^*_{n,2}$ and $O^*_{n,2.5}$ denote the constructed oracles in H_2 and $H_{2.5}$, respectively. Use the same initial oracle O_n and the same sampled data W in both experiments. Couple the remaining oracle randomness as follows:

- At every δ_i with $i \in \mathcal{I}_1$, use the same fresh uniform output from $\{0,1\}^n \setminus Y_1$.
- At every δ_i with $i \in \mathcal{I}_0$, both constructed oracles output y_i.
- At every input outside $\bigcup_{i \in [p]} \mathsf{PreImage}^{O_n}(y_i)$, use the same fresh uniform output from $\{0,1\}^n \setminus Y$.
- At the remaining inputs, sample the oracle values according to their respective prescribed distributions.

All fresh samples at distinct inputs are independent within each oracle construction. Since the labels are distinct on B, the designated-input cases do not conflict. This coupling therefore preserves both conditional experiment distributions and satisfies

$$O^*_{n,2}(z) = O^*_{n,2.5}(z) \qquad \text{for every } z \notin S. \tag{26}$$

The agreement outside the selected preimage sets uses the fact that H_2 already resamples its background entries uniformly from $\{0,1\}^n \setminus Y$.

The hidden set remains hidden in $H_{2.5}$. Fix any attainable value of W under the conditioning on B. The complete oracle $O^*_{n,2.5}$ can be generated from this fixed information and independent fresh randomness: at each designated input its distribution is determined by W, and at every other input it is independently uniform over $\{0,1\}^n \setminus Y$. In particular, the construction does not otherwise use the initial oracle O_n.

It follows that

$$O^*_{n,2.5} \text{ and } S \text{ are conditionally independent given } (W, B). \tag{27}$$

This remains true despite conditioning on B, since B depends only on the initial sampling, not on the fresh randomness used to construct $O^*_{n,2.5}$.

Intuitively, after fixing W and conditioning on B and $|S| = s$, every s-element subset of $\{0,1\}^n \setminus D$ should be equally likely to be S. We justify this symmetry by reshuffling the truth table outside D, preserving the sampling probabilities while permuting the locations of S.

Formally, consider any permutation π of $\{0,1\}^n$ that fixes every point in D, and replace the initial oracle table f by $f \circ \pi^{-1}$. Uniform oracle sampling is invariant under this transformation. Moreover, it preserves the image of f, the cardinality of every preimage set, and the values at all designated inputs. Consequently, the probability of generating the fixed data W is unchanged. Indeed, for $i \in \mathcal{I}_0$, sampling y_i uniformly from the image and then δ_i uniformly from its preimage has the same probability before and after the transformation. For $i \in \mathcal{I}_1$, the same holds for sampling δ_i uniformly from $\{0,1\}^n$ and setting $y_i = f(\delta_i)$.

The event B is also invariant under this transformation, whereas the hidden set S is mapped to $\pi(S)$. Consequently, for every attainable s,

$$S \mid (W, B, |S| = s) \quad \text{is uniform among the } s\text{-subsets of } \{0,1\}^n \setminus D. \tag{28}$$

Indeed, permutations fixing D act transitively on these s-subsets.

Combining Eq. (27) and (28), we conclude that conditioning additionally on the complete table $O^*_{n,2.5}$ does not change this uniform distribution. Thus, even the complete accessible oracle does not reveal the locations of S outside D.

Bounding the probability of a measured query. Consider an execution of $\mathcal{A}$ with oracle $O^*_{n,2.5}$ and its prescribed indexed inputs. For each $j \in [q]$, consider the experiment that runs $\mathcal{A}$ until immediately before its j-th oracle query and measures that query's input register in the computational basis. An execution making fewer than j queries is treated as a failure to hit S.

Fix attainable values of

$$W, \quad O^*_{n,2.5}, \quad |S| = s$$

under the conditioning on B. Let $\mu_j(z)$ denote the probability that the measured query is z under this conditioning. The adversary's execution uses only its indexed inputs, the oracle $O^*_{n,2.5}$, and its own randomness. Hence its query distribution does not depend further on S.

By the conditional uniformity established above,

$$\begin{aligned}\Pr\left[\text{measured query } j \in S \,\middle|\, W, O^*_{n,2.5}, B, |S| = s\right] \\ = \sum_{z \notin D} \mu_j(z) \frac{s}{2^n - p} \le \frac{s}{2^n - p} \le \frac{pn}{2^n - p}.\end{aligned}$$

The last inequality follows from Eq. (25). Averaging over the conditioned variables preserves the bound.

Let P_{hit} be the probability of obtaining an element of S when choosing j uniformly from $[q]$ and performing the preceding measurement in the $H_{2.5}$ execution, with the initial sampling conditioned on B. Then

$$P_{\text{hit}} \le \frac{pn}{2^n - p}. \tag{29}$$

Applying O2H. Let $\mathcal{A}[H_b]$ denote the output of $\mathcal{A}$ in experiment H_b, and define the conditional distinguishing advantage

$$\varepsilon_B := \left|\Pr[\mathcal{A}[H_2] = 1 \mid B] - \Pr[\mathcal{A}[H_{2.5}] = 1 \mid B]\right|.$$

The coupled oracles agree outside S by Eq. (26). The O2H bound, with measured queries taken from the execution using $O^*_{n,2.5}$, therefore gives

$$\varepsilon_B \le 2q\sqrt{P_{\text{hit}}}. \tag{30}$$

The bound permits arbitrary correlations among the two oracle tables, the set S, and the auxiliary input. The conditional independence proved above is needed to bound P_{hit}, not to apply O2H itself.

Removing the conditioning. Finally, the two experiments have the same initial sampling distribution and therefore the same probability of B. Decomposing their acceptance probabilities according to B and $\neg B$, we obtain

$$\begin{aligned}
&|\Pr[\mathcal{A}[H_2] = 1] - \Pr[\mathcal{A}[H_{2.5}] = 1]| \\
&\quad \le \Pr[B]\varepsilon_B + \Pr[\neg B] \\
&\quad \le 2q(n)\sqrt{\frac{p(n)n}{2^n - p(n)}} + \mathsf{negl}(n) = \mathsf{negl}(n).
\end{aligned}$$

Since $\mathcal{A}$ was arbitrary, this establishes $H_2 \overset{\text{c}_\text{q}}{\approx} H_{2.5}$.

This completes the proof of Claim 3.

5 NIZK in the URS-PRHF Model

In this section, we construct a post-quantum NIZK proof from a strong δ-dense post-quantum PRHF, using a uniform random string (URS) as setup.

This construction is not a URS NIZK in the standard sense: both the prover and the verifier additionally require access to an honestly generated (strong δ-dense post-quantum) PRHF as per Def. 6. The PRHF sampling algorithm samples a function H from a family $\mathcal{H}_n$. If this algorithm is public-coin, its random coins can be included in the URS. However, future potential instantiations of PRHFs from concrete assumptions may require private-coin sampling. In that case, our construction does not yield a URS NIZK in the standard sense. We therefore formalize the notion achieved by our construction in Def. 7: in addition to a standard URS, the prover and the verifier are given access to an honestly generated PRHF. We call this the *URS-PRHF model*.

This model serves only as an intermediate step toward our final construction in the QROM in Sec. 6, where we implement both the URS and the PRHF using a random oracle. The URS-PRHF model suffices for this purpose.

Definition 7 (NIZK Proof in the URS-PRHF Model). *Let $\{\mathcal{H}_n\}_{n\in\mathbb{N}}$ be a (strong δ-dense post-quantum) PRHF as per Def. 6. An (unbounded-prover) non-interactive zero knowledge (NIZK) proof system in the URS-PRHF model for a language $\mathcal{L}$ is a tuple of algorithms $\Pi = (P, V)$. The setup consists of a uniform random string $\sigma \xleftarrow{\$} \{0,1\}^{\ell(n)}$, where ℓ is a polynomial, and an independently sampled PRHF $H_n \leftarrow \mathcal{H}_n$, where $n = |x|$.*

- *$P(H_n, \sigma, x)$ is an (unbounded, randomized) honest prover algorithm that takes as input the code of H_n, the URS σ, and $x \in \mathcal{L}$, and outputs a proof π.*
- *$V(H_n, \sigma, x, \pi)$ is a polynomial-time deterministic verifier that takes as input the code of H_n, the URS σ, x, and a proof π, and outputs either 1 (accept) or 0 (reject).*

We require Π to satisfy the following completeness, soundness, and zero knowledge requirements:

- ***Perfect Completeness:** For all $x \in \mathcal{L}$ with $|x| = n$, it holds that*

$$\Pr\left[V(H_n, \sigma, x, \pi) = 1 : \begin{array}{r} H_n \leftarrow \mathcal{H}_n, \\ \sigma \xleftarrow{\$} \{0,1\}^{\ell(n)}, \\ \pi \leftarrow P(H_n, \sigma, x) \end{array}\right] = 1.$$

– ***Statistical Soundness:*** *It holds that*

$$\Pr_{\substack{H_n \leftarrow \mathcal{H}_n \\ \sigma \xleftarrow{\$} \{0,1\}^{\ell(n)}}} [\exists (x,\pi) \text{ such that } x \notin \mathcal{L},\ |x| = n,\ \text{and } V(H_n, \sigma, x, \pi) = 1] \leq \mathsf{negl}(n).$$

– ***Zero Knowledge:*** *There exists a QPT simulator* Sim *such that, for all $x \in \mathcal{L}$ with $|x| = n$ and all QPT algorithms $\mathcal{A}$,*

$$\left| \Pr\left[\mathcal{A}(H_n, \sigma, x, \pi) = 1 : \begin{array}{l} H_n \leftarrow \mathcal{H}_n,\ \sigma \xleftarrow{\$} \{0,1\}^{\ell(n)}, \\ \pi \leftarrow P(H_n, \sigma, x) \end{array}\right] \right.$$
$$\left. - \Pr\left[\mathcal{A}(H_n, \sigma^*, x, \pi^*) = 1 : \begin{array}{l} H_n \leftarrow \mathcal{H}_n, \\ (\sigma^*, \pi^*) \leftarrow \mathrm{Sim}(H_n, x) \end{array}\right] \right| \leq \mathsf{negl}(n).$$

The simulator is given the code of the honestly sampled H_n and simulates the URS and the proof, without modifying H_n.

5.1 Construction

In this subsection, we present a construction satisfying Def. 7. The construction and its security proof also apply when H_{PRHF} is instantiated by a random oracle: The proof relativizes by replacing each evaluation of H_{PRHF} with an oracle query. We use this oracle formulation in the QROM construction of Sec. 6.5.

We first recall the following existence result.

Lemma 16 ([GIK+23, Theorem 5.1]). *For every language $L \in \mathsf{NP}$ and every efficiently approximable constant $\gamma \in (0,1)$, there exists a NIZK proof system in the Z-Tamperable-Hidden-Bits model,*

$$\mathsf{ZHB\text{-}NIZK}_\gamma = \left(\mathsf{ZHB.Gen}_\gamma,\ \mathsf{ZHB.P}_\gamma,\ \mathsf{ZHB.V}_\gamma\right),$$

with simulator $\mathsf{ZHB.Sim}_\gamma$, satisfying the requirements of Def. 3.

The construction closely follows the framework of [GIK+23]. The main differences are that we instantiate it with our strong δ-dense post-quantum PRHF in place of the δ-dense hash function used there, and establish security against quantum adversaries.

Algorithm 5.1: URS Generator $\mathsf{URS.Gen}(1^n)$

$\mathsf{URS.Gen}(1^n)$:

1. For $i \in [p_{\mathsf{hbs}}(n)]$, sample a uniformly random $y_i \in \{0,1\}^{\ell(n)}$
2. Output $\mathsf{urs} = y_1 \cdots y_{p_{\mathsf{hbs}}(n)}$

Algorithm 5.2: URS Prover $\mathsf{URS.P}$

$\mathsf{URS.P}(\mathsf{urs}, H_{\mathsf{PRHF}}, x)$:

1. Parse urs as $y_1, \ldots, y_{p_{\mathsf{hbs}}(n)}$ where $n = |x|$.
2. For $i \in [p_{\mathsf{hbs}}(n)]$,
   (a) If $\exists z \in \{0,1\}^n$ such that $H_{\mathsf{PRHF}}(z) = y_i$, sample $v_i \xleftarrow{\$} \{z \in \{0,1\}^n : H_{\mathsf{PRHF}}(z) = y_i\}$, and set $r_i = 0$.
   (b) If $\nexists z \in \{0,1\}^n$ such that $H_{\mathsf{PRHF}}(z) = y_i$, set $v_i = \perp$ and $r_i = 1$.

3. $(\mathsf{ZHB.I}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB.P}_\gamma(r, x)$ where $r = r_1 \dots r_{p_{\mathsf{hbs}}(n)}$.
4. Return $\pi := (\mathsf{ZHB.I}, \{v_i\}_{i \in \mathsf{ZHB.I}}, \mathsf{ZHB}.\pi)$.

Algorithm 5.3: URS Verifier $\mathsf{URS.V}$

$\mathsf{URS.V}(\mathsf{urs}, H_{\mathsf{PRHF}}, x, \pi)$:

1. Parse urs as $y_1, \dots, y_{p_{\mathsf{hbs}}(n)}$ where $n = |x|$.
2. Parse π as $(\mathsf{ZHB.I}, \{v_i\}_{i \in \mathsf{ZHB.I}}, \mathsf{ZHB}.\pi)$.
3. For $i \in \mathsf{ZHB.I}$,
   - If $v_i = \perp$, set $\widetilde{r}_i = 1$.
   - If $v_i \neq \perp$, and $H_{\mathsf{PRHF}}(v_i) = y_i$, set $\widetilde{r}_i = 0$.
   - If $v_i \neq \perp$, and $H_{\mathsf{PRHF}}(v_i) \neq y_i$, reject.
4. Return $\mathsf{ZHB.V}_\gamma(x, \mathsf{ZHB.I}, \{\widetilde{r}_i\}_{i \in \mathsf{ZHB.I}}, \mathsf{ZHB}.\pi)$

Throughout the following analysis, let $L \in \mathsf{NP}$ denote the language for which the underlying $\mathsf{ZHB\text{-}NIZK}_\gamma$ is defined. The constructed $\mathsf{URS\text{-}NIZK}$ proves membership in the same language L, and we write $n = |x|$ for the length of a statement x.

5.2 Security Proof

Note that the completeness and soundness of the above NIZK proof system directly follows from [GIK+23] as the quantum-accessibility of zero-knowledge adversary will not affect the the above two properties, we will then focus on the zero-knowledge property.

Lemma 17. *If H_{PRHF} is a δ-dense post-quantum PRHF for some* $\mathsf{Efficiently\text{-}Approximable}$ *constant $\delta \in (0,1)$ and $\mathsf{ZHB.NIZK}^\gamma$ with $\gamma = 1 - \delta$ satisfies statistical zero knowledge in the Z-Tamperable-Hidden-Bits model, then there exists a QPT simulator* $\mathsf{URS.Sim}$ *and for all $x \in \mathcal{L}$ and all non-uniform QPT adversary $\mathcal{A}$,*

$$\left| \Pr\left[\mathcal{A}(\mathrm{urs}, H_{\mathsf{PRHF}}, x, \pi) = 1 \ : \ \begin{array}{l} H_{\mathsf{PRHF}} \leftarrow \mathcal{H}_n; \\ \mathrm{urs} \leftarrow \mathsf{URS.Gen}(1^{|x|}); \\ \pi \leftarrow \mathsf{URS.P}(\mathrm{urs}, H_{\mathsf{PRHF}}, x) \end{array} \right] - \Pr\left[\mathcal{A}(\mathrm{urs}^*, H_{\mathsf{PRHF}}, x, \pi^*) = 1 \ : \ \begin{array}{l} H_{\mathsf{PRHF}} \leftarrow \mathcal{H}_n; \\ (\mathrm{urs}^*, \pi^*) \leftarrow \mathrm{URS.Sim}(H_{\mathsf{PRHF}}, x); \end{array} \right] \right| \leq \mathsf{negl}(n).$$

Proof. We complete the proof by giving the explicit construction of the simulator $\mathsf{URS.Sim}$ and proving that it is indistinguishable from the real prover $\mathsf{URS.P}$ through hybrid arguments.

Algorithm 5.4: URS Simulator $\mathsf{URS.Sim}$

$\mathsf{URS.Sim}(H_{\mathsf{PRHF}}, x)$:

1. Let $n = |x|$.
2. $(\mathsf{ZHB.I}, \{r_i\}_{i \in \mathsf{ZHB.I}}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB.Sim}_\gamma(x)$.
3. For $i \in [p_{\mathsf{hbs}}(n)]$, sample $v_i \leftarrow \{0,1\}^n$, and set $y_i = H_{\mathsf{PRHF}}(v_i)$.
4. For $i \in \mathsf{ZHB.I}$,
   (a) If $r_i = 1$, set $v'_i = \perp$

(b) If $r_i = 0$, set $v'_i = v_i$.

5. Output $(\mathsf{urs} = (y_1 \dots y_{p_{\mathsf{hbs}}(n)}), \pi = (\mathsf{ZHB.I}, \{v'_i\}_{i \in \mathsf{ZHB.I}}, \mathsf{ZHB}.\pi))$.

Hybrid H_0: This is the real-world game where the adversary interacts with the real prover.We formally describe it in Expr. 22.

Experiment 22: Hybrid H_0

1. Let $n = |x|$.
2. For $i \in [p_{\mathsf{hbs}}(n)]$,
   (a) Sample $y_i \xleftarrow{\$} \{0,1\}^{\ell(n)}$.
   (b) If $\exists z \in \{0,1\}^n$ such that $H_{\mathsf{PRHF}}(z) = y_i$, and sample $v_i \xleftarrow{\$} \{z \in \{0,1\}^n : H_{\mathsf{PRHF}}(z) = y_i\}$, and set $r_i = 0$.
   (c) If $\nexists z \in \{0,1\}^n$ such that $H_{\mathsf{PRHF}}(z) = y_i$, set $v_i = \perp$ and $r_i = 1$.
3. $(\mathsf{ZHB}.\mathcal{I}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB.P}_\gamma(r, x)$ where $r = r_1 \dots r_{p_{\mathsf{hbs}}(n)}$.
4. Output $(\mathsf{urs} = (y_1 \dots y_{p_{\mathsf{hbs}}(n)}), \pi = (\mathsf{ZHB}.\mathcal{I}, \{v_i\}_{i \in \mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi))$.

Hybrid: H_1 We present H_1 in Expr. 23 and highlight its difference from H_0 in red color. In this hybrid, we just interchange the order of sampling y_i and sampling v_i for those $r_i = 0$ without changing the distribution.

Experiment 23: Hybrid H_1

1. Let $n = |x|$.
2. For $i \in [p_{\mathsf{hbs}}(n)]$,
   (a) Sample $r_i \in \{0,1\}$ such that $\Pr[r_i = 1] = \gamma'$ where $\gamma' = 1 - \Pr_{y \leftarrow \{0,1\}^{\ell(n)}}[\exists z \text{ s.t. } H_{\mathsf{PRHF}}(z) = y]$
3. $(\mathsf{ZHB}.\mathcal{I}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB.P}_\gamma(r, x)$ where $r = r_1 \dots r_{p_{\mathsf{hbs}}(n)}$.
4. Define $\begin{cases} \mathsf{Image} := \{y \in \{0,1\}^{\ell(n)} : \exists z \in \{0,1\}^n \text{ such that } H_{\mathsf{PRHF}}(z) = y\} \\ \mathsf{PreImage}(y) := \{z \in \{0,1\}^n : H_{\mathsf{PRHF}}(z) = y\} \end{cases}$
5. For $i \in [p_{\mathsf{hbs}}(n)]$,
   (a) If $r_i = 0$, sample $y_i \leftarrow \mathsf{Image}$ and $v_i \leftarrow \mathsf{PreImage}(y_i)$, and set $v'_i = v_i$.
   (b) If $r_i = 1$, sample $y_i \leftarrow \{0,1\}^{\ell(n)} \setminus \mathsf{Image}$, and set $v'_i = \perp$.
6. Output $(\mathsf{urs} = (y_1 \dots y_{p_{\mathsf{hbs}}(n)}), \pi = (\mathsf{ZHB}.\mathcal{I}, \{v'_i\}_{i \in \mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi))$.

Claim 20. *It holds that $H_0 \overset{i.d.}{=\!=} H_1$.*

Proof. The above hybrid is the same as the previous hybrid, except that we have changed the way of generating the (r_i, v_i, y_i).

Observe the following statistical facts in hybrid H_1:

– The distribution of $(\mathsf{ZHB}.\mathcal{I}, \mathsf{ZHB}.\pi)$ only depends on the random r and the input x.

– For any $i \neq j$, the distribution of (r_i, v_i, y_i) is independently and identically distributed as (r_j, v_j, y_j).

– For any i, we have $\Pr[r_i = 1] = \Pr_{y \xleftarrow{\$} \{0,1\}^{\ell(n)}}[\nexists z \in \{0,1\}^n \text{ such that } H_{\mathsf{PRHF}}(z) = y]$.

– For any i, conditioned on $r_i = 0$, y_i is uniformly distributed over all elements in the image of H_{PRHF}, and v_i is uniformly distributed over all pre-images of y_i.

– For any i, conditioned on $r_i = 1$, y_i is uniformly distributed over all elements that are not in the image of H_{PRHF}, and $v_i = \perp$.

Therefore, it is identical for us to first sample each r_i based on the distribution of H_{PRHF}, and then sample y_i and v_i based on conditional probability induced by r_i. Also, as $(\mathsf{ZHB}.\mathcal{I}, \mathsf{ZHB}.\pi)$ only depends on r, as long as the distribution of r is the same, the distribution of final output for $(\mathsf{ZHB}.\mathcal{I}, \mathsf{ZHB}.\pi)$ will also be the same. □

Hybrid: H_2 We present H_2 in Expr. 24 and highlight its difference from H_1 in red color. In this hybrid, we replace the exact sampling probability of r_i with its approximation, changing the resulting distribution by only a negligible statistical distance.

Experiment 24: Hybrid H_2

1. Let $n = |x|$.
2. For $i \in [p_{\mathsf{hbs}}(n)]$,
   (a) Sample $r_i \in \{0,1\}$ such that $\Pr[r_i = 1] = 1 - \mathsf{Approx}_\delta(1^n)$
   Notice that this change reflects the δ-density property in Property 1, with δ replaced by its efficiently computable approximation $\mathsf{Approx}_\delta(1^n)$ from Def. 2.
3. $(\mathsf{ZHB}.\mathcal{I}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB}.\mathsf{P}_\gamma(r, x)$ where $r = r_1 \dots r_{p_{\mathsf{hbs}}(n)}$.
4. Define $\begin{cases} \mathsf{Image} := \{y \in \{0,1\}^{\ell(n)} : \exists z \in \{0,1\}^n \text{ such that } H_{\mathsf{PRHF}}(z) = y\} \\ \mathsf{PreImage}(y) := \{z \in \{0,1\}^n : H_{\mathsf{PRHF}}(z) = y\} \end{cases}$
5. For $i \in [p_{\mathsf{hbs}}(n)]$,
   (a) If $r_i = 0$, sample $y_i \leftarrow \mathsf{Image}$ and $v_i \leftarrow \mathsf{PreImage}(y_i)$, and set $v_i' = v_i$.
   (b) If $r_i = 1$, sample $y_i \leftarrow \{0,1\}^{\ell(n)} \setminus \mathsf{Image}$, and set $v_i' = \perp$.
6. Output $(\mathsf{urs} = (y_1 \dots y_{p_{\mathsf{hbs}}(n)}), \pi = (\mathsf{ZHB}.\mathcal{I}, \{v_i'\}_{i \in \mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi))$.

Claim 21. *It holds that* $H_1 \overset{s}{\approx} H_2$.

Proof. Let $D_{0,x}$ and $D_{1,x}$ be the distributions of binary string r sampled in hybrid $H_1(x)$ and $H_2(x)$ respectively.
Based on the δ-density property of H_{PRHF}, we have:

$$\begin{aligned} \gamma' &= 1 - \Pr_{y \xleftarrow{\$} \{0,1\}^{\ell(n)}}[\exists z \in \{0,1\}^n \text{ such that } H_{\mathsf{PRHF}}(z) = y] \\ &= 1 - \delta \pm \mathsf{negl}(n) && \text{(Property 1 of Def. 6)} \\ &= 1 - \mathsf{Approx}_\delta(1^n) \pm \mathsf{negl}(n) && \text{(Def. 2)} \end{aligned}$$

As the distribution of each r_i is independently and identically distributed, we can bound the statistical distance between $D_{0,x}$ and $D_{1,x}$ with Lem. 37 as follows:

$$\Delta(D_{0,x}, D_{1,x}) \leq p_{\mathsf{hbs}}(n) \cdot \mathsf{negl}_1(n) = \mathsf{negl}_2(n)$$

As these two hybrids only differ in the way of sampling r, and thus there exist a negligible function $\mathsf{negl}(n)$ such that:

$$\Delta(H_1(x), H_2(x)) \leq \mathsf{negl}(n)$$

□

Hybrid: H_3 We present H_3 in Expr. 25 and highlight its difference from H_2 in red color. In this hybrid, we changed the way of sampling v'_i as we defined in the Strong-PreImage Pseudorandomness of the δ-dense post-quantum PRHF .

