# On Quantum Query Complexities of Collision-Finding in Non-Uniform Random Functions[⋆]

Tianci Peng, Shujiao Cao, and Rui Xue

State Key Laboratory of Information Security, Institute of Information Engineering,
Chinese Academy of Sciences, Beijing 100093, China,
School of Cyber Security, University of CAS, Beijing 100049, China
{pengtianci, caoshujiao, xuerui}@iie.ac.cn

**Abstract.** Collision resistance and collision finding are now extensively exploited in Cryptography, especially in the case of quantum computing. For any function $f : [M] \to [N]$ with $f(x)$ uniformly distributed over $[N]$, Zhandry has shown that the number $\Theta(N^{1/3})$ of queries is both necessary and sufficient for finding a collision in $f$ with constant probability. However, there is still a gap between the upper and the lower bounds of query complexity in general non-uniform distributions.

In this paper, we investigate the quantum query complexity of collision-finding problem with respect to general non-uniform distributions. Inspired by previous work, we pose the concept of collision domain and a new parameter $\gamma$ that heavily depends on the underlying non-uniform distribution. We then present a quantum algorithm that uses $O(\gamma^{1/6})$ quantum queries to find a collision for any non-uniform random function. By making a transformation of a problem in non-uniform setting into a problem in uniform setting, we are also able to show that $\Omega(\gamma^{1/6} \log^{-1/2} \gamma)$ quantum queries are necessary in collision-finding in any non-uniform random function.

The upper bound and the lower bound in this work indicates that the proposed algorithm is nearly optimal with query complexity in general non-uniform case.

**Keywords:** Quantum, Query complexity, Collision-finding algorithm, Compressed oracle technique, Non-uniform distribution, Lower bound

## 1 Introduction

The quantum computation has brought threats to classical cryptography since Shor's seminal article [25]. In the black-box model, the advantage of quantum computing embodied the fact that a quantum adversary may take input or output in the form of a quantum superposition, which potentially allows to gain advantages that might not be possible in traditional computations. As a result,

[⋆] This work is supported by National Natural Science Foundation of China (No.61772514, 62172405).

many classical public key encryption systems, including Diffie-Hellman protocol [12] and RSA cryptosystem [24], are broken by Shor's factoring and discrete-log algorithms [25]. The Grover's algorithm [18] allows greatly to improve the efficiency of the adversary by accelerating the speed for solving the search problems. Many cryptographic schemes that are secure in classical computation may no longer be applicable to the quantum world [6, 13, 19, 21].

Faced with the threats from quantum computing, people widely investigate, as in classical computation, the complexity of quantum computing. One expects to figure out, to each problem, the problem-solving capabilities and limitations of quantum computing by considering the number of queries in the black-box model.

In this work, we manage to further explore both upper and lower bound of quantum query complexity of collision-finding problem in generally non-uniform random functions.

Collision resistance is one of important properties in cryptography. For any positive integers $M, N$, let $[M]$ be the set $\{1, \ldots, M\}$. A collision to a function $f$ from $[M]$ to $[N]$ is a couple of distinct inputs $x_1, x_2 \in [M]$ such that $f(x_1) = f(x_2)$. The collision-resistant hash functions in cryptography emphasize the difficulty in finding collisions. They are broadly employed in various cryptographic primitives [8, 9, 15].

It has been well studied on the complexity of finding a collision in quantum setting. In view of the upper bounds, Brassard et al. [11] showed that $O(N^{1/3})$ queries are sufficient to find a collision in a two-to-one function. Ambainis [3] proved that $O(M^{2/3})$ quantum queries are sufficient to achieve a collision with constant probability in a function $f : [M] \to [N]$ by quantum walk, in which $f$ should be guaranteed to have at least one pair of collisions.

Zhandry [30] proposed an algorithm that takes $O(N^{1/3})$ quantum queries to find a collision in a uniformly random function. Targhi et al. [14] and Balogh et al. [7] proved, separately, that $O(2^{\beta/3})$ quantum queries are sufficient to find a collision in a non-uniform random function, where $\beta$ is the collision-entropy of distribution $D$ (cf. Definition 1.).

In terms of lower bounds, Aaronson and Shi [1] proved that $\Omega(N^{1/3})$ quantum query is necessary to find a collision in any function $f : [N] \to [N]$, where $f$ is a two-to-one function. Which has been further extended to the case of small range by Kutin [22] and Ambainis [4] independently. Yuen [28] proved that $\Omega(N^{1/5}/\texttt{poly}\log N)$ quantum queries are necessary in finding a collision in a uniformly random function $f : [N] \to [N]$. Zhandry [30] improved this bound further to $\Omega(N^{1/3})$ in a uniform function from $[M]$ to $[N]$.

Targhi et al. proved a lower bound $\Omega(2^{\mathtt{k}/9})$ in a non-uniform random function [26], in the case that for any $x \in [M]$ the output of $f(x)$ is selected according to a distribution $D$ over $[N]$, which possesses the min-entropy $\mathtt{k}$. Targhi and Unruh improved the previous lower bound to $\Omega(2^{\mathtt{k}/5})$ [14]. Balogh et al. [7] recently improved that lower bound to $\Omega(2^{\mathtt{k}/3})$.

One should note that Zhandry's results claim that, in a uniform random function $f$ from $[M]$ to $[N]$, that $\Theta(N^{1/3})$ quantum queries are both necessary and sufficient to find a collision [30], and thus give the tight upper and lower bounds in uniform case.

In the research of collision-finding problem with respect to general non-uniform distributions, however, from the reviews above and also the summarization in Table 1, we see that there lacks currently a tight upper and lower query complexity bound characterizing as in uniform settings.

