

Tight Quantum Time-Space Tradeoffs for Function Inversion

Kai-Min Chung* Siyao Guo† Qipeng Liu‡ Luowen Qian§

Abstract

In function inversion, we are given a function $f : [N] \mapsto [N]$, and want to prepare some advice of size S , such that we can efficiently invert any image in time T . This is a well studied problem with profound connections to cryptography, data structures, communication complexity, and circuit lower bounds. Investigation of this problem in the quantum setting was initiated by Nayebi, Aaronson, Belovs, and Trevisan (2015), who proved a lower bound of $ST^2 = \tilde{\Omega}(N)$ for random permutations against classical advice, leaving open an intriguing possibility that Grover’s search can be sped up to time $\tilde{O}(\sqrt{N/S})$. Recent works by Hhan, Xagawa, and Yamakawa (2019), and Chung, Liao, and Qian (2019) extended the argument for random functions and quantum advice, but the lower bound remains $ST^2 = \tilde{\Omega}(N)$.

In this work, we prove that even with quantum advice, $ST + T^2 = \tilde{\Omega}(N)$ is required for an algorithm to invert random functions. This demonstrates that Grover’s search is optimal for $S = \tilde{O}(\sqrt{N})$, ruling out any substantial speed-up for Grover’s search even with quantum advice. Further improvements to our bounds would imply a breakthrough in circuit lower bounds, as shown by Corrigan-Gibbs and Kogan (2019).

To prove this result, we develop a general framework for establishing quantum time-space lower bounds. We further demonstrate the power of our framework by proving the following results.

- Yao’s box problem: We prove a tight quantum time-space lower bound for classical advice. For quantum advice, we prove a first time-space lower bound using shadow tomography. These results resolve two open problems posted by Nayebi, Aaronson, Belovs, and Trevisan (2015).
- Salted cryptography: We show that “salting generically provably defeats preprocessing,” a result shown by Coretti, Dodis, Guo, and Steinberger (2018), also holds in the quantum setting. In particular, we prove quantum time-space lower bounds for a wide class of salted cryptographic primitives in the quantum random oracle model. This yields the first quantum time-space lower bound for salted collision-finding, which in turn implies that $\text{PWPP}^{\mathcal{O}} \not\subseteq \text{FBQP}^{\mathcal{O}}/\text{qpoly}$ relative to a random oracle \mathcal{O} .

*Academia Sinica. Email: kmchung@iis.sinica.edu.tw

†New York University Shanghai. Email: siyao.guo@nyu.edu

‡Princeton University & NTT Research. Email: qipengl@cs.princeton.edu

§Boston University. Email: luowenq@bu.edu

Contents

1	Introduction	1
1.1	Function Inversion	1
1.2	Yao’s Box Problem	3
1.3	Post-Quantum Non-Uniform Security	4
1.4	Technical Overview	6
1.4.1	From Non-uniform Algorithms to Multi-Instance Games	6
1.4.2	Analyzing Multi-Instance Game via “Compressed Oracles”	9
1.4.3	Yao’s Box Problem	11
1.5	Related Works	12
1.6	Open Problems	13
2	Preliminaries	14
2.1	Quantum Computation	14
2.2	Quantum Random Oracle Model	15
2.3	Compressed Oracle	15
3	Security Games	19
3.1	Security Game with Advice	19
3.2	Multi-Instance Security	21
3.3	Helper Lemma for Multi-Instance Security	22
4	Reducing Games with Advice to Multi-Instance Games	22
4.1	Reduction for Classical Advice	24
4.2	Reduction for Decision Games with Classical Advice	26
4.3	Reduction for Publicly-Verifiable Games	27
4.4	Reduction for Decision Games with Quantum Advice	29
5	One-Way Function	32
5.1	Multi-Instance Security for Inverting Random Image	34
5.2	Multi-Instance Indistinguishability of Sampling Image	37
5.3	Multi-Instance Security of OWF	41
5.4	Security of PRG with Advice	42
6	Yao’s Box	43
7	Salting Defeats Preprocessing	46
7.1	Multi-Instance Security of Salted Game	48
7.2	Hardness of Salted Collision-Resistant Hash Functions	50
7.3	Tightness of Salted Game Lower Bound	51
A	Extended Preliminaries for Compressed Oracle	56

1 Introduction

1.1 Function Inversion

The task of function inversion asks that given a function $f : [N] \mapsto [N]$ and a point y , find an x such that $f(x) = y$. It is easy to show that a classical inversion algorithm requires $\Omega(N)$ queries (which trivially lower bounds time) to succeed with constant probability in inverting a random function f , even with the help of randomness. Grover [Gro96] considered the same problem in the context of database search, and showed that quantum computers can invert any function in time only $\tilde{O}(\sqrt{N})$, which was subsequently shown to be tight [BBBV97].

The situation becomes intriguing when preprocessing is allowed. Namely, we allow the algorithm to take the entire truth table of f and arbitrarily preprocess an S -bit advice string $\alpha = \alpha(f)$, and then we give the algorithm a random $f(x)$ and ask the algorithm to invert it using at most T queries. Understanding the tradeoff between S and T is referred to as time-space tradeoffs for function inversion. This tradeoff is an important problem in cryptography. Recent works [GGH⁺19, KP19, CK19] showed its connections to well studied problems in data structures, communication complexity, and circuit lower bounds.

For classical algorithms, the heuristic algorithm proposed by Hellman [Hel80], and subsequently rigorously analyzed by Fiat and Naor [FN99], uses S bits of advice and T queries to invert a random function with high probability for every S, T satisfying $S^2T \geq \tilde{O}(N^2)$. For the lower bound, Yao [Yao90] and De et al. [DTT10] proved that any preprocessing algorithm that uses S bits of advice and T queries must satisfy $ST = \tilde{\Omega}(N)$, which remains the best general lower bound we know today. Corrigan-Gibbs and Kogan [CK19] recently investigated possible improvements on the lower bound, and showed that any improvements on Yao’s lower bound will lead to improved circuit lower bounds.

The study of time-space tradeoffs in the quantum setting was initiated by Nayebi, Aaronson, Belovs and Trevisan [NABT15]. When the preprocessing algorithm is quantum, it is natural to distinguish the cases of quantum versus classical advice, as analogous to the complexity classes of BQP/qpoly versus BQP/poly and QMA versus QCMA. Nayebi et al. [NABT15] showed that any quantum preprocessing algorithm that uses S bits of *classical* advice and T queries to invert a random permutation with a constant probability must satisfy $ST^2 = \tilde{\Omega}(N)$. Note that the T^2 term is necessary given Grover’s search algorithm. Recently, with motivations from post-quantum cryptography, Hhan, Xagawa, and Yamakawa [HXY19] extended the lower bound to handle general function inversion (and other cryptographic primitives). For the more challenging case of algorithms with *quantum* advice, Hhan et al. [HXY19] and Chung et al. [CLQ19] proved lower bounds for inverting random permutation and a restricted class of random functions¹. However, these lower bounds remain $ST^2 = \tilde{\Omega}(N)$. As pointed out by Nayebi et al. [NABT15], this leaves open the following intriguing possibility:

Could a piece of preprocessed advice help speed up Grover’s search algorithm?

In this work, we prove the following quantum time-space lower bound for function inversion, which shows that even quantum advice of size $S = O(\sqrt{N})$ does not help speed up Grover’s search algorithm.

Theorem 1.1. *Let $f : [N] \mapsto [N]$ be a random function. For any quantum oracle algorithms \mathcal{A} with S -qubit oracle-dependent advice $\alpha = \alpha(f)$ and T queries to f ,*

¹Specifically, the lower bound only holds for functions with roughly the same domain and image size.

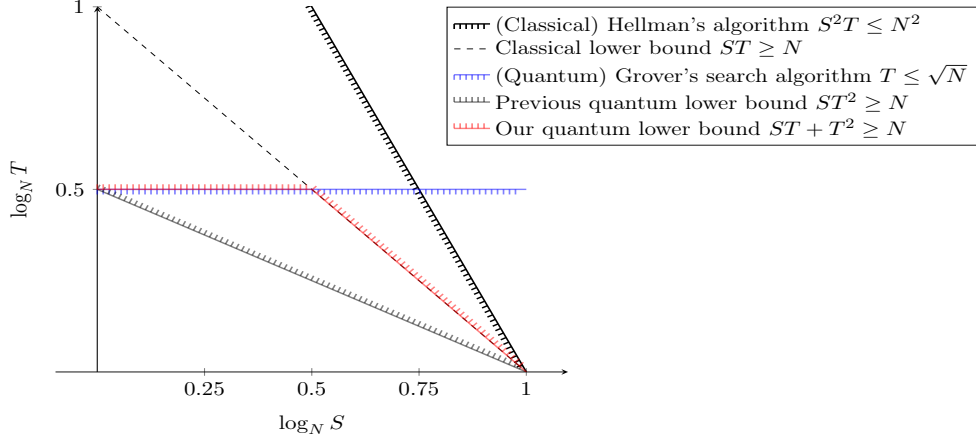


Figure 1: Time-space tradeoffs for inverting a random function $f : [N] \mapsto [N]$ with a constant successful probability. For the classical setting, the best upper bound is given by Hellman’s algorithm [Hel80, FN99], and the best lower bound is given by [Yao90, DGK17, CDGS18]. For the quantum setting, the best upper bound is given either by Grover’s search algorithm [Gro96] or (classical) Hellman’s algorithm, and the previous best lower bound is given by [HXY19, CLQ19]. For simplicity, logarithmic terms and constant factors are omitted.

- if α is classical, then

$$\Pr \left[\mathcal{A}^f(\alpha, f(x)) \in f^{-1}(f(x)) \right] = \tilde{O} \left(\frac{ST + T^2}{N} \right);$$

- if α is quantum, then

$$\Pr \left[\mathcal{A}^f(\alpha, f(x)) \in f^{-1}(f(x)) \right] = \tilde{O} \left(\sqrt[3]{\frac{ST + T^2}{N}} \right),$$

where both probabilities are over f , a uniformly random x from $[N]$, and randomness of \mathcal{A} .

Our lower bound implies that for a quantum preprocessing algorithm to invert a random function with a constant probability, it must satisfy $ST + T^2 = \tilde{\Omega}(N)$ even for the case of quantum advice. This further shows that Grover’s search is optimal for $S = \tilde{O}(\sqrt{N})$, ruling out any substantial speed-up for Grover’s search even with quantum advice. For $S = \tilde{\omega}(\sqrt{N})$, our lower bound matches Yao’s lower bound for classical algorithms and is the best provable lower bound in light of the above-mentioned barrier results of [CK19].

Furthermore, in the context of cryptography, a typical complexity measure to define the security is $S + T$ (corresponding to the program length and the running-time), with respect to which our lower bound implies a lower bound $S + T \geq \tilde{\Omega}(\sqrt{N})$. This matches Grover’s search algorithm and gives a tight characterization for function inversion.

A comparison of our bounds with known upper and lower bounds is included in Figure 1. We remark that our lower bounds can be extended to general functions $f : [N] \mapsto [M]$ (see Theorem 5.3 and Theorem 5.4).

1.2 Yao’s Box Problem

Yao’s box problem [Yao90] is another basic problem investigated in the literature of time-space tradeoffs that is closely related to the function inversion problem and non-uniform security of pseudorandom generators. In this problem, a preprocessing algorithm \mathcal{A} can compute an S -bit advice for a function $f : [N] \mapsto \{0, 1\}$ in the preprocessing phase, and then in the online phase, it is required to compute the bit $f(x) \in \{0, 1\}$ for a random point $x \in [N]$ by making at most T query but without querying f on x . Similar formulations of such problem in the classical setting are recently studied in the context of circuit complexity [ST18, MW19], which led to a new result on depth 3 circuits.

In the classical setting, Yao [Yao90] proved that for \mathcal{A} to succeed with probability $2/3$, it must satisfy $ST = \Omega(N)$, which is known to be optimal. In the quantum setting, Nayebi et al. [NABT15] showed an $ST^2 = \Omega(N)$ lower bound for solving Yao’s box problem with *classical* advice, which does not rule out the possibility that a preprocessing algorithm with S bits of advice and $T = O(\sqrt{N/S})$ queries. Subsequently, Hhan et al. [HXY19, Lemma 6] refined the analysis and showed that for any S, T, N , any quantum algorithm with classical advice can only succeed with probability $1/2 + \tilde{O}(ST^2/N)^{1/6}$. However, the following two problems posted by Nayebi et al. remain open.

Open Problem 1.2 ([NABT14, Section 3.4, Section 5]). *Is there a quantum algorithm that solves Yao’s box in time $T = O(\sqrt{N/S})$? Or equivalently, prove or disprove the optimality of the lower bound $ST^2 = \tilde{\Omega}(N)$ for Yao’s box.*

Open Problem 1.3 ([NABT14, Section 5]). *Extend the lower bound for Yao’s box to the setting where the advice can be an arbitrary quantum state.*

We prove the following theorem for Yao’s box problem, answering both open problems above.

Theorem 1.4. *Let $f : [N] \mapsto \{0, 1\}$ be a random function. For any quantum oracle algorithms \mathcal{A} with S -qubit oracle-dependent advice $\alpha = \alpha(f)$ and T queries to f except on the given challenge point x ,*

- *if α is classical, then*

$$\Pr[\mathcal{A}^f(\alpha, x) = f(x)] = \frac{1}{2} + \tilde{O}\left(\sqrt[3]{\frac{ST}{N}}\right);$$

- *if α is quantum, then*

$$\Pr[\mathcal{A}^f(\alpha, x) = f(x)] = \frac{1}{2} + \tilde{O}\left(\sqrt[19]{\frac{S^5 T}{N}}\right),$$

where both probabilities are over f , a uniformly random x from $[N]$, and randomness of \mathcal{A} .

In particular, this theorem implies that any algorithm achieving success probability $2/3$ has to satisfy $ST = \tilde{\Omega}(N)$ for classical advice, which shows that the power of quantum query does not help solving Yao’s box problem, and $S^5 T = \tilde{\Omega}(N)$ for quantum advice.

We note that while in the classical setting, time-space lower bounds for Yao’s box problem and the function inversion problem can be proved using the same techniques, such as compression [DTT10, DGK17] or presampling [Unr07, CDGS18], proving quantum time-space lower bounds

for Yao’s box problem seems to be more challenging for quantum advice. This is also the case for our framework since in Yao’s box problem, the algorithm cannot verify the answer on its own. For the case of quantum advice, we employ a novel use of online shadow tomography [AR19], which enables us to prove the first time-space lower bound for algorithms with quantum advice. We discuss the issue more carefully in Section 1.4.3.

Finally, we note that while we did not explicitly prove this, we believe that our techniques are sufficient for proving a lower bound $ST = \tilde{\Omega}(N)$ for quantum advice, if we restrict the algorithm such that it has to recover $f(x)$ for any $x \in [N]$ with probability $2/3$.

1.3 Post-Quantum Non-Uniform Security

It turns out that our techniques for proving the two results above are fairly general, and we can use it to prove a variety of quantum non-uniform lower bounds. We now turn our focus to proving non-uniform lower bounds in cryptographic ideal models, which is a topic that has gained a lot of momentum recently.

Random oracle methodology and concrete security. While theoretically we can instantiate a lot of private-key cryptography assuming only the existence of any one-way function [Lev87], the constructions are almost always way too inefficient to be any useful for practical purposes. Practical cryptographic schemes, instead are usually designed under an ideal model and are proven secure under that model. A popular model of choice is random oracle model (ROM) [BR93], which “heuristically” models a function in question $f : [N] \mapsto [M]$, say a SHA3 hash function, as a truly random function that everyone has only oracle access to. To determine the security parameter to use in the construction, usually concrete security bounds are derived in the ideal model, and we work out the calculations to make sure any adversary in the ideal model, using at most a reasonable amount of resources, can only succeed with small enough probability, usually 2^{-32} [LR10, Arc20].

While the random oracle model “heuristically” captures all attacks that do not employ the structure of any particular instantiation of the random oracle, the model itself captures neither preprocessing attackers nor quantum attackers. Moreover, security bounds obtained in the random oracle model are inaccurate or do not apply at all once preprocessing or quantum computation are allowed.

Quantum random oracle model. Quantum algorithms are known to achieve various nontrivial speedups compared to classical algorithms, sometimes even an exponential speedup. For example, Grover’s search algorithm achieves \sqrt{N} speedup over classical algorithms, and Shor’s algorithm solves factoring in quantum polynomial time.

Motivated by assessing the post-quantum security of constructions in ROM, Boneh et al. [BDF⁺11] introduced the quantum random oracle model (QROM) where the attacker can make superposition queries to the oracle, as given a classical description of any function, a quantum algorithm can trivially perform such superposition queries. Extending security proofs in classical ROM into the stronger QROM has been an active area of investigation [BDF⁺11, Zha12, BZ13, Unr15, TU16, Unr17, KLS18, Zha19, AHU19, LZ19b, DFMS19].

Auxiliary-input random oracle model. Consider the example where we use random oracle to instantiate an one-way function (OWF). As shown by Hellman’s algorithm [Hel80], a preprocessing

attacker can achieve a non-trivial saving of resources from N to $N^{2/3}$ even in the random oracle model. Because the instantiated function is usually public, there is no way of preventing the adversary from performing a heavy precomputation to speed up the online attack.

To address this mismatch, Unruh [Unr07] introduced the auxiliary-input random oracle model (AI-ROM) where the adversary is allowed to obtain a bounded length advice about the random oracle before attacking the system. Several works [Unr07, DGK17, CDGS18, CDG18] have developed various techniques for analyzing the security in this model.

Hhan et al. [HXY19] first considered the auxiliary-input quantum random oracle model (AI-QROM) and quantum auxiliary-input quantum random oracle model (QAI-QROM), which is the natural generalization of the model above for quantum adversaries. Despite the progress in QROM and AI-ROM respectively, proving post-quantum non-uniform security remains quite challenging. Only a few security bounds have been proven against classical advice, i.e. under AI-QROM [HXY19], and *no* security bounds are known against quantum advice, i.e. QAI-QROM, except for OWFs [HXY19, CLQ19]. The only known technique under these models is the (quantum) compression technique, introduced by Nayebi et al. [NABT15], and its variants. We note that this technique is only somewhat generic, and its effectiveness seems limited under QAI-QROM.

New generic framework for security bounds in (Q)AI-QROM. We generalize our techniques for function inversion and Yao’s box into a general framework for proving concrete post-quantum non-uniform security in AI-QROM and QAI-QROM. In particular, we show the following theorem.

Theorem (Informal). *For any security game G , adversary time bound T , and any $g > 0$, consider its multi-instance game $G^{\otimes g}$, which requires the adversary to break g independent challenges sequentially, and each instance is given time T .*

If the best winning probability for $G^{\otimes g}$ is δ^g in the QROM, meaning that the best adversary can only win the game with probability at most δ^g , then for adversaries with $S = g$ (qu)bits of advice:

1. G is roughly $\tilde{O}(\delta)$ -secure in AI-QROM;
2. If G is publicly verifiable, G is $\text{negl}(N)$ -secure in QAI-QROM if $\delta = \text{negl}(N)$;
3. If G is a decision game, G is $1/2 + \text{negl}(N)$ -secure in QAI-QROM if $\delta = 1/2 + \text{negl}(N)$.

The formal definition of security games and multi-instance games are given in Section 3, and the formal reduction is presented in Section 4.

We note that our bound for function inversion we show in Section 1.1 directly translates to the concrete security bound for OWFs, and our techniques for OWFs can be additionally used to show security for pseudorandom generators (PRGs) with little efforts. The concrete bounds, proven in Theorem 5.3, Theorem 5.4, and Theorem 5.14, are summarized into the table below.

	OWFs	PRGs
AI-ROM [DGK17, CDGS18]	$\frac{ST}{\alpha}$	$\left(\frac{ST}{N}\right)^{1/2} + \frac{T}{N}$
AI-QROM [HXY19]	$\left(\frac{ST^2}{\alpha}\right)^{1/2}$	$\left(\frac{ST^4}{N} + \frac{T^4}{N}\right)^{1/6}$
AI-QROM (ours)	$\frac{ST}{\alpha} + \frac{T^2}{\alpha}$	$\left(\frac{ST}{N} + \frac{T^2}{N}\right)^{1/3}$
QAI-QROM (ours)	$\left(\frac{ST}{\alpha} + \frac{T^2}{\alpha}\right)^{1/3}$	$\left(\frac{S^5T}{N} + \frac{S^4T^2}{N}\right)^{1/19}$

Table 1: Asymptotic security bounds on the security of OWFs and PRGs constructed from a random function $f : [N] \mapsto [M]$ against (S, T) -algorithms, where $\alpha := \min\{N, M\}$.

Salting defeats preprocessing in the quantum setting. Instead of proving more concrete security bounds using the framework, we show that salting, a common mechanism used in practice for defeating auxiliary input, generically extends the security of applications proven in the QROM to the (Q)AI-QROM. A similar statement was first shown to hold in the classical world by Coretti et al. [CDGS18], where they showed that the security in the ROM can be generically extended into salted security in the AI-ROM.

In this work, we prove the hardness of salted multi-instance game under QROM.

Lemma (Informal statement of Lemma 7.2). *For any security game G with security δ in the QROM against adversary running in time T , let G_S denote its salted version with salt space $[K]$. Then the best winning probability for multi-instance salted game $G_S^{\otimes g}$ is at most $(\delta + gT/K)^g$, which is tight (Section 7.3).*

Combining this lemma with our reduction theorem above, we conclude that salting generically defeats preprocessing in the AI-QROM, and defeats preprocessing against publicly-verifiable and decision games in the QAI-QROM. We believe our techniques are sufficient and both the reduction theorem and salted multi-instance bound can be generalized for proving quantum non-uniform salted security bounds under other idealized cryptographic models, e.g. the ideal-cipher model (which is by definition a salted permutation family), and the salted generic group model. However, in order to simplify presentation, in this work we only focus on the ROM.

Using this generic bootstrapping theorem, we can also easily obtain security bounds for a wide class of salted cryptographic primitives in the AI-QROM and QAI-QROM. In particular, we use it to give the first bound for salted collision-resistant hash (CRH) in AI-QROM and QAI-QROM, resolving an open problem raised by Hhan et al. [HXY19]. This in turn implies that $\text{PWPP}^{\mathcal{O}} \not\subseteq \text{FBQP}^{\mathcal{O}}/\text{qpoly}$ relative to a random oracle \mathcal{O} (Theorem 7.12).

Finally, we note that our proof can also be naturally downgraded to a classical reduction, which gives a new proof of the classical result first proven by Coretti et al. [CDGS18]

1.4 Technical Overview

1.4.1 From Non-uniform Algorithms to Multi-Instance Games

We start by considering applying a general approach in the literature for proving lower bounds for non-uniform algorithms in the context of classical function inversion, and then generalize it to quantum non-uniformity. In particular, the following argument is based off the work of Aaronson [Aar05].

Let $f : [N] \mapsto [N]$ be a function. Consider a classical (randomized) algorithm \mathcal{A} with S -bit of advice $\alpha = \alpha(f)$, T queries to f (which is a lower bound on its running time), that can invert a random image for *any* function² f with probability at least δ , i.e.

$$\delta := \Pr_{\mathcal{A}, x}[\mathcal{A}^f(\alpha, f(x)) \in f^{-1}(f(x))],$$

where the probability is taken over the measurement randomness of the algorithm \mathcal{A} and the random choice of x , and we want to give an upper bound on δ . At a high level, the approach is to reduce it to proving lower bound for the multi-instance version of the problem for algorithms *without* advice. Specifically, we reduce it to bound the success probability of an algorithm \mathcal{B} with gT queries to invert random g inputs $f(x_1), \dots, f(x_g)$ simultaneously, for some parameter $g \in [N]$ on the number of instances. For function inversion, it is easy to show that for any $g > 0$, the best success probability drops exponentially fast in g . Specifically, for any (randomized) algorithm \mathcal{B} and random functions f ,

$$\Pr_{\mathcal{B}, f, x_1, \dots, x_g}[\mathcal{B}^f(f(x_1), \dots, f(x_g)) \text{ inverts } f(x_1), \dots, f(x_g)] \leq O(gT/N)^g. \quad (1)$$

The reduction proceeds in two simple steps. We first use \mathcal{A} to construct an algorithm \mathcal{B}' using *only one copy of the original advice* for the multi-instance problem (with decent success probability), and then get rid of the advice by simply guessing a uniformly random bitstring. The algorithm $\mathcal{B}'^f(\alpha, f(x_1), \dots, f(x_g))$ simply invokes \mathcal{A} to invert each $f(x_i)$, and succeeds when $\mathcal{A}^f(f(x_i))$ succeeds on all g instances. By independence, it is easy to see that

$$\Pr_{\mathcal{B}', x_1, \dots, x_g}[\mathcal{B}'^f(\alpha, f(x_1), \dots, f(x_g)) \text{ inverts } f(x_1), \dots, f(x_g)] \geq \delta^g.$$

Next we remove the advice by guessing. Consider an algorithm \mathcal{B} first guesses a *random* advice $\alpha \in \{0, 1\}^S$ and then runs \mathcal{B}' . Clearly, \mathcal{B} guesses the advice correctly with probability 2^{-S} , in which case \mathcal{B} simulates \mathcal{B}' perfectly. Hence,

$$\Pr_{\mathcal{B}, x_1, \dots, x_g}[\mathcal{B}^f(f(x_1), \dots, f(x_g)) \text{ inverts } f(x_1), \dots, f(x_g)] \geq 2^{-S} \delta^g.$$

As this statement is for any function, the same conclusion holds for random functions. Combining this with the above upper bound (1) on the success probability of \mathcal{B} with $g = S$ shows that $\delta \leq O(ST/N)$, which matches the best known classical bound [Yao90, DGK17, CDGS18].