Experiment 25: Hybrid H_3

1. Let $n = |x|$.
2. For $i \in [p_{\mathsf{hbs}}(n)]$,
   (a) Sample $r_i \in \{0,1\}$ such that $\Pr[r_i = 1] = 1 - \mathsf{Approx}_\delta(1^n)$
3. $(\mathsf{ZHB}.\mathcal{I}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB}.\mathsf{P}_\gamma(r, x)$ where $r = r_1 \dots r_{p_{\mathsf{hbs}}(n)}$.
4. Deletion of the definition of Image and PreImage
5. For $i \in [p_{\mathsf{hbs}}(n)]$,
   (a) If $r_i = 0$, then set $v_i \xleftarrow{\$} \{0,1\}^n, y_i = H_{\mathsf{PRHF}}(v_i)$ and set $v'_i = v_i$
   (b) If $r_i = 1$, then set $v_i \xleftarrow{\$} \{0,1\}^n, y_i = H_{\mathsf{PRHF}}(v_i)$ and set $v'_i = \perp$
6. Output $(\mathsf{urs} = (y_1 \dots y_{p_{\mathsf{hbs}}(n)}), \pi = (\mathsf{ZHB}.\mathcal{I}, \{v'_i\}_{i \in \mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi))$.

Claim 22. *It holds that* $H_2 \overset{c_q}{\approx} H_3$.

Proof. In this argument, we try to reduce the distinguishability between hybrid H_2 and hybrid H_3 to breaking the Strong-PreImage Pseudorandomness of the δ-dense post-quantum PRHF.
Assume for contradiction that there exist a QPT quantum distinguisher $\mathcal{A}$ such that:

$$|\Pr[\mathcal{A}(H_2(x)) = 1] - \Pr[\mathcal{A}(H_3(x)) = 1]| \geq \frac{1}{\mathsf{poly}(n)}$$

We can then construct a non-uniform algorithm $\mathcal{B}_\rho$ that can break the Strong-PreImage Pseudorandomness of the δ-dense post-quantum PRHF w.r.t. the game defined in Box 1.

The algorithm $\mathcal{B}_\rho$'s non-uniform advice ρ contains the snapshot of the execution of H_3 at (inclusively) Step 3. In particular, it contains the statement x and the corresponding $(\mathsf{ZHB}.\mathcal{I}, \mathsf{ZHB}.\pi)$, and the bit-vector $\mathbf{r} = (r_1 \dots r_{p_{\mathsf{hbs}}(n)})$. Then, $\mathcal{B}_\rho$ starts to participate the Strong-PreImage Pseudorandomness game by receiving the challenge transcript $(\mathbf{r}, \{y_i\}_{i \in [p_{\mathsf{hbs}}(n)]}, \{v_i\}_{i \in \mathcal{I}_0})$ from the challenger. (Note that we can assume without loss of generality that this $\mathbf{r}$ in the challenge is the same $\mathbf{r}$ contained in $\mathcal{B}_\rho$'s non-uniform advice, because the strong pre-image pseudo-randomness defined in Property 3 holds against any bit-vector $\mathbf{r}$.)

Algorithm 5.5: $\mathcal{B}_\rho$ **that can break the Strong-PreImage Pseudorandomness**

$\mathcal{B}_\rho(1^n, H_{\mathsf{PRHF}}, \mathbf{r}, \{y_i\}_{i\in[p_{\mathsf{hbs}}(n)]}, \{v_i\}_{i\in\mathcal{I}_0})$:

1. Extract x and the corresponding $(\mathsf{ZHB}.\mathcal{I}, \mathsf{ZHB}.\pi)$ from the non-uniform advice ρ of $\mathcal{B}$.
2. For $i \in [p_{\mathsf{hbs}}(n)]$,
   (a) If $r_i = 0$, set $v'_i = v_i$
   (b) If $r_i = 1$, set $v'_i = \perp$
3. Set $\mathsf{urs} = (y_1 \dots y_{p_{\mathsf{hbs}}(n)}), \pi = (\mathsf{ZHB}.\mathcal{I}, \{v'_i\}_{i\in\mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi)$
4. Invoke $\mathcal{A}(\mathsf{urs}, H_{\mathsf{PRHF}}, x, \pi)$ and output whatever $\mathcal{A}$ outputs.

Refering to the strong pre-image pseudorandomness property in Item 3 of Def. 6, the challenge transcript

$$\left(\{r_i\}_{i\in[p(n)]}, \{y_i\}_{i\in[p_{\mathsf{hbs}}(n)]}, \{v_i\}_{i\in\mathcal{I}_0}\right)$$

is distributed exactly as in hybrid H_2 when sampled from $\mathcal{D}_0$, and exactly as in hybrid H_3 when sampled from $\mathcal{D}_1$. Consequently,

$$\Pr[\mathcal{B}_\rho(\mathcal{D}_0) = 1] = \Pr[\mathcal{A}(H_2(x)) = 1] \quad \text{and} \quad \Pr[\mathcal{B}_\rho(\mathcal{D}_1) = 1] = \Pr[\mathcal{A}(H_3(x)) = 1].$$

Therefore, we have:

$$|\Pr[\mathcal{B}_\rho(D_0) = 1] - \Pr[\mathcal{B}_\rho(D_1) = 1]| = |\Pr[\mathcal{A}(H_2(x)) = 1] - \Pr[\mathcal{A}(H_3(x)) = 1]| \geq \frac{1}{\mathsf{poly}(n)},$$

which contradicts the Strong-PreImage Pseudorandomness of the δ-dense post-quantum PRHF. Thus we can conclude that H_2 and H_3 are indistinguishable towards QPT adversaries with PRHF-independent non-uniform advice.

□

Hybrid H_4**:** We present H_4 in Expr. 26 and highlight its difference from H_3 in red color. In this hybrid, we replace the real proof generated by $\mathsf{ZHB}.\mathsf{P}_\gamma$ with the output of the statistical zero-knowledge simulator $\mathsf{ZHB}.\mathsf{Sim}_\gamma$.

Experiment 26: Hybrid H_4

1. Let $n = |x|$.
2. $(\mathsf{ZHB}.\mathcal{I}, \{r_i\}_{i\in\mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB}.\mathsf{Sim}_\gamma(x)$.
3. For every $i \in [p_{\mathsf{hbs}}(n)]$, sample $v_i \xleftarrow{\$} \{0,1\}^n$ and set $y_i = H_{\mathsf{PRHF}}(v_i)$.
4. For every $i \in \mathsf{ZHB}.\mathcal{I}$,
   (a) If $r_i = 0$, set $v'_i = v_i$.
   (b) If $r_i = 1$, set $v'_i = \perp$.
5. Output $(\mathsf{urs} = (y_1 \dots y_{p_{\mathsf{hbs}}(n)}), \pi = (\mathsf{ZHB}.\mathcal{I}, \{v'_i\}_{i\in\mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi))$.

Claim 23. *It holds that* $H_3 \overset{s}{\approx} H_4$.

Proof. In this argument, we try to reduce the distinguishability between hybrid H_3 and hybrid H_4 to breaking the statistical zero-knowledge property of the $\mathsf{ZHB}.\mathsf{NIZK}$.

Recall that $\gamma = 1 - \delta$. We choose the approximation algorithm for γ such that

$$\mathsf{Approx}_\gamma(1^n) = 1 - \mathsf{Approx}_\delta(1^n).$$

Therefore, the distribution of r in H_3 is identical to the distribution generated by $\mathsf{ZHB.Gen}_\gamma(1^n)$.

Assume for contradiction that there exists a quantum distinguisher $\mathcal{A}$ such that

$$|\Pr[\mathcal{A}(H_3(x)) = 1] - \Pr[\mathcal{A}(H_4(x)) = 1]| \geq \frac{1}{\mathsf{poly}(n)}.$$

We can then construct an adversary $\mathcal{A}_2$ that breaks the statistical zero-knowledge property of the ZHB.NIZK as follows:

Algorithm 5.6: $\mathcal{A}_2$ that breaks the Statistical Zero-Knowledge Property of the ZHB.NIZK

$\mathcal{A}_2(x, \mathsf{ZHB}.\mathcal{I}, \{r_i\}_{i\in\mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi)$:

1. Let $n = |x|$.
2. Sample $H_{\mathsf{PRHF}} \xleftarrow{\$} \mathcal{H}$.
3. For every $i \in [p_{\mathsf{hbs}}(n)]$, sample $v_i \xleftarrow{\$} \{0,1\}^n$ independently and set $y_i = H_{\mathsf{PRHF}}(v_i)$.
4. For every $i \in \mathsf{ZHB}.\mathcal{I}$,
   (a) If $r_i = 0$, set $v'_i = v_i$.
   (b) If $r_i = 1$, set $v'_i = \perp$.
5. Set $\mathsf{urs} = (y_1 \dots y_{p_{\mathsf{hbs}}(n)})$ and $\pi = (\mathsf{ZHB}.\mathcal{I}, \{v'_i\}_{i\in\mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi)$
6. Output $\mathcal{A}(\mathsf{urs}, H_{\mathsf{PRHF}}, x, \pi)$.

We first argue that $\mathcal{A}_2$ exactly simulates hybrid H_3 when its input is sampled from the real-world distribution of the ZHB.NIZK. In this case,

$$r \leftarrow \mathsf{ZHB.Gen}_\gamma(1^n)$$

and

$$(\mathsf{ZHB}.\mathcal{I}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB.P}_\gamma(r, x).$$

After receiving the revealed bits $\{r_i\}_{i\in\mathsf{ZHB}.\mathcal{I}}$, the adversary $\mathcal{A}_2$ samples every v_i independently, sets $y_i = H_{\mathsf{PRHF}}(v_i)$, and defines v'_i in exactly the same manner as hybrid H_3. Therefore,

$$\Pr\left[\mathcal{A}_2\left(x, \mathsf{ZHB}.\mathcal{I}, \{r_i\}_{i\in\mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi\right) = 1 : \begin{array}{l} r \leftarrow \mathsf{ZHB.Gen}_\gamma(1^n), \\ (\mathsf{ZHB}.\mathcal{I}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB.P}_\gamma(r, x) \end{array}\right] = \Pr[\mathcal{A}(H_3(x)) = 1].$$

Similarly, when the input of $\mathcal{A}_2$ is sampled from the simulated-world distribution

$$(\mathsf{ZHB}.\mathcal{I}, \{r_i\}_{i\in\mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB.Sim}_\gamma(x),$$

the output distribution of $\mathcal{A}_2$ is identical to the output distribution of $\mathcal{A}$ in hybrid H_4. Therefore,

$$\Pr\left[\mathcal{A}_2\left(x, \mathsf{ZHB}.\mathcal{I}, \{r_i\}_{i\in\mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi\right) = 1 : (\mathsf{ZHB}.\mathcal{I}, \{r_i\}_{i\in\mathsf{ZHB}.\mathcal{I}}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB.Sim}_\gamma(x)\right] = \Pr[\mathcal{A}(H_4(x)) = 1].$$

By combining the above two equalities, we have

$$\begin{aligned}&|\Pr\left[\mathcal{A}_2(\mathsf{Real}_{\mathsf{ZHB}}(x))=1\right]-\Pr\left[\mathcal{A}_2(\mathsf{Sim}_{\mathsf{ZHB}}(x))=1\right]|\\&\quad=|\Pr[\mathcal{A}(H_3(x))=1]-\Pr[\mathcal{A}(H_4(x))=1]|\geq\frac{1}{\mathsf{poly}(n)}.\end{aligned}$$

This contradicts the statistical zero-knowledge property of the ZHB.NIZK. Thus, we can conclude that hybrid H_3 and hybrid H_4 are statistically indistinguishable. □

Completes the proof of Zero-Knowledge: By combining the indistinguishability between hybrid H_0 and hybrid H_4, we can conclude that the real-world interaction between the adversary and the real prover is computationally indistinguishable from the output of the simulator towards any QPT quantum adversary with H-independent non-uniform advice. This completes the proof of the ZK property in the URS-PRHF model.

□

6 NIZK in QROM

6.1 Structure of Our Construction

We now turn our URS-PRHF-model NIZK construction into a NIZK proof system in the QROM by realizing both the PRHF and the reference string using a single quantum random oracle. Our strengthened post-quantum PRHF definition allows us to reuse the proof for the zero-knowledge property from Sec. 5 when we instantiate the PRHF using a random oracle. This differs from [GIK+23], where the use of hash-dependent non-uniform advice prevents the URS zero-knowledge proof from directly carrying over to the random-oracle setting. The remaining work is to simulate the oracle supplying the URS and to justify combining the two oracle interfaces. We organize the construction into the following three steps:

1. Instantiate the PRHF in our URS-PRHF-model NIZK construction (Sec. 5.1) with a quantum random oracle O_{PRHF}. This step relies on our result that random oracles instantiate strong post-quantum PRHFs (Sec. 4.2), and yields a URS-PRHF-model NIZK proof system relative to O_{PRHF}.
2. Replace the URS with values of an independent random oracle O_{URS}, obtaining a NIZK proof system with access to two independent quantum random oracles. The conditional compressed oracle enables efficient simulation of O_{URS} conditioned on the reference string produced by the URS simulator.
3. Apply the two-oracle-to-one-oracle technique (Sec. 3.3) to implement both oracles using a single quantum random oracle, yielding the desired NIZK proof system in the QROM.

For the random-oracle instantiation of the URS-PRHF-model NIZK construction (Sec. 5.1), the PRHF density parameter and the complementary hidden-bit parameter are

$$\delta=1-\frac{1}{e},\qquad\gamma=1-\delta=\frac{1}{e}.$$

The construction requires both constants to be efficiently approximable. It suffices to establish this property for $\gamma=1/e$: subtracting an approximation to γ from 1 yields an approximation to δ with the same error. The proof is standard and is deferred to Sec. A.3.

6.2 Instantiating the PRHF

Our analysis (Sec. 4.2) shows that a quantum random oracle $O_{\mathsf{PRHF}} \leftarrow \mathcal{O}$ instantiates a $(1 - 1/e)$-dense post-quantum PRHF. Since $1 - 1/e$ is efficiently approximable (Corollary 8), applying our URS-PRHF-model construction (Sec. 5.1) yields a NIZK proof system with both URS and RO:

$$\Pi_{\mathsf{URS}} = \left(\mathsf{URS.Gen}, \mathsf{URS.P}^{O_{\mathsf{PRHF}}}, \mathsf{URS.V}^{O_{\mathsf{PRHF}}}\right),$$

with simulator $\mathsf{URS.Sim}^{|O_{\mathsf{PRHF}}\rangle}$.

We use the zero-knowledge guarantee established in Sec. 5.1 against polynomial-time quantum distinguishers with quantum oracle access to O_{PRHF}. All security probabilities include the sampling of $O_{\mathsf{PRHF}} \leftarrow \mathcal{O}$, together with the randomness and measurement outcomes of the relevant algorithms. The oracle is sampled once and remains fixed throughout each experiment.

6.3 Replacing the URS

Lemma 18 (Eliminating the URS). *Suppose that Π_{URS}, relative to O_{PRHF}, has perfect completeness, adaptive statistical soundness with error $\varepsilon_{\mathrm{s}}(n)$, and zero knowledge against polynomial-query-bounded quantum distinguishers, with a QPT oracle simulator.*

Then there exists an unbounded-prover NIZK proof system $\Pi_{\mathsf{TWO-QROM}}$ for L with separate quantum access to two independent random oracles O_{URS} and O_{PRHF}, without an additional reference string. The system preserves perfect completeness, adaptive statistical soundness with error $\varepsilon_{\mathrm{s}}(n)$, and zero knowledge against the same class of distinguishers, with a stateful QPT simulator.

Proof. We construct Π_{TwoQROM} by deriving the reference string from O_{URS} and invoking the prover and verifier of the URS-PRHF-model construction with quantum access to O_{PRHF}. We then analyze its security properties separately.

Throughout the following analysis, an integer index i used as an input to O_{URS} is understood to denote $\langle i \rangle_n$, its padded n-bit binary representation as defined in Sec. 3. The same convention applies to integer-labelled oracle locations in sets, conditioning events, and partial tables. For simplicity, we suppress these encodings until the final construction in Sec. 6.5, where we write them explicitly.

Let $n = |x|$ and $m = p_{\mathsf{hbs}}(n)$. The reference string in the instantiated URS system consists of m independent, uniformly distributed n-bit blocks. We obtain these blocks from a random oracle O_{URS}, sampled independently of O_{PRHF}, by setting

$$y_i := O_{\mathsf{URS}}(i), \qquad i \in [m].$$

Define

$$U_n(O_{\mathsf{URS}}) := O_{\mathsf{URS}}(1) \| \cdots \| O_{\mathsf{URS}}(m).$$

The prover and verifier are defined as follows.

Algorithm 6.1: Two-Oracle QROM Prover $\mathsf{TwoQROM.P}$

$\mathsf{TwoQROM.P}^{O_{\mathsf{URS}}, O_{\mathsf{PRHF}}}(x)$:

1. Set $n = |x|$ and $m = p_{\mathsf{hbs}}(n)$.
2. For each $i \in [m]$, query O_{URS} at i to obtain $y_i = O_{\mathsf{URS}}(i)$.
3. Set $\mathsf{urs} = y_1 \| \cdots \| y_m$.
4. $\pi \leftarrow \mathsf{URS.P}^{O_{\mathsf{PRHF}}}(\mathsf{urs}, x)$ and return π.

Algorithm 6.2: Two-Oracle QROM Verifier $\mathsf{TwoQROM.V}$

$\mathsf{TwoQROM.V}^{O_{\mathsf{URS}},O_{\mathsf{PRHF}}}(x,\pi)$:

1. Set $n=|x|$ and $m=p_{\mathsf{hbs}}(n)$.
2. For each $i\in[m]$, query O_{URS} at i to obtain $y_i=O_{\mathsf{URS}}(i)$.
3. Set $\mathsf{urs}=y_1\|\cdots\|y_m$.
4. Return $\mathsf{URS.V}^{O_{\mathsf{PRHF}}}(\mathsf{urs},x,\pi)$.

Since the values of O_{URS} at distinct inputs are independent and uniform, $U_n(O_{\mathsf{URS}})$ has exactly the distribution generated by $\mathsf{URS.Gen}(1^n)$. Moreover, it is independent of O_{PRHF}. Hence,

$$\big(O_{\mathsf{PRHF}},U_n(O_{\mathsf{URS}})\big)\equiv\big(O_{\mathsf{PRHF}},\mathsf{URS.Gen}(1^n)\big),\tag{31}$$

where the reference string on the right is generated independently of O_{PRHF}.

Conditioning on the reference string fixes only the values $O_{\mathsf{URS}}(1),\dots,O_{\mathsf{URS}}(m)$. The remaining values $O_{\mathsf{URS}}(z)$, for $z\notin[m]$, remain mutually independent and uniformly distributed, and are independent of O_{PRHF}. For a fixed statement x, this property also holds after conditioning on the honest proof π: the prover accesses O_{URS} only to obtain urs, and its subsequent computation depends only on urs, x, its internal randomness, and O_{PRHF}.

More explicitly, fix x and let $z_1,\dots,z_t\notin[m]$ be distinct oracle inputs. For arbitrary $v_1,\dots,v_t\in\{0,1\}^n$,

$$\Pr\left[\bigwedge_{j=1}^{t}O_{\mathsf{URS}}(z_j)=v_j\;\middle|\;O_{\mathsf{PRHF}}=F,\ \mathsf{urs}=u_0,\ \pi=\pi_0\right]=2^{-nt},$$

whenever the conditioning event has positive probability. Thus, writing $u_0=y_1\|\cdots\|y_m$, quantum access to O_{URS} under this conditioning is exactly emulated by the partially-fixed compressed oracle initialized with the partial table

$$\{(i,y_i):i\in[m]\}.$$

6.3.1 Completeness

Lemma 19 (Perfect completeness). *If Π_{URS} has perfect completeness, then Π_{TwoQROM} has perfect completeness.*

Proof. Fix $x\in L$ and let $n=|x|$. By construction and the distributional identity in (31),

$$\begin{aligned}&\Pr\left[\mathsf{TwoQROM.V}^{O_{\mathsf{URS}},O_{\mathsf{PRHF}}}(x,\pi)=1:\pi\leftarrow\mathsf{TwoQROM.P}^{O_{\mathsf{URS}},O_{\mathsf{PRHF}}}(x)\right]\\&=\Pr\left[\mathsf{URS.V}^{O_{\mathsf{PRHF}}}(\mathsf{urs},x,\pi)=1:\begin{array}{l}\mathsf{urs}\leftarrow\mathsf{URS.Gen}(1^n),\\ \pi\leftarrow\mathsf{URS.P}^{O_{\mathsf{PRHF}}}(\mathsf{urs},x)\end{array}\right]\\&=1.\end{aligned}$$

The first probability includes the independent sampling of O_{URS} and O_{PRHF}; the second includes the independent sampling of O_{PRHF} and urs. Both probabilities also include the internal randomness of the algorithms. The final equality follows from the perfect completeness of Π_{URS}. □

6.3.2 Soundness

Lemma 20 (Adaptive statistical soundness). *If Π_{URS} has adaptive statistical soundness with error $\varepsilon_{\mathrm{s}}(n)$, then Π_{TwoQROM} has adaptive statistical soundness with the same error.*

Proof. For a fixed oracle O_{PRHF} and reference string urs, define

$$\mathsf{Bad}_n(O_{\mathsf{PRHF}}, \mathsf{urs}) := \left[\exists x \in \{0,1\}^n \setminus L,\ \exists \pi : \mathsf{URS.V}^{O_{\mathsf{PRHF}}}(\mathsf{urs}, x, \pi) = 1\right].$$

Every statement of length n uses the same reference string $U_n(O_{\mathsf{URS}})$. Therefore,

$$\begin{aligned}
&\Pr_{O_{\mathsf{URS}}, O_{\mathsf{PRHF}}}\left[\exists x \in \{0,1\}^n \setminus L,\ \exists \pi : \mathsf{TwoQROM.V}^{O_{\mathsf{URS}}, O_{\mathsf{PRHF}}}(x, \pi) = 1\right] \\
&\quad = \Pr_{O_{\mathsf{URS}}, O_{\mathsf{PRHF}}}\left[\mathsf{Bad}_n\big(O_{\mathsf{PRHF}}, U_n(O_{\mathsf{URS}})\big)\right] \\
&\quad = \Pr_{\substack{O_{\mathsf{PRHF}} \leftarrow \mathcal{O}, \\ \mathsf{urs} \leftarrow \mathsf{URS.Gen}(1^n)}}\left[\mathsf{Bad}_n(O_{\mathsf{PRHF}}, \mathsf{urs})\right] \\
&\quad \le \varepsilon_{\mathrm{s}}(n).
\end{aligned}$$

The second equality follows from (31), and the final inequality is the adaptive statistical soundness of Π_{URS}.

The existential quantifiers range over all false statements of length n and all proofs. Thus, the argument permits adaptive statement selection and computationally unbounded cheating provers. □

6.3.3 Zero Knowledge

For a reference string $\mathsf{urs} = y_1 \| \cdots \| y_m$, define

$$\mathsf{HD}_{\mathsf{urs}} := \{(i, y_i) : i \in [m]\}.$$

This table defines a consistent partial function because its inputs are distinct. By Lem. 7 , $\mathsf{Com}(\mathsf{HD}_{\mathsf{urs}})$ perfectly emulates quantum access to a random oracle conditioned on its first m values being $y_1, \ldots, y_m$. All remaining values are mutually independent, uniformly distributed n-bit strings.

Algorithm 6.3: Proof Simulator TwoQROM.SimProof

$\mathsf{TwoQROM.SimProof}(x)$:

1. Set $n = |x|$ and $m = p_{\mathsf{hbs}}(n)$.
2. Initialize the PRHF-oracle simulator: $\mathcal{C}_{\mathsf{PRHF}} \leftarrow \mathrm{Com}(\varnothing)$.
3. Run: $(\mathsf{urs}^*, \pi^*) \leftarrow \mathsf{URS.Sim}^{|\mathcal{C}_{\mathsf{PRHF}}\rangle}(x)$.
   Retain the resulting state of $\mathcal{C}_{\mathsf{PRHF}}$.
4. Parse $\mathsf{urs}^* = y_1^* \| \cdots \| y_m^*$ and set $\mathsf{HD}^* := \{(i, y_i^*) : i \in [m]\}$.
5. Initialize the URS-oracle simulator: $\mathcal{C}_{\mathsf{URS}} \leftarrow \mathrm{Com}(\mathsf{HD}^*)$.
6. Output (st, π^*), where st contains the current joint state of $\mathcal{C}_{\mathsf{URS}}$ and $\mathcal{C}_{\mathsf{PRHF}}$.