**Table 1.** Recent complexity results to collision-finding problem

| Literatures | Distribution | Upper Bound | Lower Bound |
|---|---|---|---|
| [Zha15] [30] | uniform | $O(N^{1/3})$ | $\Omega(N^{1/3})$ |
| [TTU16] [26] | non-uniform |  | $\Omega(2^{\mathrm{k}/9})$ |
| [EU18] [14] | non-uniform | $O(\min\{2^{\beta/3}, 2^{\mathrm{k}/2}\})$ | $\Omega(\max\{2^{\beta/9}, 2^{\mathrm{k}/5}\})$ |
| [BES18] [7] | non-uniform | $O(\min\{2^{\beta/3}, 2^{\mathrm{k}/2}\})$ | $\Omega(\max\{2^{\beta/6}, 2^{\mathrm{k}/3}\})$ |

Existing algorithms in this setting, for example in [7], gain optimum bounds valid to some special distributions, as is pointed out later in the context, not in generally non-uniform distributions. Hence, exploiting further characterization of query complexity of collision-finding in non-uniform functions is deserved from theoretical point of view as well as in practical applications in cryptography. For example, non-uniform functions were used in cryptographic systems such as the famous Fujisaki-Okamoto construction [16] and further discussions in [5, 27].

This work aims to work out an (almost) optimal upper and lower bounds on numbers of quantum queries for collision-finding problem in general non-uniform random functions.

### 1.1 Contributions

In this work, we firstly propose a collision-finding algorithm and analyze its quantum query complexity, and then show a quantum query complexity lower bound in any non-uniform function. Both the upper bound and the lower bound are characterized in a new proposed collision parameter, and finally result in tighter bounds.

In [14], Targhi et al. showed that, provided that only min-entropy is used in describing the collision-finding complexity in non-uniform random functions, the upper bound $O(2^{\mathrm{k}/2})$ and the lower bound $\Omega(2^{\mathrm{k}/3})$ are both the best possible ones. The gap between the upper and lower bounds there hints us that min-entropy might not be good enough in describing query complexity of the collision-finding problem.

This leads us to seek more finer parameter that may reflect more properties of underlying distributions. In this work we propose a new collision parameter $\gamma$ in investigating the quantum query complexity of collision-finding in non-uniform random functions.

For any constant $c > 1$, we start with a partition, called $c$-partition which is somewhat similar as in [17], to divide the domain $[N]$ into a sequence of subsets

$\{S_1, \ldots, S_\ell\}$ according to the weight of distribution $D : \{p_1, \cdots, p_N\}$. To denote $p(S_i)$ as the maximal probability of elements in $S_i$ of size $n_i$, then $\gamma(c)$ is defined to be the minimum one among all $1/n_i p^3(S_i)$. Although the value of $\gamma(c)$ is dependent on $c$, but later in Proposition 7, we show that the constant $c$ does not affect the magnitude of query complexity, and hence we may mention $\gamma$ now and then rather than $\gamma(c)$.

In terms of this parameter, we are able to compose a new algorithm which succeeds in collision-finding with $O(\gamma^{1/6})$ quantum queries and also prove the number $\Omega(\gamma^{1/6}/\sqrt{\log \gamma})$ of quantum queries are necessary for any collision-finding algorithms over non-uniform distributions. Since the two bounds almost meet, we believe parameter $\gamma$ is an appropriate one in describing the query complexity of collision-finding.

We should point out that the proposed parameter in calculation, as the collision-entropy does (cf. Definition 1), will need more information about the distribution than the min-entropy does. Indeed, before running the proposed collision-finding algorithm, it needs to find the right collision domain $S_i$ at first, which is somewhat different from existing algorithms which only taking the min-entropy as input. This is based on the observation that the key to improve the acceleration in algorithm is to find the most suitable collision search range (which is called collision domain in the context, cf. Definition 3), and that should be inevitable to make use of the information of underlying distribution. This might also be understood as a trade-off between getting a finer algorithm and using neat property like the min-entropy.

This requirement is often implicitly assumed (or by default) in practice. An example is in the CCA quantum security proof to the famous Fujisaki-Okamoto transformation [5,27]. In the proof there, one has to ensure that it is hard for any polynomial quantum algorithm to find collisions $(c, c')$ in non-uniform random function $g = f \circ H = f(\delta, H(\delta, c))$, where $H$ is a uniform random function and $f$ is the encryption algorithm in the asymmetric encryption scheme. An adversary not only has oracle access to $f$, but also know the underlying structure of the encryption algorithm (for example, the adversary knows in advance that $f$ is actually the ElGamal encryption algorithm, which means the construction of this scheme is based on DDH assumptions). Which means, the adversary is able to analyze the distribution of pre-image of the encryption algorithm (namely for any $\delta \in \{0,1\}^m$ and $y \in \{0,1\}^n$, and to calculate $\Pr[y = f(\delta, H(\delta, c)) : c \leftarrow \texttt{coin}]$). That implies the adversary is capable and likely to know all the information about corresponding distribution of the non-uniform random function $g$.

In addition, the enhanced lower bound here can be used in the CCA security proof of FO transformation as the replacement of the corresponding component in the proof (such as Lemma 11, in full version of [5]), which will result a tighter reducible bound in the CCA security of FO transformation.

### 1.2 Technical Overview

Now we present the main ideas and techniques in this work.

**The parameter $\gamma$ and its properties.** A novel parameter $\gamma$ is proposed in this work, aiming at accurately describing the quantum computational complexity of collision-finding problem in general non-uniform random functions.

In the course of our research, we observe that there is a significant difference between quantum and classical computing in solving the collision-finding problem in the non-uniform case. Specifically, the acceleration effects in quantum computing in more restricted search scope may be performed better. This fact is counterintuitive to the classical setting. That means, in the classical case it is always easier to find any collision in a set than in any of its subsets.