It should be clear from the above example that this approach is fairly general and reduces the non-uniform lower bound problems to analyze the success probability of the corresponding multi-instance games. Indeed, this approach and its variants have implicitly appeared in various contexts under the name of direct product theorems. We discuss this in more details in Section 1.5.

First attempt using multi-instance problems. Seeing this as a promising start, we now consider the setting of quantum algorithms with *classical* advice. Indeed, the argument works out similarly, except that now we need to consider the success probability of the best *quantum* algorithm that queries gT locations and succeed inverting g independently random images. We could hope that

²Since we are requiring the algorithm to invert any function, the lower bound we present here is slightly weaker. We intentionally make this omission for the overview to highlight the more important ideas in our proofs. Interested readers should refer to the formal proofs.

analyzing the best success probability for quantum algorithms solving the multi-instance problem would lead us to the desired bound.

Unfortunately, it turns out that this approach is destined to fail to achieve any $\delta \ll ST^2/N$. With g quantum queries, the famous Grover's search can find one element in η fraction of all elements with probability $\approx \eta q^2$. Consider the following algorithm $\tilde{\mathcal{B}}$, where it tries to find one pre-image among all the g images using the first T queries, which succeeds with probability gT^2/N . If it succeeds in finding the first pre-image, it then use the next T queries to find one pre-image among all remaining $(g - 1)$ images, which succeeds with probability $(g - 1)T^2/N$ and so on. Using this algorithm, for any function, we can find all g pre-images with probability at least roughly

$$(gT^2/N) \cdot ((g - 1)T^2/N) \cdot \dots \cdot (T^2/N) \approx (T^2/N)^g \cdot g! \approx (gT^2/N)^g.$$

This implies that the best bound we can hope to achieve for the multi-instance problem would be $O(gT^2/N)^g$, which would in turn imply the bound $O(ST^2/N)$.

Bypassing the barrier via multi-instance games. A natural question arises that whether we can go beyond ST^2/N . We claim that this is actually the case.

To see this point, we first recall the high level ideas of the argument above – we first bootstrap the best algorithm \mathcal{A} with advice α for computing f^{-1} with success probability δ , into a multi-instance algorithm \mathcal{B} with advice α with success probability δ^g , and then remove the advice and incur a loss of 2^{-S} , which we amortize into δ^g . The problem essentially reduces to proving a lower bound for the success probability of the resulting algorithm \mathcal{B} that solves the multi-instance problem.

While it may seem like that the $(gT^2/N)^g$ algorithm above seems could be an upper bound for the new problem, we observe that this iterated Grover's search algorithm $\tilde{\mathcal{B}}$ actually never arises from our reduction from \mathcal{A} to \mathcal{B} . In particular, \mathcal{B} always solves each instance one by one, while $\tilde{\mathcal{B}}$ in some sense solves *all* g instances at once. To view this issue in terms of quantum queries, the first T queries by \mathcal{B} are only searching pre-image of $f(x_1)$, but the first T queries by $\tilde{\mathcal{B}}$ are searching for all g pre-images.

We formalize this intuition by strengthening the multi-instance problem into what we call a “multi-instance game,” where the algorithm (or adversary) is instead *interacting* with a verifier (or challenger). For each round $i \in [g]$, the challenger samples a new image $f(x_i)$, and the adversary is given T queries to f , before producing an output x'_i , which the challenger checks whether $f(x_i) = f(x'_i)$. The main change we make to the multi-instance *games*, compared with multi-instance *problems*, is that the adversary gets challenges one-by-one or *sequentially*, rather than getting all challenges at once or in *parallel*.

Observe that this multi-instance game seems to rule out the algorithm $\tilde{\mathcal{B}}$ above, as the adversary does not have any information about $f(x_i)$ until he has issued $(i - 1)T$ queries.

If we assume that the probability that any quantum adversary wins such multi-instance game for function inversion is $O\left(\frac{gT+T^2}{N}\right)^g$, using the same reduction as before, we would reach the conclusion that the best success probability for function inversion with S bits of classical advice and T quantum queries is $O(ST + T^2)/N$. It turns out that we can indeed prove this assumption, but we will defer the discussion and consider quantum advice first.

Beyond classical preprocessing. The reduction above requires the algorithm to solve multiple instances using only a single copy of the advice, which is problematic in the quantum setting due to

no-cloning theorem. We resolve this problem by constructing \mathcal{B} similarly as before, and add gentle measurement to solve multiple instances. To fill in the details, \mathcal{B} does the following.

1. Prepare $k = \Theta(\log g)$ copies of the quantum advice $\beta := \alpha^{\otimes k}$.
2. Boost \mathcal{A} 's success probability from constant to $1 - o(1)$, by running \mathcal{A} on each copy of the advice α k times, and identify the correct answer using one additional query.
3. To solve g instances simultaneously, \mathcal{B} simply runs boosted \mathcal{A} for each instance, and applies measurement. As we have boosted the success probability high enough, the measurement will be “gentle” and we can recover an almost-as-good-as-new quantum advice for the next instance.
4. Finally, to remove the quantum advice, we replace β with a maximally mixed state, which gives us a multiplicative loss of 2^{-Sk} in success probability.

However, for function inversion, this idea seems to fail as function inversion problem can have non-unique correct answers. In particular, if $f^{-1}(f(x)) = \{x_1, x_2, \dots, x_\ell\}$, and the algorithm somehow prepares the answer $|x'_i\rangle$ that is a uniform superposition over all the answers x_1, \dots, x_ℓ , while it succeeds with probability 1, performing a gentle measurement on this answer seems very difficult. This difficulty of performing gentle measurements for function inversion was also acknowledged by the work of Hhan et al. [HXY19], although under a different context.

We claim that using multi-instance *games* (as opposed to multi-instance problems), there is actually a very elegant solution to this. Instead of having the adversary submitting a classical answer x'_i , we will allow the adversary to submit a *quantum state* $|x'_i\rangle$, and the challenger can compute in superposition whether $f(|x'_i\rangle) = f(x_i)$, and measure her decision (which will be gentle, as we boosted its success probability), and send back $|x'_i\rangle$. On a high level, the idea is basically having the adversary and the challenger “jointly” perform this gentle measurement. As the adversary cannot control challenger’s behavior, this change should not impact the best winning probability of the multi-instance game.

We formally define multi-instance games in Section 3.2 and highlight the difference from multi-instance problems in more details there at Remark 3.13.

1.4.2 Analyzing Multi-Instance Game via “Compressed Oracles”

To complete our time-space tradeoff for function inversion, the only remaining step is to bound the best winning probability of the multi-instance game for function inversion. We extend the techniques of Zhandry’s compressed oracles [Zha19] and combine with a new indistinguishability lemma to give a tight bound for this problem.

Compressed oracles. In the classical setting, there is a commonly used technique for arguing random functions called the lazy sampling of a random oracle. The idea is that a simulator will maintain a partial truth table about the random function. Upon an oracle query x , the simulator looks up x in the table D and returns it as the answer. If not found, the simulator freshly samples a new y as the output of x and inserts the pair (x, y) into the table D .

Zhandry observes that, if care is taken to implement the oracle correctly, a quantum analogy of the classical on-the-fly simulation is possible. Unlike the classical simulation, they simulate a random oracle as a superposition of tables, each of which partially instantiates a random function.

Below is the high level idea of their simulation. The table is initialized as an empty table. Upon a quantum query is made by an algorithm, the simulator updates the database in superposition: for a query $|x\rangle$ and a table $|D\rangle$, the simulator look up x in the table D ; if not found, it initializes a superposition of all possible output $\sum_y |y\rangle$ (up to a normalization) as the value of $D(x)$ and updates D in superposition to get $|D \cup (x, y)\rangle$; it then returns $D(x)$ as the output.

A major difference between quantum setting and classical setting is an algorithm may forget some query it made before. As an example, an algorithm can query the same input twice to uncompute everything and thus completely lose the information about the output. Therefore, to perfectly simulate a quantum random oracle, the simulator also checks after every query that if the algorithm loses all information about the query. With all these operations above, Zhandry shows a quantum random oracle can be efficiently simulated on-the-fly.

Analyzing multi-instance game. To prove the success probability of multi-instance function inversion, we consider a stronger statement regarding the success probability of inverting for any round: assume an algorithm already makes $(i - 1)T$ queries for the first $(i - 1)$ rounds, and conditioned on it having passed the first $(i - 1)$ rounds, what is the probability of succeeding in the i -th round by making T queries?

Lemma 1.5. *For any quantum algorithm making $q_0 + q$ queries to a random function $f : [N] \rightarrow [N]$, if $f(x)$ is sampled and given after the q_0 -th query, conditioned on arbitrary outcomes (with non-zero probability) of the algorithm's measurement during the first q_0 queries, the probability of inverting $f(x)$ is at most $O((q_0 + q^2)/N)$.*

We consider the lemma above as remarkable and perhaps even surprising, as intuitively, it is saying that quantum power can achieve a quadratic speed up for search *only* if you know what you are looking for, and there is no classical analogue of this.

With the above lemma, the probability of succeeding in the i -th round is at most $O((iT + T^2)/N)$. Therefore, the probability of succeeding in inverting all random images is at most $O((gT + T^2)/N)^g$.

To prove this lemma, let us start by assuming that a uniformly random image y is instead given as a challenge and without conditioning on the intermediate measurements. By Zhandry's techniques, after making q_0 queries to a random oracle, the knowledge of an algorithm about the random oracle can be viewed as a superposition of some tables with at most q_0 entries which specifies partial random functions over at most q_0 inputs. Since y is sampled uniformly, the amplitude (square root of probability) of database containing y is about $\sqrt{q_0/N}$. After given the image y , each query can increase the amplitude by at most $\sqrt{1/N}$. Therefore, the final amplitude of of database containing y is about $(\sqrt{q_0} + q)/\sqrt{N}$ which gives us the lemma above.

There are two challenges we need to overcome for proving the full lemma.

The first challenge comes from the fact that our lemma statement requires an algorithm is conditioned on some fixed measurement outcomes. To address this, we extend Zhandry's techniques to such settings in Section 2.3 and complete the proofs in Appendix A, which is a very natural but non-trivial extension and crucial for analyzing multi-instance game.

The second challenge seems even more difficult. For function inversion, the image is sampled by first sampling a random pre-image x and computing $f(x)$. The natural application of compressed oracle only gives us a probability bound when a random y is sampled instead of $f(x)$. This is a nontrivial issue as with high probability, the distribution of $f(x)$ is only supported on a constant fraction of $[N]$, so the statistical difference of the two distributions is significant. The issue is more

prominent when we consider the general random functions $f : [N] \mapsto [M]$ where $N \ll M$, where a uniform sample of y is with high probability not an image of any x .

Towards this challenge, we prove the following indistinguishability lemma to bridge the gap. The proof of the lemma does not require the compressed oracle technique and we believe that both lemmas are of independent interest.

Lemma 1.6 (Indistinguishability). *For any quantum algorithm making $q_0 + q$ queries to a random function $f : [N] \rightarrow [N]$, if $f(x)$ or a uniformly random y is sampled and given after the q_0 -th query, conditioned on arbitrary outcomes (with non-zero probability) of the algorithm’s measurement during the first q_0 queries, the advantage of distinguishing is at most $O((q_0 + q^2)/N)$.*

We can then assume a random y is sampled instead of a random $f(x)$ with only an additive loss of $O((gT + T^2)/N)$ in each round, giving us the bound that we desire.

Proving the indistinguishability lemma. We first convert the problem from distinguishing samples (random $f(x)$ or random y) into distinguishing oracles. Assume the oracle is sampled as follows: first, a uniformly random input x is sampled; then a random function f_{-x} defined on all inputs except x is sampled, together with two independently sampled y_0, y_1 ; define $f_{-x}||y$ as a function that outputs $f_{-x}(x')$ on all inputs that are not x and y on input x ; the distinguisher is ask to given either oracle access to $f_{-x}||y_0$ or $f_{-x}||y_1$ and the same challenge y_0 after the q_0 -th query, distinguish which oracle is given (without knowing x). When the function $f_{-x}||y_0$ and challenge y_0 is given, it corresponds to the case a random $f(x)$ is given; while giving the second function corresponds to the second case that a random y is given. It can be shown that the two problems have the exact same difficulty.

Intuitively, every quantum query to a function entangles a quantum algorithm with one of the output of that function in superposition. By making q_0 queries, the algorithm is entangled with at most q_0 outputs of the function in superposition. When y_0 is given, the only way to tell which oracle is given is by already making a query on x and entangling the algorithm with either y_0 or y_1 respectively. Since y_0 has not been given during the first q_0 queries, x is perfectly hidden and completely uniformly random from the algorithm’s view. Thus, such entanglement only happens with probability q_0/N . For the remaining q queries, knowing the information y_0 does help build the entanglement faster. One strategy is to use Grover’s search to check if y_0 is an image of the function since with constant probability, y_0 is not an image of $f_{-x}||y_1$. By using Grover’s search, the advantage of distinguish is about q^2/N and we show such advantage is the best one can get for these q queries. Combining with these two separate analysis, we conclude the indistinguishability lemma.

1.4.3 Yao’s Box Problem

We now focus our attention on Yao’s box problem. Assume that an algorithm \mathcal{A} , given any function $f : [N] \mapsto \{0, 1\}$, prepares an S -qubit advice $\alpha = \alpha(f)$ such that \mathcal{A} can recover $f(x)$ for a random x using α without ever querying $f(x)$ in time T with probability $1/2 + \varepsilon$.

We claim that for classical advice, our reduction from function inversion with advice to multi-instance game can also be generalized to Yao’s box, with a more careful amortizing analysis; and the multi-instance game for Yao’s box is fairly straightforward to argue using similar techniques as for function inversion. Intuitively, for Yao’s box, the only non-trivial strategy is that the adversary’s first $(i - 1)T$ quantum queries predicted the challenge x_i , so the best advantage of any algorithm,

i.e. the best winning probability minus $1/2$, can only be $O(\sqrt{gT/N})$ instead of $O(\sqrt{gT^2/N})$. This leads us to the final bound $1/2 + \tilde{O}(ST/N)^{1/3}$, where the additional exponent loss comes from the new amortizing argument.

For quantum advice, things are a lot trickier. While Aaronson [Aar05] proved a similar lower bound for a different problem against quantum advice, the techniques there only allow us to prove lower bounds against algorithms that find the correct answer with probability at least $2/3$ for *all* f and x (and indeed under this setting, our techniques combined with [Aar05] are sufficient to give query lower bound $ST \geq \tilde{\Omega}(N)$ even for quantum advice). However, this is insufficient in our settings where we need to consider the stronger lower bound where x is sampled randomly, and the algorithm can only predict some $f(x)$ and output a random guess for others.

We first revisit the idea of Aaronson [Aar05], which is to prepare $\text{poly}(\log g)$ copies of the advice, and use majority vote³ to boost the success probability from $2/3$ to $1 - o(1)$ to make the measurement gentle. We show that majority vote cannot possibly boost success probability under the average-case (over x) setting, by considering the following example: the algorithm \mathcal{A} outputs the correct answer with probability 1 on 40% of the inputs, and with probability 0.45 on the other 60% of the inputs. Overall, the success probability is 67%, but with majority vote, the success probability goes down, and can go down arbitrarily close to 40%, which is much worse than random guessing!

One way to resolve this is to instead “gently” measure the success probability of \mathcal{A} for each instance $f(x)$, and throw a biased random coin according to this distribution as our answer. We observe that the problem of “gently” measuring this probability can be reduced to shadow tomography, which is a problem introduced by Aaronson [Aar18]. As our multi-instance game requires that each challenge is given sequentially, we also require the shadow tomography to be able to handle online queries. Aaronson and Rothblum [AR19] showed that online shadow tomography indeed can be done using $S^2 \log^2 g$ copies of the advice, which leads us to the final bound $S^5 T = \tilde{\Omega}(N)$ for constant $\varepsilon > 0$.

1.5 Related Works

In this section, we compare our techniques with other related works.

The approach of reducing time-space tradeoff lower bounds to multi-instance *problems*, which is outlined in Section 1.4.1, has appeared implicitly in various works [Bea91, Kla03, KŠW07], where they usually refer to the (exponential) hardness of multi-instance problems as “(strong) direct product theorems.” While the different approaches presented in different works share some similar high-level ideas, the context and details in each work are slightly different. In this work, to avoid confusion, we use the term “multi-instance problem” instead of direct products. Recently Hamoudi and Magniez [HM20] independently applied the similar technique along with Zhandry’s compressed oracles to prove quantum time-space tradeoffs for finding multiple collisions.

Classically, this approach has also been considered in various works [GKL93, Imp11] for proving non-uniform lower bounds. Aaronson [Aar05] first showed how to employ such ideas when the non-uniform lower bounds need to hold even against quantum advice. While the problem they consider is quite simple and somewhat arbitrary, the starting point we outline in Section 1.4.1 is based off this work.

As far as we are concerned, our work is the first one to consider the stronger variant “multi-

³This idea was implicitly given, where they called it “boosting” under the context of randomized algorithms in complexity theory.

instance games” and show a separation of the two variants for function inversion in Section 1.4.1. Additionally, we present our reduction under a general framework for the stronger variant of multi-instance games.

The idea of using gentle measurements is almost ubiquitous for proving lower bounds against quantum advice. To the best of our knowledge, Aaronson [Aar05] first showed how to combine boosting and gentle measurements for quantum advice lower bounds, which we briefly discuss in Section 1.4.3. In particular, this technique was also employed by Hhan et al. [HXY19] to prove an asymptotic lower bound of $ST^2 \geq N$ for inverting random permutations, although under a different context.

1.6 Open Problems

Quantum time-space tradeoff lower bounds for permutation inversion. While our work provides substantial evidence that the quantum time-space tradeoff bound for inverting permutations is $ST + T^2 = \tilde{\Theta}(N)$, which would have resolved the open problem posted by Nayebi et al. [NABT15, Section 4.3], we are not able to formally prove this due to a lack of “compressed permutation oracles”. This is especially interesting, considering that it is easier to argue about permutations than functions using the compression argument, which is used in prior works [NABT15, HXY19, CLQ19].

Nontrivial quantum speed ups for function inversion. While our lower bound is best possible, it still leaves open the possibility that some nontrivial quantum speed ups exist under the following asymptotic regime:

$$\begin{cases} ST \gg N, \\ T^2 \ll N, \\ S^2T \ll N^2. \end{cases}$$

An especially interesting case is that there might be a quantum algorithm with advice under this regime, but it might seem extremely hard to be “dequantized”.

Our paper also gives many lower bounds, improving any of the bounds that are not tight (for example, the query bound for average-case Yao’s box against quantum advice), or showing new non-trivial attacks, are all interesting possibilities.

Suboptimal exponent on success probability for quantum advice, or is it? All the bounds we have achieved for quantum advice are not tight in terms of the exponent on the success probability for quantum advice. In particular, we note that for function inversion, our classical advice lower bound has exponent 1, which is tight – while the quantum advice bound has exponent $1/3$. Can this loss be avoided, or is there any speed up in terms of S and T for sub-constant success probability?

We make the observation that the loss in exponent ultimately comes from the use of gentle measurements. Looking back at the literature, all the quantum advice lower bound techniques [Aar05, HXY19, CLQ19] we have seen so far always require “reusing” of the advice, which in turns require gentle measurements. Is there a way to avoid reusing the quantum advice to escape this cost? We suspect that this is the case, as in the work of Chung et al. [CLQ19], they presented a reduction to quantum random access code, which by definition seems to avoid this issue, albeit in the end, the advice reusing issue somehow kicks back in.

2 Preliminaries

For any $n \in \mathbb{N}$, we denote $[n]$ to be the set $\{1, 2, \dots, n\}$. We denote $\mathbb{Z}/n\mathbb{Z} = \{0, 1, \dots, n-1\}$ as the ring of integer modulo n , and $\mathbb{F}_2 = \{0, 1\}$ as the binary finite field. For a complex vector $\mathbf{x} \in \mathbb{C}^n$, we denote the L^2 -norm $\|\mathbf{x}\| = \|\mathbf{x}\|_2 = \sum_{i \in [n]} x_i \bar{x}_i$.

Next, we recall some basic facts about quantum computation, and review the relevant literature on the quantum random oracle model.

2.1 Quantum Computation

A quantum system Q is defined over a finite set B of classical states. A **pure state** over Q is a unit vector in $\mathbb{C}^{|B|}$, which assigns a complex number to each element in B . In other words, let $|\phi\rangle$ be a pure state in Q , we can write $|\phi\rangle$ as a column vector:

$$|\phi\rangle = \sum_{x \in B} \alpha_x |x\rangle$$

where $\sum_{x \in B} |\alpha_x|^2 = 1$ and $\{|x\rangle\}_{x \in B}$ is called the “*computational basis*” of $\mathbb{C}^{|B|}$. The computational basis forms an orthonormal basis of $\mathbb{C}^{|B|}$. We define $\langle\phi|$ to be the row vector that is the conjugate of $|\phi\rangle$.

Given two quantum systems Q_1 over B_1 and Q_2 over B_2 , we can define a *product* quantum system $Q_1 \otimes Q_2$ over the set $B_1 \times B_2$. Given $|\phi_1\rangle \in Q_1$ and $|\phi_2\rangle \in Q_2$, we can define the product state $|\phi_1\rangle \otimes |\phi_2\rangle \in Q_1 \otimes Q_2$.

We say $|\phi\rangle \in Q_1 \otimes Q_2$ is *entangled* if there does not exist $|\phi_1\rangle \in Q_1$ and $|\phi_2\rangle \in Q_2$ such that $|\phi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle$. For example, consider $B_1 = B_2 = \{0, 1\}$ and $Q_1 = Q_2 = \mathbb{C}^2$, $|\phi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$ is entangled. Otherwise, we say $|\phi\rangle$ is unentangled.

A pure state $|\phi\rangle \in Q$ can be manipulated by a unitary operator $U \in \mathbb{C}^{|B| \times |B|}$. The resulting state $|\phi'\rangle = U|\phi\rangle$. We denote the trace distance operator $\|U\|_{\text{tr}}$ to be $\frac{1}{2} \text{Tr} \sqrt{U^\dagger U}$.

We extract classical information from a quantum state $|\phi\rangle$ by performing a *measurement*. A measurement specifies an orthonormal basis, typically the computational basis, and the probability of getting result x is $|\langle x|\phi\rangle|^2$. After the measurement, $|\phi\rangle$ “collapses” to the state $|x\rangle$ if the result is x .

For example, given the pure state $|\phi\rangle = \frac{3}{5}|0\rangle + \frac{4}{5}|1\rangle$ measured under $\{|0\rangle, |1\rangle\}$, with probability $9/25$ the result is 0 and $|\phi\rangle$ collapses to $|0\rangle$; with probability $16/25$ the result is 1 and $|\phi\rangle$ collapses to $|1\rangle$.

We assume quantum circuits can implement any unitary transformation (by using these basic gates, Hadamard, phase, CNOT and $\frac{\pi}{8}$ gates), in particular the following two unitary transformations:

- **Classical Computation:** Given a function $f : X \rightarrow Y$, one can implement a unitary U_f over $\mathbb{C}^{|X| \cdot |Y|} \rightarrow \mathbb{C}^{|X| \cdot |Y|}$ such that for any $|\phi\rangle = \sum_{x \in X, y \in Y} \alpha_{x,y} |x, y\rangle$,

$$U_f |\phi\rangle = \sum_{x \in X, y \in Y} \alpha_{x,y} |x, y \oplus f(x)\rangle$$

Here, \oplus is a commutative group operation defined over Y . In particular, if f is given as a classical circuit C , there exists an efficient implementation of the unitary U_f using $|C|$ ancillas, and each gate is evaluated at most twice.

- **Quantum Fourier Transform:** For every $n \in \mathbb{N}$, the quantum Fourier transform QFT_n is a unitary operation, that is given a quantum state $|\phi\rangle = \sum_{j \in \mathbb{Z}/n\mathbb{Z}} x_j |j\rangle$, outputs $|\psi\rangle = \sum_{k \in \mathbb{Z}/n\mathbb{Z}} y_k |k\rangle$ where the sequence $\{y_k\}_k$ is the Fourier transform to the sequence $\{x_j\}_j$, i.e.

$$y_k = \frac{1}{\sqrt{n}} \sum_{j \in \mathbb{Z}/n\mathbb{Z}} \omega_n^{jk} x_j$$

where $\omega_n = e^{2\pi i/n}$, and $i = \sqrt{-1}$ is the imaginary unit.

2.2 Quantum Random Oracle Model

An oracle-aided quantum algorithm can perform quantum computation as well as quantum oracle query. A quantum oracle query for an oracle $f : [N] \rightarrow [M]$ is modeled as a unitary $U_f : |x\rangle |u\rangle = |x\rangle |u + f(x)\rangle$, where $+$ denotes addition in integer ring $\mathbb{Z}/M\mathbb{Z}$ (we take the natural bijection that $M \simeq 0$, but any bijection $[M] \leftrightarrow \mathbb{Z}/M\mathbb{Z}$ suffice for our purposes).