Algorithm 6.4: Oracle Simulator TwoQROM.SimRO

$\mathsf{TwoQROM.SimRO}$ maintains st and provides two quantum oracle interfaces:

1. Upon a query to $|O_{\mathsf{URS}}\rangle$, apply the query operation of $\mathcal{C}_{\mathsf{URS}}$.
2. Upon a query to $|O_{\mathsf{PRHF}}\rangle$, apply the query operation of $\mathcal{C}_{\mathsf{PRHF}}$.

Each operation acts coherently on the query registers and the corresponding compressed-oracle state. Return the query registers and retain the resulting joint simulator state for subsequent queries.

The oracle simulation continues from the state produced during the execution of $\mathsf{URS.Sim}$, preserving its correlations with the simulated reference string and proof. Only the URS-oracle simulator is initialized with a nonempty conditioning table, namely HD^*. At initialization, conditioned on (urs^*, π^*), the unspecified URS-oracle values are fresh, mutually independent uniform strings, independent of the preceding simulation.

Lemma 21 (Zero knowledge). *Suppose that Π_{URS} satisfies zero knowledge against polynomial-query-bounded quantum distinguishers with access to $|O_{\mathsf{PRHF}}\rangle$. Then Π_{TwoQROM} satisfies zero knowledge against polynomial-query-bounded quantum distinguishers with separate access to $|O_{\mathsf{URS}}\rangle$ and $|O_{\mathsf{PRHF}}\rangle$, with an efficient stateful simulator.*

Proof. Fix a polynomial-query-bounded quantum distinguisher $\mathcal{A}$ and a statement $x \in L$, and let $n = |x|$. We construct a URS-PRHF-model distinguisher $\mathcal{B}^{|O_{\mathsf{PRHF}}\rangle}$ such that any non-negligible distinguishing advantage of $\mathcal{A}$ contradicts the zero-knowledge guarantee of Sec. 5.2.

Algorithm 6.5: URS Distinguisher $\mathcal{B}$

$\mathcal{B}^{|O_{\mathsf{PRHF}}\rangle}(\mathsf{urs}, x, \pi)$:

1. Set $n = |x|$ and $m = p_{\mathsf{hbs}}(n)$, and parse $\mathsf{urs} = y_1 \| \cdots \| y_m$.
2. Initialize
$$\mathcal{C}_{\mathsf{URS}} \leftarrow \mathrm{Com}(\mathsf{HD}_{\mathsf{urs}}).$$
3. Run $\mathcal{A}(x, \pi)$, answering its queries to $|O_{\mathsf{URS}}\rangle$ using $\mathcal{C}_{\mathsf{URS}}$ and forwarding its queries to $|O_{\mathsf{PRHF}}\rangle$ to the supplied oracle. Retain the state of $\mathcal{C}_{\mathsf{URS}}$ throughout the interaction.
4. Output the bit returned by $\mathcal{A}$.

Real experiment. When (urs, π) is a real URS transcript, $\mathcal{B}$ exactly reproduces the real experiment for Π_{TwoQROM}. Indeed, urs is uniform and independent of O_{PRHF}. Conditioned on urs, O_{PRHF}, and π, the URS-oracle values outside $[m]$ remain mutually independent and uniform, as shown above. Lem. 7 therefore gives an exact emulation of quantum access to the URS oracle under this conditioning. Consequently,

$$\begin{aligned} p_{\mathsf{real}} &:= \Pr\left[\mathcal{A}^{|O_{\mathsf{URS}}\rangle, |O_{\mathsf{PRHF}}\rangle}(x, \pi) = 1 : \pi \leftarrow \mathsf{TwoQROM.P}^{O_{\mathsf{URS}}, O_{\mathsf{PRHF}}}(x)\right] \\ &= \Pr\left[\mathcal{B}^{|O_{\mathsf{PRHF}}\rangle}(\mathsf{urs}, x, \pi) = 1 : \begin{array}{l} \mathsf{urs} \leftarrow \mathsf{URS.Gen}(1^n), \\ \pi \leftarrow \mathsf{URS.P}^{O_{\mathsf{PRHF}}}(\mathsf{urs}, x) \end{array}\right]. \end{aligned}$$

This equality also holds when queries to the two oracles are interleaved. For each fixed O_{PRHF}, its query operations are intervening operations on the distinguisher's registers, which are covered by the exact simulation in Lem. 7.

Simulated experiment. When (urs^*, π^*) is generated by $\mathsf{URS.Sim}^{|O_{\mathsf{PRHF}}\rangle}$, the reduction runs $\mathcal{A}(x, \pi^*)$ with access to the same O_{PRHF} and to $\mathrm{Com}(\mathsf{HD}_{\mathsf{urs}^*})$ as its URS-oracle interface.

Apply Lem. 7 with the empty partial function to this entire computation: the execution of $\mathsf{URS.Sim}$, the initialization of the conditional URS oracle, and the subsequent execution of $\mathcal{A}$. Replacing O_{PRHF} throughout by a single persistent $\mathrm{Com}(\varnothing)$ preserves the output distribution exactly.

The resulting computation is precisely the simulation defined above. Retaining the PRHF-oracle state throughout preserves all correlations among urs^*, π^*, and subsequent oracle responses. In the following experiment, $|\mathcal{C}_{\mathsf{URS}}\rangle$ and $|\mathcal{C}_{\mathsf{PRHF}}\rangle$ denote the two stateful interfaces maintained by $\mathsf{TwoQROM.SimRO}$ using the joint state st returned by $\mathsf{TwoQROM.SimProof}$. We obtain

$$p_{\mathsf{sim}} := \Pr\left[\mathcal{A}^{|\mathcal{C}_{\mathsf{URS}}\rangle, |\mathcal{C}_{\mathsf{PRHF}}\rangle}(x, \pi^*) = 1 : (\mathsf{st}, \pi^*) \leftarrow \mathsf{TwoQROM.SimProof}(x)\right]$$

$$= \Pr\left[\mathcal{B}^{|O_{\mathsf{PRHF}}\rangle}(\mathsf{urs}^*, x, \pi^*) = 1 : (\mathsf{urs}^*, \pi^*) \leftarrow \mathsf{URS.Sim}^{|O_{\mathsf{PRHF}}\rangle}(x)\right].$$

Reduction and efficiency. The reduction $\mathcal{B}$ makes no additional queries to O_{PRHF} beyond those made by $\mathcal{A}$. By Lem. 7, simulating the URS oracle adds only polynomial overhead in n, the query count, and the size of $\mathsf{HD}_{\mathsf{urs}}$. Since $\mathcal{A}$ is polynomial-query-bounded, $\mathcal{B}$ is also polynomial-query-bounded.

Applying the zero-knowledge guarantee of Π_{URS} to $\mathcal{B}$ and using the two identities above yields

$$\left|p_{\mathsf{real}} - p_{\mathsf{sim}}\right| \leq \mathrm{negl}(n).$$

Finally, $\mathsf{URS.Sim}$ is QPT, and a QPT distinguisher makes only polynomially many additional oracle queries. The PRHF compressed oracle handles their combined query sequence while retaining its state throughout. The URS compressed oracle starts with m prescribed entries and handles polynomially many subsequent queries. Lem. 7 therefore gives polynomial time and space for both oracle simulations.

Thus, the proof simulator is QPT, the stateful oracle simulator runs in polynomial time throughout any QPT interaction, and the transformation introduces no additional distinguishing error. □

Together, Lem. 19 to 21 prove Lem. 18.

□

6.4 Combining the Two Oracles

There exists an unbounded-prover NIZK proof system for NP in the QROM with perfect completeness, adaptive statistical soundness, and zero knowledge against quantum polynomial-time oracle-aided distinguishers.

Let $O \leftarrow \mathcal{O}$ be a single random oracle. By Lem. 1, interpret O as providing mutually independent random functions

$$O_{n+1,n} : \{0,1\}^{n+1} \to \{0,1\}^n, \qquad n \in \mathbb{N}.$$

For each n and every $z \in \{0,1\}^n$, define

$$O_{\mathsf{URS}}(z) := O_{n+1,n}(0\|z), \qquad O_{\mathsf{PRHF}}(z) := O_{n+1,n}(1\|z).$$

These restrictions are independent random functions from $\{0,1\}^n$ to $\{0,1\}^n$, since their input domains are disjoint. The resulting system therefore inherits perfect completeness and adaptive statistical soundness from Π_{TwoQROM}.

The oracle-merging lemma (Sec. 3.3) gives an exact implementation of each query to $|O_{n+1,n}\rangle$ using four queries to the separate interfaces $|O_{\mathsf{URS}}\rangle$ and $|O_{\mathsf{PRHF}}\rangle$, with polynomial-time overhead. This implementation also applies when the prefix register is in superposition or entangled with other registers. Thus, any QPT distinguisher with access to $|O\rangle$ induces a QPT distinguisher with separate access to $|O_{\mathsf{URS}}\rangle$ and $|O_{\mathsf{PRHF}}\rangle$.

6.5 Final Construction

Combining the above results and exploiting the exact construction in Sec. 5.1, we state the final construction of a NIZK proof system for NP in the QROM.

Algorithm 6.6: The Prover $\mathsf{QRO.P}^O$

$\mathsf{QRO.P}^O(x)$:

1. Let $n = |x|$.
2. For $i \in [p_{\mathsf{hbs}}(n)]$, set $y_i = O_{n+1,n}(0||\langle i \rangle_n)$
   where $\langle i \rangle_n$ is the n-bit binary representation of i as we defined in Sec. 3
3. For $i \in [p_{\mathsf{hbs}}(n)]$,
   (a) If $\exists z \in \{0,1\}^n$ such that $O_{n+1,n}(1||z) = y_i$, sample $v_i \xleftarrow{\$} \{z \in \{0,1\}^n : O_{n+1,n}(1||z) = y_i\}$, and set $r_i = 0$.
   (b) If $\nexists z \in \{0,1\}^n$ such that $O_{n+1,n}(1||z) = y_i$, set $v_i = \perp$ and $r_i = 1$.
4. $(\mathsf{ZHB.I}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB.P}_{e^{-1}}(r, x)$ where $r = r_1 \dots r_{p_{\mathsf{hbs}}(n)}$.
5. Return $\pi := (\mathsf{ZHB.I}, \{v_i\}_{i \in \mathsf{ZHB.I}}, \mathsf{ZHB}.\pi)$.

Algorithm 6.7: The Verifier $\mathsf{QRO.V}^O$

$\mathsf{QRO.V}^O(x, \pi)$:

1. Let $n = |x|$.
2. For $i \in [p_{\mathsf{hbs}}(n)]$, set $y_i = O_{n+1,n}(0||\langle i \rangle_n)$
3. Parse π as $(\mathsf{ZHB.I}, \{v_i\}_{i \in \mathsf{ZHB.I}}, \mathsf{ZHB}.\pi)$.
4. For $i \in \mathsf{ZHB.I}$,
   – If $v_i = \perp$, set $\widetilde{r}_i = 1$.
   – If $v_i \neq \perp$, and $O_{n+1,n}(1||v_i) = y_i$, set $\widetilde{r}_i = 0$.
   – If $v_i \neq \perp$, and $O_{n+1,n}(1||v_i) \neq y_i$, reject.
5. Return $\mathsf{ZHB.V}_{e^{-1}}(x, \mathsf{ZHB.I}, \{\widetilde{r}_i\}_{i \in \mathsf{ZHB.I}}, \mathsf{ZHB}.\pi)$

Algorithm 6.8: Proof Simulator $\mathsf{QRO.SimProof}$

$\mathsf{QRO.SimProof}(x)$:

1. Let $n = |x|$.
2. $(\mathsf{ZHB.I}, \{r_i\}_{i \in \mathsf{ZHB.I}}, \mathsf{ZHB}.\pi) \leftarrow \mathsf{ZHB.Sim}_{\frac{1}{e}}(x)$.
3. Create history database $\mathsf{HD} = \{\}$.
4. For $i \in [p_{\mathsf{hbs}}(n)]$
   (a) Sample $v_i \xleftarrow{\$} \{0,1\}^n$.
   (b) If $(1\|v_i, y) \in \mathsf{HD}$ for some y, set $y_i = y$. Otherwise, sample $y_i \xleftarrow{\$} \{0,1\}^n$.
   (c) Update history database $\mathsf{HD} = \mathsf{HD} \cup \{(0||\langle i \rangle_n, y_i), (1||v_i, y_i)\}$
5. For $i \in \mathsf{ZHB.I}$,
   (a) If $r_i = 1$, set $v'_i = \perp$
   (b) If $r_i = 0$, set $v'_i = v_i$.
6. Set $\pi = (\mathsf{ZHB.I}, \{v'_i\}_{i \in \mathsf{ZHB.I}}, \mathsf{ZHB}.\pi)$.

7. Return (st, π), where st contains the current state of the compressed oracle $\mathrm{Com}(\mathsf{HD})$.

Algorithm 6.9: Oracle Simulator $\mathsf{QRO.SimRO}$

$\mathsf{QRO.SimRO}(\mathsf{st})$:

1. Answer each quantum query using the compressed partially fixed oracle with regard to the current state st, retaining its updated state for subsequent queries.

7 Oracle Separation between NP ∩ coNP and BQP

We now prove the random-oracle separation informally stated in Thm. 1.

Theorem 4. *If* $\mathsf{UP} \not\subseteq \mathsf{BQP}$, *then*

$$\Pr_{O \leftarrow \mathcal{O}}\left[\mathsf{NP}^O \cap \mathsf{coNP}^O \not\subseteq \mathsf{BQP}^O\right] = 1.$$

To prove Thm. 4, we adapt the language construction of [GIK+23] using the QROM NIZK proof system from Sec. 6.5. We start with a language $\mathcal{L}_0 \in \mathsf{UP} \setminus \mathsf{BQP}$ and define an oracle-aided language $\mathcal{L}^O$ on triples (x, i, π). The proof π certifies that $x \in \mathcal{L}_0$, and membership in $\mathcal{L}^O$ asks whether the i-th bit of the unique witness for x is 1. Uniqueness and the statistical soundness of the proof system give $\mathcal{L}^O \in \mathsf{NP}^O \cap \mathsf{coNP}^O$ for almost every oracle (Lem. 24).

We then establish quantum hardness in Lem. 25. If a fixed quantum oracle machine decided $\mathcal{L}^O$ on at least a $2/3$ fraction of oracles, we could use it together with the NIZK simulator to recover the unique witness bit by bit without access to O. Verifying the recovered witness would give a BQP decider for $\mathcal{L}_0$. Finally, we apply the Bennett–Gill argument in Lem. 22 to conclude that, with probability 1 over O, no quantum polynomial-time oracle machine decides $\mathcal{L}^O$.

7.1 Technical Tool

We use a quantum formulation of the Bennett–Gill argument [BG81, Lemma 1] to pass from a bound for each fixed quantum oracle machine to an almost-sure hardness result. Under the finite-dependence and decidability conditions below, a uniform gap from probability 1 implies that each fixed machine decides the language only on a measure-zero set of oracles. Since there are countably many such machines, their success sets together still have measure zero.

This lemma serves as a key technical tool in Sec. 7.5, where we use it to show that $\mathcal{L}^O \notin \mathsf{BQP}^O$ for almost every oracle O.

Lemma 22 ([BG81, Lemma 1], Refurnished). *Let* $\mathcal{L}^{(\cdot)}$ *be an oracle-dependent language such that each bit of the oracle affects the membership of only finitely many strings in* $\mathcal{L}^{\mathcal{O}}$*; that is, for every* $j \in \mathbb{N}$ *there exists a finite set* F_j *such that for any two oracles* $\mathcal{O}$ *and* $\mathcal{O}'$ *differing only in their* j*-th bit, and for any* $(x, i, \pi) \notin F_j$*, we have* $\mathcal{L}^{\mathcal{O}}(x, i, \pi) = \mathcal{L}^{\mathcal{O}'}(x, i, \pi)$*, where*

$$\mathcal{L}^{\mathcal{O}}(x, i, \pi) := \begin{cases} 1 & (x, i, \pi) \in \mathcal{L}^{\mathcal{O}} \\ 0 & \textit{otherwise} \end{cases}.$$

Moreover, assume that $\mathcal{L}^{\mathcal{O}}$ *is decidable by an oracle Turing machine that halts on all inputs for every oracle* $\mathcal{O}$*.*

Let $\mathcal{D}^{(\cdot)} = \{D_1^{(\cdot)}, D_2^{(\cdot)}, \ldots\}$ be the set of all possible oracle-aided polynomial-time quantum Turing Machines. If there exists a constant $\varepsilon \in (0,1]$ such that for all $k \in \mathbb{N}$,

$$\Pr_{\mathcal{O}}\left[\forall (x,i,\pi) \in \{0,1\}^*, \Pr\left[D_k^{|\mathcal{O}\rangle}(x,i,\pi) = \mathcal{L}^{\mathcal{O}}(x,i,\pi)\right] \geq \frac{2}{3}\right] < 1-\varepsilon,$$

then

$$\Pr_{\mathcal{O}}\left[\exists k \in \mathbb{N} \text{ such that } \forall (x,i,\pi) \in \{0,1\}^*, \Pr\left[D_k^{|\mathcal{O}\rangle}(x,i,\pi) = \mathcal{L}^{\mathcal{O}}(x,i,\pi)\right] \geq \frac{2}{3}\right] = 0. \tag{32}$$

Proof. Since a countable union of events of probability zero still has probability zero, it suffices to prove that, for every $k \in \mathbb{N}$,

$$\Pr_{\mathcal{O}}\left[\forall (x,i,\pi) \in \{0,1\}^*,\ \Pr\left[D_k^{|\mathcal{O}\rangle}(x,i,\pi) = \mathcal{L}^{\mathcal{O}}(x,i,\pi)\right] \geq \frac{2}{3}\right] = 0.$$

Below, we fix k.

For each $n \in \mathbb{N}$, let S_n be the set of all oracles $\mathcal{O}$ such that

$$\forall (x,i,\pi) \in \{0,1\}^{\leq n},\ \Pr\left[D_k^{|\mathcal{O}\rangle}(x,i,\pi) = \mathcal{L}^{\mathcal{O}}(x,i,\pi)\right] \geq \frac{2}{3},$$

where $\{0,1\}^{\leq n}$ denotes the set of all bit strings of length at most n.

Then, it suffices to prove that

$$\Pr_{\mathcal{O}}\left[\mathcal{O} \in \bigcap_{n\in\mathbb{N}} S_n\right] = 0.$$

To this end, we establish the following claim.

Claim 24. *For any $m \in \mathbb{N}$, there exists $m' > m$ such that*

$$\Pr_{\mathcal{O}}[\mathcal{O} \in S_{m'} \mid \mathcal{O} \in S_m] < 1-\varepsilon.$$

We first complete the proof of Lem. 22 assuming Claim 24, and then proceed to prove Claim 24 itself.

By the claim, we may choose an increasing sequence $m_0 < m_1 < m_2 < \cdots$ such that

$$\Pr_{\mathcal{O}}[\mathcal{O} \in S_{m_j} \mid \mathcal{O} \in S_{m_{j-1}}] < 1-\varepsilon \quad \text{for all } j \in \mathbb{N}.$$

Since $\{S_n\}_n$ is monotonically decreasing, we have

$$\begin{aligned}
\Pr_{\mathcal{O}}[\mathcal{O} \in S_{m_j}] &= \Pr_{\mathcal{O}}[\mathcal{O} \in S_{m_j} \wedge \mathcal{O} \in S_{m_{j-1}}] \\
&= \Pr_{\mathcal{O}}[\mathcal{O} \in S_{m_j} \mid \mathcal{O} \in S_{m_{j-1}}] \Pr_{\mathcal{O}}[\mathcal{O} \in S_{m_{j-1}}] \\
&< (1-\varepsilon) \Pr_{\mathcal{O}}[\mathcal{O} \in S_{m_{j-1}}].
\end{aligned}$$

By induction, we have

$$\Pr_{\mathcal{O}}[\mathcal{O} \in S_{m_j}] < (1-\varepsilon)^j \Pr_{\mathcal{O}}[\mathcal{O} \in S_{m_0}] \leq (1-\varepsilon)^j.$$

Since $\{S_n\}_n$ is monotonically decreasing, the following holds as desired

$$\Pr_{\mathcal{O}}\left[\mathcal{O} \in \bigcap_{n\in\mathbb{N}} S_n\right] = \lim_{n\to\infty} \Pr_{\mathcal{O}}[\mathcal{O} \in S_n] \leq \lim_{j\to\infty} (1-\varepsilon)^j = 0$$

This finishes the proof of Lem. 22, modulo the proof of Claim 24, which we provide below.

Proof of Claim 24: Let $T_k(m)$ denote the maximum running time of D_k on inputs of length at most m. Then, on inputs of length at most m, the machine D_k can make oracle queries of length at most $T_k(m)$, and hence can access only the first $2^{T_k(m)}$ bits of $\mathcal{O}$. Also, let M be the oracle Turing machine guaranteed by the assumption of the lemma, which decides $L^{\mathcal{O}}$ and halts on every input for every oracle $\mathcal{O}$. For every m, there exists a finite bound $U(m)$ such that $M^{\mathcal{O}}$ halts within $U(m)$ steps on every input of length at most m, uniformly over all oracles $\mathcal{O}$. Indeed, this follows from König's lemma, since otherwise there would exist an oracle $\mathcal{O}$ and an input on which $M^{\mathcal{O}}$ does not halt.

Therefore, whether $\mathcal{O} \in S_m$ depends only on the first $2^{R_k(m)}$ bits of $\mathcal{O}$, where $R_k(m) := \max\{T_k(m), U(m)\}$.

Consequently, to prove the claim it suffices to show that there exists m' such that, for every $s \in \{0,1\}^{2^{R_k(m)}}$,

$$\Pr_{\mathcal{O}}[\mathcal{O} \in S_{m'} \mid \mathcal{O} \text{ has prefix } s] < 1-\varepsilon,$$

or equivalently,

$$\Pr_{\mathcal{O}}[s * \mathcal{O} \in S_{m'}] < 1-\varepsilon,$$

where $s * \mathcal{O}$ denotes the oracle obtained by replacing the first $2^{R_k(m)}$ bits of $\mathcal{O}$ with s.

Since there are only finitely many such strings s, and since the sets S_n are monotonically decreasing in n, we may interchange the order of quantifiers over s and m'.[11] Namely, it suffices to show that for every $s \in \{0,1\}^{2^{R_k(m)}}$ there exists m' such that

$$\Pr_{\mathcal{O}}[s * \mathcal{O} \in S_{m'}] < 1-\varepsilon.$$

Below, we fix s, and prove that the above holds for some m'.

Since $2^{R_k(m)}$ is a finite constant, by the premise of Lem. 22 there exists a finite set $B_m \subseteq \{0,1\}^*$ such that for all oracles $\mathcal{O}$,

$$\mathcal{L}^{\mathcal{O}}(x,i,\pi) = \mathcal{L}^{s*\mathcal{O}}(x,i,\pi) \quad \text{for all } (x,i,\pi) \notin B_m.$$

We define a modified oracle-aided quantum Turing machine $\widetilde{D}_k^{|\mathcal{O}\rangle}$ as follows:

– $\widetilde{D}_k^{|\mathcal{O}\rangle}$: On input (x,i,π), if $(x,i,\pi) \in B_m$, output $\mathcal{L}^{\mathcal{O}}(x,i,\pi)$; otherwise, output $D_k^{|s*\mathcal{O}\rangle}(x,i,\pi)$.

Note that $\widetilde{D}_k$ is also a QPT machine, since s and B_m are finite and, as shown above, $\mathcal{L}^{\mathcal{O}}$ is computable in oracle-independent constant time on B_m. Namely, $\widetilde{D}_k^{(\cdot)}$ belongs to the family $\mathcal{D}^{(\cdot)}$ defined in the statement of Lem. 22.