Based on this observation, the key point to effectively find a collision is to seek the most suitable search range, which is referred as collision domain in this work. Given non-uniform distribution $D$ with weight sequence $\{p_1, \ldots, p_N\}$, assuming without lose of generality that $p_1 \geq p_2 \geq \cdots \geq p_N$, we choose a constant threshold $c > 1$ (which is proved not essential to the complexity and will be discussed later in Proposition 7), and partition domain $[D]$ accordingly into a series of subsets.

For any non-uniform distribution $D$ over $[N]$, we want to make a partition of $[N]$. For this purpose, a constant $c$ is chosen to control the size of partition, so that in each part $S$ of the partition, the ratio $p_j/p_i$ for any $i, j \in S$ does not exceed the threshold $c > 1$. The larger $c$ is, the fewer parts in partition there would be.

Given the threshold $c > 1$, we start to collect all index $j \in [N]$ satisfying $p_1 \geq p_j > p_1/c$ as a set $S_1$. Let $n_1 := |S_1|$ and $p(S_1) := p_1$. And then let $S_2$ be the collection of all $j \in [N]$ such that $p(S_2) \geq p_j > p(S_2)/c$, where $p(S_2) := p_{n_1+1}$. Continuing this process, it finally divides $[N]$ into a series of subsets $S_1, S_2, \ldots, S_\ell$ with $|S_i| = n_i$ for $i = 1, \ldots, \ell$. We name this partition as $c$-partition. If let $\gamma_i(c) := 1/n_i p^3(S_i)$ for each $i \in [\ell]$, the parameter $\gamma(c)$ is defined as the smallest one among $\gamma_i(c)$. Suppose $\gamma(c) = \gamma_{k_0}(c)$, the subset $S_{k_0}$ is named as the collision domain and $k_0$ is the smallest such index among $[\ell]$.

We give the relations of $\gamma(c)$ in Section 5 (see Proposition 4 and 5), with the min-entropy $\mathtt{k}$ and collision-entropy $\beta$ used in existing algorithms and show that both the upper bound and lower bound in this work are at least as better as the best prior result in [7]. To indicate superiority of the result in this work, we supply an example such that $\max\{\beta^{1/6}, 2^{\mathtt{k}/3}\} \ll \gamma^{1/6}(c) \ll \min\{\beta^{1/3}, 2^{\mathtt{k}/2}\}$.

The proposed collision-finding algorithm in this work will take the collision domain as input, comparing to previous algorithms which mainly take min-entropy or collision-entropy as inputs. To make the computation of collision domain simple in application, we show a result in Proposition 6 to reduce the actual range to locate the collision domain.

Although $\gamma(c)$ depends on the classification of discrete sequence with constant $c$, we are able to show, in Proposition 7, the relative error between $\gamma(c_1)$ and $\gamma(c_2)$ is $O(1)$ as long as $c_1, c_2 > 1$ are constants. That claims that the parameter $\gamma(c)$ is NOT affected in the magnitude of query complexity bounds, provided $c$ is a constant.

**The algorithm and complexity upper bound.** With the parameter $\gamma(c)$ defined as above, the algorithm is intuitively designed as follows. First, to use Grover' algorithm finding a list $L$ of elements in collision domain $S_{k_0}$, and then to adopt standard collision algorithm to find a collision in $L$. It uses adaptively adjusted search domain according to the underlying distribution to achieve acceleration, which has been missed in previous algorithms.

For this purpose, some changes to the BBHT algorithm must be made so as to be used in our algorithm. The BBHT algorithm in [10] gives the expected number of queries when the algorithm succeeds, while we expect to get a relation, depending on the number of queries, to indicate the success probability of the algorithm. The changes here are actually a de-randomized version of BBHT algorithm.

Finally, we show that proposed collision-finding algorithm makes $O(\gamma^{1/6}(c))$ queries succeeds with probability $\Omega(1)$. To some extent it implies that the parameter $\gamma(c)$ exploits the more significant information of a non-uniform distribution in collision-finding, resulting in a better acceleration.

**The lower bound.** Exploring the lower bound in terms of $\gamma(c)$ is a little bit involved. To obtain the lower bound of query complexity, we firstly estimate the success probability restricted over each $S_r$ (Problem 1 in Section 4.2), where $S_r$ is any subset partitioned in $c$-partition. Then, the success probability of the problem is bounded by the sum of the upper bounds of the success probabilities over all $S_r$. In such a way, we are able to calculus a lower bound.

To attack the restricted problem, one idea is to adapt techniques like Zhandry's compressed oracle. That technique makes a quantum algorithm capable of "recording" when accesses to a quantum random oracle. That is useful to derive a quantum lower bound because of its "recording property". However, Zhandry's technique in [31] is with respect to uniform random functions. We therefore transform the problem into another one (Problem 2), a problem with respect to unform distribution. In such a way, the compressed oracle technique in quantum random oracle model (QROM) proposed by Zhandry is successfully exploited.

The transformation technique in this work might be of independent interesting in lower bound exploration in the non-uniform random case.

The transformed problem splits into several sub-problems of the same type according to $c$-partition, and each sub-problem can be calculated the corresponding upper bound of the success probability after $q$ quantum queries by compressed oracle technique. From this, we can show $\Omega(\gamma^{1/6}(c) \cdot \delta_c^{-1/2})$ as a quantum lower bound.

At last, by the properties of $c$-partition, we show $\delta_c = O(\mathtt{k})$, where $\mathtt{k}$ is the min-entropy of distribution $D$. On the other hand, we prove in Section 5 that $\mathtt{k} = \Theta(\log \gamma(c))$ (Proposition 4). We hence obtain $\Omega(\gamma^{1/6}(c)/\sqrt{\log \gamma(c)})$ as a quantum query complexity lower bound for collision-finding problem in general non-uniform distributions.