A random oracle is a random function $H : [N] \rightarrow [M]$. The random function H is chosen at the beginning. A quantum algorithm making T oracle queries to H can be modeled as the following: it has three registers $|x\rangle, |u\rangle, |z\rangle$, where $x \in [N], u \in \mathbb{Z}/M\mathbb{Z}$ and z is the algorithm's internal working memory; it starts with some input state $|0\rangle |0\rangle |\psi\rangle$, then it applies a sequence of unitary to the state: $U_0, U_H, U_1, U_H, \dots, U_{T-1}, U_H, U_T$ and a final measurement over computational basis. Each U_H is the quantum oracle query unitary $U_H : |x\rangle |u\rangle = |x\rangle |u + H(x)\rangle$ and U_i is the local quantum computation that is independent of H . We can always assume there is only one measurement which is a measurement over computational basis and applied at the last step of the algorithm.

2.3 Compressed Oracle

In this subsection, we recall the technique introduced by Zhandry [Zha19]. We will explain how to purify a random oracle in the quantum setting first, and then give equivalent forms of a quantum random oracle, namely standard oracle **StO**, phase oracle **PhO** and compressed standard oracle **CStO**. All these oracles are equivalent in the sense that for every (even unbounded) algorithm making queries to one of these oracles, the output distribution of the algorithm is exactly identical regardless of which oracle is given. Then Zhandry shows dealing with compressed standard oracle is usually easier. Roughly speaking, Zhandry shows that with compressed standard oracle, one could quantify the amount of the information about the random oracle learned by any quantum algorithm, analogous to the lazy sampling technique that is very commonly used for classical random oracles.

Note that in Zhandry's work [Zha19], they originally only considered output of size $M = 2^m$, and implementing a quantum random oracle as $U_H : |x\rangle |u\rangle = |x\rangle |u \oplus H(x)\rangle$ where \oplus is bit-wise XOR, or the addition over \mathbb{F}_2^m . Therefore, their description of the compressed oracle technique is different since the range is defined as \mathbb{F}_2^m instead of $\mathbb{Z}/M\mathbb{Z}$ considered in this paper. Two oracles are equivalent as we can simulate one with the other using two queries. For the completeness of the paper, we will reprove some of the useful lemmas under the integer ring $\mathbb{Z}/M\mathbb{Z}$.

Also note that Zhandry also showed that a compressed oracle can be efficiently implemented by a quantum computer, i.e. the running time is only polynomial in the number of queries and $\log N, \log M$. In this work, since we mainly consider query complexity and for presentation, we ignore the issue of efficiency for a simpler presentation.

Purification: standard oracle. Let H be a random oracle $[N] \rightarrow [M]$. The function H is sampled at the very beginning, or equivalently, initially we prepare a maximally mixed state $\eta \sum_H |H\rangle \langle H|$ up to some normalization factor η , and each query can be implemented by another unitary U , which reads the function H and applies U_H . However, we can “purify” the random oracle, meaning that we can replace the mixed state of $|H\rangle$ with a uniform superposition of all possible functions, i.e. $\sqrt{\eta} \sum_H |H\rangle$. Consider the truth table of H , that is $|H\rangle = |H(1)\rangle |H(2)\rangle \cdots |H(N)\rangle$. Let \mathcal{A} be any quantum algorithm. We say the algorithm can query the standard oracle if we treat the algorithm’s registers and $|H\rangle$ as a whole system, initialized as $|0\rangle |\psi\rangle \otimes \frac{1}{M^{N/2}} \sum_H |H\rangle$. An oracle query **StO** in this purified state is defined as,

$$\text{StO } |x\rangle |u\rangle |z\rangle \otimes |H\rangle = |x\rangle |u + H(x)\rangle |z\rangle \otimes |H\rangle,$$

where $|x\rangle, |u\rangle$ are the input and output register, $|z\rangle$ is an arbitrary working register and $|H\rangle$ is the random oracle. Each local quantum computation is $U_i \otimes I$ which only operates on \mathcal{A} ’s registers $|x\rangle |u\rangle |z\rangle$. Therefore the computation of any \mathcal{A} can be described as a sequence of: $U_0 \otimes I$, **StO**, \dots , $U_{T-1} \otimes I$, **StO**, $U_T \otimes I$, and a final computational measurement over \mathcal{A} ’s register. The following proposition tells that the output distribution using a standard oracle is exactly the same as using a random oracle.

Lemma 2.1 ([Zha19, Lemma 2]). *Let \mathcal{A} be an (unbounded) quantum algorithm making oracle queries. The output of \mathcal{A} given a random function H is exactly identical to the output of \mathcal{A} given access to a standard oracle. Therefore, a random oracle with quantum query access can be perfectly simulated as a standard oracle.*

Phase kickback: phase oracle. Define a unitary V as $(I_x \otimes \text{QFT}_M^\dagger \otimes I_H)$ which applies QFT_M^\dagger on the output register $|u\rangle$. Define the phase oracle operator $\text{PhO} := V^\dagger \cdot \text{StO} \cdot V$.

$$\begin{aligned} \text{PhO } |x\rangle |u\rangle \otimes |H\rangle &= V^\dagger \cdot \text{StO} \cdot \frac{1}{\sqrt{M}} \sum_y \omega_M^{-uy} |x\rangle |y\rangle \otimes |H\rangle \\ &= V^\dagger \cdot \frac{1}{\sqrt{M}} \sum_y \omega_M^{-uy} |x\rangle |y + H(x)\rangle \otimes |H\rangle \\ &= \frac{1}{M} \sum_{y, y'} \omega_M^{-uy + (y+H(x))y'} |x\rangle |y'\rangle \otimes |H\rangle \\ &= \frac{1}{M} \omega_M^{uH(x)} \sum_{y, y'} \omega_M^{(y+H(x))(y'-u)} |x\rangle |y'\rangle \otimes |H\rangle \\ &= |x\rangle |u\rangle \otimes \omega_M^{uH(x)} |H\rangle. \end{aligned}$$

Similarly, we override the notation **PhO** such that for any auxiliary register $|z\rangle$, $\text{PhO } |x\rangle |u\rangle |z\rangle \otimes |H\rangle = |x\rangle |u\rangle |z\rangle \otimes \omega_M^{uH(x)} |H\rangle$.

Observing that $VV^\dagger = I$, the following lemma tells that we can efficiently convert between a standard oracle algorithm and a phase oracle algorithm.

Lemma 2.2 ([Zha19, Lemma 3]). *Let \mathcal{A} be an (unbounded) quantum algorithm making queries to a standard oracle. Let \mathcal{B} be the algorithm that is identical to \mathcal{A} , except it performs V and V^\dagger before and after each query. Then the output distributions of \mathcal{A} (given access to a standard oracle) and \mathcal{B}*

(given access to a phase oracle) are identical. Therefore, a quantum random oracle can be perfectly simulated as a phase oracle.

We then have the following lemma for the phase oracle, that formulates the behavior of a quantum algorithm making at most T queries to the phase oracle. We have seen that every PhO query will add a phase to the $|H\rangle$ register, i.e., $\text{PhO } |x\rangle |u\rangle \otimes |H\rangle = |x\rangle |u\rangle \otimes \omega_M^{uH(x)} |H\rangle$. Define D as a truth table, or equivalently a vector in $(\mathbb{Z}/M\mathbb{Z})^N$ and $D(x)$ be the x -th entry of D . Define $|D|$ be the number of non-zero entries in D . For any D , we define $|\phi_D\rangle = \frac{1}{M^{N/2}} \sum_H \omega_M^{\langle D, H \rangle} |H\rangle$ for all $D \in (\mathbb{Z}/M\mathbb{Z})^N$ where $\langle D, H \rangle$ is defined to be the inner product $\sum_{x \in [N]} D(x)H(x)$. Note that we will only use this inner product on the exponent of ω_M so it is irrelevant whether we are computing it on the integer ring or the ring modulo M .

Lemma 2.3. *Let \mathcal{A} be a quantum algorithm making at most T queries to a phase oracle. The overall state of \mathcal{A} and the phase oracle can be written as $\sum_{z, D: |D| \leq T} \alpha_{z, D} |z\rangle \otimes \frac{1}{M^{N/2}} \sum_H \omega_M^{\langle D, H \rangle} |H\rangle = \sum_{z, D: |D| \leq T} \alpha_{z, D} |z\rangle \otimes |\phi_D\rangle$.*

Moreover, it is true even if the state is conditioned on arbitrary outcomes (with non-zero probability) of \mathcal{A} 's intermediate measurements.

For completeness, we provide the proof for this lemma in Appendix A.

Compressed standard oracle. Intuitively, compressed oracle is an analogy of classical lazy sampling method. Instead of recording all the information of H in the registers (like what it does in the standard oracle or the phase oracle), Zhandry provides a better solution which is useful to argue the amount of the information an algorithm knows about the random oracle.

The oracle register records a database/list that contains the output on each input x , the output is an element in $\mathbb{Z}/M\mathbb{Z} \cup \{\perp\}$, where \perp is a special symbol denoting that the value is “uninitialized”. The database is initialized as an empty list D_0 of length N , in other words, it is initialized as the pure state $|\emptyset\rangle := |\perp, \perp, \dots, \perp\rangle$. Let $|D|$ denote the number of entries in D that are not \perp . Define $D(x)$ to be the x -th entry.

For any D and x such that $D(x) = \perp$, we define $D \cup (x, u)$ to be the database D' , such that for every $x' \neq x$, $D'(x') = D(x)$ and at the input x , $D'(x) = u$.

The compressed standard oracle is the unitary $\text{CStO} := \text{StdDecomp} \circ \text{CStO}' \circ \text{StdDecomp}$, where

- $\text{CStO}' |x, u\rangle |D\rangle = |x, u + D(x)\rangle |D\rangle$ when $D(x) \neq \perp$, which writes the output of x defined in D to the u register.
- $\text{StdDecomp}_x |D\rangle$, which works on the x -th register of the database $D(x)$:
 - If $D(x) = \perp$, StdDecomp_x maps $|\perp\rangle$ to $\frac{1}{\sqrt{M}} \sum_y |y\rangle$, or equivalently, $\text{StdDecomp}_x |D\rangle = \frac{1}{\sqrt{M}} \sum_y |D \cup (x, y)\rangle$. Intuitively, if the database does not contain information about x , it samples a fresh y as the output of x .
 - If $D(x) \neq \perp$, StdDecomp_x works on the x -th register, and it is an identity on $\frac{1}{\sqrt{M}} \sum_y \omega_M^{uy} |y\rangle$ for all $u \neq 0$; it maps the uniform superposition $\frac{1}{\sqrt{M}} \sum_y |y\rangle$ to $|\perp\rangle$.

More formally, for a D' such that $D'(x) = \perp$,

$$\text{StdDecomp}_x \frac{1}{\sqrt{M}} \sum_y \omega_M^{uy} |D' \cup (x, y)\rangle = \frac{1}{\sqrt{M}} \sum_y \omega_M^{uy} |D' \cup (x, y)\rangle \text{ for any } u \neq 0,$$

and,

$$\text{StdDecomp}_x \frac{1}{\sqrt{M}} \sum_y |D' \cup (x, y)\rangle = |D'\rangle.$$

Since all $\frac{1}{\sqrt{M}} \sum_y \omega_M^{xy} |y\rangle$ and $|\perp\rangle$ form a basis, these requirements define a unique unitary operation. We define $\text{StdDecomp}(|x\rangle \otimes |D\rangle) := |x\rangle \otimes \text{StdDecomp}_x |D\rangle$.

Zhandry proves that, StO and CStO are perfectly indistinguishable by any *unbounded* quantum algorithm.

Lemma 2.4 ([Zha19, Lemma 4]). *Let \mathcal{A} be an (unbounded) quantum algorithm making oracle queries. The output of \mathcal{A} given access to the standard oracle is exactly identical to the output of \mathcal{A} given access to a compressed standard oracle.*

Combining this lemma with Lemma 2.1, we obtain the following corollary.

Corollary 2.5. *A quantum random oracle can be perfectly simulated as a compressed standard oracle.*

In this work, we only consider query complexity, and thus simulation efficiency is irrelevant to us. Looking ahead, we simulate a random oracle as a compressed standard oracle to help us analyze security of different games with the help from the following lemmas.

The first lemma gives a general formulation of the overall state of \mathcal{A} and the compressed standard oracle after \mathcal{A} makes T queries, analogous to Lemma 2.3 for phase oracle. We also defer the proof for this lemma to Appendix A.

Lemma 2.6. *If \mathcal{A} makes at most T queries to a compressed standard oracle, assuming the overall state of \mathcal{A} and the compressed standard oracle is $\sum_{z,D} \alpha_{z,D} |z\rangle_{\mathcal{A}} |D\rangle_H$, then it has support on all D such that $|D| \leq T$. In other words, the overall state can be written as,*

$$\sum_{z,D:|D|\leq T} \alpha_{z,D} |z\rangle_{\mathcal{A}} \otimes |D\rangle_H.$$

Moreover, it is true even if the state is conditioned on arbitrary outcomes (with non-zero probability) of \mathcal{A} 's intermediate measurements.

The second lemma provides a quantum analogue of lazy sampling in the classical ROM.

Lemma 2.7 ([Zha19, Lemma 5]). *Let H be a random oracle from $[N] \rightarrow [M]$. Consider a quantum algorithm \mathcal{A} making queries to the standard oracle and outputting tuples $(x_1, \dots, x_k, y_1, \dots, y_k, z)$. Supposed the random function H is measured after \mathcal{A} produces its output. Let R be an arbitrary set of such tuples. Suppose with probability p , \mathcal{A} outputs a tuple such that (1) the tuple is in R and (2) $H(x_i) = y_i$ for all i . Now consider running \mathcal{A} with the compressed standard oracle CStO, and supposed the database D is measured after \mathcal{A} produces its output. Let p' be the probability that (1) the tuple is in R and (2) $D(x_i) = y_i$ (in particular, $D(x_i) \neq \perp$) for all i . Then $\sqrt{p} \leq \sqrt{p'} + \sqrt{k/M}$.*

Moreover, it is true even if it is conditioned on arbitrary outcomes (with non-zero probability) of \mathcal{A} 's intermediate measurements.

3 Security Games

3.1 Security Game with Advice

Definition 3.1 (Algorithm with Advice). An (S, T) (query) algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with (oracle-dependent) advice consists of two procedures:

- $|\alpha\rangle \leftarrow \mathcal{A}_1(H)$, which is an arbitrary (unbounded) function of H , and outputs an S -qubit quantum state $|\alpha\rangle$;
- $|\text{ans}\rangle \leftarrow \mathcal{A}_2^{\tilde{H}}(|\alpha\rangle, \text{ch})$, which is an unbounded algorithm that takes advice $|\alpha\rangle$, a challenge ch , makes at most T quantum queries to \tilde{H} , and outputs an answer, which we measure in the computational basis to obtain the classical answer ans if needed.

Furthermore, we distinguish the following cases:

- If both the output of \mathcal{A}_1 and all queries of \mathcal{A}_2 are classical queries⁴, we call it a classical algorithm with (classical) advice, or an (S, T) algorithm in the AI-ROM (auxiliary input random oracle model);
- If only the output of \mathcal{A}_1 is classical, we call it a quantum algorithm with classical advice, or an (S, T) algorithm in the AI-QROM (auxiliary input quantum random oracle model);
- Otherwise, we call it a quantum algorithm with (quantum) advice, or an (S, T) algorithm in the QAI-QROM (quantum auxiliary input quantum random oracle model).
- If $S = 0$, we call it an classical/quantum algorithm without advice, or an algorithm in the ROM (random oracle model)/QROM (quantum random oracle model) respectively.

Remark 3.2. In the above definition, we assume the advice is a pure state without loss of generality. Mentioned in [Aar05], by Kraus' Theorem, every S -qubit mixed state can be realized as half of a $2S$ -qubit pure state.

Below, we will use the word "adversary" and "algorithm" interchangeably, especially when we consider interactive security games shortly after.

Definition 3.3 (Security Game). Let H be a random oracle $[N] \rightarrow [M]$. A (non-interactive) (classical) security game $G = (C)$ is specified by a challenger $C = (\text{Samp}, \text{Query}, \text{Ver})$, where:

1. $\text{ch} \leftarrow \text{Samp}^H(r)$ is a deterministic classical algorithm that takes randomness $r \in R$ as input, and outputs a challenge ch .
2. $\text{Query}^H(r, \cdot)$ is a deterministic classical algorithm that hardcodes the challenge and provides adversary's online queries⁵.
3. $b \leftarrow \text{Ver}^H(r, \text{ans})$ is a deterministic classical algorithm that takes the input ans and outputs a decision b indicating whether the game is won.

⁴Here, we do not distinguish whether $\mathcal{A}_2^{\tilde{H}}$ is a quantum algorithm making classical queries or a completely classical algorithm making classical queries, since a quantum algorithm can always be simulated in classical finite time.

⁵For almost all applications below, $\text{Query}^H(r, \cdot) = H(\cdot)$. The only exception is Yao's box, where the adversary cannot query the challenge point.

For every algorithm with advice, i.e. $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we define

$$\mathcal{A} \iff C(H) := \text{Ver}^H \left(r, \mathcal{A}_2^{\tilde{H}}(\mathcal{A}_1(H), \text{Samp}^H(r)) \right)$$

to be the binary variable indicating whether \mathcal{A} successfully makes the challenger output 1, or equivalently if \mathcal{A} wins the security game, where $\tilde{H}(\cdot) := \text{Query}^H(r, \cdot)$.

Definition 3.4 (Security in the AI-ROM/AI-QROM/QAI-QROM). We define the security in the AI-ROM, AI-QROM, QAI-QROM of a security game G to be

$$\delta = \delta(S, T) := \sup_{\mathcal{A}} \Pr_{H, r, \mathcal{A}} [\mathcal{A} \iff C(H) = 1],$$

where \mathcal{A} in the probability denotes the randomness of the algorithm, and supremum is taken over all \mathcal{A} in the AI-ROM/AI-QROM/QAI-QROM (respectively).

Additionally, we can say a security game G is δ -secure if its security is at most δ .

For some games, it is helpful to consider how much advantage an algorithm can gain by getting more queries or more advice. This is formalized as below.

Definition 3.5. We call the security game a **decision game** if $\text{ans} \in \{0, 1\}$.

Definition 3.6 (Advantage against Decision Games). We define the advantage of \mathcal{A} for a decision game G to be

$$\varepsilon = \varepsilon(S, T) := \delta(S, T) - 1/2,$$

if it has winning probability $\delta(S, T)$.

Definition 3.7 (Best Advantage of Decision Games). We define the best advantage of a decision game G in model \mathcal{M} to be $\varepsilon(S, T) := \delta(S, T) - 1/2$ if G has security $\delta(S, T)$ in \mathcal{M} , where the \mathcal{M} could be AI-ROM, AI-QROM, or QAI-QROM.

Another class of security games that we would like to consider is when the adversary can verify the answer by itself, which we formalize as below.

Definition 3.8. We call a security game to be **publicly-verifiable** if $\text{Ver}^H(r, \cdot) = \widetilde{\text{Ver}}^{\tilde{H}}(\text{ch}, \cdot)$ for some classical algorithm $\widetilde{\text{Ver}}^{\tilde{H}}$ where $\tilde{H}(\cdot) = \text{Query}^H(r, \cdot)$.

Since by our definition of security games, an algorithm has only access to $\tilde{H}(\cdot)$ which may not be the same as $H(\cdot)$. Therefore, publicly-verifiable means an algorithm can verify an answer by only making queries to \tilde{H} .

Remark 3.9. In general, interesting decision games are not publicly-verifiable, otherwise, there exists an adversary that breaks the game using only as many queries as it takes for the challenger to verify the answer.

3.2 Multi-Instance Security

In this section, we introduce multi-instance game and multi-instance security.

Definition 3.10 (Multi-Instance Security Game). *For any security game $G = (C)$ and any positive integer g , we define the multi-instance game $G^{\otimes g} = (C^{\otimes g})$, where $C^{\otimes g}$ is given as follows:*

1. For $i \in [g]$, do:
 - (a) Sample fresh randomness $r_i \leftarrow_{\S} R$;
 - (b) Compute $\text{ch}_i \leftarrow C.\text{Samp}^H(r_i)$ and send it to the adversary;
 - (c) Give adversary access to oracle $C.\text{Query}^H(r_i, \cdot)$ until the adversary submits a quantum state (register) $|\text{ans}_i\rangle$;
 - (d) Let $\{P_0, P_1\}$ be a projective measurement where P_1 defines all ans that $C.\text{Ver}^H(r_i, \text{ans}) = 1$ and $P_0 = I - P_1$. Measure $|\text{ans}_i\rangle$ in $\{P_0, P_1\}$ to get the quantum state $|\text{ans}'_i\rangle$ and store the result in b_i ;
 - (e) Send $|\text{ans}'_i\rangle$ back to the adversary.
2. Output $b_1 \wedge b_2 \wedge \dots \wedge b_g$.

Definition 3.11. *A (g, S, T) adversary with advice for a multi-instance security game $G^{\otimes g} = (C^{\otimes g})$ is defined as $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, where the interactions between $\mathcal{A}_2(|\alpha\rangle)$ and $C^{\otimes g}$ is defined as follows.*

1. $|\alpha\rangle \leftarrow \mathcal{A}_1(H)$, which is an arbitrary (unbounded) function of H , and outputs an S -qubit quantum state $|\alpha\rangle$ for \mathcal{A}_2 ;
2. For each $i \in [g]$,
 - (a) \mathcal{A}_2 is given a challenge ch_i and an oracle \tilde{H}_i from $C^{\otimes g}$;
 - (b) \mathcal{A}_2 makes at most T queries to \tilde{H}_i and prepares $|\text{ans}_i\rangle$;
 - (c) \mathcal{A}_2 sends $|\text{ans}_i\rangle$ to $C^{\otimes g}$ and gets $|\text{ans}'_i\rangle$ back;
3. Finally, $C^{\otimes g}$ outputs a bit b .

In particular, if $S = 0$, we also call it (g, T) classical/quantum adversary (without advice), or a (g, T) algorithm in the ROM/QROM respectively.

For any \mathcal{A} , which is a (g, S, T) adversary with advice, we define

$$\mathcal{A} \iff C^{\otimes g}(H)$$

to be the binary variable indicating whether \mathcal{A} successfully makes the challenger output 1 in the game defined above, or equivalently if \mathcal{A} wins the multi-instance security game.

Definition 3.12 (Multi-Instance Security). *We say a multi-instance game $G^{\otimes g}$ is δ -secure in the QROM (quantum random oracle model) if for any (g, T) adversary \mathcal{A} ,*

$$\Pr_{H, \mathcal{A}, C^{\otimes g}} [\mathcal{A} \iff C^{\otimes g}(H)] \leq \delta^g = \delta(g, T)^g,$$

where \mathcal{A} in the probability denotes the randomness of the algorithm, $C^{\otimes g}$ in the probability denotes the randomness of the challenger.

Remark 3.13 (Comparison with multi-instance problems). *We expand our discussion in Section 1.4.1 by noting that our multi-instance game is different from multi-instance problems (also similar to direct product theorem) in two important aspects:*

1. *The adversary gets back $|\text{ans}'_i\rangle$ after the challenger measures her decision. For a classical adversary (i.e. if $|\text{ans}'_i\rangle$ is classical), this makes no difference. For a quantum adversary, this makes the game easier (δ might go up). Looking ahead, this change is only necessary for Theorem 4.3 but we state this for general games as it does not hurt the general bound in all the cases we consider.*
2. *The adversary gets access to ch_{i+1} only after submitting his $|\text{ans}_i\rangle$. This makes the game harder, especially for quantum adversaries, as discussed in Section 1.4.1.*

3.3 Helper Lemma for Multi-Instance Security

We now show a helpful lemma, which shows that to prove multi-instance security, it suffices to prove conditional security for each round.

Lemma 3.14. *Let G be any security game. Let B_i be the random variable for b_i for the multi-instance game $G^{\otimes g}$ as defined in Definition 3.10, and let $X := (B_1, B_2, \dots, B_{i-1}, Y)$ be the random variable that denote the outcomes of all the measurements during the first $(i-1)$ rounds. For any δ , if for any multi-instance adversary \mathcal{A} and for any x that $\Pr[X = x] > 0$, we have that*

$$\Pr[B_i = 1 | X = x] \leq \delta,$$

then $G^{\otimes g}$ is δ -secure.

Proof. We note that the success probability of \mathcal{A} is exactly

$$\begin{aligned} & \Pr[B_1 = B_2 = \dots = B_g = 1] \\ &= \prod_{i=1}^g \Pr[B_i = 1 | B_1 = \dots = B_{i-1} = 1] \\ &= \prod_{i=1}^g \mathbb{E}_Y [\Pr[B_i = 1 | B_1 = \dots = B_{i-1} = 1, Y]] \\ &\leq \delta^g. \end{aligned}$$

Here in the last inequality, we implicitly assume that the expectation is only taken over $\Pr[Y | B_1 = \dots = B_{i-1} = 1] > 0$, as otherwise the term will trivially evaluate to zero. \square

4 Reducing Games with Advice to Multi-Instance Games

In this section, we give four reductions from security against (g, T) multi-instance adversaries to security against (S, T) adversaries with advice.

We first present the theorem that bootstraps security of multi-instance game to security against quantum adversaries with classical advice or completely classical adversaries.