[11] Suppose that for each s there exists m'_s such that $\Pr_{\mathcal{O}}[s*\mathcal{O} \in S_{m'_s}] < 1-\varepsilon$. Let $m' := \max_s m'_s$. Then $S_{m'} \subseteq S_{m'_s}$ for all s, and hence

$$\Pr_{\mathcal{O}}[s*\mathcal{O} \in S_{m'}] \leq \Pr_{\mathcal{O}}[s*\mathcal{O} \in S_{m'_s}] < 1-\varepsilon.$$

Thus, it follows from the premise of Lem. 22 that

$$\Pr_{\mathcal{O}}\left[\forall(x,i,\pi)\in\{0,1\}^*,\ \Pr\left[\widetilde{D}_k^{|\mathcal{O}\rangle}(x,i,\pi)=\mathcal{L}^{\mathcal{O}}(x,i,\pi)\right]\geq\frac{2}{3}\right]<1-\varepsilon.$$

By construction, for all $\mathcal{O}$ and all $(x,i,\pi)\in B_m$, it holds that

$$\widetilde{D}_k^{|\mathcal{O}\rangle}(x,i,\pi)=\mathcal{L}^{\mathcal{O}}(x,i,\pi).$$

Moreover, for for all $\mathcal{O}$ and all $(x,i,\pi)\notin B_m$,

$$\widetilde{D}_k^{|\mathcal{O}\rangle}(x,i,\pi)=D_k^{|s*\mathcal{O}\rangle}(x,i,\pi)\quad\text{and}\quad\mathcal{L}^{\mathcal{O}}(x,i,\pi)=\mathcal{L}^{s*\mathcal{O}}(x,i,\pi).$$

Therefore,

$$\Pr_{\mathcal{O}}\left[\forall(x,i,\pi)\notin B_m,\ \Pr\left[D_k^{|s*\mathcal{O}\rangle}(x,i,\pi)=\mathcal{L}^{s*\mathcal{O}}(x,i,\pi)\right]\geq\frac{2}{3}\right]<1-\varepsilon.$$

This immediately implies

$$\Pr_{\mathcal{O}}\left[\forall(x,i,\pi)\in\{0,1\}^*,\ \Pr\left[D_k^{|s*\mathcal{O}\rangle}(x,i,\pi)=\mathcal{L}^{s*\mathcal{O}}(x,i,\pi)\right]\geq\frac{2}{3}\right]<1-\varepsilon,$$

or equivalently (by the definition of S_n),

$$\Pr_{\mathcal{O}}\left[s*\mathcal{O}\in\bigcap_{n\in\mathbb{N}}S_n\right]<1-\varepsilon.$$

Since the sets S_n are monotonically decreasing,

$$\lim_{n\to\infty}\Pr_{\mathcal{O}}[s*\mathcal{O}\in S_n]=\Pr_{\mathcal{O}}\left[s*\mathcal{O}\in\bigcap_{n\in\mathbb{N}}S_n\right]<1-\varepsilon.$$

Hence, there exists m' such that

$$\Pr_{\mathcal{O}}[s*\mathcal{O}\in S_{m'}]<1-\varepsilon.$$

This completes the proof of Claim 24.

□

7.2 Post-Quantum injOWF Implies UP $\not\subseteq$ BQP

For a better illustration of the oracle separation result between NP ∩ coNP and BQP, we start by first establishing the connection between the existence of injective post-quantum one-way function and the separation between UP and BQP, instead of directly starting with a vague language that lies in UP \ BQP. It does not mean that the existence of injective post-quantum one-way function is necessary for the separation between NP ∩ coNP and BQP, but it provides us a more clear starting point to construct such oracle separation result.

Lemma 23. *If there exists an injective post-quantum one-way function $f:\{0,1\}^*\to\{0,1\}^*$, then* UP $\not\subseteq$ BQP.

Proof. Consider the language built from the function f as follows:

$$\mathcal{L} \coloneqq \{(y,i) \in \{0,1\}^* \times \mathbb{N} : \exists x \in \{0,1\}^*,\ f(x) = y \wedge x_i = 1 \wedge i \leq |x|\}\,.$$

Here, the index i is encoded in binary.

It is easy to see that $\mathcal{L} \in \mathsf{NP}$ since we can verify the membership of every y in $\mathcal{L}$ with the corresponding x as the witness. The uniqueness of the witness x is guaranteed by the injectivity of f. Also, the BQP hardness of $\mathcal{L}$ follows from the post-quantum one-wayness of f, otherwise the pre-image can be inverted with bit-by-bit recovery alongside with the standard amplification of BQP. We can then conclude that $\mathcal{L} \in \mathsf{UP} \setminus \mathsf{BQP}$. □

7.3 The Oracle-Aided Language $\mathcal{L}^O$

Following [GIK+23], we construct an oracle-aided language whose instances ask for a bit of a unique witness. Fix a base language $\mathcal{L}_0 \in \mathsf{UP} \setminus \mathsf{BQP}$ (Existence is guaranteed by the assumption $\mathsf{UP} \nsubseteq \mathsf{BQP}$) and a witness relation $R_{\mathcal{L}_0}$. Let $\Pi_{\mathsf{NIZK}} = (\mathcal{P}^{(\cdot)}, \mathcal{V}^{(\cdot)})$ be the QROM NIZK proof system from Sec. 6.5 instantiated for $\mathcal{L}_0$.

We form an oracle-aided language by attaching a proof of $x \in \mathcal{L}_0$ to each query about a bit of its unique witness. For a fixed oracle O, define

$$\mathcal{L}^{\mathcal{O}} \coloneqq \left\{(x,i,\pi) \,\middle|\, \left(\exists w :\ (x,w) \in R_{\mathcal{L}_0} \wedge w_i = 1\right) \wedge \Pi_{\mathsf{NIZK}}.\mathcal{V}^{\mathcal{O}}(x,\pi) = 1\right\}. \tag{33}$$

We proceed to show that $\mathcal{L}^O$ belongs to $\mathsf{NP}^O \cap \mathsf{coNP}^O$ for almost every oracle in Sec. 7.4. We establish its quantum hardness in Sec. 7.5.

7.4 Membership in $\mathsf{NP}^O \cap \mathsf{coNP}^O$

The following lemma follows directly from [GIK+23, Lemma 8.4]. As in their construction, our QROM proof system has a classical prover and verifier and satisfies perfect completeness and adaptive statistical soundness. We only use its zero-knowledge guarantee against quantum adversaries in Sec. 7.5.

Lemma 24 ([GIK+23, Lemma 8.4]). *If $\mathsf{UP} \nsubseteq \mathsf{BQP}$, then with probability 1 over the choice of a random oracle $O \leftarrow \mathcal{O}$, the language $\mathcal{L}^O$ defined in Eq. (33) belongs to $\mathsf{NP}^O \cap \mathsf{coNP}^O$. Equivalently,*

$$\Pr_{O \leftarrow \mathcal{O}}\left[\mathcal{L}^O \in \mathsf{NP}^O \cap \mathsf{coNP}^O\right] = 1.$$

7.5 Non-Membership in BQP^O

We now show that the oracle-aided language defined in Eq. (33) remains hard for quantum polynomial-time algorithms with superposition access to the oracle. The proof uses the QROM NIZK simulator to turn a machine that decides $\mathcal{L}^O$ on every input for at least a $2/3$ fraction of oracles into an oracle-free BQP decider for $\mathcal{L}_0$. We then apply Lem. 22 to turn the resulting bound for each fixed machine into an almost-sure hardness result.

Lemma 25. *If $\mathsf{UP} \nsubseteq \mathsf{BQP}$, then with probability 1 over the choice of a random oracle $O \leftarrow \mathcal{O}$, the language $\mathcal{L}^O$ defined in Eq. (33) lies outside BQP^O. Equivalently,*

$$\Pr_{O \leftarrow \mathcal{O}}\left[\mathcal{L}^O \notin \mathsf{BQP}^O\right] = 1.$$

Proof. Fix any oracle-aided QPT machine $D^{(\cdot)}$, and define the set of good oracles on which D decides $\mathcal{L}^{\mathcal{O}}$ on all inputs as

$$G_D \coloneqq \left\{ \mathcal{O} : \forall (x,i,\pi) \in \{0,1\}^*, \Pr\left[D^{|\mathcal{O}\rangle}(x,i,\pi) = \mathcal{L}^{\mathcal{O}}(x,i,\pi)\right] \geq \frac{2}{3} \right\}.$$

To apply Lem. 22 for the separation results, we first have to show that $\Pr_{\mathcal{O}}[\mathcal{O} \in G_D] < 2/3$, note that $\frac{2}{3}$ here can be replaced by any real number ε where $\frac{1}{2} < \varepsilon < 1$

Assume for contradiction that $\Pr_{\mathcal{O}}[\mathcal{O} \in G_D] \geq 2/3$ We then build a quantum decider in BQP for the base language $\mathcal{L}_0$ with the help of the zero-knowledge simulator constructed in the previous section.

In particular, the real oracle-aided decider for the i-th bit is defined as follows:

Algorithm 7.1: Real-decider for i-th bit: $D^{|\mathcal{O}\rangle}_{(i)}$

$D^{|\mathcal{O}\rangle}_{(i)}(x,\pi)$

1. Prepare an empty multiset $\mathsf{Bits} = \{\}$
2. for $j \in [|x|]$
   (a) Invoke $D^{|\mathcal{O}\rangle}(x,i,\pi)$, add the output to Bits
3. Return the majority element in Bits.

Lemma 26. *For all $(x,w) \in R_{\mathcal{L}_0}$ and all $i \in [p(|x|)]$, we have*

$$\Pr\left[D^{|\mathcal{O}\rangle}_{(i)}(x,\pi) = w_i \ : \ \begin{array}{l} O \leftarrow \mathcal{O}, \\ \pi \leftarrow P^O(x) \end{array}\right] \geq \frac{2}{3} - \mathsf{negl}(|x|).$$

Here, P is the honest NIZK prover, and the decider uses the same sampled oracle O. The probability also includes the internal randomness and measurement outcomes of the algorithms.

Proof. By perfect completeness, an honestly generated proof $\pi \leftarrow P^O(x)$ satisfies $V^O(x,\pi) = 1$ with probability 1. Hence, conditioned on $O \in G_D$, each invocation of $D^{|\mathcal{O}\rangle}(x,i,\pi)$ outputs w_i with probability at least $2/3$.

Let X be the random variable of the count of w_i in Bits and $X_j = 1$ if $D^{|\mathcal{O}\rangle}(x,i,\pi) = w_i$ in the j-th iteration, otherwise $X_j = 0$. Notice the event that the output of $D^{|\mathcal{O}\rangle}_{(i)}(x,\pi)$ is w_i is equivalent to the event that X is larger than $\frac{|\mathsf{Bits}|}{2}$, and the bad event probability is bounded by the Chernoff bound as follows:

$$\begin{aligned}
\Pr\left[D^{|\mathcal{O}\rangle}_{(i)}(x,\pi) \neq w_i\right] &= \Pr\left[X \leq \frac{|\mathsf{Bits}|}{2}\right] \\
&= \Pr\left[\sum_{j=1}^{|x|} X_j \leq \frac{|x|}{2}\right] \\
&= \Pr\left[\sum_{j=1}^{|x|} X_j \leq (1-\frac{1}{4})\frac{2|x|}{3}\right] \\
&\leq \Pr\left[\sum_{j=1}^{|x|} X_j \leq (1-\frac{1}{4})\mathbb{E}[X]\right]
\end{aligned}$$

$$
\begin{aligned}
&\leq \exp\left(-\mathbb{E}[X]\cdot(\frac{1}{4})^2\cdot\frac{1}{2}\right) && \text{(Chernoff Bound)}\\
&\leq \exp\left(-\frac{2|x|}{3}\cdot\frac{1}{32}\right)\\
&= \exp\left(-\frac{|x|}{48}\right)\\
&= \mathsf{negl}(|x|)
\end{aligned}
$$

Averaging over the choice of $\mathcal{O}$, we obtain

$$
\begin{aligned}
\Pr\left[D_{(i)}^{|\mathcal{O}\rangle}(x,\pi)=w_i\right] &\geq \Pr_{\mathcal{O}}[\mathcal{O}\in G_D]-\mathsf{negl}(|x|)\\
&\geq \frac{2}{3}-\mathsf{negl}(|x|),
\end{aligned}
$$

as desired. □

The quantum decider $\widetilde{D}$ without actual access to the oracle is defined as follows:

Algorithm 7.2: Simulated-decider for i-th bit: $\widetilde{D}(x,i)$

$\widetilde{D}(x,i)$

1. Invoke $\mathsf{SimProof}(x)$ to get (st,π)
2. Return $D_{(i)}^{|\mathsf{SimRO}(st,\cdot)\rangle}(x,\pi)$.

Lemma 27. *For all (x,w) in the relation $R_{\mathcal{L}_0}$ and all $i\in[p(|x|)]$,*

$$
\Pr\left[\widetilde{D}(x,i)=w_i\right]\geq\frac{2}{3}-\mathsf{negl}(|x|).
$$

Proof. Apply the zero-knowledge property of the NIZK to $D_{(i)}^{|\mathcal{O}\rangle}$ which is polynomial-bound. The probability that this bit equals w_i differs by at most a negligible amount. Combining this with Lem. 26 proves the lemma. □

By Lem. 27, for some negligible function $\mathsf{negl}_{\mathsf{sim}}$, the success probability of $\widetilde{D}(x,i)$ is at least $2/3-\mathsf{negl}_{\mathsf{sim}}(|x|)$. Since $\mathsf{negl}_{\mathsf{sim}}$ is negligible, there exists a constant n_0 such that, for every $n\geq n_0$, we have $\mathsf{negl}_{\mathsf{sim}}(n)\leq 1/15$. Therefore, for every $|x|\geq n_0$,

$$
\frac{2}{3}-\mathsf{negl}_{\mathsf{sim}}(|x|)\geq\frac{2}{3}-\frac{1}{15}=\frac{3}{5}.
$$

This bound is not required to hold when $|x|<n_0$. Instead, for these input lengths we can exhaustively search for the unique witness using the efficiently decidable relation $R_{\mathcal{L}_0}$ as $\mathcal{L}_0\in\mathbf{UP}$. Although this search takes exponential time as a function of $|x|$, its running time on the branch $|x|<n_0$ is bounded by $2^{p(n_0)}\cdot\mathsf{poly}(n_0)$, which is a fixed constant.

We define the amplified quantum decider as follows:

Algorithm 7.3: Amplified decider for i-th bit: $\widehat{D}(x,i)$

$\widehat{D}(x,i)$

1. Let $n = |x|$.
2. If $n < n_0$:
   (a) Enumerate every $w \in \{0,1\}^{\leq p(n)}$.
   (b) If $(x,w) \in R_{\mathcal{L}_0}$, return w_i.
   (c) If no such witness exists, return 0.
3. Prepare an empty multiset $\mathsf{Bits} = \{\}$.
4. For $j \in [n]$:
   (a) Invoke a fresh instance of $\widetilde{D}(x,i)$, and add its output to Bits.
5. Return the majority element in Bits.

We then show that the amplified decider recovers each witness bit with overwhelming probability.

Lemma 28. *For all (x,w) in the relation $R_{\mathcal{L}_0}$ and all $i \in [p(|x|)]$,*

$$\Pr\left[\widehat{D}(x,i) = w_i\right] \geq 1 - \mathsf{negl}(|x|).$$

Proof. For the case that $|x| < n_0$, it is clearly that $\widehat{D}(x,i)$ decide the witness bit w_i perfectly, which means

$$\Pr\left[\widehat{D}(x,i) = w_i\right] = 1$$

For the case that $|x| \geq n_0$,

Let X_j be the indicator random variable for the event that the j-th invocation of $\widetilde{D}(x,i)$ outputs w_i, and let $X = \sum_{j=1}^{|x|} X_j$. Every invocation uses fresh randomness for the simulator and the quantum machine. By Lem. 27, for all sufficiently large $|x|$ we have $\mathbb{E}[X] \geq 3|x|/5$. Therefore, by a Chernoff bound,

$$\begin{aligned}
\Pr\left[\widehat{D}(x,i) \neq w_i\right] &= \Pr\left[X \leq \frac{|x|}{2}\right] \\
&\leq \Pr\left[X \leq \left(1 - \frac{1}{6}\right)\mathbb{E}[X]\right] \\
&\leq \exp\left(-\frac{\mathbb{E}[X]}{72}\right) \\
&\leq \exp\left(-\frac{|x|}{120}\right) \\
&= \mathsf{negl}(|x|).
\end{aligned}$$

which means

$$\Pr\left[\widehat{D}(x,i) = w_i\right] \geq 1 - \mathsf{negl}(|x|).$$

This completes the proof of Lem. 28.

□

We then give the final decision algorithm $D^*(x)$ for deciding the membership of the language $\mathcal{L}_0$ as follows:

Algorithm 7.4: Final witness-recovering decider: $D^*(x)$

$D^*(x)$

1. For $i \in [p(|x|)]$
   (a) $w_i \leftarrow$ output of $\widehat{D}(x, i)$.
2. If there exists a prefix u of w (possibly $u = w$) such that $(x, u) \in R_{\mathcal{L}_0}$, output 1.
   Else output 0

We claim that the above algorithm $D^*(x)$ is a BQP decider for $\mathcal{L}_0$.

For those $x \notin \mathcal{L}_0$

There does not exist any w such that $(x, w) \in R_{\mathcal{L}_0}$ by the definition of **UP**, thus the output of $D^*(x)$ is always 0.
Therefore, we conclude that $D^*(x)$ correctly decides membership in $\mathcal{L}_0$, which is

$$\forall x \notin \mathcal{L}_0, \Pr[D^*(x) = 0] = 1$$

For those $x \in \mathcal{L}_0$

Notice that D^* decides x only if the corresponding uniquen witness w satisfies $(x, w) \in R_{\mathcal{L}_0}$ is recovered.

Formally, for any fixed $x \in L_0$, we analyze the probability of the event that $D^*(x)$ outputs 1:

$$\begin{aligned}
&\Pr\left[D^*(x) = 1\right] \\
&\geq \Pr\left[\forall i \in [p(|x|)], \widehat{D}(x, i) = w_i\right] \\
&= 1 - \Pr\left[\exists i \in [p(|x|)], \widehat{D}(x, i) \neq w_i\right] \\
&\geq 1 - p(|x|) \Pr\left[\widehat{D}(x, i) \neq w_i\right] && \text{(Union Bound)} \\
&\geq 1 - p(|x|) \cdot \mathsf{negl}(|x|) && \text{(Lem. 28)} \\
&\geq 1 - \mathsf{negl}(|x|)
\end{aligned}$$

for some negligible function $\mathsf{negl}(\cdot)$.

Thus, we conclude that $D^*(x)$ correctly decides membership in $\mathcal{L}_0$.

The above two cases imply that D^* is a BQP decider for the language L_0, contradicting the assumption that $L_0 \in \mathbf{UP} \setminus \mathsf{BQP}$. Therefore, for every fixed oracle-aided QPT machine $D^{(\cdot)}$,

$$\Pr_{\mathcal{O}}[\mathcal{O} \in G_D] < \frac{2}{3} = 1 - \frac{1}{3}.$$

It remains to check that $\mathcal{L}^{\mathcal{O}}$ satisfies the two structural conditions of Lem. 22. The language is decidable by an oracle Turing machine that always halts: on input (x, i, π), the machine can enumerate the polynomially bounded witnesses for $\mathcal{L}_0$, check the relation, and run the NIZK verifier. Moreover, a fixed bit of the random oracle belongs to a fixed input-length component. The verifier on an instance x accesses only this component, and the index i and every well-formed proof π have polynomially bounded length; Hence, changing one oracle bit affects only finitely many strings in $\mathcal{L}^{\mathcal{O}}$.

We can now invoke Lem. 22 with $\varepsilon = 1/3$. It follows that, with probability 1 over the choice of $\mathcal{O}$, no oracle-aided QPT machine decides $\mathcal{L}^{\mathcal{O}}$ on all inputs. Equivalently,

$$\Pr_{\mathcal{O}}[\mathcal{L}^{\mathcal{O}} \notin \mathsf{BQP}^{\mathcal{O}}] = 1.$$

□

Finishing the proof of Thm. 4. Assume $\mathsf{UP} \not\subseteq \mathsf{BQP}$. By Sec. 7.4 and 7.5, the events $\mathcal{L}^O \in \mathsf{NP}^O \cap \mathsf{coNP}^O$ and $\mathcal{L}^O \notin \mathsf{BQP}^O$ each have probability 1 over $O \leftarrow \mathcal{O}$. It follows that

$$\Pr_{O \leftarrow \mathcal{O}}\left[\mathsf{NP}^O \cap \mathsf{coNP}^O \not\subseteq \mathsf{BQP}^O\right] = 1.$$

8 Black-Box Impossibility of Constructing Quantum-Hard NP ∩ coNP Problems

We next establish our main impossibility result.

Theorem 5. *There is no fully black-box quantum reduction of any (even worst-case) quantum-hard problem in* $\mathsf{NP} \cap \mathsf{coNP}$ *to post-quantum iO and post-quantum inj-OWFs.*

To prove Thm. 5, we follow the *two-oracle paradigm* introduced by Hsiao and Reyzin [HR04]. This paradigm rules out fully black-box constructions of a primitive Y from a primitive X by exhibiting two oracles: an oracle F that realizes X and an auxiliary oracle G that breaks the proposed construction of Y, while X remains secure even given access to both F and G. A fully black-box construction and its security reduction would yield a contradiction in this oracle world. We use the quantum formulation of this framework due to Hosoyamada and Yamakawa [HY20], in which quantum algorithms may have superposition access to both oracles.

Our auxiliary oracle is a universal decision oracle for complementary verifier pairs. Importantly, it is required to decide only those pairs that remain complementary relative to every implementation of the source primitive under consideration. This is precisely the structure guaranteed by a fully black-box construction.

We start by specifying necessary preliminaries (Sec. 8.1). We then show a warm-up case of separating injOWFs only, without considering iO (Sec. 8.2). In this warm-up, the base oracle is a family of random injective functions. We show that it remains one-way against quantum adversaries even in the presence of the quantum-accessible decision oracle. We then extend the oracle construction to incorporate iO (Sec. 8.3).

8.1 Black-Box Framework and Technical Tools

We first specify the oracle interfaces and the black-box model. We then define the decision oracle and establish the structural property that makes it possible to analyze its effect on source security. Finally, we give a quantum hybrid lemma that converts this structural property into a bound against superposition queries.

8.1.1 Oracle Access and the Black-Box Model

Throughout, a sampled classical oracle is a deterministic function. Quantum access to a classical function h means access to its unitary evaluation oracle

$$O_h\,|k, v\rangle = |k, v \oplus h(k)\rangle\,.$$

When an oracle implements a randomized classical procedure, its randomness is supplied as an explicit input. In particular, an obfuscation oracle has the form $iO(C, r)$, where r is the obfuscator's random tape. Any randomness used to sample an oracle table is sampled once at the beginning of the experiment; the table is then held fixed throughout all queries.

We use post-quantum iO for classical oracle-aided circuits. The adversary may make quantum oracle queries both before and after receiving the obfuscation.

Definition 8 (Post-Quantum Indistinguishability Obfuscation). *Let $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ be a class of classical oracle-aided circuits, where every $C \in \mathcal{C}_n$ has size n. Let $\Gamma = \{\Gamma_n\}_{n\in\mathbb{N}}$ be an oracle distribution ensemble. In the experiments below, the oracle tuple is sampled from Γ_n and is held fixed throughout the experiment.*

A PPT algorithm $i\mathcal{O}$ is a post-quantum indistinguishability obfuscator for $\mathcal{C}$ relative to Γ if the following hold.

1. ***Functionality.*** *For every $n \in \mathbb{N}$ and every $C \in \mathcal{C}_n$,*

$$\Pr_{\Gamma, i\mathcal{O}}\left[\widehat{C}^{\Gamma} \equiv C^{\Gamma} \ : \ \widehat{C} \leftarrow i\mathcal{O}^{\Gamma}(1^n, C)\right] = 1.$$

2. ***Post-Quantum Indistinguishability.*** *For every non-uniform QPT oracle-aided adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that produces $C_0, C_1 \in \mathcal{C}_n$ satisfying $C_0^{\Gamma} \equiv C_1^{\Gamma}$ with probability one,*

$$\mathrm{Adv}^{\mathsf{iO}}_{\Gamma, i\mathcal{O}, \mathcal{C}, \mathcal{A}}(n) = \left|\Pr\left[\mathsf{Exp}^{\mathsf{iO}}_{\Gamma, i\mathcal{O}, \mathcal{C}, \mathcal{A}}(n) = 1\right] - \frac{1}{2}\right| \leq \mathsf{negl}(n),$$

where $\mathsf{Exp}^{\mathsf{iO}}_{\Gamma, i\mathcal{O}, \mathcal{C}, \mathcal{A}}(n)$ is the experiment defined in Expr. 27.

Experiment 27: Security for Post-Quantum iO

$\mathsf{Exp}^{\mathsf{iO}}_{\Gamma, i\mathcal{O}, \mathcal{C}, \mathcal{A}}(n)$:

1. Sample the oracle tuple according to Γ_n.
2. $(C_0, C_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{|\Gamma\rangle}(1^n)$. (The register st may contain quantum side information.)
3. $b \xleftarrow{\$} \{0, 1\}$.
4. $\widehat{C} \leftarrow i\mathcal{O}^{\Gamma}(1^n, C_b)$.
5. $b' \leftarrow \mathcal{A}_2^{|\Gamma\rangle}(\mathsf{st}, \widehat{C})$.
6. Output 1 if $b' = b$, and output 0 otherwise.