The structure of the paper is as follows. In Section 2, we give some definitions (including $c$-partition) and preliminaries. In Section 3, a new collision-finding algorithm and its correctness are presented. The quantum query complexity of the algorithm is also analyzed. In Section 4, a lower bound to collision-finding problem is proved by adapting Zhandry's compressed oracle technique. In the last Section, the relations of parameter $\gamma(c)$ with min-entropy and collision-entropy are shown, and some other properties are discussed.

## 2 Preliminaries

### 2.1 Notations and Definitions

In this paper, $M, N$ are positive integers, and $[N]$ is the set $\{1, \ldots, N\}$ and $[M..N]$ is the set $\{M, M+1, \ldots, N\}$ if $M \leq N$. The set $N^M$ is the collection of all functions from $[M]$ to $[N]$, and $f \leftarrow N^M$ refers to the uniformly random sampling from $N^M$. If $D$ is a distribution over $[N]$, then $D^M$ represents the distribution over $N^M$ such that $\Pr_{f \leftarrow D^M}[f(x) = y] = D(y)$ for any $x \in [M]$ independently. We denote $p_i := D(i)$ and assume, without loss of generality, that $p_1 \geq p_2 \geq \cdots \geq p_N$ due to the property of symmetry.

The following definitions appeared in the literatures and will be referred later.

**Definition 1 (Min-Entropy& Collision-Entropy [7]).** *The* min-entropy *of a probabilistic distribution $D$ is $\mathtt{k} := -\log_2(\max_y D(y))$ which is $-\log_2 p_1$ in our setting. The* collision-entropy *of $D$ is defined as $\beta := -\log_2(\sum_{i=1}^{N} p_i^2)$.*

In the literature of collision-finding, the upper and lower bounds of query complexities were described in terms of min-entropy $\mathtt{k}$ or collision-entropy $\beta$. For example, in classical computation, the best upper bound is $O(2^{\beta/2})$ and the lower bound is $\Omega(2^{\mathtt{k}/2})$. In this paper, however, we show that in quantum world, the complexity of collision-finding in non-uniform distribution may not be completely characterized by these two parameters. In order to narrow the gap between upper and lower bounds mentioned above, more properties of non-uniform distributions have to be considered. With such a point of view, we divide the set $[N]$ into several parts according to $\{p_1, \ldots, p_n\}$, which is described as follows.

For any probabilistic distribution $D$ over $[N]$, we assume its weights satisfying $p_1 \geq p_2 \geq \cdots \geq p_N$ in the whole paper. For any $c > 1$, we divide $[N]$ into a series of subsets $S_1, S_2, \ldots, S_\ell$ with respect to $D$: $S_1$ is the collection of index $i \in [N]$ such that $p_i > p_1/c$, with $p(S_1) := p_1$ and $|S_1| = n_1$; $S_2$ then contains all indexes $j \in [N] - S_1$ such that $p_j > p(S_2)/c$, and $p(S_2)$ is the largest one among $\{p_i \mid i \in S_2\}$, and hence $p(S_2) = p_{n_1+1}$ (since $p_1 \geq p_2 \geq \cdots \geq p_N$), and so forth. In other words, the subset $S_i$'s are in some sense the maximal sets of indexes whose corresponding weight differ by a constant factor $c > 1$. Formally,

**Definition 2 ($c$-partition).** *Given constant $c > 1$ and a distribution $D$ over $[N]$ as above. The* $c$-partition of $[N]$ with respect to $D$ *is a partition $\{S_1, \ldots, S_\ell\}$ of $[N]$ such that*

$$|S_i| := n_i, \qquad [N] = \cup_{i=1}^{\ell} S_i, \qquad S_i \cap S_j = \emptyset \ (\textit{for any} \quad i \neq j),$$

*where $S_i$, $(i = 1, \ldots)$ recursively defined as follows:*

- *Let $S_1 := \{j \in [N] \mid p_1 \geq p_j > p_1/c\}$. To denote $n_1 := |S_1|$ and $p(S_1) := p_1$.*
- *For $i \geq 2$, let*

$$S_i := \{j \in [N] \mid p(S_i) \geq p_j > \frac{p(S_i)}{c}\}, \qquad n_i := |S_i|. \tag{1}$$

*Where $p(S_i) \in \{p_1, \ldots, p_N\}$ satisfies*

$$p(S_i) := \max\{p_j : j \in [N] - \bigcup_{k=1}^{i-1} S_k\}. \tag{2}$$

For any constant $c > 1$ and a (non-uniform) distribution $D$ over $[N]$, we define the so called collision parameter $\gamma(c)$ with respect to $c$-partition in the same notations as above, as follows.

**Definition 3 (Collision Parameters & Collision Domains).** *For any real number $c > 1$ and a probabilistic distribution over [N], let c-partition of $[N]$ with respect to $D$ as above. The $\gamma(c)$ defined as follows is called* the collision parameter *of $D$ with respect to $c$, for $i = 1, \ldots, \ell$,*

$$\gamma_i(c) := 1/n_i p^3(S_i), \quad \gamma(c) := \min_{i \in [\ell]}\{\gamma_i(c)\}. \tag{3}$$

*Let $k_0$, referred as the index of collision domain later in context, be the smallest $k_0 \in [\ell]$ such that $\gamma(c) = \gamma_{k_0}(c)$, then $S_{k_0}$ is called* the collision domain *of $D$ with respect to $c$.*

*Remark 1.* The notion "collision parameter" proposed here will be in place of "min-entropy and collision-entropy" appeared in current literatures. It heavily dependants on the distribution $D$ in evaluations. This is similar as "collision-entropy" in prior work, that also needs the whole information of $D$ to calculate. In addition, it is not hard to check that for any uniform distribution $D$ and $c > 1$, $\gamma^{1/6}(c)$ is just the same as $2^{\beta/3}$ and $2^{\mathrm{k}/2}$ in magnitude, the latter is proved optimal in that case.