Theorem 4.1. *Given a security game G . Let δ_0 be the winning probability of an adversary that outputs a random answer without advice or making any query. Assume that multi-instance game $G^{\otimes g}$ is $\delta(g, T)$ -secure in QROM (or ROM), G 's security against (S, T) adversaries in AI-QROM (or AI-ROM, respectively) is*

$$\delta' \leq 4 \cdot \delta(S + \log(1/\delta_0) + 1, T).$$

Remark. *Looking ahead, we show in Lemma 5.2 that for OWF, its multi-instance security in the QROM is $\delta(g, T) = O\left(\frac{gT+T^2}{\min\{N, M\}}\right)$. The winning probability of random guess δ_0 is at least $1/N$. Plug these parameters into the corollary above, we have that OWF's security in the AI-QROM is at most*

$$O\left(\frac{(S+T+\log N) \cdot T}{\min\{N, M\}}\right) = \tilde{O}\left(\frac{ST+T^2}{\min\{N, M\}}\right).$$

While this theorem is very general, it yields no meaningful bound when $\delta \geq 1/4$, which is the case for decision games as a random guessing adversary achieves success probability $1/2$! We show that this is not an issue, by showing that a more careful application of the reduction that is used for proving the theorem above, actually yields the following bound for decision games against classical advice, although with worse exponents on $\varepsilon' = \delta' - 1/2$.

Theorem 4.2. *Let G be a decision game. Let ε_0 be a lower bound on the advantage for an adversary with T_0 queries and S_0 bits of advice. Assume that multi-instance game $G^{\otimes g}$ is $(1/2 + \varepsilon(g, T))$ -secure (or has best advantage $\varepsilon(g, T)$) in QROM (or ROM). For any (S, T) adversaries in AI-QROM (or AI-ROM, respectively), the best advantage against G is ε' , which satisfies,*

$$\varepsilon' \leq 4 \cdot \varepsilon\left(\frac{10 \ln 2}{\varepsilon'} \cdot (S + S_0 + \log(1/\varepsilon_0) + 2), T + T_0\right).$$

Remark. *Looking ahead, we show in Lemma 6.2 that for Yao's box, its multi-instance best advantage is $\varepsilon(g, T) = O\left(\sqrt{\frac{gT}{N}}\right)$. When $S_0 = 1$ and $T_0 = 0$, $\varepsilon_0 \geq \frac{1}{2N}$ (which is simply remembering $H(1)$). Plug these parameters into the theorem above, we have that Yao's box's best advantage in the AI-QROM ε' satisfies*

$$\varepsilon' \leq O\left(\sqrt{\frac{(S+1+\log N+2) \cdot T}{\varepsilon' N}}\right).$$

Therefore, we conclude that

$$\varepsilon' \leq O\left(\frac{(S+\log N) \cdot T}{N}\right)^{1/3} = \tilde{O}\left(\frac{ST}{N}\right)^{1/3}.$$

These two theorems conclude the bounds for adversaries with classical advice. Next, we consider bounds for quantum adversaries with advice, which is slightly worse than their classical counterparts.

We present a different reduction for quantum adversaries with advice, albeit the reduction only works for publicly-verifiable games.

Theorem 4.3. *Given a publicly-verifiable security game G , let T_{Ver} be the upper bound on the number of \tilde{H} queries for computing $\text{Ver}^{\tilde{H}}(ch, \cdot)$. Assume that $\delta_0 = 1/\text{poly}(N, M)$ is the winning probability by random guess, and assume that multi-instance game $G^{\otimes g}$ is $\delta(g, T)$ -secure in the*

QROM. For any (S, T) adversaries in the QAI-QROM, the best advantage against G is δ' , which satisfies

$$\delta' \leq 8 \cdot \delta \left(\tilde{O}(S)/\delta', \tilde{O}(T + T_{\text{Ver}})/\delta' \right),$$

where \tilde{O} absorbs $\text{poly}(\log N, \log M, \log S)$ factors.

Remark. Recall that for OWF, its multi-instance security is $\delta(g, T) = O\left(\frac{gT+T^2}{\min\{N, M\}}\right)$ and $T_{\text{Ver}} = 1$; for random guess, the winning probability is $\delta_0 \geq 1/N$. Plug these parameters into the corollary above, we have that OWF's security δ' in the QAI-QROM satisfies

$$\delta' \leq \tilde{O} \left(\frac{ST + T^2}{\delta'^2 \cdot \min\{N, M\}} \right).$$

In conclusion,

$$\delta' \leq \tilde{O} \left(\frac{ST + T^2}{\min\{N, M\}} \right)^{1/3}.$$

As noted before that publicly-verifiable games and decision games are essentially disjoint, we finally present the fourth reduction for decision games.

Theorem 4.4. Given a decision security game G . Assume that $\varepsilon_0 = 1/\text{poly}(N, M)$ is a non-negligible lower bound on the advantage for an adversary with T_0 queries and S_0 qubits of advice, and assume that multi-instance game $G^{\otimes g}$ is $\varepsilon(g, T)$ -secure in the QROM. For any (S, T) adversaries in QAI-QROM, the best advantage against G is ε' , which satisfies

$$\varepsilon' \leq 8 \cdot \varepsilon \left(\tilde{O}(S + S_0)^3/\varepsilon'^9, \tilde{O}((T + T_0)(S + S_0)^2)/\varepsilon'^8 \right),$$

where \tilde{O} absorbs $\text{poly}(\log N, \log M, \log S)$ factors.

Remark. Recall that for Yao's box, its multi-instance advantage is $\varepsilon(g, T) = O\left(\sqrt{\frac{gT}{N}}\right)$; when $S_0 = 1$ and $T_0 = 0$, $\varepsilon_0 \geq \frac{1}{2N}$ (which is simply remembering $H(1)$). Plug these parameters into the corollary above, we have that Yao's box's best advantage ε' in the QAI-QROM satisfies

$$\varepsilon' \leq \tilde{O} \left(\sqrt{\frac{S^5 T}{\varepsilon'^{17} N}} \right),$$

therefore, we conclude that

$$\varepsilon' \leq \tilde{O} \left(\frac{S^5 T}{N} \right)^{1/19}.$$

4.1 Reduction for Classical Advice

Fact 4.5. Given events E_1, E_2, \dots, E_N , any real number $B \geq 0$, ε , let I be a uniformly random integer in $[N]$. If $\Pr[E_I] \geq B + \varepsilon$, there exists a subset $S \subseteq [N]$ of size $|S| \geq \varepsilon N/2$, such that for any $i \in S$, $\Pr[E_i] \geq B + \varepsilon/2$.

Proof. Without loss of generality, assume $\varepsilon > 0$. Let S^* be the subset of $[N]$ that contains all $i \in [N]$ whose $\Pr[E_i] \geq B + \varepsilon/2$. Assume for contradiction, $|S^*| < \varepsilon N/2$. We have,

$$\begin{aligned} B + \varepsilon &\leq \Pr[E_I] \\ &\leq 1 \cdot \Pr[I \in S^*] + \Pr[E_I | I \notin S^*] \cdot \Pr[I \notin S^*] \\ &< 1 \cdot \varepsilon/2 + (B + \varepsilon/2) \cdot 1 \\ &= B + \varepsilon, \end{aligned}$$

which is a contradiction. \square

Proposition 4.6. *Given a security game G . Given an (S, T) adversary with classical advice \mathcal{A} for a security game G with winning probability δ , then for all real $B \geq 0$, let $\delta = B + \varepsilon$, there exists an (g, S, T) adversary with classical advice \mathcal{A}' for the multi-instance game $G^{\otimes g}$ with winning probability $\delta' \geq \varepsilon/2 \cdot (B + \varepsilon/2)^g$ for any $g > 0$.*

Furthermore, this reduction is classical, meaning that if \mathcal{A} is a classical algorithm with advice, so is \mathcal{A}' .

Proof. Without loss of generality, assume $B \leq \delta$ or $\varepsilon \geq 0$. Consider the following adversary with advice \mathcal{A}' :

1. Compute $\alpha \leftarrow \mathcal{A}_1(H)$;
2. For $i \in [g]$, on receiving challenge ch_i and given oracle access to \tilde{H}_i , computes $\mathcal{A}_2^{\tilde{H}_i}(\alpha, \text{ch}_i)$, sends its output, and ignores the answer sent back by the challenger.

Consider the event E_H to be the event when \mathcal{A} wins the game conditioned on the random oracle is fixed as H . Using Fact 4.5, we can find a subset H_{good} of all possible functions such that whenever the function H is in H_{good} , $\Pr_{r, \mathcal{A}}[\mathcal{A} \iff C(H) = 1] \geq B + \varepsilon/2$. Since H_{good} takes up at least $\varepsilon/2$ fraction of all possible functions, we have the theorem by observing that \mathcal{A}' given above have the success probability lower bounded by $\varepsilon/2 \cdot (B + \varepsilon/2)^g$. \square

Proposition 4.7. *Given a security game G and a (g, S, T) adversary with (classical or quantum) advice \mathcal{A} for the multi-instance game $G^{\otimes g}$ with winning probability δ , there exists an (g, T) adversary (without advice) \mathcal{A}' for the same game with winning probability at least $2^{-S} \cdot \delta$.*

Furthermore, the reduction is classical.

Proof. \mathcal{A}' works simply by invoking the second-stage adversary with a maximally mixed state I_{2^S} as its advice. Note that a maximally mixed state is a mixed state in any basis, in particular, assume that the correct advice is $|\alpha\rangle$, we can write I_{2^S} as a maximally mixed state in basis states v_1, v_2, \dots, v_{2^S} where $v_1 = |\alpha\rangle$. Using this observation, we conclude that \mathcal{A}' has winning probability at least $2^{-S} \cdot \delta$.

The reduction is classical, as picking a maximally mixed state I_{2^S} is equivalent to choosing a S -bit bit-string uniformly at random. \square

Combining the two propositions above yields the following corollary.

Corollary 4.8. *Let G be a security game. Given an (S, T) adversary with classical advice \mathcal{A} for the security game G with success probability δ , then for all real $B \geq 0$, let $\delta = B + \varepsilon$, there exists an (g, T) adversary (without advice) \mathcal{A}' for the multi-instance game $G^{\otimes g}$ with winning probability $\delta' \geq 2^{-S} \cdot \varepsilon/2 \cdot (B + \varepsilon/2)^g$ for any $g > 0$.*

Furthermore, the reduction is classical.

Theorem 4.1. *Given a security game G . Let δ_0 be the winning probability of an adversary that outputs a random answer without advice or making any query. Assume that multi-instance game $G^{\otimes g}$ is $\delta(g, T)$ -secure in QROM (or ROM), G 's security against (S, T) adversaries in AI-QROM (or AI-ROM, respectively) is*

$$\delta' \leq 4 \cdot \delta(S + \log(1/\delta_0) + 1, T).$$

Proof. Let \mathcal{A} be the (S, T) adversary with classical advice \mathcal{A} for G with the best winning probability δ' , then using Corollary 4.8 with $B = 0$, there exists an (g, T) adversary (without advice) \mathcal{A}' for the multi-instance game $G^{\otimes g}$ with winning probability at least $2^{-S} \cdot \delta'/2 \cdot (\delta'/2)^g$ for any $g > 0$. By multi-instance security, we know that

$$\delta(g, T)^g \geq 2^{-S} \cdot \delta'/2 \cdot (\delta'/2)^g \geq 2^{-S} \cdot \delta_0/2 \cdot (\delta'/2)^g.$$

Take $g = S + \log(1/\delta_0) + 1$, we have

$$\delta(g, T)^g \geq 2^{-g} \cdot (\delta'/2)^g,$$

and thus we have the theorem. □

4.2 Reduction for Decision Games with Classical Advice

Fact 4.9. *For any real $C \geq 0$, $0 \leq x \leq 1$, we have that $\frac{C}{C+x} \leq 1 - \frac{x}{C+1}$.*

Proof. By monotonicity and convexity of $\frac{C}{C+x}$ on x . □

Theorem 4.2. *Let G be a decision game. Let ε_0 be a lower bound on the advantage for an adversary with T_0 queries and S_0 bits of advice. Assume that multi-instance game $G^{\otimes g}$ is $(1/2 + \varepsilon(g, T))$ -secure (or has best advantage $\varepsilon(g, T)$) in QROM (or ROM). For any (S, T) adversaries in AI-QROM (or AI-ROM, respectively), the best advantage against G is ε' , which satisfies,*

$$\varepsilon' \leq 4 \cdot \varepsilon \left(\frac{10 \ln 2}{\varepsilon'} \cdot (S + S_0 + \log(1/\varepsilon_0) + 2), T + T_0 \right).$$

Proof. Let \mathcal{A} be the (S, T) adversary with classical advice \mathcal{A} for G with the best advantage ε' . By security of multi-instance game and Corollary 4.8 with $B = 1/2$, we know that there exists an (g, T) adversary (without advice) \mathcal{A}' for the multi-instance game $G^{\otimes g}$ with winning probability at least $2^{-S} \cdot \varepsilon'/2 \cdot (1/2 + \varepsilon'/2)^g$ for any $g > 0$. Moreover, if $S \geq S_0$ and $T \geq T_0$, we have $\varepsilon' \geq \varepsilon_0$. Therefore,

$$(1/2 + \varepsilon(S, T))^g \geq 2^{-S} \cdot (\varepsilon'/2) \cdot (1/2 + \varepsilon'/2)^g \geq 2^{-S} \cdot (\varepsilon_0/2) \cdot (1/2 + \varepsilon'/2)^g.$$

Therefore, in general,

$$(1/2 + \varepsilon(g, T + T_0))^g \geq 2^{-S-S_0} \cdot (\varepsilon_0/4) \cdot (1/2 + \varepsilon'/2)^g.$$

Take $g = \frac{10 \ln 2}{\varepsilon'} \cdot (S + S_0 + \log(1/\varepsilon_0) + 2)$, using Fact 4.9, and the fact that, we have that $(1 - 1/x)^x \leq 1/e$ for any real $x \geq 1$, we have

$$\begin{aligned}
\left(\frac{1/2 + \varepsilon'/4}{1/2 + \varepsilon'/2}\right)^g &\leq \left(1 - \frac{\varepsilon'/4}{1/2 + \varepsilon'/4 + 1}\right)^g \\
&= \left(1 - \frac{1}{4(1/2 + 1)/\varepsilon' + 1}\right)^g \\
&\leq \left(1 - \frac{1}{5(1/2 + 1)/\varepsilon'}\right)^g \\
&\leq (1 - \varepsilon'/10)^g \\
&\leq 2^{-(S+S_0+\log(1/\varepsilon_0)+2)} \\
&= 2^{-S-S_0} \cdot \varepsilon_0/4.
\end{aligned}$$

Combining the two formulae above, we obtain that

$$(1/2 + \varepsilon(g, T + T_0))^g \geq (1/2 + \varepsilon'/4)^g,$$

where $g = \frac{10 \ln 2}{\varepsilon'} \cdot (S + S_0 + \log(1/\varepsilon_0) + 2)$. Thus we have the theorem. \square

4.3 Reduction for Publicly-Verifiable Games

Lemma 4.10 (Gentle measurement lemma [Aar05, Lemma 2.2]). *Suppose a 2-outcome measurement of a mixed state ρ yields outcome 0 with probability $1 - \varepsilon$. Then after the measurement, we can recover a state $\tilde{\rho}$ such that $\|\tilde{\rho} - \rho\|_{\text{tr}} \leq \sqrt{\varepsilon}$. This is true even if the measurement is a POVM (that is, involves arbitrarily many ancilla qubits).*

Lemma 4.11 (Quantum union bound [AR19, Corollary 11]). *Let ρ be a mixed state and let S_1, \dots, S_m be any quantum operations. Suppose that for all i , we have*

$$\|S_i(\rho) - \rho\|_{\text{tr}} \leq \varepsilon_i.$$

Then

$$\|S_m(S_{m-1}(\dots S_1(\rho)\dots)) - \rho\|_{\text{tr}} \leq \varepsilon_1 + \dots + \varepsilon_m.$$

Lemma 4.12 ([BV97, Lemma 3.6]). *Let $\mathcal{D}(\psi)$ denote the probability distribution that results from a measurement of $|\psi\rangle$ in the computational basis. If $\|\phi - \psi\|_{\text{tr}} \leq \varepsilon$, then $\|\mathcal{D}(\phi) - \mathcal{D}(\psi)\|_1 \leq 4\varepsilon$.*

Proposition 4.13. *There exists a universal constant $c > 0$. Given a publicly-verifiable security game G , let T_{Ver} be the upper bound on the number of \tilde{H} queries for computing $\widetilde{\text{Ver}}^{\tilde{H}}(\text{ch}, \cdot)$. Given an (S, T) adversary with advice \mathcal{A} for G with winning probability δ , there exists an (g, S', T') adversary with advice \mathcal{A}' for the multi-instance game $G^{\otimes g}$ with winning probability at least $(\delta/4)^{g+1}$ for any $g > 0$, where*

$$\begin{cases} S' = kS, \\ T' = 2k(T + T_{\text{Ver}}), \\ k = c \cdot \log(g + 1)/\delta. \end{cases}$$

Proof. Consider the following adversary \mathcal{A}' :

1. At the pre-processing phase, compute $|\alpha_j\rangle \leftarrow \mathcal{A}_1(H)$ for all $j \in [k]$.
2. At the online phase, on receiving challenge ch_i and given oracle access to $\tilde{H}(\cdot) := \text{Query}^H(\text{ch}_i, \cdot)$, do:
 - (a) Run $|\text{ans}_j\rangle \leftarrow \mathcal{A}_2^{\tilde{H}}(|\alpha_j\rangle, \text{ch}_i)$ for all $j \in [k]$;
 - (b) Compute in superposition $|v_j\rangle \leftarrow \widetilde{\text{Ver}}^{\tilde{H}}(\text{ch}_i, |\text{ans}_j\rangle)$ for all $j \in [k]$.
 - (c) In superposition, find the first j such that $v_j = 1$, and store ans_j into an empty register ans .
 - (d) Send $|\text{ans}\rangle$ to challenger, and receives $|\text{ans}'\rangle$ back;
 - (e) Un-compute (c), (b), (a).

First, it is easy to see that for \mathcal{A}' , its advice is of length $S' = kS$ and it requires to make at most $T' = 2k(T + T_{\text{Ver}})$ oracle queries ($T + T_{\text{Ver}}$ queries to compute each v_j and $T + T_{\text{Ver}}$ queries to un-compute the previous round).

Using Fact 4.5, we can find a subset H_{good} that is at least $\delta/2$ fraction of all functions, such that any function H in H_{good} , $\Pr_{r, \mathcal{A}}[\mathcal{A} \iff C(H) = 1] \geq \delta/2$.

Fix any function in H_{good} . Using Fact 4.5 again, we can find a subset R_{good} that is of size at least $\delta|R|/4$ (which consequently is of at least $\delta/4$ fraction), such that for any randomness r in R_{good} , $\Pr_{\mathcal{A}}[\mathcal{A} \iff C(H) = 1] \geq \delta/4$.

Assume ch_1 is generated using the randomness in R_{good} , then we know that the challenger will reject the first answer with probability at most $(1 - \delta/4)^k \leq \frac{1}{100g^4}$ if c is a large enough constant. Let ρ be $|\alpha_1\rangle, \dots, |\alpha_k\rangle$. By Lemma 4.10, after one round, the advice state we recovered from un-computing ρ_1 satisfies $\|\rho_1 - \rho\|_{\text{tr}} \leq 1/\sqrt{100g^4} = 1/10g^2$.

If all challenges are generated using randomness in R_{good} , by Lemma 4.11, the overall advice state starting in each round ρ_j satisfies $\|\rho_j - \rho\|_{\text{tr}} \leq g \cdot \frac{1}{10g^2}$. Therefore, in each round, since every operation besides the measurement is unitary, by Lemma 4.12, the challenger accepts with probability at least $1 - \frac{1}{2g}$. By union bound, the challenger accepts with probability at least $1/2$, finishing the proof that the winning probability is at least $\delta/2 \cdot (\delta/4)^g \cdot 1/2$. \square

Combining this with Proposition 4.7 (to guess a quantum advice by a maximally mixed state), we obtain the following.

Corollary 4.14. *There exists a universal constant $c > 0$. Given a publicly-verifiable security game G , let T_{Ver} be the upper bound on the number of \tilde{H} queries for computing $\widetilde{\text{Ver}}^{\tilde{H}}(\text{ch}, \cdot)$. Given an (S, T) adversary with advice \mathcal{A} for G with winning probability δ , there exists an (g, T') adversary \mathcal{A}' for the multi-instance game $G^{\otimes g}$ with winning probability at least $2^{-kS} \cdot (\delta/4)^{g+1}$ for any $g > 0$, where $T' = 2k(T + T_{\text{Ver}})$ and $k = c \cdot \log(g + 1)/\delta$.*

Fact 4.15. *Given any real $C \geq 0, D \geq 2$, if $g_0 = C + D + 14$ and $g = 2g_0 \log g_0$, $g \geq C \log(g + 1) + D$.*

Proof. Since $C \geq 0, D \geq 2$, we have $g_0 \geq 16$ and $\log g_0 \geq 4$.

$$\begin{aligned}
g - (C \log(g + 1) + D) &\geq g - (C \log(2g) + D) \\
&\geq 2g_0 \log g_0 - (C \log(4g_0 \log g_0) + D) \\
&\geq 2(C + D + 14) \log g_0 - (C \log(4g_0 \log g_0) + D) \\
&\geq C(2 \log g_0 - (2 + \log g_0 + \log \log g_0)) + D(2 \log g_0 - 1).
\end{aligned}$$

We complete the proof by noting that coefficients of C and D are both non-negative when $\log g_0 \geq 4$. \square

Theorem 4.3. *Given a publicly-verifiable security game G , let T_{Ver} be the upper bound on the number of \tilde{H} queries for computing $\text{Ver}^{\tilde{H}}(\text{ch}, \cdot)$. Assume that $\delta_0 = 1/\text{poly}(N, M)$ is the winning probability by random guess, and assume that multi-instance game $G^{\otimes g}$ is $\delta(g, T)$ -secure in the QROM. For any (S, T) adversaries in the QAI-QROM, the best advantage against G is δ' , which satisfies*

$$\delta' \leq 8 \cdot \delta \left(\tilde{O}(S)/\delta', \tilde{O}(T + T_{\text{Ver}})/\delta' \right),$$

where \tilde{O} absorbs $\text{poly}(\log N, \log M, \log S)$ factors.

Proof. Let \mathcal{A} be the (S, T) adversary with advice for G with the best advantage δ' , then there exists an (g, T') adversary (without advice) \mathcal{A}' for the multi-instance game $G^{\otimes g}$ with winning probability at least $2^{-kS} \cdot (\delta'/4)^{g+1}$ for any $g > 0$, where $T' = 2k(T + T_{\text{Ver}})$ and $k = c \cdot \log(g+1)/\delta'$. Then we have,

$$\begin{aligned} \delta(g, T')^g &\geq 2^{-kS} \cdot (\delta'/4)^{g+1} \\ &\geq 2^{-kS} \cdot (\delta_0/4) \cdot (\delta'/4)^g \end{aligned}$$

Take $g_0 = \frac{c}{\delta'} \cdot S + \log(1/\delta_0) + 16$ and $g = 2g_0 \cdot \log g_0$. By Fact 4.15, let $C = \frac{c}{\delta'} S$ and $D = \log(1/\delta_0) + 2$, we have $g \geq C \log(g+1) + D = c \log(g+1)S/\delta' + \log(1/\delta_0) + 2$.

Therefore, we have

$$\begin{aligned} \delta(g, T')^g &\geq 2^{-kS} \cdot (\delta_0/4) \cdot (\delta'/4)^g \\ &= 2^{-c \log(g+1)S/\delta'} \cdot 2^{-\log(1/\delta_0)-2} \cdot (\delta'/4)^g \\ &\geq (\delta'/8)^g. \end{aligned}$$

Therefore, we conclude that

$$\delta' \leq 8 \cdot \delta(g, T').$$

where $g = \tilde{O}(S)/\delta'$ and $T' = \tilde{O}(T + T_{\text{Ver}})/\delta'$. \square

4.4 Reduction for Decision Games with Quantum Advice

Theorem 4.16 (Online shadow tomography [AR19, Theorem 9]). *There exists an explicit procedure that performs shadow tomography, namely on input n copies of an arbitrary unknown d -dimensional mixed state ρ , and any two-outcome measurements E_1, \dots, E_m , it estimates $\Pr[E_i(\rho) = 1]$ within an additive error of $\pm \varepsilon$ for every $i \in [m]$ with overall success probability $1 - \beta$, if*

$$n = \Omega \left(\frac{\log^2 m \cdot \log^2 d \cdot \log \frac{1}{\beta}}{\varepsilon^8} \right).$$

Furthermore, this procedure satisfies the following properties:

1. It is online, meaning that it can be given by a pair of quantum algorithms: an initialization algorithm $|\text{st}\rangle \leftarrow \text{QPMW.Init}(n, \rho^{\otimes n}, m, \varepsilon, \beta)$, and an estimation algorithm $(|\text{st}\rangle, p_i) \leftarrow \text{QPMW.Est}(|\text{st}\rangle, i, E_i)$. $|\text{st}\rangle$ will be updated by the estimation algorithm QPMW.Est .

2. If the measurement E_i can be written as (1). prepare a state $|\text{ans}\rangle$ which is initialized as 0, (2). apply a unitary U_i over the joint state ρ and $|\text{ans}\rangle$, (3). measure $|\text{ans}\rangle$ in the computational basis. Then U_i (or its inverse) will be run at most $4n$ times⁶.

First, we will present the following useful fact which says that additive error estimation is robust with respect to arbitrary distribution.