We next formalize the black-box constructions ruled out by our result. The verifiers are classical, as required for an $\mathsf{NP} \cap \mathsf{coNP}$ language, whereas the security reduction may be quantum. The universal structure requirement below is essential: the same verifier pair must remain complementary for every functionally correct implementation of the source primitives.

Definition 9 (Quantum Fully Black-Box Construction). *A post-quantum fully black-box construction of a hard $\mathsf{NP} \cap \mathsf{coNP}$ problem from injOWFs and iO for a class $\mathcal{C}$ of classical circuits with injOWF-gates consists of two classical polynomial-time oracle-aided verifiers (V_0, V_1) and a quantum oracle-aided reduction $\mathcal{R}$ satisfying the following conditions. In the efficient formulation, $\mathcal{R}$ is polynomial-time. For the quantitative formulation, we allow an otherwise inefficient reduction and bound relevant oracle-query complexity explicitly with parameters $q_{\mathcal{R}}(n)$ and $\varepsilon_{\mathcal{R}}(n)$ below.*

1. $\mathsf{NP} \cap \mathsf{coNP}$ ***Structure.*** *Let $\mathfrak{S}$ be the family of all oracle tuples (f, iO) satisfying the prescribed syntax and output-length bounds, such that f is injective and iO is a deterministic function of the circuit and its random tape with $\widehat{C}^f \equiv C^f$ for every $C \in \mathcal{C}$, every random tape r, and $\widehat{C}^{(\cdot)} := iO(C, r)$. For every $(f, iO) \in \mathfrak{S}$ and every input z, exactly one bit $b \in \{0,1\}$ satisfies*

$$\exists w : \ V_b^{f,iO}(z, w) = 1.$$

The witnesses have polynomially bounded length. Thus, defining membership by the verifier V_1, the pair (V_0, V_1) defines a language

$$L^{f,iO} \in \mathsf{NP}^{f,iO} \cap \mathsf{coNP}^{f,iO}.$$

2. ***Quantum Black-Box Security Proof.*** *There exist functions $q_{\mathcal{R}}(\cdot) \geq 1$ and $\varepsilon_{\mathcal{R}}(\cdot) > 0$ such that the following holds. (In the efficient formulation, $q_{\mathcal{R}}(n)$ is polynomial and $\varepsilon_{\mathcal{R}}(n)$ is inverse-polynomial.) Let Γ be any distribution over oracle tuples supported on $\mathfrak{S}$. For any quantum oracle-aided adversary $\mathcal{A}$ that decides $L^{f,iO}$ exactly in the worst case, namely, for every input z,*

$$\Pr_{(f,iO)\leftarrow\Gamma}\left[\mathcal{A}^{|f\rangle,|iO\rangle}(z) = b \quad \forall z \in \{0,1\}^n, b \in \{0,1\} \ s.t. \ \exists w : V_b^{f,iO}(z, w) = 1\right] = 1,$$

the reduction breaks either f or iO. More precisely, for infinitely many $n \in \mathbb{N}$, either

$$\Pr_{\substack{(f,iO)\leftarrow\Gamma \\ x\xleftarrow{\$}\{0,1\}^n}}\left[\mathcal{R}^{|\mathcal{A}\rangle,|f\rangle,|iO\rangle}(1^n, f(x)) = x\right] \geq \varepsilon_{\mathcal{R}}(n),$$

or

$$\mathrm{Adv}^{\mathsf{iO}}_{\Gamma,iO,\mathcal{C},\mathcal{R}^{|\mathcal{A}\rangle}}(n) \geq \varepsilon_{\mathcal{R}}(n).$$

In both experiments, $\mathcal{R}$ makes at most $q_{\mathcal{R}}(n)$ queries in total to its oracle interfaces. Moreover, for every computational-basis input z appearing in a query to $\mathcal{A}$, every $(f, iO) \in \mathfrak{S}$, every witness w, and each $b \in \{0,1\}$, the execution $V_b^{f,iO}(z, w)$ makes at most $q_{\mathcal{R}}(n)$ oracle queries.

The oracle interface supplied for a quantum adversary is part of the black-box model; the notation $|\mathcal{A}\rangle$ does not assert a canonical coherent implementation of an arbitrary quantum channel. For a deterministic classical adversary, we use its unitary function-evaluation interface. This is the only adversary interface needed for the decision-oracle attack below.

Remark 3 (Role of the auxiliary quantitative formulation). Def. 9 includes both an efficient and a quantitative formulation. The efficient formulation is the standard quantum fully black-box notion and is the notion targeted by our main separation theorem. The quantitative formulation serves only as an auxiliary framework for proving this separation.

Our quantitative results (e.g., Thm. 6 and Thm. 7) concern this auxiliary formulation *under additional restrictions* on the permitted queries, including bounds on evaluation-query lengths and on the resources of verifiers submitted to the decision oracle. These results exclude otherwise inefficient reductions with certain super-polynomial query bounds $q_{\mathcal{R}}$ satisfying those restrictions. They do *not* rule out the quantitative formulation in general, nor do they establish lower bounds on bare oracle-query complexity for unrestricted inefficient reductions.

These restricted quantitative results nevertheless suffice for our main goal: For every fixed efficient construction and reduction, we show that composition with our separating adversary satisfies all the additional resource restrictions for some polynomial bound $p(n)$. Since this polynomial bound is eventually dominated by the superpolynomial bounds covered by our quantitative results, those

results rule out the standard efficient formulation (e.g., see Corollaries 4 and 5). The inefficient formulation is therefore a tool for establishing the efficient separation, rather than a stronger notion that we claim to rule out in full generality.

The corresponding definition for constructions from injOWFs alone is obtained by omitting the obfuscation oracle and the alternative of breaking iO.

8.1.2 The Universal Decision Oracle

We now define the auxiliary oracle that breaks a proposed $\mathsf{NP} \cap \mathsf{coNP}$ construction. Its validity test depends only on the oracle family $\mathfrak{S}$, rather than on the particular realized oracle. This universal validity requirement is what allows us to control how its answer changes when the realized oracle is modified.

Definition 10 ([BDV17, Definition 4.4]). *For a family of classical oracles $\mathfrak{S}$ and an oracle $\Psi \in \mathfrak{S}$, define the classical function $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$ as follows. Its input is a pair of classical oracle-aided verifier circuits (V_0, V_1) and an input z.*

1. *Check that the circuits have compatible input formats and, for every $\Psi' \in \mathfrak{S}$ and every input z' of their common input length, exactly one bit $b \in \{0,1\}$ satisfies*

$$\exists w : \; V_b^{\Psi'}(z', w) = 1.$$

   If this condition fails, output $\perp$. We call a verifier pair satisfying this condition valid *relative to $\mathfrak{S}$.*

2. *Output the unique bit $b \in \{0,1\}$ for which*

$$\exists w : \; V_b^{\Psi}(z, w) = 1.$$

Malformed inputs are answered by $\perp$. The oracle is defined only for realizations $\Psi \in \mathfrak{S}$.

The definition specifies a function and imposes no computational bound on the validity test or the search for an accepting witness. We give quantum algorithms access to the reversible evaluation oracle of this function.

If a fixed verifier pair remains complementary for every $\Psi' \in \mathfrak{S}$, then its associated language is decidable using one call to $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$. In particular, this applies to every verifier pair satisfying the structure requirement of Def. 9. It does not apply to an arbitrary pair that is complementary only for the realized oracle Ψ, nor does it apply to verifiers that themselves query $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$.

To analyze source security, we distinguish the number of queries made by an adversary from the number made by the verifiers in a decision-oracle query.

Definition 11 (Bounded Decision-Oracle Queries). *A query $Q = (V_0, V_1, z)$ is r-*bounded *relative to $\mathfrak{S}$ if, for every $\Psi \in \mathfrak{S}$, every witness w, and each $b \in \{0,1\}$, the execution $V_b^{\Psi}(z, w)$ makes at most r queries to Ψ.*

*An adversary making at most T quantum queries to the base oracle and the decision oracle is required to have only r-bounded well-formed decision queries in its computational-basis query support, for every oracle realization under consideration. Malformed queries are allowed and receive the fixed answer $\perp$. When $T = r = q$, we call such an adversary q-*query*.*

Although the decision oracle itself is computationally unbounded, its answer to a valid r-bounded query is certified by at most r base-oracle answers. The following consequence of the $\mathsf{NP} \cap \mathsf{coNP}$ structure is the key property used in our analysis.

Lemma 29 ([BDV17, Claim 4.6]). *For every valid r-bounded query $Q = (V_0, V_1, z)$ and every $\Psi \in \mathfrak{S}$, there is a set $C = C(\Psi, V_0, V_1, z)$ of at most r classical oracle queries such that, for every $\Psi' \in \mathfrak{S}$,*

$$\Psi|_C = \Psi'|_C \quad \Longrightarrow \quad \mathsf{Decide}_{\mathfrak{S}}^{\Psi}(Q) = \mathsf{Decide}_{\mathfrak{S}}^{\Psi'}(Q).$$

Proof. We include the proof for completeness. Let $b = \mathsf{Decide}_{\mathfrak{S}}^{\Psi}(V_0, V_1, z)$, and fix an accepting witness w for $V_b^{\Psi}(z, w)$. Let C contain the oracle queries made in this verification. Since the query is r-bounded, $|C| \le r$.

If Ψ' agrees with Ψ on C, the same verification follows the same execution and accepts relative to Ψ'. Since the verifier pair remains complementary for every oracle in $\mathfrak{S}$, the decision bit relative to Ψ' must also be b.

□

8.1.3 A Quantum Hybrid Bound for Coupled Oracles

The classical argument of [BDV17] bounds the probability that an adversary encounters a query on which two oracle realizations disagree. This reasoning does not directly apply to quantum adversaries, whose queries may be superpositions of many inputs: measuring a query to identify a differing point can disturb the computation.

We instead establish a conditional-coupling lemma for the quantum oracle hybrid argument. If the probability of disagreement at every fixed input is at most γ, conditioned on the reference oracle, then any T-query quantum adversary has distinguishing advantage at most $2T\sqrt{\gamma}$. The proof follows the hybrid and projector method underlying [ARU14, Lemmas 35 and 37]. Combined with Lem. 29, this formulation lets us use classical pointwise disagreement bounds even when the adversary has superposition access to the decision oracle.

We first recall the trace-distance estimate used in the proof. Throughout this section, TD denotes trace distance with the $\frac{1}{2}\|\cdot\|_1$ convention.

Lemma 30 ([Unr14, ARU14]). *Let $|\Psi_0\rangle$ and $|\Psi_1\rangle$ be normalized quantum states that can be written as*

$$|\Psi_b\rangle = |\Psi_b^*\rangle + |\Phi^*\rangle\,, \qquad b \in \{0,1\},$$

where both $|\Psi_b^*\rangle$ are orthogonal to $|\Phi^*\rangle$. Then

$$\mathrm{TD}(\,|\Psi_0\rangle\,,\,|\Psi_1\rangle) \le 2\,\||\Psi_1^*\rangle\|\,.$$

Lemma 31 (Quantum Queries to Coupled Oracle Distributions). *Let $\gamma \in [0,1]$, and let Γ_1 and Γ_2 be distributions over classical functions from $\{0,1\}^n$ to $\{0,1\}^m$. Suppose there is a joint distribution μ on pairs (f_1, f_2) whose marginals are Γ_1 and Γ_2, respectively, such that for every f_2' in the support of Γ_2 and every $z \in \{0,1\}^n$,*

$$\Pr_{(f_1,f_2)\leftarrow\mu}\left[f_1(z) \ne f_2(z) \,\middle|\, f_2 = f_2'\right] \le \gamma. \tag{34}$$

Let $\mathcal{A}$ be a quantum oracle algorithm making at most T queries, whose initial state is independent of the sampled oracle pair. If ρ_f denotes its final state with oracle f, then

$$\mathop{\mathbb{E}}_{(f_1,f_2)\leftarrow\mu}\left[\mathrm{TD}(\rho_{f_1}, \rho_{f_2})\right] \le 2T\sqrt{\gamma}.$$

In particular, if $\mathcal{A}$ outputs a bit, then

$$\left|\Pr_{f_1\leftarrow\Gamma_1}[\mathcal{A}^{|f_1\rangle} = 1] - \Pr_{f_2\leftarrow\Gamma_2}[\mathcal{A}^{|f_2\rangle} = 1]\right| \le 2T\sqrt{\gamma}.$$

Proof. We purify the algorithm's initial state and intermediate operations, retaining any purification registers as part of its workspace. It suffices to prove the bound for these purified final states, since discarding registers cannot increase trace distance.

Let A, K, V denote the workspace, query-input, and query-output registers, respectively. For a function f, write

$$O_f \left|a, k, v\right\rangle_{AKV} = \left|a, k, v \oplus f(k)\right\rangle_{AKV}.$$

Absorbing the initial oracle-independent computation into $\left|\Psi^0\right\rangle$, let

$$\left|\Psi_f^0\right\rangle = \left|\Psi^0\right\rangle, \qquad \left|\Psi_f^i\right\rangle = U_i O_f \left|\Psi_f^{i-1}\right\rangle, \quad i \in \{1, \ldots, T\},$$

where the U_i are oracle-independent unitaries. Algorithms making fewer queries can be padded with inactive query slots; the same argument applies to their active-query subspaces.

Hybrid bound. For fixed (f_1, f_2), define $D_i^{f_1,f_2} \coloneqq \mathrm{TD}(\left|\Psi_{f_1}^i\right\rangle, \left|\Psi_{f_2}^i\right\rangle)$. By the triangle inequality and unitary invariance of trace distance,

$$\begin{aligned} D_i^{f_1,f_2} &= \mathrm{TD}\left(O_{f_1}\left|\Psi_{f_1}^{i-1}\right\rangle, O_{f_2}\left|\Psi_{f_2}^{i-1}\right\rangle\right) \\ &\le D_{i-1}^{f_1,f_2} + \mathrm{TD}\left(O_{f_1}\left|\Psi_{f_2}^{i-1}\right\rangle, O_{f_2}\left|\Psi_{f_2}^{i-1}\right\rangle\right). \end{aligned}$$

Since $D_0^{f_1,f_2} = 0$, it follows that

$$D_T^{f_1,f_2} \le \sum_{i=0}^{T-1} \mathrm{TD}\left(O_{f_1}\left|\Psi_{f_2}^i\right\rangle, O_{f_2}\left|\Psi_{f_2}^i\right\rangle\right).$$

Projector bound. For $z \in \{0,1\}^n$, define $Q_z \coloneqq I_A \otimes \left|z\right\rangle\left\langle z\right|_K \otimes I_V$, and define the disagreement projector

$$Q^{f_1,f_2} = \sum_{z:\, f_1(z) \neq f_2(z)} Q_z.$$

Fix i, and abbreviate $Q = Q^{f_1,f_2}$ and $\left|\psi\right\rangle = \left|\Psi_{f_2}^i\right\rangle$. The two states $O_{f_1}\left|\psi\right\rangle$ and $O_{f_2}\left|\psi\right\rangle$ have the common component

$$\left|\Phi^*\right\rangle = O_{f_1}(I - Q)\left|\psi\right\rangle = O_{f_2}(I - Q)\left|\psi\right\rangle.$$

Their remaining components are

$$\left|\Psi_b^*\right\rangle = O_{f_b} Q \left|\psi\right\rangle, \qquad b \in \{1, 2\}.$$

Both oracle unitaries preserve the query-input register. Consequently, each $\left|\Psi_b^*\right\rangle$ is orthogonal to $\left|\Phi^*\right\rangle$ and has norm $\left\|Q\left|\psi\right\rangle\right\|$. By Lem. 30,

$$\mathrm{TD}\left(O_{f_1}\left|\Psi_{f_2}^i\right\rangle, O_{f_2}\left|\Psi_{f_2}^i\right\rangle\right) \le 2\left\|Q^{f_1,f_2}\left|\Psi_{f_2}^i\right\rangle\right\|.$$

Averaging over the coupling. Taking expectation over $(f_1, f_2) \leftarrow \mu$ gives

$$\mathop{\mathbb{E}}_{\mu}[D_T^{f_1,f_2}] \le 2\sum_{i=0}^{T-1} \mathop{\mathbb{E}}_{\mu}\left[\left\|Q^{f_1,f_2}\left|\Psi_{f_2}^i\right\rangle\right\|\right]. \tag{35}$$

By Jensen's inequality,

$$\mathop{\mathbb{E}}_{\mu}\left[\left\|Q^{f_1,f_2}\left|\Psi_{f_2}^i\right\rangle\right\|\right] \le \sqrt{\mathop{\mathbb{E}}_{\mu}\left[\left\|Q^{f_1,f_2}\left|\Psi_{f_2}^i\right\rangle\right\|^2\right]}.$$

The state $|\Psi^i_{f_2}\rangle$ depends only on the reference oracle f_2. Since the projectors Q_z are mutually orthogonal,

$$\begin{aligned}\mathop{\mathbb{E}}_{\mu}\left[\left\|Q^{f_1,f_2}\,|\Psi^i_{f_2}\rangle\right\|^2\right] &= \mathop{\mathbb{E}}_{f_2\leftarrow\Gamma_2}\left[\sum_z \left\|Q_z\,|\Psi^i_{f_2}\rangle\right\|^2 \Pr_{\mu}[f_1(z)\neq f_2(z)\mid f_2]\right]\\ &\le \gamma \mathop{\mathbb{E}}_{f_2\leftarrow\Gamma_2}\left[\sum_z \left\|Q_z\,|\Psi^i_{f_2}\rangle\right\|^2\right]\\ &= \gamma,\end{aligned}$$

where the inequality uses Inequality (34). Substituting into Inequality (35), we obtain

$$\mathop{\mathbb{E}}_{\mu}[D_T^{f_1,f_2}] \le 2T\sqrt{\gamma}.$$

Finally, for any fixed final measurement, the difference in the probability of an outcome is at most the trace distance. Since μ has marginals Γ_1 and Γ_2,

$$\left|\Pr_{f_1\leftarrow\Gamma_1}[\mathcal{A}^{|f_1\rangle}=1]-\Pr_{f_2\leftarrow\Gamma_2}[\mathcal{A}^{|f_2\rangle}=1]\right| \le \mathop{\mathbb{E}}_{(f_1,f_2)\leftarrow\mu}\left[D_T^{f_1,f_2}\right]\le 2T\sqrt{\gamma}.$$

This completes the proof of Lem. 31.

□

8.2 Warm-Up: Separation from injOWFs

This subsection serves as a warm-up for the Sec. 8.3 by demonstrating the main technical tool underlying our black-box separation. We construct an injective function oracle that remains one-way against quantum queries even in the presence of the universal decision oracle. Since the decision oracle decides every verifier pair arising from a fully black-box construction, this suffices to rule out such constructions from injOWFs alone. In the next subsection, we strengthen the result by incorporating iO.

Notation. Let $\mathbf{I}_n^m$ denote the uniform distribution on all injective functions from $\{0,1\}^n$ to $\{0,1\}^m$. In this subsection, $\mathfrak{S}$ denotes the family of full oracle ensembles

$$f=\{f_k\}_{k\in\mathbb{N}},\qquad f_k:\{0,1\}^k\longrightarrow\{0,1\}^{k+1},$$

where each f_k is injective. The random injOWF oracle is sampled by choosing independently

$$f_k\leftarrow\mathbf{I}_k^{k+1}\qquad\text{for every } k\in\mathbb{N}.$$

We write $f(z)=f_{|z|}(z)$.

For the security proof at input length n, we prove a stronger statement: all components

$$f_{-n}\coloneqq\{f_k : k\neq n\}$$

may be fixed arbitrarily, and only f_n is sampled at random. This allows the adversary and the verifiers in its decision queries to access the entire oracle family, including input lengths other than n.

The following Thm. 6 is the post-quantum analogue of the injOWF security statement in [BDV17, Theorem 4.8]. Its role in the separation is to establish source security in the two-oracle construction. The other requirement follows from the definition of $\mathsf{Decide}^f_{\mathfrak{S}}$: every universally complementary verifier pair can be decided with one oracle query. We give the resulting black-box contradiction after the proof.

Theorem 6. *Fix $n \in \mathbb{N}$, arbitrary injective components f_{-n}, and integers $T \geq 0$ and $r \geq 1$. Let $\mathcal{A}$ be an otherwise computationally unbounded quantum oracle adversary making at most T queries in total to f and $\mathsf{Decide}^f_{\mathfrak{S}}$, with all well-formed decision-oracle queries r-bounded. The description and initial state of $\mathcal{A}$ are independent of the sampled component f_n. Then*

$$\Pr_{\substack{f_n \leftarrow \mathbf{I}_n^{n+1} \\ x \xleftarrow{\$} \{0,1\}^n}} \left[\mathcal{A}^{|f\rangle, |\mathsf{Decide}^f_{\mathfrak{S}}\rangle}(f(x)) = x \right] \leq 2(T+1)\sqrt{\frac{r}{2^n}}.$$

In particular, if $T, r \leq q(n) = O(2^{n/6})$, the inversion probability is $O(2^{-n/4})$. The same bounds hold when all components of f are sampled independently as above.

Proof. Fix n, f_{-n}, and $\mathcal{A}$ as in the theorem. The proof first samples a fresh value y outside $\mathrm{Image}(f_n)$ and plants it at a uniformly random input $x \in \{0,1\}^n$. The planted experiment has the same distribution as the real inversion experiment. We then remove the planting while keeping the challenge y. In the resulting experiment, y has no preimage under f_n. Thus, a successful inverter distinguishes the two experiments by making one additional query to check its proposed preimage.

The key structural ingredient is Lem. 29. For each fixed reference oracle f, every valid r-bounded decision query has a critical set of at most r inputs. Planting at a uniformly random n-bit input therefore changes the answer to any fixed decision query with probability at most $r/2^n$. We apply Lem. 31 to the combined oracle $(f, \mathsf{Decide}^f_{\mathfrak{S}})$ to obtain a quantum distinguishing bound of

$$2(T+1)\sqrt{r/2^n}.$$

Hybrid H_0. This is the real inversion experiment. The components f_{-n} remain fixed throughout all hybrids.

Experiment 28: Hybrid H_0

1. $f_n \leftarrow \mathbf{I}_n^{n+1}$.
2. $x \xleftarrow{\$} \{0,1\}^n$.
3. Invoke $\mathcal{A}^{|f\rangle, |\mathsf{Decide}^f_{\mathfrak{S}}\rangle}(f(x))$.

The adversary wins if its output equals x.

Hybrid H_1. We sample y uniformly outside $\mathrm{Image}(f_n)$ and plant it at a uniformly random input x. The resulting full oracle g agrees with f everywhere except at x. The difference from H_0 is highlighted in red color.

Experiment 29: Hybrid H_1

1. $f_n \leftarrow \mathbf{I}_n^{n+1}$.
2. $x \xleftarrow{\$} \{0,1\}^n$.
3. $y \xleftarrow{\$} \{0,1\}^{n+1} \setminus \mathrm{Image}(f_n)$.
4. Define the full oracle g by $g(z) = \begin{cases} y & \text{if } z = x, \\ f(z) & \text{otherwise.} \end{cases}$
5. Invoke $\mathcal{A}^{|g\rangle, |\mathsf{Decide}^g_{\mathfrak{S}}\rangle}(y)$.

Again, the adversary wins if its output equals x. Since $y \notin \mathrm{Image}(f_n)$, the modified component g_n is injective and $g \in \mathfrak{S}$.

Claim 25. *The joint distributions of the base oracle, the planted point, and the challenge satisfy*

$$(f, x, f(x)) \text{ in } H_0 \overset{i.d.}{=\!=} (g, x, y) \text{ in } H_1.$$

Consequently, the adversary's winning probabilities in H_0 and H_1 are equal.

Proof. For each fixed x, the modified function g_n is uniformly distributed over $\mathbf{I}_n^{n+1}$. Indeed, for any fixed injective g_n, its challenge value must be $y = g_n(x)$, and its possible predecessors f_n are obtained by replacing $g_n(x)$ with any value outside $\mathrm{Image}(g_n)$. There are exactly 2^n such predecessors, each having the same sampling probability.

Thus, g_n is uniform and independent of the uniform point x. Since $y = g_n(x)$ and $g_{-n} = f_{-n}$, the stated joint distributions are identical. The decision oracle is determined by the full underlying oracle, so including it preserves this equality of distributions.

This completes the proof of Claim 25.

□

Hybrid H_2. We remove the planting and give the adversary access to the reference oracle f, while keeping the challenge y outside $\mathrm{Image}(f_n)$. The difference from H_1 is highlighted in red color.