In general case (namely in arbitrary non-uniform distribution), comparing with existing collision variables, the collision parameter proposed here will give more concise characterizations for both upper and lower queries bounds to collision-finding problem. Moreover, although the parameter $\gamma(c)$ is formally related to the classification of discrete sequence, we are able to prove the fact that the query complexity in $\gamma(c)$ is NOT affected in the magnitude as long as $c$ is a constant. The analysis of these two points is presented in Section 5.

### 2.2 Grover's Algorithm and BBHT Algorithm

In [18], Grover proposed a quantum algorithm for database search problem, demonstrating the powerful acceleration effect of quantum computing on this issue.

**Lemma 1 (Grover's Algorithm [18]).** *Given a boolean function $f : [N] \to \{0,1\}$ such that there is only one $x_0 \in [N]$ satisfying $f(x_0) = 1$, there is a quantum algorithm that requires $O(\sqrt{N})$ queries to find $x_0$ in constant probability.*

Boyer et al. [10] proposed a generalized algorithm so as to be applied to the case of multiple solutions to the search problem even without knowing the number of solutions in advance. That surmounts the restriction in Grover's algorithm that there exists only one $x_0 \in [N]$ satisfying $f(x_0) = 1$. The generalized algorithm is now referred as BBHT algorithm.

**Lemma 2 (BBHT [10]).** *Given a boolean function $f : [N] \to \{0,1\}$ and $t = |f^{-1}(1)|$, there is a quantum algorithm that may find $x_0 \in [N]$ with $f(x_0) = 1$ with $O(\sqrt{N/t})$ expected queries.*

BBHT algorithm starts with uniform superposition state $|\psi_0\rangle = \sum\limits_{i=0}^{N-1} \frac{1}{\sqrt{N}}|i\rangle$ in the workspace. In this algorithm, let $\mathcal{T}$ be the solution space of $f$, that is, the set of all $x$ that satisfy $f(x) = 1$, and $\mathcal{F} := [N]\backslash\mathcal{T}$, then the input also can be written as

$$|\psi_0\rangle = \alpha_0 \sum_{i\in\mathcal{T}} \frac{1}{\sqrt{t}}|i\rangle + \beta_0 \sum_{j\in\mathcal{F}} \frac{1}{\sqrt{N-t}}|j\rangle\,.$$

Where $\alpha_0 = \sqrt{\frac{t}{N}} = \sin\theta$.

After $q$ quantum queries, one gets the superposition as

$$|\psi_q\rangle = \alpha_q \sum_{i\in\mathcal{T}} \frac{1}{\sqrt{t}}|i\rangle + \beta_q \sum_{j\in\mathcal{F}} \frac{1}{\sqrt{N-t}}|j\rangle\,.$$

Where $\alpha_q = \sin(2q+1)\theta$. It was proved that the algorithm may find the preimage of 1 with constant probability.

### 2.3 Some Probabilistic Inequalities

The following are some probabilistic inequalities used in subsequent sections.

**Lemma 3 (Höffding's Inequality).** *Let $X_1, \dots, X_n$ be a sequence of independent random variables such that $X_i$ with values in $[a_i, b_i]$ for $i \in [n]$ and $X = \sum_{i=1}^n X_i$. If the expectation $\mathrm{E}(X) = \mu$, we have, for any $t$,*

$$\Pr[\mu - X \geq t] \leqslant \exp\left(\frac{-2\,t^2}{\sum_{i=1}^n (a_i - b_i)^2}\right).$$

If we know something about the variance of random variables in addition to the expectation, in some cases, we can get a tighter bound.

**Lemma 4 (Bernstein Inequality).** *Let $X_1, \dots, X_n$ be a sequence of independent random variables with values in $[0,1]$ and $X = \sum_{i=1}^n X_i$. If the expectation $\mathrm{E}(X) = \mu$ and the variances $\mathrm{Var}(X) = \sigma^2$, we then have, for any $t$,*

$$\Pr[\mu - X \geq t] \leqslant \exp\left(\frac{-t^2/2}{\sigma^2 + t/3}\right).$$

## 3 A New Algorithm and Its Query Complexity

In Section 3.1, we will review an algorithm in [7], which achieve the optimal acceleration effect in some specific non-uniform distributions. That is, the query complexity of the algorithm matches that of the lower bound in some cases. After careful inspection, we find it is not optimal in general case. That leads us to some insightful observations to the collision-finding problem of non-uniform random functions, and motivates the algorithm in this work. In Section 3.3, a new collision-finding algorithm and its query complexity upper bound in terms of $\gamma(c)$ will be presented.

### 3.1 Motivations

Balogh, Eaton, and Song [7] proposed an excellent collision-finding algorithm (referred as BES algorithm) based on the collision-entropy $\beta$. It applies Ambainis's quantum walk algorithm [3] as a subroutine for element distinctness. Which may be sketched (among others) as follows:

**BES Algorithm (sketch)**

1. Choose $M' \subset M$ arbitrarily such that $|M'| = 2^{\beta/2}$.
2. Run Ambainis's algorithm for function $f|_{M'} : [M'] \to [N]$
3. Output $(x, x')$ if Ambainis's algorithm output $(x, x')$; Otherwise output $\perp$

The excellent point of this algorithm lies that, based on the *Element Distinctness* problem, the requirement for $M$ is minimal in the sense that even if $f \leftarrow D^M$ has only one collision, the algorithm will be able to find it successfully.