Fact 4.17. *Given $2N$ events E_1, \dots, E_N , and E'_1, \dots, E'_N , and some real ε , such that for any $i \in [N]$, $|\Pr[E_i] - \Pr[E'_i]| \leq \varepsilon$. Then for arbitrary distribution μ over $[N]$ that is independent of the $2N$ events, $|\Pr[E_\mu] - \Pr[E'_\mu]| \leq \varepsilon$.*

Proof.

$$\begin{aligned}
|\Pr[E_\mu] - \Pr[E'_\mu]| &= \left| \sum_{i \in [N]} (\Pr[E_\mu \wedge \mu = i] - \Pr[E'_\mu \wedge \mu = i]) \right| \\
&\leq \sum_{i \in [N]} |\Pr[E_\mu \wedge \mu = i] - \Pr[E'_\mu \wedge \mu = i]| \\
&= \sum_{i \in [N]} \Pr[\mu = i] \cdot |\Pr[E_i | \mu = i] - \Pr[E'_i | \mu = i]| \\
&= \sum_{i \in [N]} \Pr[\mu = i] \cdot |\Pr[E_i] - \Pr[E'_i]| \\
&\leq \sum_{i \in [N]} \Pr[\mu = i] \cdot \varepsilon \\
&= \varepsilon.
\end{aligned}$$

□

Proposition 4.18. *There exists a universal constant $c > 0$. Given a decision game G . Given an (S, T) adversary with advice \mathcal{A} for G with winning probability $1/2 + \varepsilon$, there exists an (g, S', T') adversary with advice \mathcal{A}' for the multi-instance game $G^{\otimes g}$ with winning probability at least $\varepsilon/4 \cdot (1/2 + \varepsilon/4)^g$ for any $g > 0$, where*

$$\begin{cases} S' = kS, \\ T' = 4kT, \\ k = \frac{c}{\varepsilon^8} \cdot S^2 \log^2 g. \end{cases}$$

Proof. Consider the following adversary \mathcal{A}' :

1. At the pre-processing phase, compute $|\alpha_j\rangle \leftarrow \mathcal{A}_1(H)$ for all $j \in [k]$. We denote $\rho^\otimes = |\alpha_1\rangle, \dots, |\alpha_k\rangle$ as the advice.
2. At the online phase, compute $\text{st} \leftarrow \text{QPMW.Init}(k, \rho^\otimes, g, \varepsilon/4, 1/2)$;
3. On receiving challenge ch_i and given oracle access to $\tilde{H}(\cdot) := \text{Query}^H(\text{ch}_i, \cdot)$, do:

⁶This can be verified by looking at [AR19, Figure 1].

- (a) Compute $(|\text{st}\rangle, p_i) \leftarrow \text{QPMW.Est}(|\text{st}\rangle, i, E_i)$, where the measurement E_i is written as the unitary that computes $\mathcal{A}_2^{\tilde{H}}(\rho, \text{ch}_i)$;
- (b) Flip a biased coin that yields 1 with probability p_i and 0 with probability $1 - p_i$, and send the result to the challenger;
- (c) Ignore challenger's response.

Note that each variable in Theorem 4.16 is set to be: $m = g$, $d = 2^S$, $\beta = 1/2$, additive error is set to be $\varepsilon/4$, and $\beta = 1/2$. Therefore the number of copies n in this theorem, or in other words, k in our proposition is $\frac{c}{\varepsilon^8} S^2 \log^2 g$. The advice is of length $S' = kS$ and the number of queries is at most $T' = 4kT$ because for each E_i , the unitary (or its inverse) that computes E_i will be applied at most $4k$ times (by Theorem 4.16).

Using Fact 4.5, we can find a subset H_{good} of all possible functions such that whenever the function H is in H_{good} , $\Pr_{r, \mathcal{A}}[\mathcal{A} \iff C(H) = 1] \geq 1/2 + \varepsilon/2$. Since H_{good} takes up $\varepsilon/2$ fraction of all possible functions, and this shadow tomography algorithm succeeds with probability $\geq 1 - \beta = 1/2$. We have the theorem by observing that \mathcal{A}' given above has the success probability lower bounded by $\varepsilon/2 \cdot 1/2 \cdot (1/2 + \varepsilon/2 - \varepsilon/4)^g$ (by Fact 4.17 and the fact that each challenge is generated independently). \square

Combining this theorem with Proposition 4.7 (to replace a quantum advice with a maximally mixed state), we obtain the following.

Corollary 4.19. *There exists a universal constant $c > 0$. Given a decision security game G . Given an (S, T) adversary with advice \mathcal{A} for G with winning probability $1/2 + \varepsilon$, there exists an (g, T') adversary \mathcal{A}' for the multi-instance game $G^{\otimes g}$ with winning probability at least $2^{-kS} \cdot \varepsilon/4 \cdot (1/2 + \varepsilon/4)^g$ for any $g > 0$, where $T' = 4kT$ and $k = \frac{c}{\varepsilon^8} \cdot S^2 \log^2 g$.*

Fact 4.20. *Given any real $C \geq 0, D \geq 2$, if $g_0 = C + D$ and $g = 16g_0 \log g_0$, $g \geq C \log^2 g + D$.*

Proof. First, since $C \geq 0, D \geq 2$, we have $g_0 \geq 2$.

$$\log g = \log(2g_0 \log^2 g_0) = 1 + \log g_0 + 2 \log \log g_0 \leq 3.9 \cdot \log g_0$$

The above inequality is true because for $g_0 \geq 2$, $2 \log \log g_0 \leq 1.9 \cdot \log g_0$ and $1 \leq \log g_0$. Then we have,

$$\begin{aligned} g - (C \log^2 g + D) &\geq 16 \cdot g_0 \log^2 g_0 - (C \cdot (3.9)^2 \log^2 g_0 + D) \\ &\geq C(16 - 3.9^2) \log^2 g_0 + D(16 \log^2 g_0 - 1) \end{aligned}$$

We complete the proof by noting that coefficients of C and D are both non-negative when $\log g_0 \geq 1$. \square

Theorem 4.4. *Given a decision security game G . Assume that $\varepsilon_0 = 1/\text{poly}(N, M)$ is a non-negligible lower bound on the advantage for an adversary with T_0 queries and S_0 qubits of advice, and assume that multi-instance game $G^{\otimes g}$ is $\varepsilon(g, T)$ -secure in the QROM. For any (S, T) adversaries in QAI-QROM, the best advantage against G is ε' , which satisfies*

$$\varepsilon' \leq 8 \cdot \varepsilon \left(\tilde{O}(S + S_0)^3 / \varepsilon'^9, \tilde{O}((T + T_0)(S + S_0)^2) / \varepsilon'^8 \right),$$

where \tilde{O} absorbs $\text{poly}(\log N, \log M, \log S)$ factors.

Proof. Let \mathcal{A} be the (S, T) adversary with advice for G with the best advantage ε' , then there exists an (g, T) adversary (without advice) \mathcal{A}' for the multi-instance game $G^{\otimes g}$ with winning probability at least $2^{-kS} \cdot \varepsilon'/4 \cdot (1/2 + \varepsilon'/4)^g$ for any $g > 0$, where $T' = 4kT$ and $k = \frac{c}{\varepsilon_0} \cdot S^2 \log^2 g$.

By the security of multi-instance game, if $T' \geq T_0$ and $S \geq S_0$, we know that

$$(1/2 + \varepsilon(g, T'))^g \geq 2^{-kS} \cdot \varepsilon'/4 \cdot (1/2 + \varepsilon'/4)^g \geq 2^{-kS} \cdot \varepsilon_0/4 \cdot (1/2 + \varepsilon'/4)^g.$$

Therefore, in general,

$$(1/2 + \varepsilon(g, T'))^g \geq 2^{-k(S+S_0)} \cdot \varepsilon_0/4 \cdot (1/2 + \varepsilon'/4)^g.$$

Let $C = \frac{15 \ln 2}{\varepsilon'} \cdot c/\varepsilon'^8 \cdot S^2(S + S_0)$ and $D = \frac{15 \ln 2}{\varepsilon'}(\log(1/\varepsilon_0) + 2) \geq 2$. Let

$$\begin{aligned} g_0 &= \frac{15 \ln 2}{\varepsilon'} \cdot (c/\varepsilon'^8 \cdot S^2(S + S_0) + \log(1/\varepsilon_0) + 2) = C + D, \\ g &= 16g_0 \log^2 g_0 \geq C \log^2 g + D = \frac{15 \ln 2}{\varepsilon'}(k(S + S_0) + \log(1/\varepsilon_0) + 2), \end{aligned}$$

where the inequality for g comes from Fact 4.20.

$$\begin{aligned} \left(\frac{1/2 + \varepsilon'/8}{1/2 + \varepsilon'/4}\right)^g &\leq \left(1 - \frac{\varepsilon'/8}{1/2 + \varepsilon'/8 + 1}\right)^g \\ &= \left(1 - \frac{1}{8(1/2 + 1)/\varepsilon' + 1}\right)^g \\ &\leq \left(1 - \frac{\varepsilon'}{15}\right)^g \\ &\leq 2^{-k(S+S_0)} \cdot \varepsilon_0/4, \end{aligned}$$

where the first inequality comes from Fact 4.9 and the last inequality comes from $g \geq \frac{15 \ln 2}{\varepsilon'}(k(S + S_0) + \log(1/\varepsilon_0) + 2)$. Overall, we have,

$$(1/2 + \varepsilon(g, T'))^g \geq 2^{-k(S+S_0)} \cdot \varepsilon_0/4 \cdot (1/2 + \varepsilon'/4)^g \geq (1/2 + \varepsilon'/8)^g.$$

In conclusion, $\varepsilon'(S, T) \leq 8 \cdot \varepsilon(g, T')$ where $g = \tilde{O}(S + S_0)^3/\varepsilon'^9$ and $T' = 4kT = \tilde{O}((T + T_0)(S + S_0)^2)/\varepsilon'^8$. \square

5 One-Way Function

In this section, we show the security of OWF in the AI-QROM and QAI-QROM. Let us start with reformalizing the security of OWF in QROM in our language of security games.

Definition 5.1 (OWF Security Game). *Let H be a random oracle $[N] \rightarrow [M]$. The security game $G_{\text{OWF}} = (C_{\text{OWF}})$ is specified by three procedures (Samp, Query, Ver), where:*

1. **Samp** ^{H} (x) takes a random element $x \in [N]$, and outputs $y = H(x)$.
2. **Query** ^{H} (x, x') ignores the challenge point x and simply outputs $H(x')$.

3. $\text{Ver}^H(x, x')$ outputs 1 if and only if $H(x) = H(x')$.

To fill in the rest of the details from Definition 3.3, the security game defined here is that given a random oracle, the challenger samples a uniformly random element $x \in [N]$ and computes $y = H(x)$ as the challenge. The adversary is allowed to make oracle queries to H and it wins if and only if it finds any pre-image of y . This is exactly the security of OWF in QROM. This is also exactly the problem of inverting a random image.

In the rest of the section, we are going to prove the following lemma.

Lemma 5.2. $G_{\text{OWF}}^{\otimes g}$ is $\delta(g, T) = O\left(\frac{gT+T^2}{\min\{N, M\}}\right)$ -secure in the QROM.

Combining this lemma with Theorem 4.1 (or Theorem 4.3), we can show the security of G_{OWF} in the AI-QROM (or QAI-QROM, respectively).

Theorem 5.3. G_{OWF} is $\delta(S, T) = O\left(\frac{(S+T+\log N)T}{\min\{N, M\}}\right) = \tilde{O}\left(\frac{ST+T^2}{\min\{N, M\}}\right)$ -secure in the AI-QROM.

Proof. In Theorem 4.1, δ_0 is at least $1/N$ because there is at least one pre-image of y , so random guess has winning probability at least $1/N$. Plugging in $\delta_0 \geq 1/N$, and $G_{\text{OWF}}^{\otimes g}$ is $O\left(\frac{gT+T^2}{\min\{N, M\}}\right)$ -secure from Lemma 5.2, we have G_{OWF} is $\delta(S, T)$ -secure where,

$$\begin{aligned}\delta(S, T) &\leq 4 \cdot O\left(\frac{(S + \log(1/\delta_0) + 1)T + T^2}{\min\{N, M\}}\right) \\ &= O\left(\frac{(S + \log N) \cdot T + T^2}{\min\{N, M\}}\right).\end{aligned}$$

□

Theorem 5.4. G_{OWF} is $\delta(S, T) = \tilde{O}\left(\left(\frac{ST+T^2}{\min\{N, M\}}\right)^{1/3}\right)$ -secure in the QAI-QROM, where \tilde{O} absorbs $\text{poly}(\log N, \log M, \log S)$ factors.

Proof. It is a publicly-verifiable game. For G_{OWF} , we can verify that $T_{\text{Ver}} = 2$ and δ_0 is again at least $1/N$. Since $G_{\text{OWF}}^{\otimes g}$ is $\delta'(S, T) = O\left(\frac{gT+T^2}{\min\{N, M\}}\right)$ -secure from Lemma 5.2, we have G_{OWF} is $\delta(S, T)$ -secure where,

$$\begin{aligned}\delta(S, T) &\leq 8 \cdot \delta' \left(\tilde{O}(S)/\delta, \tilde{O}(T + T_{\text{Ver}})/\delta \right) \\ &\leq 8 \cdot \frac{\tilde{O}(ST + T^2)}{\delta^2 \cdot \min\{N, M\}}.\end{aligned}$$

Therefore, $\delta(S, T) = \tilde{O}\left(\left(\frac{ST+T^2}{\min\{N, M\}}\right)^{1/3}\right)$. □

In the following two subsections, we will first show multi-instance security for a slightly different game with $\delta(g, T) = O\left(\frac{gT+T^2}{M}\right)$. Then in the second subsection, we show that these two games are almost indistinguishable even for a quantum adversary, and conclude that the multi-instance security for $G_{\text{OWF}}^{\otimes g}$ is $\delta(g, T) = O\left(\frac{gT+T^2}{\min\{N, M\}}\right)$.

5.1 Multi-Instance Security for Inverting Random Image

Consider the following OWF game variant, where the main difference is that instead of uniformly sampling $x \in [N]$ and telling the adversary $H(x)$, we are directly sampling a uniformly random $y \in [M]$ and telling the adversary y .

Definition 5.5 (OWF _{y} Security Game). *Let H be a random oracle $[N] \rightarrow [M]$. The security game $G_{\text{OWF}_y} = (C_{\text{OWF}_y})$ is specified by the challenger $C_{\text{OWF}_y} = (\text{Samp}, \text{Query}, \text{Ver})$, where:*

1. $\text{Samp}^H(y) = y$ takes a randomness $y \in [M]$, and outputs y as the challenge.
2. $\text{Query}^H(y, x') = H(x')$.
3. $\text{Ver}^H(y, x')$ outputs 1 if and only if $H(x') = y$.

In other words, given a random oracle, the challenger samples a random output y as the challenge. The adversary is allowed to make oracle queries to H and it wins if and only if it finds a pre-image of y .

In OWF _{y} problem, a challenge y may not have a valid answer. For example, when $N \ll M$, a random element y in $[M]$ is not the image of any inputs with high probability.

Lemma 5.6. $G_{\text{OWF}_y}^{\otimes g}$ is $\delta(g, T) = O\left(\frac{gT+T^2}{M}\right)$ -secure in the QROM.

Proof. Let \mathcal{A} be any (g, T) adversary. By definition of security, to prove the lemma, we would need to establish an upper bound on $\Pr[\mathcal{A} \iff C_{\text{OWF}_y}^{\otimes g}(H)]$. Without loss of generality, we can write \mathcal{A} as a sequence of unitary operators $(U_{i,j})_{i \in [g], j \in [T+1]}$ operating on registers $|x\rangle, |y\rangle, |\text{aux}\rangle$, initialized to empty, where

- $x \in [N], u \in [M]$ and aux is the auxiliary register whose size is unbounded.
- $|u\rangle$ serves as the challenge as input to $U_{i,1}$.
- For $1 \leq t \leq T$, $U_{i,t}$ prepares a query to $\text{Query}^H(y, \cdot) = H(\cdot)$ at $|x\rangle$ and receives the output at $|u\rangle$.
- $U_{i,T+1}$ prepares the final output for i -th game in register $|x\rangle$, clears the register $|u\rangle$, and receives the measured answer also back in the register $|x\rangle$ and the next challenge in the register $|u\rangle$.

Consider the simulator that instead simulates the interaction of \mathcal{A} and the challenger with the compressed oracle. Formally, the simulator initializes the state $|\psi'_{1,0}\rangle := |0\rangle_{\mathcal{A}} \otimes |\emptyset\rangle_H$. For each round $i \in [g]$:

1. Starting at $|\psi'_{i,0}\rangle$, sample the next challenge $y_i \leftarrow_{\S} [M]$, and let $|u\rangle \leftarrow y_i$. Let the resulting state *conditioned on* the challenge being y_i be $|\psi_{i,0}\rangle$.
2. We define $|\psi_{i,t}\rangle$ and $|\psi'_{i,t}\rangle$:

$$\begin{aligned} |\psi'_{i,t+1}\rangle &= U_{i,t+1} |\psi_{i,t}\rangle && \text{for all } t = 0, \dots, T; \\ |\psi_{i,t}\rangle &= U_H |\psi'_{i,t}\rangle && \text{for all } t = 1, \dots, T; \end{aligned}$$

where U_H is CStO operating on registers $|x\rangle, |u\rangle, |H\rangle$.

3. By our assumption on $U_{i,T+1}$, in state $|\psi'_{i,T+1}\rangle, |x\rangle$ holds the adversary's answer and $|u\rangle = |0\rangle$. Verify the answer by doing the following: apply U_H on $|x\rangle, |u\rangle$. Let us call the resulting state as $|\psi_{i,T+1}\rangle$, then

$$|\psi_{i,T+1}\rangle = \sum_{x,u,\mathbf{aux},D} \gamma_{x,u,\mathbf{aux},D} |x, u, \mathbf{aux}\rangle_{\mathcal{A}} |D\rangle_H$$

4. Compute in superposition $b_i = [u = y_i]$ and measure b_i .

5. Uncompute U_H , let the resulting state *conditioned on* measurement result being b_i be $|\phi'_{i+1,0}\rangle$.

By Corollary 2.5, the probability that this algorithm obtains $b_1 = \dots = b_g = 1$ is exactly the probability that $\Pr[\mathcal{A} \iff C_{\text{OWF}_y}^{\otimes g}(H)]$. We have the following proposition.

Proposition 5.7. *For all $i \in [g]$, let Y_1, Y_2, \dots, Y_{i-1} be the random variable of the first $(i-1)$ challenges and B_1, \dots, B_i be the random variable for the decision bits. For any (g, T) algorithm in the QROM, y_1, y_2, \dots, y_{i-1} and b_1, b_2, \dots, b_i , we have,*

$$\Pr[B_i = 1 \mid Y_1 = y_1, \dots, Y_{i-1} = y_{i-1}, B_1 = b_1, \dots, B_{i-1} = b_{i-1}] \leq O\left(\frac{iT + T^2}{M}\right),$$

given that $\Pr[Y_1 = y_1, \dots, Y_{i-1} = y_{i-1}, B_1 = b_1, \dots, B_{i-1} = b_{i-1}] > 0$.

Proof. We start by observing that these conditioning correspond to the pure state $|\psi'_{i,0}\rangle$ that we have described above. The state is well defined as these classical outcomes occur with non-zero probability as we have assumed. Since the total number of queries up to this point is at most $(i-1)(T+2)$, by Lemma 2.6, there exists some complex numbers α_* such that

$$|\psi'_{i,0}\rangle = \sum_{z,D:|D|\leq(i-1)(T+2)} \alpha_{z,D} |z\rangle |D\rangle.$$

For any $y_i \in [M]$, define P_{y_i} to be a projection of finding y_i in the database:

$$P_{y_i} = \sum_{D:\exists x \in [N], D(x)=y_i} |D\rangle \langle D|.$$

Let $p_{y_i}^{(t)}$ be the probability that after t queries to CStO, measuring the database register gives a database containing y_i , or formally,

$$p_{y_i}^{(t)} := |P_{y_i} |\psi_{i,t}\rangle|^2.$$

Since $|\psi'_{i,0}\rangle$ only has non-zero amplitude over D with size at most $(i-1)(T+2)$, for each z, D pair, $|\alpha_{z,D}|^2$ contributes to at most $(i-1)(T+2)$ different y_i 's. We observe that the same conclusion also holds for $|\psi_{i,0}\rangle$, as the only difference between these two states is that u is initialized to some element in $[M]$, and H register is unchanged. In other words,

$$\sum_{y_i=1}^M p_{y_i}^{(0)} \leq (i-1)(T+2). \quad (2)$$

We then have the following lemma that bounds $p_{y_i}^{(t)}$ for all $t > 0$, which is the same as [Zha19, Theorem 1] and similar to [LZ19a, Lemma 8 and Lemma 9].

Lemma 5.8. For all $t = 0, 1, \dots, T$ and all $y_i \in [M]$, $\sqrt{p_{y_i}^{(t+1)}} \leq \sqrt{p_{y_i}^{(t)}} + \sqrt{1/M}$.

Proof. Let us define $q_{y_i}^{(t)} = \left| P_{y_i} |\psi'_{i,t}\rangle \right|^2$. First, similar to [LZ19a, Lemma 8], we have $q_{y_i}^{(t+1)} = p_{y_i}^{(t)}$, since

$$\sqrt{q_{y_i}^{(t+1)}} = |P_{y_i} |\psi'_{i,t+1}\rangle| = |P_{y_i} U_{i,t+1} |\psi_{i,t}\rangle| \stackrel{(*)}{=} |P_{y_i} |\psi_{i,t}\rangle| = \sqrt{p_{y_i}^{(t)}}.$$

The equality (*) follows from the fact that $U_{i,t+1}$ only applies on \mathcal{A} 's register but P_{y_i} only applies on the database register and therefore they commute.

Second, we have $\sqrt{p_{y_i}^{(t+1)}} \leq \sqrt{q_{y_i}^{(t+1)}} + \sqrt{1/M}$ [LZ19a, Lemma 9]:

$$\begin{aligned} \sqrt{p_{y_i}^{(t+1)}} &= |P_{y_i} |\psi_{i,t+1}\rangle| = |P_{y_i} U_H |\psi'_{i,t+1}\rangle| \\ &= |P_{y_i} U_H (P_{y_i} + (I - P_{y_i})) |\psi'_{i,t+1}\rangle| \\ &\leq |P_{y_i} U_H P_{y_i} |\psi'_{i,t+1}\rangle| + |P_{y_i} U_H (I - P_{y_i}) |\psi'_{i,t+1}\rangle|. \end{aligned}$$

Intuitively, the first term corresponds to the case that before making the $(t+1)$ -th query, the database already has y_i . The second term corresponds to the case that before making the $(t+1)$ -th query, the database does not have y_i .

The first term $|P_{y_i} U_H P_{y_i} |\psi'_{i,t+1}\rangle| \leq |U_H P_{y_i} |\psi'_{i,t+1}\rangle| = |P_{y_i} |\psi'_{i,t+1}\rangle| = \sqrt{q_{y_i}^{(t+1)}}$.

For the second term, let $|\phi\rangle := (I - P_{y_i}) |\psi'_{i,t+1}\rangle$, by definition of P_{y_i} , there exists some complex numbers β_* such that

$$|\phi\rangle = \sum_{\substack{x, u, \mathbf{aux}, \\ D: \forall x', D(x') \neq y_i}} \beta_{x, u, \mathbf{aux}, D} |x, u, \mathbf{aux}\rangle |D\rangle.$$

We can equivalently write the second term as

$$\begin{aligned}
& |P_{y_i} U_H |\phi\rangle| \\
&= \left| P_{y_i} \text{CStO} \sum_{\substack{x,u,\text{aux}, \\ D:\forall x', D(x') \neq y_i}} \beta_{x,u,\text{aux},D} |x, u, \text{aux}\rangle |D\rangle \right| \\
&= \left| P_{y_i} \sum_{\substack{x,u,\text{aux}, \\ D:\forall x', D(x') \neq y_i}} \beta_{x,u,\text{aux},D} \text{StdDecomp}_x \circ \text{CStO}' \circ \text{StdDecomp}_x |x, u, \text{aux}\rangle |D\rangle \right| \\
&= \left| P_{y_i} \sum_{\substack{x,u,\text{aux}, \\ D:\forall x', D(x') \neq y_i}} \sum_y \frac{\beta_{x,u,\text{aux},D}}{\sqrt{M}} \text{StdDecomp}_x |x, u + y, \text{aux}\rangle |D \cup (x, y)\rangle \right| \\
&\stackrel{(*)}{\leq} \left| P_{y_i} \sum_y \sum_{\substack{x,u,\text{aux}, \\ D:\forall x', D(x') \neq y_i}} \frac{\beta_{x,u,\text{aux},D}}{\sqrt{M}} |x, u + y, \text{aux}\rangle |D \cup (x, y)\rangle \right| \\
&= \left| \sum_{\substack{x,u,\text{aux}, \\ D:\forall x', D(x') \neq y_i}} \frac{\beta_{x,u,\text{aux},D}}{\sqrt{M}} |x, u + y_i, \text{aux}\rangle |D \cup (x, y_i)\rangle \right| \\
&= \frac{1}{\sqrt{M}} ||\phi\rangle| \leq \frac{1}{\sqrt{M}}.
\end{aligned}$$

The inequality (*) comes from the fact that $D \cup (x, y)$ already contains x , therefore StdDecomp_x will either remove the entry for x or not change the database at all. \square

Using an induction on the lemma, we have

$$\sqrt{p_{y_i}^{(T+1)}} \leq \sqrt{p_{y_i}^{(0)}} + (T+1)/\sqrt{M}. \quad (3)$$

Define \tilde{p}_{y_i} as the probability that $B_i = 1$ conditioned on the challenge is y_i . In other words, \tilde{p}_{y_i} is the probability that the final state x, u when measured gives $u = H(x) = y_i$. By Lemma 2.7, we have $\sqrt{\tilde{p}_{y_i}} \leq \sqrt{p_{y_i}^{(T+1)}} + 1/\sqrt{M}$. Combining this with (2), (3), and Cauchy-Schwarz inequality,

$$\frac{1}{M} \sum_{y_i=1}^M \tilde{p}_{y_i} \leq \frac{2}{M} \sum_{y_i=1}^M \left(p_{y_i}^{(0)} + (T+2)^2/M \right) \leq \frac{O(iT + T^2)}{M}.$$

We complete the proof by observing that the left hand side is exactly the probability in the statement. \square

Combine Proposition 5.7 and Lemma 3.14, we have proven the lemma. \square

5.2 Multi-Instance Indistinguishability of Sampling Image

The analysis for multi-instance security of $G_{\text{OWF}_y}^{\otimes g}$ is not enough for $G_{\text{OWF}}^{\otimes g}$. This is because in the standard one-way function game, y is sampled in the way that it is correlated with the random

oracle. Therefore, we can not use Lemma 2.7, since the challenge image y is almost always in the database. In this subsection, we will instead show that the advantage of indistinguishing whether y_i is sampled uniformly at random or from $H(x_i)$ for a uniformly random x_i is at most $O((iT + T^2)/N)$.