Experiment 30: Hybrid H_2
1. $f_n \leftarrow \mathbf{I}_n^{n+1}$.
2. $x \xleftarrow{\$} \{0,1\}^n$.
3. $y \xleftarrow{\$} \{0,1\}^{n+1} \setminus \mathrm{Image}(f_n)$.
4. Invoke $\mathcal{A}^{\lvert f\rangle, \lvert \mathsf{Decide}^{f}_{\mathfrak{S}}\rangle}(y)$.

To compare these hybrids, we turn the inverter into a bit-output distinguisher. Given a challenge y and access to an underlying oracle h and its decision oracle, define $\mathcal{B}$ as follows:

1. Run $\mathcal{A}^{\lvert h\rangle, \lvert \mathsf{Decide}^{h}_{\mathfrak{S}}\rangle}(y)$ and obtain an output a.
2. If $a \notin \{0,1\}^n$, output 0.
3. Query $h(a)$ and output 1 if and only if $h(a) = y$.

The algorithm $\mathcal{B}$ makes at most $T + 1$ oracle queries, and its decision-oracle queries remain r-bounded. In the following claim, the subscripts H_1 and H_2 specify the sampling of the oracle and challenge in the corresponding hybrids.

Claim 26. *It holds that*

$$\left| \Pr_{H_1}\left[\mathcal{B}^{\lvert g\rangle, \lvert \mathsf{Decide}^{g}_{\mathfrak{S}}\rangle}(y) = 1\right] - \Pr_{H_2}\left[\mathcal{B}^{\lvert f\rangle, \lvert \mathsf{Decide}^{f}_{\mathfrak{S}}\rangle}(y) = 1\right] \right| \le 2(T+1)\sqrt{\frac{r}{2^n}}.$$

Proof. We apply Lem. 31 to a tagged oracle combining the base oracle and the decision oracle. For every $h \in \mathfrak{S}$, define

$$\widetilde{F}_h(0, z) = h(z), \qquad \widetilde{F}_h(1, Q) = \mathsf{Decide}^{h}_{\mathfrak{S}}(Q),$$

where $Q = (V_0, V_1, z)$ is an r-bounded query. Malformed inputs receive the fixed answer $\perp$. Encodings of non-r-bounded decision queries can also be assigned $\perp$ when extending $\widetilde{F}_h$ to a total function, since they never occur in the query support of the algorithms under consideration.

For each fixed input length, the querying algorithm has finite query registers. We pad all relevant encodings and answers to fixed lengths. Thus $\widetilde{F}_h$ can be viewed as a classical function on a finite fixed-length domain, and access to the two oracle interfaces is simulated by access to $\widetilde{F}_h$ without increasing the number of queries.

Conditioning on the challenge. Fix $y \in \{0,1\}^{n+1}$. Since every injective f_n has exactly 2^n values outside its image, the conditional distribution of f_n given the challenge y is

$$f_n \leftarrow \mathbf{I}_n^{n+1} \mid y \notin \mathrm{Image}(f_n).$$

The point x remains uniform and independent of f under this conditioning.

Let $\Gamma_{1,y}$ and $\Gamma_{2,y}$ be the corresponding distributions of merged oracles:

$$\begin{aligned}
\Gamma_{1,y}: \quad & f_n \leftarrow \mathbf{I}_n^{n+1} \mid y \notin \mathrm{Image}(f_n), \qquad x \xleftarrow{\$} \{0,1\}^n, \\
& g = f_{x \to y}, \qquad \text{output } \widetilde{F}_g; \\
\Gamma_{2,y}: \quad & f_n \leftarrow \mathbf{I}_n^{n+1} \mid y \notin \mathrm{Image}(f_n), \qquad \text{output } \widetilde{F}_f.
\end{aligned}$$

Here f_{-n} is the fixed collection of components from the theorem statement, and $f_{x \to y}$ changes only the value at x. We couple these distributions by sampling the same f and a uniform x and outputting

$$(\widetilde{F}_g, \widetilde{F}_f), \qquad g = f_{x \to y}.$$

Pointwise disagreement. Fix the full reference oracle f in the support of the conditional distribution, and fix a merged-oracle input w. We claim that

$$\Pr_{x \xleftarrow{\$} \{0,1\}^n} \left[\widetilde{F}_g(w) \neq \widetilde{F}_f(w) \right] \leq \frac{r}{2^n}. \tag{36}$$

There are three cases.

– If $w = (0, z)$ is a direct query to the base oracle and $|z| = n$, then

$$\Pr_x[g(z) \neq f(z)] = \Pr_x[x = z] = 2^{-n} \leq r 2^{-n}.$$

If $|z| \neq n$, the disagreement probability is zero.

– If $w = (1, Q)$ is an r-bounded decision query whose verifier pair is invalid relative to $\mathfrak{S}$, both answers are $\perp$. Indeed, the validity test depends only on the verifier pair and the family $\mathfrak{S}$, not on the realized oracle. Malformed inputs likewise have disagreement probability zero.

– Suppose $w = (1, Q)$ is a valid r-bounded query, where $Q = (V_0, V_1, z)$. Let $C_Q = C(f, V_0, V_1, z)$ be the critical set from Lem. 29. Then $|C_Q| \leq r$. Since $g \in \mathfrak{S}$ and f and g differ only at x, the condition $x \notin C_Q$ implies $\mathsf{Decide}^f_{\mathfrak{S}}(Q) = \mathsf{Decide}^g_{\mathfrak{S}}(Q)$. Consequently,

$$\Pr_x \left[\mathsf{Decide}^f_{\mathfrak{S}}(Q) \neq \mathsf{Decide}^g_{\mathfrak{S}}(Q) \right] \leq \Pr_x[x \in C_Q] = \frac{|C_Q \cap \{0,1\}^n|}{2^n} \leq \frac{r}{2^n}.$$

Inputs outside the allowed query domain have a fixed answer in both merged oracles and hence also have disagreement probability zero. This proves Inequality (36).

The estimate holds for every fixed full reference oracle f. Since $\widetilde{F}_f$ is a deterministic function of f, averaging over reference oracles with the same merged-oracle realization gives the same estimate conditioned on $\widetilde{F}_f$. Thus the coupling satisfies the hypothesis of Lem. 31 with $\gamma = \min\{1, r2^{-n}\}$.

Applying the quantum hybrid bound. Let $\mathcal{B}_y$ be the algorithm obtained by hardwiring y as the challenge input to $\mathcal{B}$ and routing its queries through the merged oracle. It makes at most $T+1$ queries. By Lem. 31,

$$\left| \Pr_{\widetilde{F}_g \leftarrow \Gamma_{1,y}} [\mathcal{B}_y^{|\widetilde{F}_g\rangle} = 1] - \Pr_{\widetilde{F}_f \leftarrow \Gamma_{2,y}} [\mathcal{B}_y^{|\widetilde{F}_f\rangle} = 1] \right| \leq 2(T+1)\sqrt{\gamma} \leq 2(T+1)\sqrt{\frac{r}{2^n}}.$$

Finally, the challenge has the same uniform marginal distribution in both hybrids. For every fixed $y \in \{0,1\}^{n+1}$,

$$\Pr[y \text{ is the sampled challenge}] = \frac{\Pr_{f_n}[y \notin \text{Image}(f_n)]}{2^n} = \frac{1/2}{2^n} = 2^{-(n+1)}.$$

Averaging the preceding bound over y completes the proof of Claim 26.

□

Completing the proof of Thm. 6. In H_1, the function g_n is injective and $g_n(x) = y$. Therefore,

$$\Pr_{H_1}\left[\mathcal{B}^{|g\rangle,|\mathsf{Decide}^g_{\mathfrak{S}}\rangle}(y) = 1\right] = \Pr[H_1 \text{ wins}].$$

In H_2, the challenge y is outside $\text{Image}(f_n)$, and $\mathcal{B}$ rejects every output outside $\{0,1\}^n$. Hence

$$\Pr_{H_2}\left[\mathcal{B}^{|f\rangle,|\mathsf{Decide}^f_{\mathfrak{S}}\rangle}(y) = 1\right] = 0.$$

Combining Claims 25 and 26, we obtain

$$\Pr[H_0 \text{ wins}] = \Pr[H_1 \text{ wins}] \leq 2(T+1)\sqrt{\frac{r}{2^n}}.$$

The probability is also at most 1, proving the quantitative bound.

If $T, r \leq q(n) = O(2^{n/6})$, then

$$2(T+1)\sqrt{\frac{r}{2^n}} \leq 2(q(n)+1)\sqrt{\frac{q(n)}{2^n}} = O(2^{-n/4}).$$

The bound holds uniformly over the fixed components f_{-n}, so averaging over independently sampled components gives the same result for the full random oracle ensemble.

This completes the proof of Thm. 6.

□

Consequence for black-box constructions. We now relate the security theorem to the goal of this subsection.

Corollary 4. *There is no fully black-box quantum construction of a worst-case quantum-hard language in* $\mathsf{NP} \cap \mathsf{coNP}$ *from post-quantum injOWFs, in the sense of the injOWF-only version of Def. 9.*

Proof. Suppose such a construction $(V_0, V_1, \mathcal{R})$ exists. Instantiate its source oracle by the random injective family f defined above.

By the structure requirement, (V_0, V_1) remains complementary for every $f \in \mathfrak{S}$. Thus an adversary can decide the constructed language exactly by submitting the corresponding verifier circuits and input to $\mathsf{Decide}^f_{\mathfrak{S}}$. For each input length, the polynomial-time verifier machines are compiled into classical oracle-aided circuits. The resulting decision procedure has a unitary implementation using the decision oracle, so it also supplies the coherent interface needed by the quantum reduction.

Replace every call by $\mathcal{R}$ to this adversary with its decision-oracle implementation. The resulting inverter makes polynomially many queries to f and $\mathsf{Decide}^f_{\mathfrak{S}}$, and every decision-oracle query contains verifiers making polynomially many base-oracle queries. By Thm. 6, its inversion probability is $O(2^{-n/4})$.

On the other hand, the black-box security requirement gives inverse-polynomial inversion probability for infinitely many input lengths, since the adversary decides the constructed language exactly. This is a contradiction.

This completes the proof of Corollary 4.

□

8.3 Separation from iO and injOWFs

We now extend the warm-up construction to include iO. Following [BDV17], we augment the injective function oracle with an injective encoding oracle and an evaluation oracle. We show that both source primitives remain secure against quantum adversaries with access to the universal decision oracle.

We first specify the oracle family and the resource bounds used in the security analysis. We then prove one-wayness by a coherent simulation using the warm-up construction, and prove obfuscation security by a sequence of coupled-oracle hybrids. Finally, we show that every efficient fully black-box reduction induces an adversary satisfying these resource bounds for some polynomial parameter.

The quantitative bounds below concern a resource model that controls evaluation-query lengths as well as oracle-query counts. They do not assert a lower bound on bare oracle-query complexity for unrestricted inefficient reductions.

8.3.1 Oracle Construction and Resource Bounds

Oracle setting. We use full oracle families indexed by all input lengths. As in the warm-up, let

$$f = \{f_k\}_{k\in\mathbb{N}}, \qquad f_k : \{0,1\}^k \longrightarrow \{0,1\}^{k+1},$$

where each f_k is injective. Let

$$\mathcal{O} = \{\mathcal{O}_k\}_{k\in\mathbb{N}}, \qquad \mathcal{O}_k : \{0,1\}^{2k} \longrightarrow \{0,1\}^{5k},$$

where each $\mathcal{O}_k$ is injective. We parse an input to $\mathcal{O}_k$ as a pair $(C, r) \in \{0,1\}^k \times \{0,1\}^k$.

We fix a padded circuit-description convention under which a k-bit description specifies a classical circuit with f-gates and at most k oracle queries on every input. Malformed circuit descriptions denote a fixed zero circuit with the prescribed interface. Incompatible evaluation inputs receive $\perp$. These conventions make circuit evaluation a total function and apply equally to circuits encoded by $\mathcal{O}$ and to the backup circuits defined below.

A *backup map* is a family

$$\varphi = \{\varphi_k\}_{k\in\mathbb{N}}, \qquad \varphi_k : \{0,1\}^{5k} \longrightarrow \{0,1\}^k.$$

For $\widehat{C} \in \{0,1\}^{5k}$, define

$$\mathsf{Eval}_{\varphi}^{f,\mathcal{O}}(\widehat{C}, z) = \begin{cases} C^f(z) & \text{if } \mathcal{O}_k(C,r) = \widehat{C} \text{ for some } (C,r), \\ \big(\varphi_k(\widehat{C})\big)^f(z) & \text{if } \widehat{C} \notin \mathrm{Image}(\mathcal{O}_k). \end{cases}$$

Injectivity of $\mathcal{O}_k$ makes the first case unambiguous. Malformed oracle inputs receive $\perp$.

Write

$$\Psi(f, \mathcal{O}, \varphi) \coloneqq \big(f, \mathcal{O}, \mathsf{Eval}_{\varphi}^{f,\mathcal{O}}\big), \qquad \Psi_{\varphi} = \Psi(f, \mathcal{O}, \varphi).$$

The underlying oracle family is

$$\mathfrak{S}_{\Psi} \coloneqq \left\{ \Psi(f,\mathcal{O},\varphi) : \begin{array}{c} f_k : \{0,1\}^k \to \{0,1\}^{k+1} \text{ is injective for every } k, \\ \mathcal{O}_k : \{0,1\}^{2k} \to \{0,1\}^{5k} \text{ is injective for every } k, \\ \varphi_k : \{0,1\}^{5k} \to \{0,1\}^k \text{ is arbitrary for every } k \end{array} \right\}.$$

In particular, $\mathfrak{S}_{\Psi}$ ranges over all backup maps, even when a security experiment fixes one particular φ. We use this same family in the validity test of $\mathsf{Decide}_{\mathfrak{S}_{\Psi}}$ throughout every experiment and hybrid.

For a fixed backup map φ, the random oracle realization is obtained by sampling independently

$$f_k \leftarrow \mathbf{I}_k^{k+1}, \qquad \mathcal{O}_k \leftarrow \mathbf{I}_{2k}^{5k} \qquad \text{for every } k \in \mathbb{N}.$$

All sampled tables remain fixed throughout the experiment. Define the augmented oracle tuple

$$\Gamma(f, \mathcal{O}, \varphi) \coloneqq \left(f, \mathcal{O}, \mathsf{Eval}_{\varphi}^{f,\mathcal{O}}, \mathsf{Decide}_{\mathfrak{S}_{\Psi}}^{\Psi(f,\mathcal{O},\varphi)}\right),$$

and let Γ_{φ} denote its distribution under this sampling.

Obfuscation and the implementation interface. On input a circuit description $C \in \{0,1\}^n$, the obfuscator samples $r \xleftarrow{\$} \{0,1\}^n$, computes $\widehat{C} = \mathcal{O}_n(C, r)$, and returns the canonical oracle-aided wrapper $E_{\widehat{C}}$ that evaluates $E_{\widehat{C}}^{\Psi_{\varphi}}(z) = \mathsf{Eval}_{\varphi}^{f,\mathcal{O}}(\widehat{C}, z)$. We denote this obfuscator by $i\mathcal{O}$. For every realization in $\mathfrak{S}_{\Psi}$, every circuit C, and every random tape r, $E_{\mathcal{O}_n(C,r)}^{\Psi_{\varphi}} \equiv C^f$. Thus functionality holds for every admissible implementation, not only with high probability over the sampled tables.

The input circuit has only f-gates, whereas the returned wrapper has an Eval-gate. Accordingly, returned programs execute relative to the ambient oracle tuple. This is the implementation convention used when applying the black-box framework in this subsection: calls to obfuscation and executions of returned programs are compiled into calls to $\mathcal{O}$ and Eval, respectively. It is not a claim that $E_{\widehat{C}}$ is an f-only circuit.

The wrapper has size polynomial in n, and its description can be constructed from $\widehat{C}$ and parsed to recover $\widehat{C}$. Hence giving the adversary $\widehat{C}$ in the security experiment is equivalent to giving it the canonical wrapper $E_{\widehat{C}}$.

Reprogramming notation. For a fixed $r \in \{0,1\}^n$, write

$$T = (*, r) \coloneqq \{(C, r) : C \in \{0,1\}^n\},$$

ordered lexicographically by C. If D is an ordered tuple of 2^n distinct elements of $\{0,1\}^{5n}$, then $\mathcal{O}_{T \mapsto D}$ maps each element of T to the corresponding entry of D and agrees with $\mathcal{O}$ elsewhere. Sampling an ordered subset means sampling uniformly without replacement, including a uniformly random ordering.

Similarly, $\mathcal{O}_{(C,r) \mapsto \widehat{C}}$ changes only the value at (C, r), and $\varphi_{\widehat{C} \mapsto C}$ changes only the backup circuit assigned to $\widehat{C}$. All these modifications leave components at other lengths unchanged.

Resource bounds. Unlike the warm-up in Sec. 8.2, a single evaluation query can invoke a circuit containing many f-gates. We therefore bound the lengths of evaluation encodings in addition to the number of external oracle queries.

Definition 12 (Resource-Bounded Queries). *Let $q \geq 1$. In this subsection, a quantum adversary with access to $\Gamma(f, \mathcal{O}, \varphi)$ is* q-query *if it satisfies the following conditions.*

1. ***Direct function queries.*** *It makes at most q queries to f.*
2. ***Encoding and evaluation queries.*** *It makes at most q queries in total to $\mathcal{O}$ and* Eval*. Every well-formed evaluation query $(\widehat{C}, z)$ has $|\widehat{C}| \leq 5q$. Thus the selected circuit, whether recovered from $\mathcal{O}$ or supplied by φ, makes at most q queries to f.*
3. ***Decision queries.*** *It makes at most q queries to $\mathsf{Decide}_{\mathfrak{S}_\Psi}$. For every well-formed query (V_0, V_1, z), every $\Psi \in \mathfrak{S}_\Psi$, every witness w, and each $b \in \{0,1\}$, the execution $V_b^{\Psi}(z, w)$ satisfies the first two conditions above. The verifiers have access only to Ψ, not to the decision oracle.*

For quantum queries, the restrictions apply to every computational-basis query in the support of the query state, including in the hybrid executions used below. Malformed queries are allowed and receive $\perp$.

For a two-stage adversary, the query bounds apply to the two stages combined.

Thus a q-query adversary makes at most $3q$ queries to the augmented tuple. Each verifier in a decision query makes at most $2q$ queries to the underlying tuple Ψ. Consequently, Lem. 29 supplies a critical set of size at most $2q$ for every valid decision query.

The adversaries considered below may otherwise be computationally unbounded. Their descriptions and initial states are independent of the sampled oracle component, apart from information obtained through the stated experiment.

8.3.2 One-Wayness in the Presence of the Augmented Oracle

We first show that access to the encoding, evaluation, and decision interfaces does not destroy one-wayness. The proof fixes $\mathcal{O}$ and φ and simulates their interfaces using the warm-up oracle. The simulation must preserve the validity test of the full decision oracle, including on invalid verifier pairs.

Theorem 7. *Fix $n \in \mathbb{N}$, an integer $q \geq 1$, a backup map φ, an injective encoding family $\mathcal{O}$, and arbitrary injective components f_{-n}. Let $\mathcal{A}$ be a q-query quantum adversary whose description and initial state are independent of f_n. Then*

$$\Pr_{\substack{f_n \leftarrow \mathbf{I}_n^{n+1} \\ x \xleftarrow{\$} \{0,1\}^n}} \left[\mathcal{A}^{|\Gamma(f,\mathcal{O},\varphi)\rangle}(f_n(x)) = x \right] \leq O\left(q^3 2^{-n/2}\right).$$

In particular, if $q(n) = O(2^{n/12})$, the inversion probability is $O(2^{-n/4})$. The same bounds hold when all components of f and $\mathcal{O}$ are sampled independently as above, with φ fixed.

Proof. Fix $n, q, \varphi, \mathcal{O}, f_{-n}$, and $\mathcal{A}$ as in the theorem. Let $\mathfrak{S}_f$ denote the injective-function family from Sec. 8.2. We construct a quantum adversary $\mathcal{B}^{|f\rangle, |\mathsf{Decide}^f_{\mathfrak{S}_f}\rangle}$ that perfectly simulates $\mathcal{A}^{|\Gamma(f,\mathcal{O},\varphi)\rangle}$. The simulator makes $O(q^2)$ queries in total, and each translated decision query is $O(q^2)$-bounded relative to $\mathfrak{S}_f$. Applying Thm. 6 with these two bounds gives the result.

Fixed tables and oracle-independent computation. At the fixed security parameter, all query registers have finite length. The relevant portions of $\mathcal{O}$, its inverse, and φ can therefore be hardwired into a finite simulation circuit. This circuit may be inefficient, as permitted by the query model, but its description is independent of the sampled component f_n.

The same observation applies to the predicate $\mathsf{Valid}_{\mathfrak{S}_\Psi}(V_0, V_1)$ from Def. 10. On the finite set of verifier descriptions that fit in the query registers, this predicate has a finite truth table. We hardwire that table and evaluate it reversibly without queries to f. This does not require an efficient or uniform algorithm for deciding universal validity.

Simulation of the base interfaces. Given the challenge $y = f_n(x)$, the simulator runs $\mathcal{A}(y)$ and answers its queries as follows.

1. **f-queries.** The simulator forwards the query to its own f oracle.
2. **$\mathcal{O}$-queries.** The simulator answers using the fixed encoding tables. This requires no queries to f.
3. **Eval-queries.** On input $(\widehat{C}, z)$, the simulator determines the circuit

$$C = \begin{cases} \pi_1\big(\mathcal{O}^{-1}(\widehat{C})\big) & \text{if } \widehat{C} \in \mathrm{Image}(\mathcal{O}), \\ \varphi(\widehat{C}) & \text{otherwise.} \end{cases}$$

It then evaluates $C^f(z)$. By the evaluation-length restriction, this circuit makes at most q queries to f.

To implement the oracle response coherently, the simulator reversibly selects C, evaluates it while retaining the computation history, XORs the result into the answer register, and reverses the evaluation and selection. All workspace is restored to zero. The resulting map is

$$|\widehat{C}, z, v\rangle\ |0\rangle \longmapsto\ |\widehat{C}, z, v \oplus \mathsf{Eval}_\varphi^{f,\mathcal{O}}(\widehat{C}, z)\rangle\ |0\rangle\,.$$

This costs $O(q)$ queries to f per evaluation query, including uncomputation.

The same construction handles superpositions of different query inputs. Controlled routing between interfaces incurs at most constant-factor query overhead.

Simulation of the decision interface. For a well-formed query $Q = (V_0, V_1, z)$, the simulator first reversibly computes $\mathsf{Valid}_{\mathfrak{S}_\Psi}(V_0, V_1)$. If the pair is invalid, it returns $\perp$.

Suppose the pair is valid. The simulator compiles each V_b into an f-aided verifier D_b by replacing its $\mathcal{O}$- and Eval-queries with the classical simulations described above. For every $f' \in \mathfrak{S}_f$, every input z', and every witness w,

$$D_b^{f'}(z', w) = V_b^{\Psi(f', \mathcal{O}, \varphi)}(z', w).$$

Since $\Psi(f', \mathcal{O}, \varphi) \in \mathfrak{S}_\Psi$ and (V_0, V_1) is valid relative to $\mathfrak{S}_\Psi$, the pair (D_0, D_1) is valid relative to $\mathfrak{S}_f$. Therefore,

$$\mathsf{Decide}_{\mathfrak{S}_f}^{f}(D_0, D_1, z) = \mathsf{Decide}_{\mathfrak{S}_\Psi}^{\Psi(f, \mathcal{O}, \varphi)}(V_0, V_1, z).$$

The original validity test is essential here. An invalid pair relative to $\mathfrak{S}_\Psi$ can become valid after $\mathcal{O}$ and φ are fixed. Querying the translated pair without first checking its original validity would therefore not simulate the full decision oracle.

The translation of a valid query is computed reversibly while retaining the original query. The simulator evaluates the translated decision oracle into the answer register and then uncomputes the

translation and validity flag. For invalid or malformed queries, it instead returns the fixed answer $\perp$ and cleans its workspace. Thus this simulation implements the exact unitary function-evaluation interface, including on superpositions of valid and invalid queries.

Query accounting. The simulator forwards at most q direct f-queries and simulates at most q evaluation queries, each at cost $O(q)$. It also makes $O(q)$ queries to $\mathsf{Decide}^f_{\mathfrak{S}_f}$. Hence its total query count satisfies $T_{\mathcal{B}} = O(q^2)$. For a translated verifier D_b, the direct f-queries contribute at most q, and the simulated evaluation queries contribute at most q^2. Thus every translated decision query is $r_{\mathcal{B}}$-bounded for $r_{\mathcal{B}} = O(q^2)$. This bound holds for every $f' \in \mathfrak{S}_f$ and every witness on the submitted input. The verifier's internal queries are not additional queries made by $\mathcal{B}$; they enter the warm-up theorem through the separate parameter $r_{\mathcal{B}}$.