Though with very well performance in many cases, however after careful inspection, the approach is found to have space to improve the acceleration effect in some setting. The following is an example, though somewhat artificial. For $M, N > 1$, let $D$ be a distribution over $[N]$ such that

$$p_1 := \frac{1}{2^n}, p_2 = p_3 = \cdots = p_N := \frac{2^n - 1}{(N-1)2^n}.$$

for any integer $n > 0$. Then for $N = 2^{2n}$, to find a collision by BES algorithm requires $\Theta(2^{\beta/3}) = \Theta(2^{2n/3}) = \Theta(2^{2\mathtt{k}/3})$ queries since the min-entropy $\mathtt{k} = n$ in this case. The query complexity is even much higher than the other bound $2^{\mathtt{k}/2}$ by Grover's algorithm at this time.

**Observations and motivations.** The random distribution will bring us more information when exploring the query complexity, which is reflected on the fact that a random function provides additional specific information about the sampling distribution, namely $D$: $\{p_1, p_2, \ldots, p_N\}$, and how to optimally make use of this information to a random function should be the key point to finding more efficient algorithm than to an arbitrary given function.

For a uniform random function $f \leftarrow N^M$, the information is expressed in a concise manner, namely $p_1 = p_2 = \cdots = p_N = \frac{1}{N}$. All the information now can be explained by the sole parameter $N$, so the query complexity of a uniform function is only determined by $N$ (and the other parameter $M$ is to guarantee the existence of collision). While in the case of general non-uniform random functions, the probabilistic weights of the non-uniform distribution avoids this advantage. The query complexity of a non-uniform function would heavily dependent on $p_1, \ldots, p_N$. That is the main ingredients in comparing a non-uniform distribution with a uniform distribution.

Let's try to figure out the obstacle in non-uniform case when investigating the query complexity. For uniformly chosen $f \leftarrow N^M$ and for any $S_1, S_2 \subseteq [N]$ with $|S_1| = |S_2|$, we have

$$\Pr_{f \leftarrow N^M}[f(x) = f(x'), f(x) \in S_1] = \Pr_{f \leftarrow N^M}[f(x) = f(x'), f(x) \in S_2]$$

for any $x, x' \in [M]$. However, the equation would do not hold with non-uniform distributions. This fact may conduct a problem when the former collision-finding algorithm is adopted to achieve acceleration: For any two sets $S_1, S_2 \subseteq [N]$ such that $|S_1| = |S_2|$, when a quantum collision-finding algorithm is adopted, the cost for finding a collision in $S_1$ may be much less than the cost for finding a collision in $S_2$. Therefore, it's necessary to try to avoid spending numerous quantum queries which may not have much effect in collision-finding. An effective approach might be to use the Grover's algorithm while limiting the search scope, which means that the search range $S$ should not be too large.

On the other side, the search domain $S$ should not be too small. Informally, the reason why the collision-finding problem is (arguably) potentially easier than the inverting problem is that there is no prefixed point in the collision problem, which allows the collision problem to have greater searching freedom, and hence might reduce its computational complexity to certain extent.

Above all, a balance should be taken when selecting research domain $S$, and the specific equilibrium result depends on the underlying non-uniform distribution.

In view of current algorithms in literatures, the BES algorithm introduced as above sets $[N]$ as the search domain, while the others set the search domain to $S = \{1\}$ using Grover algorithm. All of them do not make full use of the information provided by the distributions.

That motivates us to propose the notion of $c$-partition in this work. By dividing $[N]$ into a sequence of subsets, the part with the best quantum acceleration effect is determined according to distribution. An (almost) optimal quantum collision-finding algorithm is designed for non-uniform functions based on these arguments.

### 3.2 A Modification of BBHT Algorithm

In our collision-finding algorithm, the BBHT algorithm will be loaded as a subroutine. It expects to have a relation depending on the number of queries to describe the success probability, while BBHT in [10] provides only the expected number of queries when the algorithm succeeds. We have to modify BBHT to adapt the requirement.

To this purpose, the BBHT Algorithm is slightly changed so as to control the number of queries allowed. Actually the modification can be seen as a derandomised version of BBHT algorithm. The modified algorithm is presented as Algorithm 1.

**Algorithm 1** Modified BBHT Algorithm

```
Input: A boolean function $f : [N] \to \{0,1\}$ and an integer $q > 0$.
Output: An element $x \in [N]$ such that $f(x) = 1$ or $\perp$.
Do $\texttt{Expt}_0$: Choose $x_0 \leftarrow [N]$ uniformly at random
For $i = 1$ to $\lfloor \log_3 q \rfloor + 1$ to do
        Experiment $\texttt{Expt}_i$: to perform $3^{i-1}$ Grover iterations for uniform superposition state, and then get $x_i$.
Check if there is an $x^* \in \{x_0, x_1, \ldots, x_{\lfloor \log_3 q \rfloor + 1}\}$ such that $f(x^*) = 1$. If so, output $x^*$; otherwise, output $\perp$.
```

The algorithm above is composed of several experiments $\texttt{Expt}_i$, and the algorithm succeeds as long as one of these experiments succeeds. The total number $Q$ of queries for the above algorithm is

$$Q := \sum_{i=1}^{\lfloor \log_3 q \rfloor + 1} 3^{i-1} + \lfloor \log_3 q \rfloor + 2 < 3.6\, q\,.$$

The success probability of modified BHHT algorithm is concluded as following.