First, we need the following lemma, which says for any quantum algorithm \mathcal{A} making at most q queries to a random oracle before getting a challenge, and at most q' queries after, it can not distinguish if it gets a random image $H(x)$ or a random element y in the range. Furthermore, the statement is true even after conditioning on arbitrary intermediate measurements.

Lemma 5.9. *Let \mathcal{D} be a quantum algorithm that makes at most $q + q'$ queries to a random oracle. Let $\mathbf{Y}_{[p]} = Y_1, Y_2, \dots, Y_p$ be the random variable of the outcomes of p arbitrary intermediate measurements, all of which occur before the q -th query. After \mathcal{D} making q queries, it is given a challenge y , which we crudely denote as $\mathcal{D}^H(y)$. It then makes another q' queries and simply outputs a bit. For any list of outcomes $\mathbf{y}_{[p]} = y_1, \dots, y_p$,*

$$\left| \sqrt{\Pr_{H, y \leftarrow \mathfrak{s}[M]} [\mathcal{D}^H(y) = 1 \mid \mathbf{Y}_{[p]} = \mathbf{y}_{[p]}]} - \sqrt{\Pr_{H, x \leftarrow \mathfrak{s}[N]} [\mathcal{D}^H(H(x)) = 1 \mid \mathbf{Y}_{[p]} = \mathbf{y}_{[p]}]} \right| \leq 2(\sqrt{q} + q')/\sqrt{N}.$$

Proof. We will first present two slightly different oracles or experiments, and argue that each of them perfectly simulates $\mathcal{D}^H(y)$ and $\mathcal{D}^H(H(x))$ respectively.

Case 1. The challenge is a uniformly random y . We implement the following changes to the oracle:

1. We replace the random oracle with a phase oracle. By Lemma 2.2, this change perfectly preserves the probability that \mathcal{D} outputs 1 as he only interacts with $|H\rangle$ using the phase oracle.
2. We consider the modified oracle $\tilde{H} : [N + 1] \mapsto [M]$, where $\tilde{H} := H \parallel h'$ for some random element $h' \in [M]$, and we pick $h' = \tilde{H}(N + 1)$ as our challenge. As \mathcal{D} only gets phase oracle access to H , and we will only use h' as classical randomness, this change again perfectly preserves \mathcal{D} 's functionality.
3. We additionally prepare a random $x \in [N]$ at a new register x but we do not use it. This change is clearly only conceptual.

Conditioned on $\mathbf{Y}_{[p]} = \mathbf{y}_{[p]}$, after \mathcal{D} has made q queries to the phase oracle and before generating the challenge, by Lemma 2.3, we know that the overall state of \mathcal{D} and the oracle internal state can be written as⁷

$$\sum_{z, D: |D| \leq q} \alpha_{z, D} |z\rangle_{\mathcal{D}} |0\rangle_y \otimes \frac{1}{\sqrt{N}} \sum_x |x\rangle \otimes \frac{1}{M^{(N+1)/2}} \sum_{H, h'} \omega_M^{\langle D, H \rangle} |H, h'\rangle,$$

⁷Here we write $|x\rangle$ for simplicity, but to see correctness, you could really think of $|x, x\rangle$ so we have two computational-basis “copies” of the same uniform superposition $1/\sqrt{N} \sum_x |x\rangle$. As we will never touch the second copy, the first copy behaves exactly like a classical random string to the rest of the algorithm.

where x and h' have not been used and thus is completely un-entangled with \mathcal{D} . After generating the challenge, the overall state becomes

$$|\phi_1\rangle := \frac{1}{N^{1/2}M^{(N+1)/2}} \sum_{z,D:|D|\leq q} \sum_x \sum_{H,h'} \alpha_{z,D} |z\rangle_{\mathcal{D}} |h'\rangle_y \otimes |x\rangle \otimes \omega_M^{\langle D,H\rangle} |H,h'\rangle.$$

Case 2. The challenge is the image for a uniformly random pre-image. We implement the following changes to the oracle:

1. We again replace the random oracle with a phase oracle using Lemma 2.2.
2. We again augment H to be $\tilde{H} : [N + 1] \mapsto [M]$ for a new random element $h' \in [M]$, and prepare a new random $x \in [N]$. For now, this is only a conceptual change (we only added two new independent registers).
3. Define a new classical oracle $H'_x : [N] \mapsto [M]$ to be

$$H'_x(x') := H_{\langle x||h'\rangle} = \begin{cases} \tilde{H}(N+1) = h', & x' = x; \\ \tilde{H}(x') = H(x'), & x' \neq x. \end{cases}$$

Instead of giving \mathcal{D} phase oracle to H , we are giving \mathcal{D} phase oracle to H'_x , where x is pulled from the new register $|x\rangle$ that we just add. For any fixed x , this change is perfectly consistent with the original oracle distribution; and as $|x\rangle$ is just a classical random string, the overall change is also perfectly consistent.

4. Instead of sampling a new pre-image for generating the challenge, we instead use x as the pre-image, that is, we “compute” $H'_x(x) = h'$ and send h' as the challenge for \mathcal{D} . It is easy to see that \mathcal{D} cannot obtain any information on x for the first q queries, thus this change still perfectly preserves \mathcal{D} 's functionality.

Therefore, conditioned on $\mathbf{Y}_{[p]} = \mathbf{y}_p$, before giving \mathcal{D} the challenge, the overall state of \mathcal{D} and the oracle internal memory is

$$\sum_{z,D:|D|\leq q} \alpha_{z,D} |z\rangle_{\mathcal{D}} |0\rangle_y \otimes \frac{1}{\sqrt{N}} \sum_x |x\rangle \otimes \frac{1}{M^{(N+1)/2}} \sum_{H,h'} \omega_M^{\langle D,H'_x\rangle} |H,h'\rangle.$$

When the challenge is given, the overall state becomes

$$|\phi_2\rangle := \frac{1}{N^{1/2}M^{(N+1)/2}} \sum_{z,D:|D|\leq q} \sum_x \sum_{H,h'} \alpha_{z,D} |z\rangle_{\mathcal{D}} |h'\rangle_y \otimes |x\rangle \otimes \omega_M^{\langle D,H'_x\rangle} |H,h'\rangle.$$

By now, we have shown two perfect simulator for $\mathcal{D}(y)$ and $\mathcal{D}(H(x))$ respectively, but instead of the two cases getting different challenges (y vs $H(x)$), we have designed the simulator in the way that the difference now is instead which oracle \mathcal{D} is interacting with (H vs H'_x). Immediately after \mathcal{D} is given the challenge, the overall states of the two simulators after conditioning are $|\phi_1\rangle, |\phi_2\rangle$ respectively. Intuitively, the problem now reduces to distinguishing these two oracles.

Lemma 5.10. $\|\phi_1\rangle - |\phi_2\rangle\|_2 \leq 2 \cdot \sqrt{q/N}$.

Proof. We compare two vectors component-wise. Fixing z, D, H, h' , since H and H'_x only differ at point x , if x satisfies $D(x) = 0$, then by how we defined the inner product,

$$\omega_M^{\langle D, H \rangle} = \omega_M^{\langle D, H'_x \rangle}.$$

Therefore, we can write these two vectors as $|\phi_1\rangle = |\phi\rangle + |\text{err}_1\rangle$ and $|\phi_2\rangle = |\phi\rangle + |\text{err}_2\rangle$, where $|\phi\rangle$ only sums over x such that $D(x) = 0$. Their difference is bounded by the following,

$$\begin{aligned} \left\| |\phi_1\rangle - |\phi_2\rangle \right\|_2 &= \left\| |\text{err}_1\rangle - |\text{err}_2\rangle \right\|_2 \\ &\leq \left| \frac{1}{N^{1/2}M^{(N+1)/2}} \sum_{z, D: |D| \leq q} \sum_{H, h'} \sum_{x: D(x) \neq 0} \alpha_{z, D} \omega_M^{\langle D, H \rangle} |z, h'\rangle |x\rangle |H, h'\rangle \right|_2 + \\ &\quad \left| \frac{1}{N^{1/2}M^{(N+1)/2}} \sum_{z, D: |D| \leq q} \sum_{H, h'} \sum_{x: D(x) \neq 0} \alpha_{z, D} \omega_M^{\langle D, H'_x \rangle} |z, h'\rangle |x\rangle |H, h'\rangle \right|_2 \\ &\leq 2 \cdot \sqrt{\frac{1}{M^{N+1}} \sum_{z, D: |D| \leq q} \sum_{H, h'} \frac{|D|}{N} \cdot |\alpha_{z, D}|^2} \\ &\leq 2 \cdot \sqrt{\frac{q}{N} \cdot \sum_{z, D: |D| \leq q} |\alpha_{z, D}|^2} \\ &= 2\sqrt{q/N}, \end{aligned}$$

where the last equality is due to the fact that the α_* 's are defined as Lemma 2.3. \square

Therefore, before making the next q' queries, the overall state is $|\phi_1\rangle$ if it is in *Case 1* or $|\phi_2\rangle$ if it is in *Case 2*. Let U_H be the oracle query to H and $U_{H'_x}$ be the oracle query to H'_x . We can write the rest of \mathcal{D}^H as $U_{q'}U_H \cdots U_1U_HU_0$. First, by the fact that unitary preserves L^2 -norm and triangle inequality, we have,

$$\begin{aligned} &\left| U_{q'}U_H \cdots U_1U_HU_0 |\phi_1\rangle - U_{q'}U_{H'_x} \cdots U_1U_{H'_x}U_0 |\phi_2\rangle \right|_2 \\ &\leq \left| U_{q'}U_H \cdots U_1U_HU_0 |\phi_1\rangle - U_{q'}U_{H'_x} \cdots U_1U_{H'_x}U_0 |\phi_1\rangle \right|_2 + 2\sqrt{q/N} \\ &\leq \sum_{j=0}^{q'-1} \left\| |\psi_j\rangle - |\psi_{j+1}\rangle \right\|_2 + 2\sqrt{q/N}, \end{aligned}$$

where $|\psi_j\rangle$ is defined as applying U_H on $|\phi_1\rangle$ for the first j oracle queries and then $U_{H'_x}$ for the rest $(q' - j)$ oracle queries.

Lemma 5.11. *For all $j = 0, \dots, q' - 1$, $\left\| |\psi_j\rangle - |\psi_{j+1}\rangle \right\|_2 \leq 2/\sqrt{N}$.*

Proof. Before making the $(j + 1)$ -th oracle query, let $z = (x', u, \text{aux})$ where x', u corresponds to the input to the phase oracle, by Lemma 2.3, the overall state can be written as

$$\frac{1}{M^{N/2}} \sum_{z, D: |D| \leq q+i} \sum_H \beta_{z, D} |x', u, \text{aux}\rangle_{\mathcal{D}} \otimes \omega_M^{\langle D, H \rangle} |H, h'\rangle \otimes \frac{1}{\sqrt{N}} \sum_x |x\rangle.$$

Note that we can make this reordering since H'_x is never invoked, and $|x\rangle$ is never used and therefore unentangled with other registers.

For the $(j+1)$ -th oracle query, either U_H or $U_{H'_x}$ is applied. We find that two oracle queries are different if and only if $x = x'$. Therefore, the difference is bounded by two times the norm of the component whose $x = x'$. Since $|x\rangle$ is un-entangled with other registers, the norm is at most $1/\sqrt{N}$. Therefore, $|\psi_j\rangle - |\psi_{j+1}\rangle| \leq 2/\sqrt{N}$. \square

Since each $|\psi_j\rangle - |\psi_{j+1}\rangle| \leq 2/\sqrt{N}$, we conclude that

$$|U_{q'}U_H \cdots U_1U_HU_0|\phi_1\rangle - U_{q'}U_{H'} \cdots U_1U_{H'}U_0|\phi_2\rangle|_2 \leq 2(q' + \sqrt{q})/\sqrt{N}.$$

By perfect simulation, we have

$$\begin{aligned} \sqrt{\Pr_{y \leftarrow \mathfrak{s}[M]} [\mathcal{D}^{(H)}(y) = 1 \mid \mathbf{Y}_{[p]} = \mathbf{y}_{[p]}]} &= |P_1U_{q'}U_H \cdots U_1U_HU_0|\phi_1\rangle|_2 \\ \sqrt{\Pr_{x \leftarrow \mathfrak{s}[M]} [\mathcal{D}^{(H)}(H(x)) = 1 \mid \mathbf{Y}_{[p]} = \mathbf{y}_{[p]}]} &= |P_1U_{q'}U_{H'} \cdots U_1U_{H'}U_0|\phi_2\rangle|_2. \end{aligned}$$

Our lemma follows. \square

5.3 Multi-Instance Security of OWF

Now with Proposition 5.7 and Lemma 5.9, we are ready to prove the multi-instance security of $G_{\text{OWF}}^{\otimes g}$ that we have claimed at the beginning.

Lemma 5.2. $G_{\text{OWF}}^{\otimes g}$ is $\delta(g, T) = O\left(\frac{gT+T^2}{\min\{N, M\}}\right)$ -secure in the QROM.

Proof. Let B_i be the random variable for b_i , indicating if \mathcal{A} passes the i -th round of the multi-instance game. Let Y_i be the random variable for the i -th challenge. Define B'_i be the random variable that \mathcal{A} finds a pre-image of the i -th challenge if Y_i is instead distributed uniformly at random from $[M]$.

From Lemma 5.9 and setting $q = (i-1)(T+2)$ and $q' = T+1$, we have, for any challenge $\mathbf{y}_{[i-1]}$ (with non-zero probability conditioned on $\mathbf{B}_{[i-1]} = \mathbf{1}^{i-1}$),

$$\begin{aligned} &\left| \sqrt{\Pr[B_i = 1 \mid \mathbf{Y}_{[i-1]} = \mathbf{y}_{[i-1]}, \mathbf{B}_{[i-1]} = \mathbf{1}^{i-1}]} \right. \\ &\quad \left. - \sqrt{\Pr[B'_i = 1 \mid \mathbf{Y}_{[i-1]} = \mathbf{y}_{[i-1]}, \mathbf{B}_{[i-1]} = \mathbf{1}^{i-1}]} \right| \leq O\left(\frac{\sqrt{iT} + T}{\sqrt{N}}\right). \end{aligned}$$

Since $B'_i = 1$ is the single stage G_{OWF_y} game, by Proposition 5.7, we have,

$$\Pr[B'_i = 1 \mid \mathbf{Y}_{[i-1]} = \mathbf{y}_{[i-1]}, \mathbf{B}_{[i-1]}] = O\left(\frac{iT + T^2}{M}\right).$$

Then, by Cauchy-Schwarz, we have,

$$\begin{aligned} &\Pr[B_i = 1 \mid \mathbf{Y}_{[i-1]} = \mathbf{y}_{[i-1]}, \mathbf{B}_{[i-1]} = \mathbf{1}^{i-1}] \\ &\leq 2\Pr[B'_i = 1 \mid \mathbf{Y}_{[i-1]} = \mathbf{y}_{[i-1]}, \mathbf{B}_{[i-1]}] + O\left(\frac{iT + T^2}{N}\right) \\ &\leq O\left(\frac{iT + T^2}{M} + \frac{iT + T^2}{N}\right) = O\left(\frac{gT + T^2}{\min\{M, N\}}\right). \end{aligned}$$

Finally, we complete the proof by invoking Lemma 3.14. \square

5.4 Security of PRG with Advice

We note that Lemma 5.9 essentially implies a security bound for PRG with advice. Let us start by reformalizing PRG security in our security game framework.

Definition 5.12 (PRG Security Game). *Let H be a random oracle $[N] \rightarrow [M]$ where $N < M$. The security game $G_{\text{PRG}} = (C_{\text{PRG}})$ is specified by three procedures (Samp, Query, Ver), where:*

1. $\text{Samp}^H(b, x, y)$ takes a random bit $b \in \{0, 1\}$, an element $x \in [N]$ and an $y \in [M]$, and outputs $H(x)$ if $b = 0$, and y otherwise.
2. $\text{Query}^H((b, x, y), x') = H(x')$ simply queries H directly.
3. $\text{Ver}^H((b, x, y), b')$ outputs 1 if and only if $b = b'$.

To fill in the rest of the details from Definition 3.3, the security game defined here is that given a random oracle, the adversary is asked to distinguish a random $y \in [M]$ from a random image $H(x)$. The adversary is allowed to make oracle queries to H and it wins if and only if he distinguishes the two distributions correctly. This is exactly the security of PRG in QROM.

Using Lemma 5.9, we have the following lemma:

Lemma 5.13. $G_{\text{PRG}}^{\otimes g}$ is $\delta(g, T) = \frac{1}{2} + O\left(\sqrt{\frac{gT+T^2}{N}}\right)$ -secure in the QROM.

Proof. Consider in round i , conditioned on random outcomes in the previous rounds $(b_j, x_j, y_j, b'_j)_{j < i}$, which we simply denote as event E . Let $p_{b,b'}$ be the probability that challenge is b and the algorithm outputs b' . By Lemma 5.9 and Cauchy-Schwarz, for any b' we have that

$$\begin{aligned} |p_{0,b'} - p_{1,b'}| &\leq |\sqrt{p_{0,b'}} - \sqrt{p_{1,b'}}| \cdot |\sqrt{p_{0,b'}} + \sqrt{p_{1,b'}}| \\ &\leq 2 \frac{\sqrt{(i-1)(T+1)} + T}{\sqrt{N}} \cdot 2 \\ &\leq 4\sqrt{2} \cdot \sqrt{\frac{g(T+1) + T^2}{N}}. \end{aligned}$$

Therefore, $|\Pr[b = b'|E] - \Pr[b \neq b'|E]| \leq |p_{0,0} - p_{1,0}| + |p_{0,1} - p_{1,1}| \leq 8\sqrt{2} \cdot \sqrt{\frac{g(T+1)+T^2}{N}}$, and we conclude that $\Pr[b = b'|E] \leq \frac{1}{2} + 4\sqrt{2} \cdot \sqrt{\frac{g(T+1)+T^2}{N}}$. \square

Consider the algorithm that queries $H(1)$, and outputs 0 if $H(1) = \text{ch}$ and 1 otherwise, then it will succeed with probability $1/2 + \Omega(1/N)$. Combining these with Theorem 4.2 (or Theorem 4.4), we obtain a security bound for PRG in the AI-QROM (or QAI-QROM, respectively).

Theorem 5.14. G_{PRG} is $\frac{1}{2} + \tilde{O}\left(\frac{ST+T^2}{N}\right)^{1/3}$ -secure in the AI-QROM, and $\frac{1}{2} + \tilde{O}\left(\frac{S^5T+S^4T^2}{N}\right)^{1/19}$ -secure in the QAI-QROM, where \tilde{O} absorbs $\text{poly}(\log N, \log M, \log S)$ factors.

6 Yao's Box

In this section, we show the security of Yao's box problem in the AI-QROM and QAI-QROM. Let us start with reformalizing Yao's box problem using our language of security games.

Definition 6.1 (Yao's box security game). *Let H be a random oracle $[N] \rightarrow [2]$. The security game $G_{\text{YB}} = (C_{\text{YB}})$ is specified by three procedures (Samp, Query, Ver), where:*

1. $\text{Samp}^H(x) = x$ takes some randomness $x \in [N]$, and outputs x as the challenge pre-image.
2. $\text{Query}^H(x, \cdot)$: for all input $x' \neq x$, the output is $H(x')$; $\text{Query}^H(x, x) = 1$.
3. $\text{Ver}^H(x, b)$ outputs 1 if and only if $H(x) = b$.

To fill in the rest of the details from Definition 3.3, given a random oracle with binary range, the challenger samples a random input x as the challenge. The adversary is allowed to make oracle queries to H except on input x and it wins if and only if it finds $H(x)$ without making any queries on x (or equivalently, even if he makes query to x , he will just get a constant 1).

We are going to prove the following lemma, the multi-instance security of $G_{\text{YB}}^{\otimes g}$.

Lemma 6.2. $G_{\text{YB}}^{\otimes g}$ is $\delta(g, T) = 1/2 + O(\sqrt{gT/N})$ -secure in the QROM.

Combining this lemma together with Theorem 4.2 (or Theorem 4.4), we can show the security of G_{YB} in the AI-QROM (or QAI-QROM, respectively).

Theorem 6.3. G_{YB} is $\delta(S, T) = 1/2 + \tilde{O}\left((ST/N)^{1/3}\right)$ -secure in the AI-QROM.

Proof. Let a trivial algorithm with 0 queries and 1 bit of advice do the following: store $H(1)$ as its advice; answer $H(1)$ if the challenge is 1 and answer a random bit otherwise. The advantage of this algorithm is $1/2 + 1/(2N)$ because with probability $1 - 1/N$, it succeeds with probability $1/2$; with probability $1/N$, it succeeds with probability 1.

Therefore, in Theorem 4.2, we have $\varepsilon_0 = 1/(2N)$, $S_0 = 1$, $T_0 = 0$ and $G_{\text{YB}}^{\otimes g}$ is $1/2 + \varepsilon(g, T) = 1/2 + \tilde{O}(\sqrt{gT/N})$ -secure. We have G_{YB} is $1/2 + \varepsilon'(S, T)$ -secure where,

$$\begin{aligned} \varepsilon'(S, T) &\leq 4 \cdot \varepsilon \left(\frac{10 \ln 2}{\varepsilon'(S, T)} (S + S_0 + \log(1/\varepsilon_0) + 2, T + T_0) \right) \\ &= \tilde{O} \left(\sqrt{\frac{ST}{\varepsilon'(S, T) N}} \right). \end{aligned}$$

We conclude that $\varepsilon'(S, T) = \tilde{O}\left((ST/N)^{1/3}\right)$. □

Theorem 6.4. G_{YB} is $\delta(S, T) = 1/2 + \tilde{O}\left((S^5 T/N)^{1/19}\right)$ -secure in the QAI-QROM.

Proof. Let a trivial algorithm with 0 queries and 1 bit of advice do the following: store $H(1)$ as its advice; answer $H(1)$ if the challenge is 1 and answer a random bit otherwise. The advantage of this algorithm is $1/2 + 1/(2N)$ because with probability $1 - 1/N$, it succeeds with probability $1/2$; with probability $1/N$, it succeeds with probability 1.

In Theorem 4.4, we have $S_0 = 1, T_0 = 0, \varepsilon_0 = 1/(2N)$ and $\varepsilon(g, T) = \tilde{O}(\sqrt{gT/N})$. We have G_{YB} is $1/2 + \varepsilon'(S, T)$ -secure where,

$$\begin{aligned}\varepsilon'(S, T) &\leq 8 \cdot \varepsilon \left(\tilde{O}(S + S_0)^3 / \varepsilon^{19}, \tilde{O}((T + T_0)(S + S_0)^2) / \varepsilon^{18} \right) \\ &= \tilde{O} \left(\sqrt{\frac{S^5 T}{\varepsilon'(S, T)^{17} \cdot N}} \right).\end{aligned}$$

We conclude that $\varepsilon'(S, T) = \tilde{O} \left((S^5 T / N)^{1/19} \right)$. □

Now the problem reduces to proving its multi-instance security.

Proof of Lemma 6.2. Let \mathcal{A} be any (g, T) adversary in the QROM. We again simulate the random oracle H with a compressed standard oracle, which by Corollary 2.5 perfectly preserves the functionality of $\mathcal{A} \iff C_{\text{YB}}^{\otimes g}$. We will also implement the projective measurement by directly measuring \mathcal{A} 's output, as the output in Yao's box is binary.

Assume after passing the first $i - 1$ challenges, conditioned on the challenges are x_1, \dots, x_{i-1} and the algorithm passes all the challenges, the overall state of the algorithm and compressed oracle is $|\phi_0\rangle = \sum_{z, D} \alpha_{z, D} |z\rangle |D\rangle$. By Lemma 2.6, we know every possible D with non-zero weight is of size at most $(i - 1)(T + 1) < g(T + 1)$, which is the total number of queries to CStO .