Applying the warm-up theorem. The simulation is exact and its initial state is independent of f_n. By Thm. 6,

$$\begin{aligned}\Pr\left[\mathcal{A}^{|\Gamma(f,\mathcal{O},\varphi)\rangle}(f_n(x)) = x\right] &= \Pr\left[\mathcal{B}^{|f\rangle,|\mathsf{Decide}^f_{\mathfrak{S}_f}\rangle}(f_n(x)) = x\right] \\ &\le 2(T_{\mathcal{B}}+1)\sqrt{\frac{r_{\mathcal{B}}}{2^n}} \\ &= O\left(q^3 2^{-n/2}\right).\end{aligned}$$

Here the probabilities average over $f_n \leftarrow \mathbf{I}_n^{n+1}$ and $x \xleftarrow{\$} \{0,1\}^n$.

For $q(n) = O(2^{n/12})$, the last expression is $O(2^{-n/4})$. The estimate is uniform over the fixed choices of f_{-n} and $\mathcal{O}$, so averaging over their independent sampling gives the same bound for the full random oracle ensemble.

This completes the proof of Thm. 7.

□

8.3.3 Obfuscation Security in the Presence of the Augmented Oracle

We next prove post-quantum obfuscation security. The proof follows the row-reprogramming hybrids of [BDV17], using Lem. 31 to control quantum queries.

The first reprogramming hides all encodings using a random tape r during the pre-challenge phase. This allows us to replace the challenge encoding by a fresh value without changing the joint distribution of the challenge, post-challenge oracle, and residual quantum state. We then restore the original pre-challenge oracle. Finally, we replace the challenge encoding by a backup evaluation of C_0. Functional equivalence of C_0 and C_1 makes the resulting experiment independent of the challenge bit.

Theorem 8. *Fix $n \in \mathbb{N}$, an integer $q \ge 1$, a backup map φ, an injective family f, and injective encoding components $\mathcal{O}_{-n}$. Sample $\mathcal{O}_n \leftarrow \mathbf{I}_{2n}^{5n}$. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a q-query quantum adversary whose description and initial state are independent of $\mathcal{O}_n$. In the real experiment, $\mathcal{A}_1$ outputs equal-size circuit descriptions $C_0, C_1 \in \{0,1\}^n$ satisfying $C_0^f \equiv C_1^f$ with probability one. Then, it hold that*

$$\left|\Pr\left[\mathsf{Exp}^{\mathsf{iO}}(n) = 1\right] - \frac{1}{2}\right| \le O\left(q^{3/2} 2^{-n/2}\right).$$

The experiment uses the augmented oracle $\Gamma(f, \mathcal{O}, \varphi)$ and the obfuscator $i\mathcal{O}$.

In particular, for $q(n) = O(2^{n/6})$, the advantage is $O(2^{-n/4})$. Averaging over independently sampled f and $\mathcal{O}_{-n}$ gives the same bound for $\mathrm{Adv}^{\mathsf{iO}}_{\Gamma_\varphi, i\mathcal{O}, \mathcal{C}, \mathcal{A}}(n)$.

Proof. Fix $n, q, \varphi, f, \mathcal{O}_{-n}$, and $\mathcal{A}$ as in the theorem. Only $\mathcal{O}_n$ is sampled in the experiments below. We write $\mathrm{Image}(\mathcal{O}_n)$ explicitly when referring to the component being modified.

Merged-oracle notation. For fixed f, define the tagged classical function

$$\begin{aligned}
\mathsf{G}_{\mathcal{O},\varphi}(0,z) &:= f(z),\\
\mathsf{G}_{\mathcal{O},\varphi}(1,(C,r)) &:= \mathcal{O}(C,r),\\
\mathsf{G}_{\mathcal{O},\varphi}(2,(\widehat{C},z)) &:= \mathsf{Eval}_{\varphi}^{f,\mathcal{O}}(\widehat{C},z),\\
\mathsf{G}_{\mathcal{O},\varphi}(3,Q) &:= \mathsf{Decide}_{\mathfrak{S}_\Psi}^{\Psi(f,\mathcal{O},\varphi)}(Q).
\end{aligned}$$

For the fixed query-register lengths, we pad encodings and answers to obtain a function on a finite fixed-length domain. Malformed queries receive $\perp$. Inputs outside the resource-bounded domain may also be assigned $\perp$, since they never occur in the executions under consideration.

Access to the four interfaces is thereby represented by at most $3q$ queries to $\mathsf{G}_{\mathcal{O},\varphi}$. All comparisons below use the same family $\mathfrak{S}_\Psi$ in the decision-oracle validity test.

Hybrid H_0. This is the real obfuscation experiment.

Experiment 31: Hybrid H_0

1. $\mathcal{O}_n \leftarrow \mathbf{I}_{2n}^{5n}$.
2. $b \xleftarrow{\$} \{0,1\}$ and $r \xleftarrow{\$} \{0,1\}^n$.
3. $(C_0, C_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{|\Gamma(f,\mathcal{O},\varphi)\rangle}(1^n)$.
4. $\widehat{C} \leftarrow \mathcal{O}_n(C_b, r)$.
5. $b' \leftarrow \mathcal{A}_2^{|\Gamma(f,\mathcal{O},\varphi)\rangle}(\mathsf{st}, \widehat{C})$.
6. Output 1 if $b' = b$, and output 0 otherwise.

The register st may contain quantum side information. We say that a hybrid wins when its output is 1.

Hybrid H_1. We change only the pre-challenge oracle. For a uniformly random tape r, we replace the entire row $T = (*, r)$ by fresh encodings D. The challenge and post-challenge oracle still use the original table $\mathcal{O}$. Its difference from H_0 is highlighted in red color.

Experiment 32: Hybrid H_1

1. $\mathcal{O}_n \leftarrow \mathbf{I}_{2n}^{5n}$.
2. $b \xleftarrow{\$} \{0,1\}$ and $r \xleftarrow{\$} \{0,1\}^n$.
3. Sample D uniformly as an ordered subset of size 2^n of $\{0,1\}^{5n} \setminus \mathrm{Image}(\mathcal{O}_n)$.
4. Set $T = (*, r)$.
5. $(C_0, C_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{|\Gamma(f,\mathcal{O}_{T\mapsto D},\varphi)\rangle}(1^n)$.
6. $\widehat{C} \leftarrow \mathcal{O}_n(C_b, r)$.
7. $b' \leftarrow \mathcal{A}_2^{|\Gamma(f,\mathcal{O},\varphi)\rangle}(\mathsf{st}, \widehat{C})$.
8. Output 1 if $b' = b$, and output 0 otherwise.

We first establish the pointwise estimate needed to compare the pre-challenge states. This is the quantum-hybrid input corresponding to [BDV17, Claim 4.18].

Lemma 32 (Row-Reprogramming Disagreement). *Fix a full reference encoding family $\mathcal{O}$. Sample $r \xleftarrow{\$} \{0,1\}^n$ and an ordered subset D as in H_1, and let $T = (*, r)$. For every fixed allowed merged-oracle input w,*

$$\Pr_{r,D}\left[\mathsf{G}_{\mathcal{O}_{T\mapsto D},\varphi}(w) \neq \mathsf{G}_{\mathcal{O},\varphi}(w)\right] \leq 4q\,2^{-n}.$$

Proof. Fix w. Since D consists of distinct values outside $\mathrm{Image}(\mathcal{O}_n)$, the modified encoding oracle remains injective. Both underlying tuples therefore belong to $\mathfrak{S}_\Psi$.

We consider the four query types.

1. **Queries to f.** The function family is unchanged, so the disagreement probability is zero.
2. **Queries to $\mathcal{O}$.** A query at a length other than n is unchanged. At length n, a fixed input (C, s) changes only if $s = r$. Hence its disagreement probability is at most 2^{-n}.
3. **Queries to Eval.** An evaluation query whose encoding has length other than $5n$ is unchanged. At length $5n$, its answer can change only if its encoding $\widetilde{C}$ belongs to $\mathcal{O}_n(T)$ or to D.
   By injectivity, a fixed $\widetilde{C}$ has at most one preimage under $\mathcal{O}_n$, so

   $$\Pr_r[\widetilde{C} \in \mathcal{O}_n(T)] \leq 2^{-n}.$$

   Also,

   $$\Pr_D[\widetilde{C} \in D] \leq \frac{2^n}{2^{5n} - 2^{2n}} \leq 2^{1-4n}.$$

   Thus the disagreement probability is at most

   $$2^{-n} + 2^{1-4n} \leq 2^{1-n}.$$

4. **Queries to Decide.** If the verifier pair is invalid relative to $\mathfrak{S}_\Psi$, both answers are $\perp$. Otherwise, let

   $$C_Q = C\big(\Psi(f, \mathcal{O}, \varphi), V_0, V_1, z\big)$$

   be the reference critical set from Lem. 29. By Def. 12, $|C_Q| \leq 2q$.
   If the two underlying tuples agree on C_Q, their decision answers agree. A union bound using the preceding direct-query estimates gives

   $$\Pr_{r,D}\left[\mathsf{Decide}^{\Psi(f,\mathcal{O}_{T\mapsto D},\varphi)}_{\mathfrak{S}_\Psi}(Q) \neq \mathsf{Decide}^{\Psi(f,\mathcal{O},\varphi)}_{\mathfrak{S}_\Psi}(Q)\right] \leq |C_Q|\,2^{1-n} \leq 4q\,2^{-n}.$$

Malformed inputs and inputs outside the allowed domain have the same fixed answer in both oracles. This completes the proof of Lem. 32. □

For subsequent comparisons, set

$$\gamma_{\mathrm{pre}} := 4q\,2^{-n}, \qquad \Delta_{\mathrm{pre}} := 6q\sqrt{\gamma_{\mathrm{pre}}}.$$

Claim 27. *It holds that*

$$|\Pr[H_1 \text{ wins}] - \Pr[H_0 \text{ wins}]| \leq \Delta_{\mathrm{pre}}.$$

Proof. Fix the full reference oracle $\mathcal{O}$, in addition to the already fixed f and φ. Couple the pre-challenge oracles by sampling (r, D) as in H_1 and taking

$$(F_1, F_2) = (\mathsf{G}_{\mathcal{O}_{T\mapsto D},\varphi}, \mathsf{G}_{\mathcal{O},\varphi}).$$

The reference oracle F_2 is fixed. By Lem. 32, this coupling satisfies the conditional disagreement hypothesis of Lem. 31 with $\gamma = \gamma_{\mathrm{pre}}$.

Let $\rho_{r,D}$ denote the complete classical–quantum output state of $\mathcal{A}_1$ under the modified oracle, including C_0, C_1 and its residual state. Let ρ_{ref} denote the corresponding state under the reference oracle. Since $\mathcal{A}_1$ makes at most $3q$ merged-oracle queries, Lem. 31 gives

$$\mathop{\mathbb{E}}_{r,D}[\mathrm{TD}(\rho_{r,D}, \rho_{\mathrm{ref}})] \le 6q\sqrt{\gamma_{\mathrm{pre}}} = \Delta_{\mathrm{pre}}.$$

For each fixed (r, D, b), challenge generation and the execution of $\mathcal{A}_2$ form the same channel in the two experiments: the challenge is $\mathcal{O}_n(C_b, r)$ and the post-challenge oracle is $\Gamma(f, \mathcal{O}, \varphi)$. Contractivity of trace distance therefore bounds the difference in winning probabilities for this continuation by $\mathrm{TD}(\rho_{r,D}, \rho_{\mathrm{ref}})$. Averaging over (r, D, b) and then over $\mathcal{O}_n$ proves the claim.

The expected trace-distance conclusion is important here. The continuation depends on r, so closeness of only the averaged pre-challenge states would not suffice.

This completes the proof of Claim 27. □

Hybrid H_2. We keep the reprogrammed pre-challenge oracle, but replace the challenge by a fresh encoding and reprogram its post-challenge entry accordingly. Its difference from H_1 is highlighted in red color.

Experiment 33: Hybrid H_2

1. $\mathcal{O}_n \leftarrow \mathbf{I}_{2n}^{5n}$.
2. $b \xleftarrow{\$} \{0,1\}$ and $r \xleftarrow{\$} \{0,1\}^n$.
3. Sample D uniformly as an ordered subset of size 2^n of $\{0,1\}^{5n} \setminus \mathrm{Image}(\mathcal{O}_n)$.
4. Set $T = (*, r)$.
5. $(C_0, C_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{|\Gamma(f,\mathcal{O}_{T\mapsto D},\varphi)\rangle}(1^n)$.
6. $\widehat{C} \xleftarrow{\$} \{0,1\}^{5n} \setminus \big(\mathrm{Image}(\mathcal{O}_n) \cup D\big)$.
7. $b' \leftarrow \mathcal{A}_2^{|\Gamma(f,\mathcal{O}_{(C_b,r)\mapsto\widehat{C}},\varphi)\rangle}(\mathsf{st}, \widehat{C})$.
8. Output 1 if $b' = b$, and output 0 otherwise.

In set expressions involving D, we identify the ordered tuple with its underlying set.

Claim 28. *The joint distributions of the challenge bit, the pre-challenge output state, the challenge, and the post-challenge oracle are identical in H_1 and H_2. Consequently,*

$$\Pr[H_1 \text{ wins}] = \Pr[H_2 \text{ wins}].$$

Proof. Condition first on

$$f, \varphi, \mathcal{O}_{-n}, b, r, D, \mathcal{O}_{T\mapsto D}. \tag{37}$$

These variables determine the entire oracle supplied to $\mathcal{A}_1$. Its classical outputs and conditional residual density operator are therefore independent of the erased values $\mathcal{O}_n|_T$.

Condition further on the classical outputs C_0, C_1, and write $u = (C_b, r) \in T$. Fix all entries $\mathcal{O}_n(v)$ for $v \neq u$, and define

$$V := \{0,1\}^{5n} \setminus \left(\{\mathcal{O}_n(v) : v \in \{0,1\}^{2n},\, v \neq u\} \cup D\right).$$

Under the original sampling, the remaining entry $\mathcal{O}_n(u)$ is uniform over V. Indeed, every injective completion compatible with the conditioned data has the same probability: $\mathcal{O}_n$ is uniform, and the number of possible ordered choices of D outside its image depends only on the image size.

In H_1, the challenge is $\widehat{C} = \mathcal{O}_n(u)$ and is therefore uniform over V. In H_2, first $a = \mathcal{O}_n(u)$ is uniform over V, and then $\widehat{C}$ is uniform over $V \setminus \{a\}$. For every $h \in V$,

$$\Pr_{H_2}[\widehat{C} = h] = \sum_{a \in V \setminus \{h\}} \frac{1}{|V|} \frac{1}{|V| - 1} = \frac{1}{|V|}.$$

Thus the challenge is uniform over V in both hybrids.

In either experiment, the post-challenge encoding oracle agrees with the fixed table away from u and maps u to $\widehat{C}$. Its evaluation and decision interfaces are determined by this table, the fixed f, φ, and the fixed family $\mathfrak{S}_\Psi$. The conditional residual state of $\mathcal{A}_1$ is the same and independent of the hidden entry a. Hence the relevant joint classical–quantum distributions are identical.

Averaging over the conditioned variables completes the proof of Claim 28.

□

Hybrid H_3. We restore the original pre-challenge oracle. The challenge sampling and post-challenge reprogramming remain unchanged. Its difference from H_2 is highlighted in red color.

Experiment 34: Hybrid H_3

1. $\mathcal{O}_n \leftarrow \mathbf{I}_{2n}^{5n}$.
2. $b \xleftarrow{\$} \{0,1\}$ and $r \xleftarrow{\$} \{0,1\}^n$.
3. Sample D uniformly as an ordered subset of size 2^n of $\{0,1\}^{5n} \setminus \mathrm{Image}(\mathcal{O}_n)$.
4. $(C_0, C_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{|\Gamma(f, \mathcal{O}, \varphi)\rangle}(1^n)$.
5. $\widehat{C} \xleftarrow{\$} \{0,1\}^{5n} \setminus \left(\mathrm{Image}(\mathcal{O}_n) \cup D\right)$.
6. $b' \leftarrow \mathcal{A}_2^{|\Gamma(f, \mathcal{O}_{(C_b, r) \mapsto \widehat{C}}, \varphi)\rangle}(\mathsf{st}, \widehat{C})$.
7. Output 1 if $b' = b$, and output 0 otherwise.

Claim 29. *It holds that*

$$|\Pr[H_3 \text{ wins}] - \Pr[H_2 \text{ wins}]| \leq \Delta_{\mathrm{pre}}.$$

Proof. Fix the full reference oracle $\mathcal{O}$. The pre-challenge states are exactly $\rho_{r,D}$ and ρ_{ref} from the proof of Claim 27. Hence

$$\mathop{\mathbb{E}}_{r,D}\left[\mathrm{TD}(\rho_{r,D}, \rho_{\mathrm{ref}})\right] \leq \Delta_{\mathrm{pre}}.$$

For each fixed (r, D, b), both experiments use the same continuation: sample $\widehat{C}$ uniformly outside $\mathrm{Image}(\mathcal{O}_n) \cup D$, and run $\mathcal{A}_2$ with oracle $\Gamma(f, \mathcal{O}_{(C_b, r) \mapsto \widehat{C}}, \varphi)$. This is a common channel on the complete pre-challenge output state, including the classical circuit descriptions. Contractivity followed by averaging over (r, D, b) and $\mathcal{O}_n$ proves the claim.

This completes the proof of Claim 29.

□

Hybrid H_4. We remove the post-challenge encoding reprogramming. Instead, we set the backup circuit at $\widehat{C}$ to C_0. Since $\widehat{C}$ is outside the image of $\mathcal{O}_n$, the challenge evaluates to C_0^f in this experiment. Its difference from H_3 is highlighted in red color.

Experiment 35: Hybrid H_4

1. $\mathcal{O}_n \leftarrow \mathbf{I}_{2n}^{5n}$.
2. $b \xleftarrow{\$} \{0,1\}$ and $r \xleftarrow{\$} \{0,1\}^n$.
3. Sample D uniformly as an ordered subset of size 2^n of $\{0,1\}^{5n} \setminus \mathrm{Image}(\mathcal{O}_n)$.
4. $(C_0, C_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{|\Gamma(f,\mathcal{O},\varphi)\rangle}(1^n)$.
5. $\widehat{C} \xleftarrow{\$} \{0,1\}^{5n} \setminus \big(\mathrm{Image}(\mathcal{O}_n) \cup D\big)$.
6. $b' \leftarrow \mathcal{A}_2^{|\Gamma(f,\mathcal{O},\varphi_{\widehat{C}\mapsto C_0})\rangle}(\mathsf{st}, \widehat{C})$.
7. Output 1 if $b' = b$, and output 0 otherwise.

The next estimate corresponds to [BDV17, Claim 4.22]. Its reference oracle uses the modified backup map, which is why $\mathfrak{S}_\Psi$ must include all backup maps.

Lemma 33 (Challenge-Reprogramming Disagreement). *Fix $f, \mathcal{O}, \varphi$, a bit b, circuits $C_0, C_1 \in \{0,1\}^n$ with $C_0^f \equiv C_1^f$, and*

$$\widehat{C} \in \{0,1\}^{5n} \setminus \mathrm{Image}(\mathcal{O}_n).$$

For every fixed allowed merged-oracle input w,

$$\Pr_{r \xleftarrow{\$} \{0,1\}^n} \left[\mathsf{G}_{\mathcal{O}_{(C_b,r)\mapsto\widehat{C}},\varphi}(w) \neq \mathsf{G}_{\mathcal{O},\varphi_{\widehat{C}\mapsto C_0}}(w) \right] \leq 2q\, 2^{-n}.$$

Proof. Define the reference tuple

$$\Psi_2 := \Psi(f, \mathcal{O}, \varphi_{\widehat{C}\mapsto C_0})$$

and, for each $r \in \{0,1\}^n$, define

$$\Psi_1^{(r)} := \Psi(f, \mathcal{O}_{(C_b,r)\mapsto\widehat{C}}, \varphi).$$

The reference tuple belongs to $\mathfrak{S}_\Psi$ because the family permits arbitrary backup maps. Since $\widehat{C} \notin \mathrm{Image}(\mathcal{O}_n)$, the reprogrammed encoding oracle remains injective, so $\Psi_1^{(r)} \in \mathfrak{S}_\Psi$ for every r.

Fix w and consider its query type.

1. **Queries to f.** Both tuples use the same f, so the disagreement probability is zero.
2. **Queries to $\mathcal{O}$.** At length n, a fixed input (C, s) changes only if $(C, s) = (C_b, r)$. Its disagreement probability is therefore at most 2^{-n}. Other lengths are unchanged.
3. **Queries to Eval.** At the challenge encoding $\widehat{C}$, the first tuple returns $C_b^f(z)$ and the reference tuple returns $C_0^f(z)$. These answers agree because $C_0^f \equiv C_1^f$.
   At any other encoding $\widetilde{C}$, the evaluation answers can differ only if

   $$\widetilde{C} = \mathcal{O}_n(C_b, r).$$

   This is the encoding removed by the reprogramming. Injectivity implies that a fixed $\widetilde{C}$ equals this value for at most one choice of r. Thus every fixed evaluation query has disagreement probability at most 2^{-n}. Encodings at other lengths are unchanged.

4. **Queries to** Decide**.** Invalid verifier pairs receive $\perp$ in both tuples. For a valid query $Q = (V_0, V_1, z)$, choose the critical set at the reference tuple, $C_Q = C(\Psi_2, V_0, V_1, z)$. This set is independent of r and has size at most $2q$. By Lem. 29, disagreement of the decision answers requires disagreement at some point of C_Q. Hence

$$\begin{aligned}\Pr_r\left[\mathsf{Decide}^{\Psi_1^{(r)}}_{\mathfrak{S}_\Psi}(Q) \neq \mathsf{Decide}^{\Psi_2}_{\mathfrak{S}_\Psi}(Q)\right] &\le \sum_{u\in C_Q} \Pr_r[\Psi_1^{(r)}(u) \neq \Psi_2(u)] \\ &\le |C_Q|\, 2^{-n} \\ &\le 2q\, 2^{-n}.\end{aligned}$$

Malformed inputs and inputs outside the allowed domain again have disagreement probability zero. This completes the proof of Lem. 33.

□

Set

$$\gamma_{\text{post}} := 2q\, 2^{-n}, \qquad \Delta_{\text{post}} := 6q\sqrt{\gamma_{\text{post}}}.$$

Claim 30. *It holds that*

$$|\Pr[H_4 \text{ wins}] - \Pr[H_3 \text{ wins}]| \le \Delta_{\text{post}}.$$

Proof. The pre-challenge oracle is the original oracle in both experiments. Condition on the classical data

$$f, \mathcal{O}, \varphi, b, D, C_0, C_1, \widehat{C},$$

and let ρ_{st} denote the resulting conditional residual density operator.

The sampling of D and $\widehat{C}$, and the entire pre-challenge execution, are independent of r. Consequently, after this conditioning, r remains uniform over $\{0,1\}^n$ and ρ_{st} is independent of r. Moreover, admissibility in the real experiment implies $C_0^f \equiv C_1^f$ with probability one here, since the pre-challenge execution uses the same original oracle.

Couple the post-challenge merged oracles by sampling this uniform r and taking

$$(F_1, F_2) = \left(\mathsf{G}_{\mathcal{O}_{(C_b,r)\mapsto\widehat{C}},\varphi}, \mathsf{G}_{\mathcal{O},\varphi_{\widehat{C}\mapsto C_0}}\right).$$

The reference oracle F_2 is fixed under the conditioning. By Lem. 33, the coupling satisfies the hypothesis of Lem. 31 with $\gamma = \gamma_{\text{post}}$.

Apply that lemma to $\mathcal{A}_2$ with initial state $\rho_{\mathsf{st}} \otimes |\widehat{C}\rangle\langle\widehat{C}|$. This state is independent of the sampled oracle pair. Since $\mathcal{A}_2$ makes at most $3q$ merged-oracle queries, the difference in its probability of outputting the fixed bit b is at most $6q\sqrt{\gamma_{\text{post}}} = \Delta_{\text{post}}$. Averaging over the common distribution of the conditioned classical data proves the claim.

This completes the proof of Claim 30. □

Completing the proof of Thm. 8. In H_4, the pre-challenge state, challenge, and post-challenge oracle are all independent of b. The decision-oracle validity predicate also uses the same fixed family $\mathfrak{S}_\Psi$, independently of b. Thus

$$\Pr[H_4 \text{ wins}] = \frac{1}{2}.$$

Combining Claims 27 to 30, we obtain

$$\left|\Pr[H_0 \text{ wins}] - \frac{1}{2}\right| \le 2\Delta_{\text{pre}} + \Delta_{\text{post}}$$

$$\leq 12q\sqrt{4q\,2^{-n}} + 6q\sqrt{2q\,2^{-n}}$$
$$= O\left(q^{3/2}2^{-n/2}\right).$$

For $q(n) = O(2^{n/6})$, this is $O(2^{-n/4})$.