**Theorem 1.** *For any boolean function $f : [N] \to \{0,1\}$ and $t_0 := |f^{-1}(1)| > 0$, the algorithm above makes at most $3.6q$ queries to find a pre-image of $1$ with probability at least $\min\{1/2, \frac{q^2 t_0}{3N}\}$.*

The proof of Theorem 1 will use the following fact, and the proof is easy and hence omitted.

**Proposition 1.** *If the increasing sequence $\{a_i\}_{i \in \mathbb{N}}$ of reals satisfies $a_{i+1} < 3a_i$ for all $i \in \mathbb{N}$, and $\lim_{i \to +\infty} a_i = +\infty$, then for any $b > a_1$, there is an integer $j \in \mathbb{N}$ such that $b/3 \leq a_j < b$.*

*Proof of Theorem 1.* From Lemma 2 and the note there, we see that $\texttt{Expt}_0$ succeeds with probability $\sin^2 \theta = t_0/N$. For any $i \geq 1$, the success probability of $\texttt{Expt}_i$ is $\sin^2 \Big(2 \cdot 3^{i-1} + 1\Big)\theta$.

For our purpose, let $a_0 := \theta, a_i := (2 \cdot 3^{i-1} + 1)\theta$ and $b := 3\pi/4$. We have $0 < a_0 = \theta \leq \pi/2 < 3\pi/4 = b$, and $a_{i+1} = (2 \cdot 3^i + 1)\theta < 3(2 \cdot 3^{i-1} + 1)\theta = 3a_i$ for all $i$. Proposition 1 tells that there is a $j$ such that $\pi/4 \leq a_j < 3\pi/4$. Let $j$ be the least such index. We then have $\sin^2 a_j \geq 1/2$. There are two possible cases for $j$.

1. If $j \le \lfloor \log_3 q \rfloor + 1$, then $\texttt{Expt}_j$ succeeds with probability at least $1/2$. Hence, the algorithm succeeds with probability at least $1/2$.
2. If $j > \lfloor \log_3 q \rfloor + 1$, we will have $0 < a_{\lfloor \log_3 q \rfloor + 1} < \pi/4$, since $a_i$ is an increasing sequence and $j$ is the least index such that $\pi/4 \le a_j < 3\pi/4$. In this case, $0 < \frac{2q\theta}{3} < (2 \cdot 3^{\lfloor \log_3 q \rfloor} + 1)\theta = a_{\lfloor \log_3 q \rfloor + 1} < \pi/4$. Hence

   $$\begin{aligned}\Pr[\texttt{Expt}_{\lfloor \log_3 q \rfloor + 1} \text{ succeeds}] &= \sin^2\left((2 \cdot 3^{\lfloor \log_3 q \rfloor} + 1)\theta\right) \\ &> \sin^2 \frac{2q\theta}{3} \overset{*}{\ge} \left(\frac{2\sqrt{2}}{\pi} \cdot \frac{2\,q}{3} \cdot \sin\theta\right)^2 > \frac{q^2 t_0}{3N}\,.\end{aligned}$$

   Where inequality (*) holds since $\sin \alpha\theta \ge \frac{2\sqrt{2}}{\pi} \cdot \alpha \cdot \sin\theta$ provided $0 < \alpha\theta < \pi/4$ for any $\alpha > 0$.

The combination of the two cases above establishes the conclusion. □

### 3.3 A New Collision-Finding Algorithm

We now propose a new collision-finding algorithm for random functions. According to Definition 2, given $D$ is a distribution over $[N]$, let $D^M$ represent the distribution over $N^M$ such that $\Pr_{f \leftarrow D^M}[f(x) = y] = D(y)$ for any $x \in [M]$ independently. The codomain $[N]$ of $f \leftarrow D^M$ is partitioned into $\ell$ parts $S_1, \dots, S_\ell$. We will use collision domain $S := S_{k_0}$ as the input of the algorithm. The algorithm is presented as Algorithm 2.

**Algorithm 2** Collision-Finding Algorithm in Non-uniform Functions

```
Input: Collision domain $S$ and a function $f : [M] \to [N]$ with $f \leftarrow D^M$.
Output: A collision $(x_1, x_2)$ or $\perp$.
Construct a function $F_1 : [M] \to \{0,1\}$ such that $F_1(x) = 1$ iff $f(x) \in S$. Let $L$ be
  a dynamic constructed set which initially is emptyset $L = \emptyset$.
Run Algorithm 1 with $F_1$ and $q_1$ to search for $x$ such that $F_1(x) = 1$. Query
  $y := f(x)$ and check whether $y \in S$. If yes, add $(x, y)$ into $L$; otherwise discard it.
      The process repeats until $L$ contains $t$ pairs of elements and to go to the next
  step; or repeats $4\,t$ times, and Algorithm halts with $|L| < t$ and outputs $\perp$.
Check the elements in $L$. If there exist $(x_1, y_1), (x_2, y_2) \in L$ such that $x_1 \neq x_2$ and
  $y_1 = y_2$, output $(x_1, x_2)$ and halt. Otherwise to go to the next step.
Construct a function $F_2 : [M] \to \{0,1\}$ such that $F_2(x) = 1$ iff there exists
  $(x_0, y_0) \in L$ such that $f(x) = y_0$ and $x \neq x_0$. Invoke the modified BBHT algorithm
  with $F_2$ and $q_2$ to get an $x_1 \in [M]$.
If there is $(x_2, y_2) \in L$ such that $f(x_1) = y_2$, then output $(x_1, x_2)$; otherwise $\perp$.
```

The parameters $t$, $q_1$ and $q_2$ in Algorithm 2 will be discussed and determined later in the context. Essentially, $t = \sqrt[3]{n_{k_0}}$ (see discussions after Theorem 4), $q_1 = O(1/\sqrt{n_{k_0} p(S_{k_0})})$ (Lemma 5), and $q_2 = O(1/\sqrt{t\, p(S_{k_0})})$ (Lemma 7).