Fixing an x_i (which is chosen with probability $1/N$), let $p_{x_i}^{(t)}$ be the probability that after T queries, measuring the database register, it gives a database containing x_i . In other words, define

$$P_{x_i} = \sum_{D: D(x_i) \neq \perp} |z, D\rangle \langle z, D|.$$

Then $p_{x_i}^{(0)} = |P_{x_i} |\phi_0\rangle|^2$. Similar to (2), we have,

$$\sum_{x_i=1}^N p_{x_i}^{(0)} \leq g(T + 1).$$

As defined by $\text{Query}^H(x_i, \cdot)$, each query made by \mathcal{A} will be sent to the challenger first and the challenger will apply the oracle query only on inputs that are not x_i . Therefore, assume before the first query the state is $|\phi_0\rangle = \sum_{x, u, \text{aux}, D} \alpha_{x, u, \text{aux}, D} |x, u, \text{aux}\rangle |D\rangle$. After the first query, it becomes

$$|\phi_1\rangle := U_H \sum_{\substack{x \neq x_i, u, \text{aux} \\ D}} \alpha_{x, u, \text{aux}, D} |x, u, \text{aux}\rangle |D\rangle + \sum_{\substack{u, \text{aux} \\ D}} \alpha_{x_i, u, \text{aux}, D} |x_i, u, \text{aux}\rangle |D\rangle.$$

where U_H is defined as CStO applied on x, u and D register. Recall that $p_{x_i}^{(0)} = |P_{x_i} |\phi_0\rangle|^2$ and $p_{x_i}^{(1)} = |P_{x_i} |\phi_1\rangle|^2$. We want to show the following lemma.

Lemma 6.5. *Let $p_{x_i}^{(0)}$ and $p_{x_i}^{(1)}$ be the probability defined above. Then $p_{x_i}^{(0)} = p_{x_i}^{(1)}$.*

Proof. This can be shown by a direct calculation.

$$\begin{aligned}
p_{x_i}^{(1)} &= |P_{x_i}|\phi_1\rangle|^2 \\
&= \left| \sum_{\substack{x \neq x_i, u, \text{aux} \\ D}} \alpha_{x, u, \text{aux}, D} P_{x_i} U_H |x, u, \text{aux}\rangle |D\rangle + \sum_{\substack{u, \text{aux} \\ D}} \alpha_{x_i, u, \text{aux}, D} P_{x_i} |x_i, u, \text{aux}\rangle |D\rangle \right|^2 \\
&= \left| \sum_{\substack{x \neq x_i, u, \text{aux} \\ D}} \alpha_{x, u, \text{aux}, D} P_{x_i} U_H |x, u, \text{aux}\rangle |D\rangle \right|^2 + \left| \sum_{\substack{u, \text{aux} \\ D}} \alpha_{x_i, u, \text{aux}, D} P_{x_i} |x_i, u, \text{aux}\rangle |D\rangle \right|^2 \\
&\stackrel{(*)}{=} \sum_{\substack{x \neq x_i, u, \text{aux}, D \\ D(x_i) \neq \perp}} |\alpha_{x, u, \text{aux}, D}|^2 + \sum_{\substack{u, \text{aux}, D \\ D(x_i) \neq \perp}} |\alpha_{x_i, u, \text{aux}, D}|^2 \\
&= \sum_{\substack{x, u, \text{aux}, D \\ D(x_i) \neq \perp}} |\alpha_{x, u, \text{aux}, D}|^2 = p_{x_i}^{(0)}.
\end{aligned}$$

The equation (*) comes from the fact that if U_H applies on $|x, u, D\rangle$ such that $x \neq x_i$, it does not change $D(x_i)$; in other words, if $D(x_i) = \perp$, then after applying U_H it is still \perp ; if $D(x_i) = v$, then after applying U_H it is still v . \square

Let $p^{(t)} := |P_{x_i}|\phi_i\rangle|$, where $|\phi_i\rangle$ is the overall state after i -th query. As we do not assume any structure of $|\phi_0\rangle$, and the algorithm's local computation does not impact the projector value, we can use this lemma t times and conclude that $p_{x_i}^{(t)} = p_{x_i}^{(0)}$ for all t .

By Lemma 2.7, let the probability that \mathcal{A} wins the game in the i -th round is p_{x_i} , after T queries we have, $\sqrt{p_{x_i}} \leq \sqrt{p_{x_i}^{(T)}} + \sqrt{1/M} = \sqrt{p_{x_i}^{(T)}} + \sqrt{1/2} = \sqrt{p_{x_i}^{(0)}} + \sqrt{1/2}$. Therefore, the overall probability is,

$$\begin{aligned}
\frac{1}{N} \sum_{x_i=1}^N p_{x_i} &= \frac{1}{N} \sum_{x_i=1}^N \left(\sqrt{p_{x_i}^{(0)}} + \sqrt{1/2} \right)^2 \\
&= \frac{1}{2} + \sum_{x_i=1}^N p_{x_i}^{(0)}/N + \sqrt{2} \cdot \sum_{x_i=1}^N \sqrt{p_{x_i}^{(0)}}/N \\
&\leq \frac{1}{2} + (g(T+1))/N + \sqrt{2g(T+1)/N} \\
&\leq \frac{1}{2} + (1 + \sqrt{2})\sqrt{g(T+1)/N}.
\end{aligned}$$

The above inequalities come from $\sum_{x_i=1}^N p_{x_i}^{(0)} \leq g(T+1)$, Cauchy-Schwarz, and assuming $g(T+1)/N \leq 1$ (which implies $g(T+1)/N \leq \sqrt{g(T+1)/N}$). On the other hand, if $g(T+1)/N > 1$, the bound above trivially holds.

Let B_i be the random variable for b_i , indicating if \mathcal{A} passes the i -th round of the multi-instance game. Let X_i be the random variable for the i -th challenge. The above statement says for any $\mathbf{x}_{[i-1]}$,

$$\Pr [B_i = 1 \mid \mathbf{B}_{[i-1]} = \mathbf{1}^{i-1}, \mathbf{X}_{[i-1]} = \mathbf{x}_{[i-1]}] \leq 1/2 + (1 + \sqrt{2})\sqrt{g(T+1)/N}$$

It implies $\Pr[B_1 = 1, B_2 = 1, \dots, B_g = 1] \leq (1/2 + (1 + \sqrt{2})\sqrt{g(T+1)/N})^g$. Therefore, for any (g, T) algorithm \mathcal{A} in the QROM, the succeeding probability is at most $(1/2 + O(\sqrt{gT/N}))^g$. \square

7 Salting Defeats Preprocessing

In this section, we show generic salting defeats quantum pre-processing with classical/quantum advice. There exist schemes that are secure in the QROM but not in the AI-QROM. One example is collision-resistant hash function. For a random oracle $H : [N] \rightarrow [M]$ with large enough N , finding a collision without any pre-processing is hard [AS04, LZ19a], which requires $T = \Omega(M^{1/3})$ queries. However, in the AI-QROM, as long as $S \geq 2 \cdot \log N$, the adversary can find a pair of collision in the pre-processing procedure and store it as the advice/leakage; in the online phase, it simply answers the pair. Therefore, collision-resistance is broken in the AI-QROM.

Definition 7.1 (Public-Salt Security Game). *Given a security game $G = (C)$, where $C = (\text{Samp}, \text{Query}, \text{Ver})$. Consider a “salted” oracle where we enlarge the input space: $H_S : [K] \times [N] \mapsto [M]$, for some parameter K . The salted security game $G_S = (C_S)$ defined on H_S , where $C_S = (\text{Samp}_S, \text{Query}_S, \text{Ver}_S)$ is specified as below:*

1. $\text{Samp}_S^{H_S}(s, r)$: on input uniformly random salt $s \in [K]$ and $r \in R$, runs $\text{ch} \leftarrow \text{Samp}^{H_S(s, \cdot)}(r)$, returns $\text{ch}_S := (s, \text{ch})$.
2. $\text{Query}_S^{H_S}((s, r), (s', x))$: on input salt s and challenger randomness r , query salt s' and some input x , if $s = s'$, output $\text{Query}^{H_S(s, \cdot)}(r, x)$; otherwise, output $H_S(s', x)$.
3. $\text{Ver}_S^{H_S}((s, r), \text{ans})$: output $\text{Ver}^{H_S(s, \cdot)}(r, \text{ans})$.

Lemma 7.2. *For any security game G with security $\delta' = \delta'(T)$ in the QROM. For any integer g and salt space $[K]$, the multi-instance game $G_S^{\otimes g}$ is δ -secure, where*

$$\delta = \delta(g, T) = 2\delta' + \frac{g(T+1)}{K}. \quad (4)$$

$G_S^{\otimes g}$ is also $\tilde{\delta}$ -secure, where

$$\tilde{\delta} = \tilde{\delta}(g, T) = \delta' + 3\sqrt{\frac{g(T+1)}{K}}. \quad (5)$$

We usually want to use (5) for decision games as a trivial random guessing adversary already wins with probability $1/2$, and the first bound trivially holds.

Combining the first part of Lemma 7.2 with Theorem 4.1, we have the following theorem. Roughly speaking, the theorem says if a security game G is $\delta(T)$ -secure in the QROM, then the salted game G_S with salt space $[K]$ is $2(\delta(T) + ST/K)$ -secure against (S, T) adversary in the AI-QROM. In other words, as long as $K \gg ST$, pre-processing does not help break the salted game G_S .

Theorem 7.3. *Let G be a security game with security $\delta'(T)$ in the QROM. Let G_S be the salted security game with parameter K . Let δ_0 be the winning probability of an adversary that outputs a random answer in the game G_S without advice or making any query. Then G_S 's security $\delta(S, T)$ against (S, T) -adversaries in the AI-QROM is*

$$\delta(S, T) \leq 8 \cdot \left(\frac{(S + \log(1/\delta_0) + 1)(T + 1)}{K} + \delta'(T) \right)$$

Proof. If G has security $\delta'(T)$, then $G_S^{\otimes g}$ is $\delta''(g, T) = 2(\delta'(T) + g(T+1)/K)$ -secure. By Theorem 4.1, G_S is $\delta(S, T)$ -secure in the AI-QROM, where $\delta(S, T) \leq 4 \cdot \delta''(S + \log 1/\varepsilon_0 + 1, T)$. \square

Combining the first part of Lemma 7.2 with Theorem 4.3, we have the following theorem about salted public-verifiable security games in the QAI-QROM.

Theorem 7.4. *Let G be a publicly-verifiable security game with security $\delta'(T)$ in the QROM. Let $T_{\text{Ver}} = \text{poly}(\log M, \log N)$ be the upper bound on the number of \tilde{H} queries for computing $\text{Ver}^{\tilde{H}}(\text{ch}, \cdot)$, i.e., the number of queries required to do the public verification. Let G_S be the salted security game with parameter K . Let $\delta_0 = 1/\text{poly}(N, M)$ be a non-negligible lower bound on the winning probability of an adversary that outputs a random answer in the G_S without advice or making any query. Then G_S 's security $\delta = \delta(S, T)$ against (S, T) adversaries in the QAI-QROM satisfying*

$$\delta \leq \tilde{O} \left(\frac{ST}{K\delta} + \delta' \left(\tilde{O}(T)/\delta \right) \right),$$

where \tilde{O} absorbs $\text{poly}(\log N, \log M, \log S)$ factors.

Combining the second part of Lemma 7.2 with Theorem 4.2, we have the following theorem about salted decision security games in the AI-QROM. Roughly speaking, the theorem says if a decision game G is $1/2 + \varepsilon(T)$ -secure in the QROM, then the salted game G_S with salt space $[K]$ is $1/2 + O(\varepsilon(T) + (ST/K)^{1/3})$ -secure against (S, T) adversary in the AI-QROM. In other words, as long as $K \gg ST$, pre-processing does not help break the salted game G_S .

Theorem 7.5. *Let G be a decision game with security $1/2 + \varepsilon'(T)$. Let G_S be the salted decision game with parameter K . Let ε_0 be a lower bound on the advantage for an adversary for G_S with T_0 queries and S_0 bits of advice. Then G_S 's security $\delta(S, T) = 1/2 + \varepsilon(S, T)$ against (S, T) -adversaries in the AI-QROM satisfies that*

$$\varepsilon(S, T) = O \left(\varepsilon'(T + T_0) + \left(\frac{(S + S_0 + \log(1/\varepsilon_0))(T + T_0)}{K} \right)^{1/3} \right).$$

Combining the second part of Lemma 7.2 with Theorem 4.4, we have the following theorem about salted decision security games in the QAI-QROM.

Theorem 7.6. *Let G be a decision game with security $1/2 + \varepsilon'(T)$. Let G_S be the salted decision game with parameter K . Let $\varepsilon_0 = 1/\text{poly}(N, M)$ be a non-negligible lower bound on the advantage for an adversary for G_S with T_0 queries and S_0 bits of advice. Then G_S 's security $\delta(S, T) = 1/2 + \varepsilon(S, T)$ against (S, T) -adversaries in the QAI-QROM satisfies that*

$$\varepsilon = \tilde{O} \left(\varepsilon' \left(\frac{\tilde{O}((T + T_0)(S + S_0)^2)}{\varepsilon^8} \right) + \sqrt{\frac{(S + S_0)^5(T + T_0)}{K\varepsilon^{17}}} \right),$$

where \tilde{O} absorbs $\text{poly}(\log N, \log M, \log S)$ factors.

7.1 Multi-Instance Security of Salted Game

In this section, we prove Lemma 7.2. We first argue some general facts about the salting game, and then show that (4) and (5) holds respectively, and therefore concluding the proof.

Let \mathcal{A} be any (g, T) adversary in the QROM for $G_S^{\otimes g}$. We again start by considering the following algorithm that perfectly simulates the original interaction:

1. Instead of directly sampling the oracle distribution $H_S : [K] \times [N] \mapsto [M]$, we instead consider the equivalent sampling procedure where we instead sample a random function $H_s : [K] \mapsto \mathcal{R}_N$, where \mathcal{R}_N denotes the randomness for sampling the original oracle $H : [N] \mapsto [M]$. Let $\mathcal{S} : \mathcal{R}_N \mapsto ([N] \mapsto [M])$ be the original oracle sampler, that is \mathcal{S} on input some randomness, samples (according to the distribution) the entire truth table of $H : [N] \mapsto [M]$. $H_S([K], [N])$ has the exact same distribution as $\mathcal{S}(H_s([K]))([N])$ as long as the sampler \mathcal{S} is correct.

For random functions or random oracle model, $\mathcal{R}_N = [M^N]$, and any arbitrary bijection $[M^N] \leftrightarrow [M]^N$ is a correct sampler.

2. By definition we have $\text{Ver}_S^{H_S}((s, r), \text{ans}) = \text{Ver}^{H_S(s, \cdot)}(r, \text{ans})$, therefore, the verification algorithm will only ever query $H_S(s, \cdot) = \mathcal{S}(H_s(s))(\cdot)$. Using this observation, whenever we need to run $\text{Ver}_S^{H_S}((s, r), \text{ans})$, we instead first obtain the (entire truth table of) oracle $H' \leftarrow \mathcal{S}(H_s(s))$, and run $b \leftarrow \text{Ver}^{H'}(r, \text{ans})$ directly.

Note that after this change, each round will only ever issue $(T + 1)$ queries to the oracle H_s .

3. We replace the oracle access to H_s with the compressed standard oracle operator CStO, which preserves the functionality by Corollary 2.5.

Conditioned on passing the first $(i - 1)$ challenges, and the challenges being $(s_1, \text{ch}_1), \dots, (s_{i-1}, \text{ch}_{i-1})$ and the algorithm having passed all the challenges (we again only consider the s 's and ch 's that have non-zero probability), we denote the overall state of the algorithm and compressed oracle to be $|\phi_0\rangle = \sum_{z, D} \alpha_{z, D} |z\rangle |D\rangle$. By Lemma 2.6, since the total number of queries made by \mathcal{A} and $C^{\otimes g}$ is at most $(i - 1)(T + 1)$, we know every possible D with non-zero weight is of size at most $g(T + 1)$.

Fixing a salt s and challenger randomness r . Let $p_{s, r}$ be the probability that \mathcal{A} wins conditioned on salt s and randomness r . Let q_s be the probability that measuring the database register gives a database containing s . In other words, define

$$Q_s = \sum_{z, D: D(s) \neq \perp} |z, D\rangle \langle z, D|$$

Then $q_s = |Q_s |\phi_0\rangle|^2$. Similar to (2), we have, $\sum_{s \in [K]} q_s \leq g(T + 1)$.

Let $U_{s, r}$ be the unitary describing the rest of the computation for this round, that is including: \mathcal{A} 's oracle queries (which also includes the oracle expansion in superposition using \mathcal{S}) and local computation, our added query and \mathcal{S} for computing H' in change 2, and the final local computation $\text{Ver}^{H'}(r, \text{ans})$. Let $P_{s, r}$ be the projective measurement that Ver outputs 1. Using triangular

inequality, we then have

$$\begin{aligned}
\sqrt{p_{s,r}} &= |P_{s,r}U_{s,r}|\phi\rangle_0| \\
&\leq |P_{s,r}U_{s,r}Q_s|\phi\rangle_0| + |P_{s,r}U_{s,r}(I - Q_s)|\phi\rangle_0| \\
&\leq |Q_s|\phi\rangle_0| + |P_{s,r}U_{s,r}(I - Q_s)|\phi\rangle_0| \\
&= \sqrt{q_s} + |P_{s,r}U_{s,r}(I - Q_s)|\phi\rangle_0|.
\end{aligned}$$

Proof of (4). Taking expectation over the randomness of s, r , the probability of \mathcal{A} succeeding in round i is

$$\begin{aligned}
\frac{1}{K \cdot R} \sum_{s,r} p_{s,r} &\leq \frac{1}{K \cdot R} \sum_{s,r} (\sqrt{q_s} + |P_{s,r}U_{s,r}(I - Q_s)|\phi\rangle_0|)^2 \\
&\leq \frac{2}{K \cdot R} \sum_{s,r} \left(q_s + |P_{s,r}U_{s,r}(I - Q_s)|\phi\rangle_0|^2 \right) \\
&\leq \frac{2 \cdot g(T+1)}{K} + \frac{2}{K \cdot R} \sum_{s,r} |P_{s,r}U_{s,r}(I - Q_s)|\phi\rangle_0|^2.
\end{aligned} \tag{6}$$

Next, we are going to show the second term is at most $2\delta'(T)$. We are going to construct an algorithm \mathcal{B} making at most T queries which takes \mathcal{A} as a subroutine and wins the game G with probability at least $\frac{1}{K \cdot R} \sum_{s,r} |P_{s,r}U_{s,r}(I - Q_s)|\phi\rangle_0|^2$, which can not be greater than $\delta'(T)$ (the security of G).

Consider the following algorithm \mathcal{B} breaking $G = (C)$, \mathcal{B} is going to simulate the multi-instance game for the first $(i - 1)$ rounds and simulate the i -th round by making queries to C . Therefore, if \mathcal{A} win the i -th round of $G^{\otimes g}$, then \mathcal{B} wins G .

1. \mathcal{B} takes \mathcal{A} as a subroutine. Let H be the oracle in the game G .
2. \mathcal{B} simulates the random oracle H_S exactly the same as our simulator (in particular, using CStO on H_s and does not use H) and interacts with \mathcal{A} up to round $(i - 1)$.
3. \mathcal{B} checks that \mathcal{A} passes the first $(i - 1)$ rounds.
4. \mathcal{B} samples a salt $s \leftarrow [K]$ and it checks (or measures) whether $D(s) = \perp$.
5. If any of the two checks fails, \mathcal{B} clears all registers and start back at the beginning.

If both checks pass, note that the overall state now conditioned on (s_i, r_i, b_i) 's is exactly $\frac{1}{\ell_s}(I - Q_s)|\phi_0\rangle$ where $\ell_s = |(I - Q_s)|\phi_0\rangle| \leq 1$ is the normalization factor.

6. \mathcal{B} simulates the i -th round using a modified oracle H^* , where H^* is consistent with H_S , except on salt s where the queries are instead forwarded to Query^H . In particular,
 - The challenge sent to \mathcal{A} is instead $\text{Samp}_S^{H^*}(s, r) = \text{Samp}^{H^*(s, \cdot)}(r) = \text{Samp}^H(r) = \text{ch}$ where ch is the challenge received from C .
 - When \mathcal{A} makes a quantum oracle query $\text{Query}_S^{H^*}((s, r), (s', x))$, if $s \neq s'$, \mathcal{B} can simulate the query using CStO on H_s ; if $s = s'$, \mathcal{B} makes an oracle query (r, x) to C , since $\text{Query}_S^{H^*}((s, r), (s', x)) = \text{Query}^H(r, x)$ if $s = s'$.

Note that this simulation does not require knowledge of r and only uses query access to Query^H , as desired.

7. Finally, \mathcal{B} output $|\text{ans}\rangle$ from the overall state $\frac{1}{\ell_s} U_{s,r}(I - Q_s) |\phi_0\rangle$.

Since we condition on step 4 passing ($D(s) = \perp$), H_S and H^* are perfectly indistinguishable for \mathcal{A} . Therefore the succeeding probability of \mathcal{B} is,

$$\frac{1}{K \cdot R} \sum_{s,r} \left| \frac{1}{\ell_s} P_{s,r} U_{s,r}(I - Q_s) |\phi_0\rangle \right|^2 \geq \frac{1}{K \cdot R} \sum_{s,r} |P_{s,r} U_{s,r}(I - Q_s) |\phi_0\rangle|^2.$$

By security of the unsalted game, left hand side is bounded by δ' , therefore,

$$\frac{1}{K \cdot R} \sum_{s,r} |P_{s,r} U_{s,r}(I - Q_s) |\phi_0\rangle|^2 \leq \delta'. \quad (7)$$

Putting this together with (6), we conclude that the algorithm wins i -th round with probability at most $2g(T+1)/K + 2\delta'(T)$. Therefore, invoking Lemma 3.14, for any (g, T) algorithm in the QROM, the succeeding probability is at most $(2g(T+1)/K + 2\delta'(T))^g$. \square

Proof of (5). Using (7), the probability of \mathcal{A} winning round i conditioned on (s_i, r_i, b_i) 's is,

$$\begin{aligned} \frac{1}{K \cdot R} \sum_{s,r} p_{s,r} &\leq \frac{1}{K \cdot R} \sum_{s,r} (\sqrt{q_s} + |P_{s,r} U_{s,r}(I - Q_s) |\phi_0\rangle|)^2 \\ &= \frac{1}{K \cdot R} \sum_{s,r} \left(q_s + |P_{s,r} U_{s,r}(I - Q_s) |\phi_0\rangle|^2 + 2\sqrt{q_s} |P_{s,r} U_{s,r}(I - Q_s) |\phi_0\rangle| \right) \\ &\leq \frac{g(T+1)}{K} + \delta'(T) + \frac{2}{K \cdot R} \sum_{s,r} \sqrt{q_s} \cdot 1 \\ &\leq \frac{g(T+1)}{K} + \delta'(T) + 2\sqrt{\frac{g(T+1)}{K}} \\ &\leq \delta'(T) + 3\sqrt{\frac{g(T+1)}{K}}, \end{aligned}$$

The last inequality holds assuming $g(T+1)/K \leq 1$, since otherwise (5) also trivially holds. Therefore, using Lemma 3.14, for any (g, T) algorithm in the QROM, the succeeding probability is at most $\left(\delta'(T) + 3\sqrt{\frac{g(T+1)}{K}} \right)^g$. \square

7.2 Hardness of Salted Collision-Resistant Hash Functions

Definition 7.7 (Collision-Resistant Hash Security Game). *Security game $G_{\text{CRH}} = (C_{\text{CRH}})$ is specified by three procedures (Samp, Query, Ver), where:*

1. $\text{Samp}^H(\perp) = \perp$.
2. $\text{Query}^H(\perp, \cdot) = H(\cdot)$ provides query access to H .
3. $\text{Ver}^H(\perp, (x_1, x_2))$ outputs 1 if and only if $x_1 \neq x_2$ and $H(x_1) = H(x_2)$.

It is easy to see that $S = 2 \log M$ suffices to break this security game with optimal probability. However, without advice, the problem is hard even against quantum computations.

Proposition 7.8 ([Zha19, Corollary 2]). G_{CRH} has security $\Theta(T^3/M)$ in QROM.

It is easy to verify that this game is publicly-verifiable, as $\text{Ver}^H(\perp, (x_1, x_2)) = \text{Ver}^{\text{Query}^H}(\perp, (x_1, x_2))$. Denote the salted collision-resistant hash game as $G_{\text{CRH},S}$. Combining this lemma with Theorem 7.3, and the fact that random guessing has winning probability $1/M$, we obtain the following corollary.

Corollary 7.9. $G_{\text{CRH},S}$ is $O(T^3/M + (S + \log M)T/K)$ -secure in the AI-QROM, and $\tilde{O}(T^3/M + ST/K)^{1/4}$ -secure in the QAI-QROM, where \tilde{O} absorbs $\text{poly}(\log N, \log M, \log K, \log S)$ factors.

We now proceed to show the implication of this corollary in complexity theory for function problems.

Definition 7.10 (Polynomial Weak Pigeonhole Principle). A binary relation $P(x, y)$ is in PWPP if and only if P is Karp-reducible to $\text{COLLISION}(\mathcal{C}, (x_1, x_2))$, where \mathcal{C} is a circuit with n inputs and $m < n$ outputs, and $(\mathcal{C}, (x_1, x_2))$ is in COLLISION if and only if $\mathcal{C}(x_1) = \mathcal{C}(x_2)$.

Definition 7.11 (Function Bounded-error Quantum Polynomial time with Polynomial-size Quantum advice). A binary relation $P(x, y)$ is in FBQP/qpoly if and only if there exists a polynomial-time uniform family of quantum circuits $\{C_n\}_{n \in \mathbb{N}}$ such that for any x , there exists some quantum state ρ of dimension $2^{\text{poly}(|x|)}$, such that $\Pr[P(x, C_{|x|}(\rho, x))] \geq 2/3$.