The bound is uniform over the fixed f and $\mathcal{O}_{-n}$. Averaging over their independent sampling gives the claimed security bound for Γ_φ. The advantage here is the absolute bias after averaging over the oracle distribution, as in Def. 8; we do not interchange this quantity with the expected absolute bias of individual oracle realizations.

This completes the proof of Thm. 8.

□

8.3.4 Consequence for Efficient Black-Box Reductions

We now connect the two source-security theorems to the black-box separation. The implementation convention specified in Sec. 8.3.1 is essential: source circuits have only f-gates, but returned obfuscated programs execute relative to the ambient oracle tuple.

Corollary 5. *Under this ambient-oracle implementation convention, there is no efficient fully black-box quantum construction of a worst-case quantum-hard language in* $\mathsf{NP} \cap \mathsf{coNP}$ *from post-quantum injOWFs and post-quantum iO for classical circuits with injOWF-gates.*

Proof. Suppose an efficient fully black-box construction $(V_0, V_1, \mathcal{R})$ exists. Fix a backup map φ, and instantiate the source primitives using the independently sampled families $f, \mathcal{O}$ and the obfuscator from Sec. 8.3.1.

Compiling the construction's interfaces. Compile the construction's verifiers into classical Ψ-aided verifiers $(\widetilde{V}_0, \widetilde{V}_1)$. An obfuscation query on (C, r) is implemented by querying $\mathcal{O}(C, r)$ and forming the canonical wrapper. Executing such a wrapper is implemented by the corresponding Eval-query. The compilation has polynomial overhead and preserves the verifiers' behavior.

Every tuple in $\mathfrak{S}_\Psi$ gives an injective function and a functionally correct obfuscation implementation. The universal structure requirement therefore implies that $(\widetilde{V}_0, \widetilde{V}_1)$ is valid relative to $\mathfrak{S}_\Psi$. Consequently, the deterministic adversary

$$\mathcal{A}_{\mathrm{dec}}(z) := \mathsf{Decide}^{\Psi}_{\mathfrak{S}_\Psi}(\widetilde{V}_0, \widetilde{V}_1, z)$$

decides the constructed language exactly on every input. At each input length, the verifier machines are compiled into oracle-aided circuits. Reversibly forming this decision query and uncomputing its description supplies the coherent function-evaluation interface required by the reduction.

Polynomial resource inclusion. Replace each call by $\mathcal{R}$ to $\mathcal{A}_{\mathrm{dec}}$ with this decision-oracle implementation, and compile the source interfaces as above. Since the reduction is efficient, it uses polynomially many oracle calls and polynomial-length query registers. The construction's verifier circuits also have polynomial size, including after interface compilation.

It follows that a single polynomial $p(n)$ bounds:

1. the number of queries to each augmented-oracle interface;
2. the lengths of all evaluation encodings queried by the composed reduction; and
3. the corresponding query counts and evaluation lengths inside every verifier submitted to $\mathsf{Decide}_{\mathfrak{S}_\Psi}$, for every witness and every underlying realization.

These are syntactic bounds on the composed algorithms and circuits, so they also hold in the hybrid executions. Increasing p if necessary, the induced source adversaries are $p(n)$-query in the sense of Def. 12. There is no requirement that $p(n)$ equal the reduction's original oracle-query bound.

Contradiction with source security. By Thm. 7 and 8, the induced inverter and admissible obfuscation distinguisher have, respectively, success and advantage at most

$$O\left(p(n)^3 2^{-n/2}\right) \qquad \text{and} \qquad O\left(p(n)^{3/2} 2^{-n/2}\right).$$

Both are negligible; in particular, both are $O(2^{-n/4})$ for sufficiently large n.

On the other hand, $\mathcal{A}_{\mathrm{dec}}$ decides the constructed language exactly in the worst case. The black-box security requirement therefore makes $\mathcal{R}$ break one of the source primitives with inverse-polynomial success or advantage for infinitely many input lengths. This contradicts the preceding bounds.

This completes the proof of Corollary 5.

□

The polynomial-resource inclusion argument is sufficient for the efficient-reduction impossibility. The quantitative source-security theorems also apply to inefficient adversaries satisfying Def. 12, but they do not by themselves give an exponential lower bound on the bare number of oracle queries made by an arbitrary inefficient reduction.

9 Acknowledgments and AI Disclosure

We thank Yuval Ishai and Alexis Korb for helpful discussions about their work [GIK+23].

GPT-5.6 and GPT-6 were used to improve the clarity and quality of the writing, correct spelling and grammatical errors, and proofread the manuscript. No AI tools were used to generate any of the ideas or techniques presented in this work, except for minor fixes to bugs identified by AI tools.

References

AHU19. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 269–295, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-26951-7_10. 10, 11, 20

AR04. Dorit Aharonov and Oded Regev. Lattice problems in NP cap coNP. In *45th Annual Symposium on Foundations of Computer Science*, pages 362–371, Rome, Italy, October 17–19, 2004. IEEE Computer Society Press. doi:10.1109/FOCS.2004.35. 2

ARU14. Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th Annual Symposium on Foundations of Computer Science*, pages 474–483, Philadelphia, PA, USA, October 18–21, 2014. IEEE Computer Society Press. doi:10.1109/FOCS.2014.57. 15, 85

AT03. Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *35th Annual ACM Symposium on Theory of Computing*, pages 20–29, San Diego, CA, USA, June 9–11, 2003. ACM Press. doi:10.1145/780542.780546. 4

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. doi:10.1137/S0097539796300933. 2

BDV17. Nir Bitansky, Akshay Degwekar, and Vinod Vaikuntanathan. Structure vs. hardness through the obfuscation lens. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 696–723, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-63688-7_23. 1, 3, 14, 15, 84, 85, 87, 92, 96, 97, 101

BG81. Charles H. Bennett and John Gill. Relative to a random oracle A, $\mathbf{P}^A \neq \mathbf{NP}^A \neq \mathbf{coNP}^A$ with probability 1. *SIAM Journal on Computing*, 10(1):96–113, 1981. doi:10.1137/0210008. 1, 6, 10, 72

BI87. Manuel Blum and Russell Impagliazzo. Generic oracles and oracle classes (extended abstract). In *28th Annual Symposium on Foundations of Computer Science*, pages 118–126, Los Angeles, CA, USA, October 12–14, 1987. IEEE Computer Society Press. doi:10.1109/SFCS.1987.30. 1

BLP93. Joe P. Buhler, Hendrik W. Lenstra, Jr., and Carl Pomerance. Factoring integers with the number field sieve. In Arjen K. Lenstra and Jr. Hendrik W. Lenstra, editors, *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*, pages 50–94. Springer, 1993. doi:10.1007/BFb0091539. 1

CJK+17. Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th Annual ACM Symposium on Theory of Computing*, pages 252–263, Montreal, QC, Canada, June 19–23, 2017. ACM Press. doi:10.1145/3055399.3055409. 4

CJL+16. Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. NIST Internal Report 8105, National Institute of Standards and Technology, 2016. doi:10.6028/NIST.IR.8105. 2

Con92. Anne Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, 1992. doi:10.1016/0890-5401(92)90048-K. 4

DMR11. Hang Dinh, Cristopher Moore, and Alexander Russell. Quantum fourier sampling, code equivalence, and the quantum security of the mceliece and sidelnikov cryptosystems, 2011. URL: https://arxiv.org/abs/1111.4382, arXiv:1111.4382. 4

GGPS23. Nicola Galesi, Joshua A. Grochow, Toniann Pitassi, and Adrian She. On the algebraic proof complexity of Tensor Isomorphism, 2023. URL: https://arxiv.org/abs/2305.19320, arXiv:2305.19320. 4

GIK+23. Riddhi Ghosal, Yuval Ishai, Alexis Korb, Eyal Kushilevitz, Paul Lou, and Amit Sahai. Hard languages in NP ∩ coNP and NIZK proofs from unstructured hardness. In Barna Saha and Rocco A. Servedio, editors, *55th Annual ACM Symposium on Theory of Computing*, pages 1243–1256, Orlando, FL, USA, June 20–23, 2023. ACM Press. doi:10.1145/3564246.3585119. i, 1, 3, 4, 5, 6, 7, 9, 10, 17, 22, 24, 26, 58, 59, 65, 72, 76, 104, 108

GLZ24. Jesse Goodman, Xin Li, and David Zuckerman. Improved condensers for Chor-Goldreich sources. In *65th Annual Symposium on Foundations of Computer Science*, pages 1513–1549, Chicago, IL, USA, October 27–30, 2024. IEEE Computer Society Press. doi:10.1109/FOCS61266.2024.00096. 107

Gor93. Daniel M. Gordon. Discrete logarithms in $GF(p)$ using the number field sieve. *SIAM Journal on Discrete Mathematics*, 6(1):124–138, 1993. doi:10.1137/0406010. 1

HR04. Chun-Yuan Hsiao and Leonid Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 92–105, Santa Barbara, CA, USA, August 15–19, 2004. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-540-28628-8_6. 81

HXY19. Minki Hhan, Keita Xagawa, and Takashi Yamakawa. Quantum random oracle model with auxiliary input. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 584–614, Kobe, Japan, December 8–12, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-34578-5_21. 26

HY20. Akinori Hosoyamada and Takashi Yamakawa. Finding collisions in a quantum world: Quantum black-box separation of collision-resistance and one-wayness. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 3–32, Daejeon, South Korea, December 7–11, 2020. Springer, Cham, Switzerland. doi:10.1007/978-3-030-64837-4_1. 81

MV05. Peter Bro Miltersen and N. V. Vinodchandran. Derandomizing Arthur–Merlin games using hitting sets. *Computational Complexity*, 14(3):256–279, 2005. doi:10.1007/s00037-005-0197-7. 4

Pei15. Chris Peikert. A decade of lattice cryptography. Cryptology ePrint Archive, Report 2015/939, 2015. URL: https://eprint.iacr.org/2015/939. 2

Rud88. Steven Rudich. *Limits on the Provable Consequences of One-way Functions*. PhD thesis, EECS Department, University of California, Berkeley, Dec 1988. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/1988/6060.html. 1

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. doi:10.1109/SFCS.1994.365700. 2

Tar89. Gábor Tardos. Query complexity, or why is it difficult to separate $NP^A \cap coNP^A$ from P^A by random oracles A? *Combinatorica*, 9(4):385–392, 1989. doi:10.1007/BF02125350. 1

Unr14. Dominique Unruh. Revocable quantum timed-release encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 129–146, Copenhagen, Denmark, May 11–15, 2014. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-55220-5_8. 15, 85

YZ21. Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 568–597, Zagreb, Croatia, October 17–21, 2021. Springer, Cham, Switzerland. doi:10.1007/978-3-030-77886-6_20. 17

Zha15. Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Info. Comput.*, 15(7–8):557–567, May 2015. 20

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. doi:10.1007/978-3-030-26951-7_9. 9, 23, 24

A More Basic Preliminaries

A.1 Probability Inequalities

Lemma 34 (Chernoff Bound). *Let $X_1, \ldots, X_n$ be independent random variables taking values in $\{0,1\}$, and let $X = \sum_{i=1}^{n} X_i$. Then for any $\delta \geq 0$ and $\mu \geq \mathbb{E}[X]$, we have*

$$\Pr[X \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}$$

Proof. Proof for this version of Chernoff bound can be found in standard literature, e.g., [GLZ24, Theorem 4].

□

Corollary 6. *Let $X_1, \ldots, X_n$ be independent random variables taking values in $\{0,1\}$, and let $X = \sum_{i=1}^{n} X_i$. Then for any $t \geq \mu$ and $\mu \geq \mathbb{E}[X]$, we have*

$$\Pr[X \geq t] \leq e^{-\mu}\left(\frac{e\mu}{t}\right)^{t}$$

Proof. It directly follows from Lem. 34 by setting $\delta = \frac{t}{\mu} - 1$.

□

Lemma 35 (Cauchy Schwarz Inequality). *For any Random variables X and Y, we have*

$$\mathbb{E}[XY] \leq \sqrt{\mathbb{E}[X^2] \cdot \mathbb{E}[Y^2]}$$

Corollary 7. *For any Random variable X, we have*

$$\mathbb{E}[X] \leq \sqrt{\mathbb{E}[X^2]}$$

Proof. Let $Y = 1$ in Lem. 35, we have

$$\mathbb{E}[X] \leq \sqrt{\mathbb{E}[X^2] \cdot \mathbb{E}[1^2]} = \sqrt{\mathbb{E}[X^2]}$$

□

Lemma 36 (McDiarmid's Inequality). *Let $X = (X_1, \ldots, X_m)$ be a vector of independent random variables, where each X_i takes values in a finite set $\mathcal{X}_i$. Suppose that $F : \prod_{i=1}^{m} \mathcal{X}_i \to \mathbb{R}$ satisfies*

$$|F(\mathbf{x}) - F(\mathbf{x}')| \leq c_i$$

for some constants $c_1, \ldots, c_m > 0$, whenever $\mathbf{x}$ and $\mathbf{x}'$ differ only in coordinate i. Then, for every $t \geq 0$,

$$\Pr\big[|F(X) - \mathbb{E}[F(X)]| \geq t\big] \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^{m} c_i^2}\right).$$

A.2 Statistical Distance and Properties

For distributions P, Q on a finite set Ω, denote the *statistical distance* between P and Q as

$$\Delta(P, Q) \coloneqq \frac{1}{2} \sum_{x \in \Omega} |P(x) - Q(x)|.$$

Lemma 37 ([GIK+23, Lemma 3.15]). *If $A = (A_1, \ldots, A_n)$ and $B = (B_1, \ldots, B_n)$ are sequences of independent random variables such that*

$$\forall i \in [n], \qquad \Delta(A_i, B_i) \leq \varepsilon,$$

then

$$\Delta\big((A_1, \ldots, A_n), (B_1, \ldots, B_n)\big) \leq n\varepsilon.$$

Also recall that statistical distance does not increase under post-processing:

Lemma 38. *Let P and Q be probability distributions on a finite set $\mathcal{X}$, and let A be a randomized transformation from $\mathcal{X}$ to a finite set $\mathcal{Y}$. Write $\mathsf{A}(P)$ and $\mathsf{A}(Q)$ for the output distributions obtained by applying A to an input drawn from P and Q, respectively. Then*

$$\Delta(\mathsf{A}(P), \mathsf{A}(Q)) \leq \Delta(P, Q).$$

In particular, discarding coordinates cannot increase statistical distance.

Proof. For each $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, define

$$A(y \mid x) \coloneqq \Pr[\mathsf{A}(x) = y],$$

where the probability is over the internal randomness of A. For every $x \in \mathcal{X}$, we have $A(y \mid x) \geq 0$ for all $y \in \mathcal{Y}$ and $\sum_{y \in \mathcal{Y}} A(y \mid x) = 1$. Writing $P(x)$ and $Q(x)$ for the corresponding probability masses, the definition of statistical distance and the triangle inequality give

$$\begin{aligned}
\Delta(\mathsf{A}(P), \mathsf{A}(Q)) &= \frac{1}{2} \sum_{y \in \mathcal{Y}} \left| \sum_{x \in \mathcal{X}} (P(x) - Q(x)) A(y \mid x) \right| \\
&\leq \frac{1}{2} \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} |P(x) - Q(x)| A(y \mid x) \\
&= \frac{1}{2} \sum_{x \in \mathcal{X}} |P(x) - Q(x)| \sum_{y \in \mathcal{Y}} A(y \mid x) \\
&= \frac{1}{2} \sum_{x \in \mathcal{X}} |P(x) - Q(x)| \\
&= \Delta(P, Q).
\end{aligned}$$

The final assertion follows by taking A to be the deterministic projection onto the coordinates being retained.

□

A.3 Efficient Approximability of $\frac{1}{e}$ and $1 - \frac{1}{e}$

Lemma 39 (Efficient Approximability of $\frac{1}{e}$).
There exist a deterministic polynomial-time algorithms $\mathsf{Approx}_{\frac{1}{e}}$ output a rational number $\gamma_n = \frac{a_n}{b_n}$ for some $a_n, b_n \in \mathbb{N}$ such that, for every $n \geq 2$ satisfying

$$\left| \frac{1}{e} - \frac{a_n}{b_n} \right| \leq 2^{-n}$$

Consequently, $\frac{1}{e}$ is efficiently approximable.

Proof. On input 1^n, the algorithm $\mathsf{Approx}_{1/e}$ outputs γ_n which is a positive rational number and can be written as

$$\gamma_n = \frac{a_n}{b_n}, \qquad a_n := \sum_{k=0}^{n} (-1)^k \frac{n!}{k!}, \qquad b_n := n!,$$

where $a_n, b_n \in \mathbb{N}$ follows.

Consider the Taylor expansion of $f(x) = e^{-x}$ around $x = 0$, evaluated at $x = 1$. Taylor's theorem gives

$$\frac{1}{e} = \sum_{k=0}^{n} \frac{(-1)^k}{k!} + R_n,$$

where, for some $\xi \in (0, 1)$,

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}.$$

Since

$$\left| f^{(n+1)}(\xi) \right| = e^{-\xi} \leq 1,$$

we obtain

$$\left| \frac{1}{e} - \gamma_n \right| = |R_n| \leq \frac{1}{(n+1)!}.$$

Since $(n+1)! \geq 2^n$ for every $n \geq 1$, we then obtain

$$\left| \frac{1}{e} - \frac{a_n}{b_n} \right| \leq \frac{1}{(n+1)!} \leq 2^{-n}.$$

Finally, the algorithm performs $O(n)$ arithmetic operations on integers of bit length bounded by $O(\log(n!)) = O(n \log n)$. Therefore, $\mathsf{Approx}_{1/e}$ runs in deterministic polynomial time. □

Corollary 8 (Efficient Approximability of $1 - \frac{1}{e}$). *There exists a deterministic polynomial-time algorithm $\mathsf{Approx}_{1-1/e}$ that, on input 1^n, output a rational number $\gamma_n = \frac{a_n}{b_n}$ for some $a_n, b_n \in \mathbb{N}$ such that, for every $n \geq 1$ satisfying*

$$\left| \left(1 - \frac{1}{e}\right) - \frac{a_n}{b_n} \right| \leq 2^{-n}.$$

Consequently, $1 - \frac{1}{e}$ is efficiently approximable.

Proof. On input 1^n, the algorithm $\mathsf{Approx}_{1-1/e}$ invokes $\mathsf{Approx}_{1/e}(1^n)$ from Lem. 39. If

$$\mathsf{Approx}_{1/e}(1^n) = \frac{a_n^{'}}{b_n^{'}},$$

for some $a_n^{'}, b_n^{'} \in \mathbb{N}$
then it outputs

$$\gamma_n := \frac{b_n^{'} - a_n^{'}}{b_n^{'}}$$

By Lem. 39,

$$\left| \frac{1}{e} - \frac{a_n^{'}}{b_n^{'}} \right| \leq 2^{-n}.$$

Therefore,

$$\begin{aligned} \left| \left(1 - \frac{1}{e}\right) - \gamma_n \right| &= \left| \left(1 - \frac{1}{e}\right) - \frac{b_n^{'} - a_n^{'}}{b_n^{'}} \right| \\ &= \left| \frac{a_n^{'}}{b_n^{'}} - \frac{1}{e} \right| \\ &\leq 2^{-n}. \end{aligned}$$

The invocation of $\mathsf{Approx}_{1/e}$ takes polynomial time, and the additional subtraction $b - a$ can be performed in time polynomial in the bit lengths of a and b. Hence, $\mathsf{Approx}_{1-1/e}$ is a deterministic polynomial-time algorithm. □

B Length-Tripling Random Oracles as Post-Quantum injOWFs

The following two lemma justifying the assumption $\mathsf{UP} \not\subseteq \mathsf{BQP}$ underlying our subsequent oracle-separation results. In particular, we show that a length-tripling random oracle heuristically instantiates a post-quantum injective one-way function with overwhelming probability. Notice that a length-tripling random oracle can be generated independently by setting $m(n) = 3n$ in our interpretation of random oracles by Lem. 1.

Lemma 40 (Overwhelming Injectivity of length-tripling random oracle). *Fix any $n \in \mathbb{N}$, for any length-tripling random oracle $O_{\mathsf{inj}} : \{0,1\}^n \to \{0,1\}^{3n}$, we have*

$$\Pr_{O \xleftarrow{\$} \mathcal{O}} [\exists x_1, x_2 \in \{0,1\}^n \text{ such that } O_{\mathsf{inj}}(x_1) = O_{\mathsf{inj}}(x_2) \wedge x_1 \neq x_2] \leq \mathsf{negl}(n)$$

Proof. Fix any two distinct inputs $x_1, x_2 \in \{0,1\}^n$, since O_{inj} is a random oracle and its outputs are independent and uniformly distributed over $\{0,1\}^{3n}$, we have

$$\Pr_{O \xleftarrow{\$} \mathcal{O}} [O_{\mathsf{inj}}(x_1) = O_{\mathsf{inj}}(x_2)] = \frac{2^{3n}}{2^{6n}} = \frac{1}{2^{3n}}.$$

There are

$$\binom{2^n}{2} < 2^{2n}$$

pairs of distinct elements of $\{0,1\}^n$. Therefore,

$$\begin{aligned}
&\Pr_{O\xleftarrow{\$}\mathcal{O}}\left[\exists x_1,x_2\in\{0,1\}^n : x_1\neq x_2 \wedge O_{\mathsf{inj}}(x_1)=O_{\mathsf{inj}}(x_2)\right]\\
&\leq \binom{2^n}{2}\cdot 2^{-3n} && \text{(Union Bound)}\\
&< 2^{2n}\cdot 2^{-3n}\\
&= 2^{-n} = \mathsf{negl}(n)
\end{aligned}$$

□

Lemma 41 (Post-Quantum Oneway-ness of length-tripling random oracle). *Fix any $n\in\mathbb{N}$, for any length-tripling random oracle $O_{\mathsf{inj}}:\{0,1\}^n\to\{0,1\}^{3n}$ and quantum oracle-aided adversary $\mathcal{A}$ making at most $q=\mathsf{poly}(n)$ queries, we have*

$$\Pr_{O\xleftarrow{\$}\mathcal{O},x\xleftarrow{\$}\{0,1\}^n}\left[\mathcal{A}^{|O_{\mathsf{inj}}\rangle}(O_{\mathsf{inj}}(x))=x\right]\leq\mathsf{negl}(n).$$

Proof. Assume that there exists a quantum adversary $\mathcal{A}$ making at most q queries that inverts a uniformly random length-tripling oracle $O_{\mathsf{inj}}:\{0,1\}^n\to\{0,1\}^{3n}$ with success probability

$$\varepsilon := \Pr_{O_{\mathsf{inj}},x}\left[\mathcal{A}^{|O_{\mathsf{inj}}\rangle}(O_{\mathsf{inj}}(x))=x\right].$$

We construct an adversary $\mathcal{B}$ for the length-preserving preimage-search problem in Lem. 3, with parameter $m=3n$.

Adversary $\mathcal{B}$ is given quantum oracle access to a uniformly random function $H:\{0,1\}^{3n}\to\{0,1\}^{3n}$ and a challenge $y=H(w)$, where $w\xleftarrow{\$}\{0,1\}^{3n}$ is hidden. It samples $r\xleftarrow{\$}\{0,1\}^{2n}$ once and runs $\mathcal{A}$ on y, simulating its oracle by$O_r(z):=H(z\,\|\,r)$

Each quantum query to O_r can be simulated using one quantum query to H by appending the fixed string r to the query input. If $\mathcal{A}$ outputs z, then $\mathcal{B}$ outputs $z\,\|\,r$.

For the analysis, write $w=x\,\|\,s$, where $x\in\{0,1\}^n$ and $s\in\{0,1\}^{2n}$. The event $r=s$ occurs with probability 2^{-2n}. Conditioned on this event, O_r is a uniformly random function from $\{0,1\}^n$ to $\{0,1\}^{3n}$, x is uniform and independent of O_r, and $y=O_r(x)$. Hence $\mathcal{A}$ outputs x with probability ε. When $r\neq s$, $\mathcal{B}$ cannot output w. Therefore,

$$\Pr_{H,w}\left[\mathcal{B}^{|H\rangle}(H(w))=w\right]=2^{-2n}\varepsilon.$$

Applying Lem. 3 at input length $3n$, we obtain

$$\begin{aligned}
2^{-2n}\varepsilon &\leq \Pr_{H,w}\left[H\big(\mathcal{B}^{|H\rangle}(H(w))\big)=H(w)\right]\\
&\leq \frac{6n(2q+1)^2+1}{2^{3n}}.
\end{aligned}$$

Consequently,

$$\varepsilon\leq\frac{6n(2q+1)^2+1}{2^n}=\mathsf{negl}(n),$$

since $q=\mathsf{poly}(n)$.

□