Theorem 7.12. Relative to a random oracle \mathcal{O} , $\text{PWPP}^{\mathcal{O}} \not\subseteq \text{FBQP}^{\mathcal{O}}/\text{qpoly}$.

Proof. For any integer n , let $\mathcal{O}_n : [2^n] \times [2^{n+1}] \mapsto [2^n]$ be a random oracle. Consider the relation $P^{\mathcal{O}}(k, (x_1, x_2))$ that corresponds to finding collision in $\mathcal{O}(k, \cdot)$ (which is a circuit consisting of one oracle gate) for any $k \in [2^n]$ is a problem in $\text{PWPP}^{\mathcal{O}}$ as $2^{n+1} > 2^n$, however, by the corollary above, any quantum polynomial-time algorithm with polynomial-size advice can succeed with probability $\exp(-\Omega(n))$ over a random k , in particular, this implies that there exists some hard instance k such that the algorithm cannot succeed with non-negligible probability, and therefore proving that this problem is not in $\text{FBQP}^{\mathcal{O}}/\text{qpoly}$. \square

7.3 Tightness of Salted Game Lower Bound

In this subsection, we establish the tightness of Theorem 7.3. In particular, we show that the bound is asymptotically tight up to poly-logarithmic factors for the following simple game.

Definition 7.13 (Prediction Security Game). Let H be a random oracle $[1] \rightarrow [M]$, i.e. $N = 1$. The security game $G_{\text{YB0}} = (C_{\text{YB0}})$ is specified by three procedures (Samp, Query, Ver), where:

1. $\text{Samp}^H(\perp) = \perp$.
2. $\text{Query}^H(\perp, \cdot) = \perp$.
3. $\text{Ver}^H(\perp, y)$ outputs 1 if and only if $H(1) = y$.

For this game, the adversary is asked to predict $H(1)$, and therefore its security $\delta'(T) = 1/M$. Intuitively, the salted prediction game, denoted as $G_{\text{YB0},S}$ is simply Yao's box over a large alphabet $[M]$. Combining this with Theorem 7.3, we immediately obtain the following corollary.

Corollary 7.14. $G_{\text{YB0},S}$ is $\delta(S, T) = O(1/M + (S + \log M)T/K)$ -secure in the AI-QROM.

Proposition 7.15. *For any S, T such that $S(T + 1) \leq K \log M$, there exists an (S, T) -adversary in the AI-QROM that breaks G_{YB0} with winning probability at least $\Omega(1/M + S(T + 1)/(K \log M))$.*

Proof. For the purpose of this proof, we will instead view $H_S : [K] \mapsto \mathbb{Z}/M\mathbb{Z}$ as $[K] \times [1] \simeq [K]$ and $[M] \simeq \mathbb{Z}/M\mathbb{Z}$. In the pre-processing stage, the algorithm stores the following $S/\log M$ elements in $\mathbb{Z}/M\mathbb{Z}$: for the j -th element sum_j ,

$$\text{sum}_j = \sum_{k=1}^{T+1} H(j(T + 1) + k), \quad \forall j = 0, 1, \dots, S/\log M - 1.$$

In the online state, given a salt $s \in [K]$, if $s \notin \{1, \dots, S(T + 1)/\log M\}$, the algorithm randomly guesses a y . Otherwise, let j be the unique integer such that $j(T + 1) < s \leq (j + 1)(T + 1)$. The algorithm queries $H(s')$ for all $s' \in \{j(T + 1) + 1, \dots, (j + 1)(T + 1)\}$, $s' \neq s$, and subtract them from sum_j and output the result, which succeeds with probability 1. \square

Acknowledgements

Kai-Min Chung is partially supported by the Academia Sinica Career Development Award under Grant no. 23-17, and MOST QC project under Grant no. MOST 108-2627-E-002-001-.

Siyao Guo is supported by Shanghai Eastern Young Scholar Program.

Qipeng Liu is supported in part by NSF and DARPA. Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF or DARPA.

Luowen Qian is supported by the DARPA SIEVE program.

References

- [Aar05] Scott Aaronson. “Limitations of Quantum Advice and One-Way Communication”. In: *Theory of Computing* 1.1 (2005), pp. 1–28. DOI: 10.4086/toc.2005.v001a001. URL: <https://doi.org/10.4086/toc.2005.v001a001>.
- [Aar18] Scott Aaronson. “Shadow tomography of quantum states”. In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. 2018, pp. 325–338.
- [AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. “Quantum security proofs using semi-classical oracles”. In: *Annual International Cryptology Conference*. Springer. 2019, pp. 269–295.
- [AR19] Scott Aaronson and Guy N. Rothblum. “Gentle measurement of quantum states and differential privacy”. In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*. Ed. by Moses Charikar and Edith Cohen. ACM, 2019, pp. 322–333. DOI: 10.1145/3313276.3316378. URL: <https://doi.org/10.1145/3313276.3316378>.
- [Arc20] Scott Arciszewski. *XChaCha: eXtended-nonce ChaCha and AEAD_XChaCha20_Poly1305*. Internet-Draft draft-irtf-cfrg-xchacha-03. Work in Progress. Internet Engineering Task Force, Jan. 2020. 18 pp. URL: <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-xchacha-03>.

- [AS04] Scott Aaronson and Yaoyun Shi. “Quantum lower bounds for the collision and the element distinctness problems”. In: *Journal of the ACM (JACM)* 51.4 (2004), pp. 595–605.
- [BBBV97] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. “Strengths and weaknesses of quantum computing”. In: *SIAM journal on Computing* 26.5 (1997), pp. 1510–1523.
- [BDF⁺11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. “Random Oracles in a Quantum World”. In: *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. Lecture Notes in Computer Science. Springer, 2011, pp. 41–69. DOI: 10.1007/978-3-642-25385-0\3. URL: https://doi.org/10.1007/978-3-642-25385-0%5C_3.
- [Bea91] Paul Beame. “A General Sequential Time-Space Tradeoff for Finding Unique Elements”. In: *SIAM J. Comput.* 20.2 (1991), pp. 270–277. DOI: 10.1137/0220017. URL: <https://doi.org/10.1137/0220017>.
- [BR93] Mihir Bellare and Phillip Rogaway. “Random Oracles are Practical: A Paradigm for Designing Efficient Protocols”. In: *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993*. 1993, pp. 62–73. DOI: 10.1145/168588.168596. URL: <http://doi.acm.org/10.1145/168588.168596>.
- [BV97] Ethan Bernstein and Umesh V. Vazirani. “Quantum Complexity Theory”. In: *SIAM J. Comput.* 26.5 (1997), pp. 1411–1473. DOI: 10.1137/S0097539796300921. URL: <https://doi.org/10.1137/S0097539796300921>.
- [BZ13] Dan Boneh and Mark Zhandry. “Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World”. In: *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8043. Lecture Notes in Computer Science. Springer, 2013, pp. 361–379. DOI: 10.1007/978-3-642-40084-1\21. URL: https://doi.org/10.1007/978-3-642-40084-1%5C_21.
- [CDG18] Sandro Coretti, Yevgeniy Dodis, and Siyao Guo. “Non-Uniform Bounds in the Random-Permutation, Ideal-Cipher, and Generic-Group Models”. In: *Annual International Cryptology Conference*. Springer, 2018, pp. 693–721.
- [CDGS18] Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John P. Steinberger. “Random Oracles and Non-uniformity”. In: *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10820. Lecture Notes in Computer Science. Springer, 2018, pp. 227–258. DOI: 10.1007/978-3-319-78381-9\9. URL: https://doi.org/10.1007/978-3-319-78381-9%5C_9.
- [CK19] Henry Corrigan-Gibbs and Dmitry Kogan. “The function-inversion problem: Barriers and opportunities”. In: *Theory of Cryptography Conference*. Springer, 2019, pp. 393–421.

- [CLQ19] Kai-Min Chung, Tai-Ning Liao, and Luowen Qian. “Lower Bounds for Function Inversion with Quantum Advice”. In: *arXiv preprint arXiv:1911.09176* (2019).
- [DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. “Security of the fiat-shamir transformation in the quantum random-oracle model”. In: *arXiv preprint arXiv:1902.07556* (2019).
- [DGK17] Yevgeniy Dodis, Siyao Guo, and Jonathan Katz. “Fixing Cracks in the Concrete: Random Oracles with Auxiliary Input, Revisited”. In: *Advances in Cryptology - EURO-CRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10211. Lecture Notes in Computer Science. 2017, pp. 473–495. DOI: 10.1007/978-3-319-56614-6_16. URL: https://doi.org/10.1007/978-3-319-56614-6_16.
- [DTT10] Anindya De, Luca Trevisan, and Madhur Tulsiani. “Time Space Tradeoffs for Attacks against One-Way Functions and PRGs”. In: *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*. 2010, pp. 649–665. DOI: 10.1007/978-3-642-14623-7_35. URL: http://dx.doi.org/10.1007/978-3-642-14623-7_35.
- [FN99] Amos Fiat and Moni Naor. “Rigorous Time/Space Trade-offs for Inverting Functions”. In: *SIAM J. Comput.* 29.3 (1999), pp. 790–803. DOI: 10.1137/S0097539795280512. URL: <http://dx.doi.org/10.1137/S0097539795280512>.
- [GGH⁺19] Alexander Golovnev, Siyao Guo, Thibaut Horel, Sunoo Park, and Vinod Vaikuntanathan. “3SUM with Preprocessing: Algorithms, Lower Bounds and Cryptographic Applications”. In: *CoRR* abs/1907.08355 (2019). arXiv: 1907.08355. URL: <http://arxiv.org/abs/1907.08355>.
- [GKL93] Oded Goldreich, Hugo Krawczyk, and Michael Luby. “On the existence of pseudorandom generators”. In: *SIAM Journal on Computing* 22.6 (1993), pp. 1163–1175.
- [Gro96] Lov K Grover. “A fast quantum mechanical algorithm for database search”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219.
- [Hel80] Martin Hellman. “A cryptanalytic time-memory trade-off”. In: *IEEE transactions on Information Theory* 26.4 (1980), pp. 401–406.
- [HM20] Yassine Hamoudi and Frédéric Magniez. “Quantum Time-Space Tradeoffs by Recording Queries”. In: *arXiv preprint arXiv:2002.08944* (2020).
- [HXY19] Minki Hhan, Keita Xagawa, and Takashi Yamakawa. “Quantum Random Oracle Model with Auxiliary Input”. In: *AsiaCrypt*. Springer. 2019.
- [Imp11] Russell Impagliazzo. “Relativized separations of worst-case and average-case complexities for NP”. In: *2011 IEEE 26th Annual Conference on Computational Complexity*. IEEE. 2011, pp. 104–114.
- [Kla03] Hartmut Klauck. “Quantum time-space tradeoffs for sorting”. In: *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. 2003, pp. 69–76.

- [KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. “A Concrete Treatment of Fiat-Shamir Signatures in the Quantum Random-Oracle Model”. In: *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10822. Lecture Notes in Computer Science. Springer, 2018, pp. 552–586. DOI: 10.1007/978-3-319-78372-7_18. URL: https://doi.org/10.1007/978-3-319-78372-7%5C_18.
- [KP19] Tsvi Kopelowitz and Ely Porat. “The Strong 3SUM-INDEXING Conjecture is False”. In: *CoRR* abs/1907.11206 (2019). arXiv: 1907.11206. URL: <http://arxiv.org/abs/1907.11206>.
- [KŠW07] Hartmut Klauck, Robert Špalek, and Ronald de Wolf. “Quantum and classical strong direct product theorems and optimal time-space tradeoffs”. In: *SIAM Journal on Computing* 36.5 (2007), pp. 1472–1493.
- [Lev87] Leonid A. Levin. “One-way functions and pseudorandom generators”. In: *Combinatorica* 7.4 (1987), pp. 357–363. DOI: 10.1007/BF02579323. URL: <https://doi.org/10.1007/BF02579323>.
- [LR10] Gregory M. Lebovitz and Eric Rescorla. “Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)”. In: *RFC* 5926 (2010), pp. 1–15. DOI: 10.17487/RFC5926. URL: <https://doi.org/10.17487/RFC5926>.
- [LZ19a] Qipeng Liu and Mark Zhandry. “On finding quantum multi-collisions”. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2019, pp. 189–218.
- [LZ19b] Qipeng Liu and Mark Zhandry. “Revisiting Post-Quantum Fiat-Shamir.” In: *IACR Cryptology ePrint Archive* 2019 (2019), p. 262.
- [MW19] Or Meir and Avi Wigderson. “Prediction from Partial Information and Hindsight, with Application to Circuit Lower Bounds”. In: *Comput. Complex.* 28.2 (2019), pp. 145–183. DOI: 10.1007/s00037-019-00177-4. URL: <https://doi.org/10.1007/s00037-019-00177-4>.
- [NABT14] Aran Nayebi, Scott Aaronson, Aleksandrs Belovs, and Luca Trevisan. “Quantum lower bound for inverting a permutation with advice”. In: *CoRR* abs/1408.3193 (2014). arXiv: 1408.3193. URL: <http://arxiv.org/abs/1408.3193>.
- [NABT15] Aran Nayebi, Scott Aaronson, Aleksandrs Belovs, and Luca Trevisan. “Quantum lower bound for inverting a permutation with advice”. In: *Quantum Information & Computation* 15.11&12 (2015), pp. 901–913. URL: <http://www.rintonpress.com/xxqic15/qic-15-1112/0901-0913.pdf>.
- [ST18] Alexander V. Smal and Navid Talebanfard. “Prediction from partial information and hindsight, an alternative proof”. In: *Inf. Process. Lett.* 136 (2018), pp. 102–104. DOI: 10.1016/j.ipl.2018.04.011. URL: <https://doi.org/10.1016/j.ipl.2018.04.011>.

- [TU16] Ehsan Ebrahimi Targhi and Dominique Unruh. “Post-Quantum Security of the Fujisaki-Okamoto and OAEP Transforms”. In: *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*. Ed. by Martin Hirt and Adam D. Smith. Vol. 9986. Lecture Notes in Computer Science. 2016, pp. 192–216. DOI: 10.1007/978-3-662-53644-5_8. URL: https://doi.org/10.1007/978-3-662-53644-5%5C_8.
- [Unr07] Dominique Unruh. “Random Oracles and Auxiliary Input”. In: *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*. Ed. by Alfred Menezes. Vol. 4622. Lecture Notes in Computer Science. Springer, 2007, pp. 205–223. DOI: 10.1007/978-3-540-74143-5_12. URL: https://doi.org/10.1007/978-3-540-74143-5%5C_12.
- [Unr15] Dominique Unruh. “Non-Interactive Zero-Knowledge Proofs in the Quantum Random Oracle Model”. In: *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9057. Lecture Notes in Computer Science. Springer, 2015, pp. 755–784. DOI: 10.1007/978-3-662-46803-6_25. URL: https://doi.org/10.1007/978-3-662-46803-6%5C_25.
- [Unr17] Dominique Unruh. “Post-quantum Security of Fiat-Shamir”. In: *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10624. Lecture Notes in Computer Science. Springer, 2017, pp. 65–95. DOI: 10.1007/978-3-319-70694-8_3. URL: https://doi.org/10.1007/978-3-319-70694-8%5C_3.
- [Yao90] AC-C Yao. “Coherent functions and program checkers”. In: *Proceedings of the twenty-second annual ACM symposium on Theory of computing*. 1990, pp. 84–94.
- [Zha12] Mark Zhandry. “Secure Identity-Based Encryption in the Quantum Random Oracle Model”. In: *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. Vol. 7417. Lecture Notes in Computer Science. Springer, 2012, pp. 758–775. DOI: 10.1007/978-3-642-32009-5_44. URL: https://doi.org/10.1007/978-3-642-32009-5%5C_44.
- [Zha19] Mark Zhandry. “How to record quantum queries, and applications to quantum indistinguishability”. In: *Annual International Cryptology Conference*. Springer. 2019, pp. 239–268.

A Extended Preliminaries for Compressed Oracle

Lemma 2.3. *Let \mathcal{A} be a quantum algorithm making at most T queries to a phase oracle. The overall state of \mathcal{A} and the phase oracle can be written as $\sum_{z,D:|D|\leq T} \alpha_{z,D} |z\rangle \otimes \frac{1}{M^{N/2}} \sum_H \omega_M^{(D,H)} |H\rangle = \sum_{z,D:|D|\leq T} \alpha_{z,D} |z\rangle \otimes |\phi_D\rangle$.*

Moreover, it is true even if the state is conditioned on arbitrary outcomes (with non-zero probability) of \mathcal{A} ’s intermediate measurements.

Proof. We begin by assuming that there are no intermediate measurements. Note that if \mathcal{A} makes some local quantum computation $U \otimes I$ on his on register $|z\rangle$, it does not affect the statement, as

$$(U \otimes I) \sum_{z,D:|D|\leq T} \alpha_{z,D} |z\rangle |\phi_D\rangle = \sum_{D:|D|\leq T} \left(U \sum_z \alpha_{z,D} |z\rangle \right) |\phi_D\rangle.$$

We prove the lemma by induction. The base case is before \mathcal{A} makes any query. The overall state by definition is $\sum_z \alpha_z |z\rangle \otimes \frac{1}{M^{N/2}} \sum_H |H\rangle$. If we view the register $|H\rangle$ under the Fourier basis $|\phi_D\rangle$, there is only non-zero amplitude on D_0 which is an all-zero vector.

If \mathcal{A} makes an oracle query to phase oracle, using the “phase kickback” trick,

$$\begin{aligned} & \text{PhO} \sum_{\substack{z=(x,u,\text{aux}) \\ D:|D|\leq T}} \alpha_{z,D} |x, u, \text{aux}\rangle \frac{1}{M^{N/2}} \sum_H \omega_M^{\langle D, H \rangle} |H\rangle \\ &= \sum_{\substack{z=(x,u,\text{aux}) \\ D:|D|\leq T}} \alpha_{z,D} |x, u, \text{aux}\rangle \frac{1}{M^{N/2}} \sum_H \omega_M^{\langle D, H \rangle + uH(x)} |H\rangle \\ &= \sum_{\substack{z=(x,u,\text{aux}) \\ D:|D|\leq T}} \alpha_{z,D} |x, u, \text{aux}\rangle |\phi_{D \oplus (x,u)}\rangle, \end{aligned}$$

where $D \oplus (x, u)$ is defined as a new vector that is almost identical to D but the x -th entry is updated to $D(x) + u$. Therefore, it is easy to see that $|D \oplus (x, u)| \leq |D| + 1$ for all D , which implies that the resulting state has only non-zero amplitude on $|z\rangle |D'\rangle$ whose $|D'| \leq T + 1$. Using induction, we reach our conclusion for the case where there are no intermediate measurements.

Now consider the case where \mathcal{A} makes intermediate measurements. Without loss of generality, we can delay all the measurements to the last step. Before making these measurement, we know the overall state can be written as

$$\sum_{z, b_1, \dots, b_k, D:|D|\leq T} \alpha_{z, b_1, \dots, b_k, D} |z\rangle |b_1, \dots, b_k\rangle \otimes |\phi_D\rangle,$$

where each $|b_i\rangle$ is going to be measured by \mathcal{A} . Therefore, it is easy to see that conditioned on any outcomes of the measurements (of non-zero probability), the resulting state has only non-zero amplitude on $|z\rangle |D\rangle$ whose $|D| \leq T$. \square

Lemma 2.6. *If \mathcal{A} makes at most T queries to a compressed standard oracle, assuming the overall state of \mathcal{A} and the compressed standard oracle is $\sum_{z,D} \alpha_{z,D} |z\rangle_{\mathcal{A}} |D\rangle_H$, then it has support on all D such that $|D| \leq T$. In other words, the overall state can be written as,*

$$\sum_{z,D:|D|\leq T} \alpha_{z,D} |z\rangle_{\mathcal{A}} \otimes |D\rangle_H.$$

Moreover, it is true even if the state is conditioned on arbitrary outcomes (with non-zero probability) of \mathcal{A} 's intermediate measurements.

Proof. First, \mathcal{A} applies a local unitary/measurement over its register. Since the operation does not operate on D , it cannot change the probability distribution of the outcomes on measuring D . As a consequence, this operation cannot increase $\max_{\alpha_{z,D} > 0} |D|$.

Second, \mathcal{A} makes an oracle query on state $|x\rangle|u\rangle$. We claim that $\text{CStO}|x\rangle|u\rangle|D\rangle$ is a state with support on database D' such that $|D'| \leq |D| + 1$, which is implicitly stated in the proof for Lemma 4, Hybrid 4 in [Zha19], and can also be directly derived from the definition of CStO and StdDecomp . We reach our conclusion by using an induction if there are no intermediate measurements.

Finally, \mathcal{A} makes intermediate measurements. Without loss of generality, all these measurements are delayed to the last step. Before making these measurement, by the above discussion, the overall state can be written as

$$\sum_{z, b_1, \dots, b_k, D: |D| \leq T} \alpha_{z, b_1, \dots, b_k, D} |z\rangle |b_1, \dots, b_k\rangle \otimes |D\rangle$$

where each $|b_i\rangle$ is going to be measured by \mathcal{A} . Therefore, it is easy to see that conditioned on any outcomes of the measurements (of non-zero probability), the resulting state has only non-zero amplitude on $|z\rangle|D\rangle$ whose $|D| \leq T$. \square

Lemma 2.7 ([Zha19, Lemma 5]). *Let H be a random oracle from $[N] \rightarrow [M]$. Consider a quantum algorithm \mathcal{A} making queries to the standard oracle and outputting tuples $(x_1, \dots, x_k, y_1, \dots, y_k, z)$. Supposed the random function H is measured after \mathcal{A} produces its output. Let R be an arbitrary set of such tuples. Suppose with probability p , \mathcal{A} outputs a tuple such that (1) the tuple is in R and (2) $H(x_i) = y_i$ for all i . Now consider running \mathcal{A} with the compressed standard oracle CStO , and supposed the database D is measured after \mathcal{A} produces its output. Let p' be the probability that (1) the tuple is in R and (2) $D(x_i) = y_i$ (in particular, $D(x_i) \neq \perp$) for all i . Then $\sqrt{p} \leq \sqrt{p'} + \sqrt{k/M}$.*

Moreover, it is true even if it is conditioned on arbitrary outcomes (with non-zero probability) of \mathcal{A} 's intermediate measurements.

Proof. Without conditioning on intermediate measurements, this is the exact statement of [Zha19, Lemma 5]. Observe that their proof works as long as the final state of the algorithm and the oracle can be written as,

$$\sum_{\substack{\mathbf{x}, \mathbf{y}, z, D \\ \forall i, D(x_i) = \perp}} \sum_{I \subseteq [k]} \sum_{\mathbf{r}_I \in \{1, \dots, M-1\}^{|I|}} \alpha_{\mathbf{x}, \mathbf{y}, z, D, \mathbf{r}_I} \frac{1}{\sqrt{M^{|I|}}} \sum_{\mathbf{y}'_I \in (\mathbb{Z}/M\mathbb{Z})^{|I|}} \omega_M^{\langle \mathbf{y}'_I, \mathbf{r}_I \rangle} |\mathbf{x}, \mathbf{y}, z, D \cup (\mathbf{x}_I, \mathbf{y}'_I)\rangle, \quad (8)$$

where $\mathbf{x} \in [N]^k$, $\mathbf{y} \in (\mathbb{Z}/M\mathbb{Z})^k$. That is, if $D(x) \neq \perp$, then the superposition of $D(x)$ decomposed in the Fourier basis does not contain a uniform superposition (corresponding to $\mathbf{r}_I \in \{1, \dots, M-1\}^{|I|}$ in the formula, or in other words, no 0 entry).

Zhandry showed the final state satisfies the condition above. It is indeed easy to prove. Recall that $\text{CStO} = \text{StdDecomp} \circ \text{CStO}' \circ \text{StdDecomp}$. As the second StdDecomp will map any uniform $\sum_y |y\rangle$ to $|\perp\rangle$, the decomposition in the Fourier basis does not contain uniform superposition.

Now consider the case of having intermediate measurements. We claim that the state has the above form, conditioned on arbitrary outcomes (with non-zero probability) of \mathcal{A} 's intermediate measurements. Without loss of generality, we defer all measurements until the end of the computation. Before making these measurements, we know the overall state can be written as

$$\sum_{\substack{\mathbf{x}, \mathbf{y}, z, \omega, D \\ \forall i, D(x_i) = \perp}} \sum_{I \subseteq [k]} \sum_{\mathbf{r}_I \in \{1, \dots, M-1\}^{|I|}} \alpha_{\mathbf{x}, \mathbf{y}, z, \omega, D, \mathbf{r}_I} \frac{1}{\sqrt{M^{|I|}}} \sum_{\mathbf{y}'_I \in (\mathbb{Z}/M\mathbb{Z})^{|I|}} \omega_M^{\langle \mathbf{y}'_I, \mathbf{r}_I \rangle} |\mathbf{x}, \mathbf{y}, z, \omega, D \cup (\mathbf{x}_I, \mathbf{y}'_I)\rangle,$$

where $\omega \in \Omega$ is the register to be measured. It is easy to see that after conditioning on $\omega \in E$ where $E \subseteq \Omega$ being an arbitrary event, the final state can still be written as (8), as conditioning simply zeroes out some α 's for every $\omega \notin E$ and re-normalizes. The rest of the proof simply follows Zhandry's proof for [Zha19, Lemma 5]. \square