Now we justify the correctness and the complexity of the algorithm in the following theorem.

**Theorem 2.** *For any constant $c > 1$, suppose $D$ be a probabilistic distribution over $[N]$ with $M > 12c^2/p(S_{k_0})$, where $k_0$ is the index of collision domain defined in Definition 3, then with $O(\gamma^{1/6}(c))$ queries, Algorithm 2 will find a collision to $f \leftarrow D^M$ with probability $\Omega(1)$.*

*Remark 2.* The algorithm's requirements for $M$ are described in terms of $p(S_{k_0})$, which may not be particularly intuitive. In Section 5, we will show that the condition of $M$ can be relaxed to $M = \Omega(2^{3k/2})$ or $M = \Omega(N)$.

The whole subsection 3.4 is devoted to the proof of Theorem 2.

### 3.4 Proof of Theorem 2

From Definition 2 of $c$-partition, we know that $p(S_i)$ is $p_{n_1+\ldots n_{i-1}+1}$ for any $i \in [\ell]$, and is, in fact, the maximum one in $\{p_j, j \in S_i\}$. Also, we have that $p(S_i)/c \le p_j \le p(S_i)$ for arbitrary $j \in S_i$.

Let $T_f$ be the set of all the $x$ such that $f(x) \in S = S_{k_0}$, and its size as $|T_f|$. For convenience, we call a function $f \leftarrow D^M$ *well-behaved* if and only if $|T_f| > 2Mn_{k_0}p(S_{k_0})/3c$. We then have the following conclusion.

**Proposition 2.** *For any constant $c > 1$, under the condition of Theorem 2, the random function $f$ is well-behaved with constant probability.*

*Proof of Proposition 2.* For any $j \in S_i$ and the collision domain $S = S_{k_0}$, we have, according to Definition 3 that

$$p(S_i)/c \le p_j \le p(S_i), \text{ and } n_{k_0}p(S_{k_0})/c \le \sum_{i \in S_{k_0}} p_i \le n_{k_0}p(S_{k_0}).$$

Let random indicator $T_{f,x} = 1$ iff $f(x) \in S_{k_0}$. It holds that $|T_f| = \sum_{x\in[M]} T_{f,x}$, and for any $x \in [M]$,

$$\begin{aligned} \mathrm{E}[T_{f,x}] &= \sum_{i\in S_{k_0}} p_i \ge \frac{n_{k_0}p(S_{k_0})}{c}, \text{ and} \\ \mathrm{Var}[T_{f,x}] &= \Big(\sum_{i\in S_{k_0}} p_i\Big) \cdot \Big(1 - \sum_{i\in S_{k_0}} p_i\Big) < n_{k_0}p(S_{k_0}). \end{aligned}$$

By Bernstein's inequality (Lemma 4) and the definition of well-behaved function, we get

$$\begin{aligned} \Pr_{f\leftarrow D^M}[f \text{ is not well-behaved}] &\le \Pr_f\Big[M\,\mathrm{E}[T_{f,x}] - |T_f| > Mn_{k_0}p(S_{k_0})/3c\Big] \\ &\le \exp\Big(\frac{-(Mn_{k_0}p(S_{k_0}))^2/18c^2}{Mn_{k_0}p(S_{k_0}) + MN_{k_0}p(S_{k_0})/9c}\Big) \\ &= \exp\Big(-\frac{Mn_{k_0}p(S_{k_0})}{18c^2 + 2c}\Big). \end{aligned}$$

Since $M > \frac{12c^2}{p(S_{k_0})}$ and $n_{k_0} \geq 1$, we get $f$ is well-behaved with probability $1 - e^{-\frac{3}{5}} > 2/5$ from inequality above. Which concludes Proposition 2. □

Let suc denote the event that Collision-Finding Algorithm successfully finds a collision. According to Proposition 2, we have

$$\begin{aligned} \Pr_{f \leftarrow D^M}[\mathrm{suc}] &= \sum_f \Pr[f] \cdot \Pr[\mathrm{suc} \mid f] \geq \sum_{f\text{: well-behaved}} \Pr[f] \cdot \Pr[\mathrm{suc} \mid f] \\ &\geq \Big( \sum_{f\text{: well-behaved}} \Pr[f] \Big) \cdot \min_f \{\Pr[\mathrm{suc} \mid f \text{ is well-behaved}]\} \\ &> \frac{2}{5} \cdot \min_f \{\Pr[\mathrm{suc} \mid f \text{ is well-behaved}]\}. \end{aligned} \tag{4}$$

The last inequality is inherited from the proof of Proposition 2.

Notice that the key to the success of the algorithm is whether Step 2 can successfully find $t$ pairs (to denote as $\mathrm{suc}_2$) and whether Step 3 or Step 4 can actually find a collision pair (to denote as $\mathrm{suc}_3$ and $\mathrm{suc}_4$ respectively). Namely our algorithm succeeds iff $\mathrm{suc}_2$ happens and one of $\mathrm{suc}_3$, $\mathrm{suc}_4$ happens. That is,

$$\begin{aligned} &\min_f \{\Pr[\mathrm{suc} \mid f \text{ is well-behaved}]\} \\ &\quad = \min_{f:\text{well-behaved}} \big\{ \Pr[\mathrm{suc}_2 \mid f] \cdot \Pr[\mathrm{suc}_3 \vee \mathrm{suc}_4 \mid f \wedge \mathrm{suc}_2] \big\} \\ &\quad \geq \min_{f:\text{well-behaved}} \Pr[\mathrm{suc}_2 \mid f] \cdot \min_{f:\text{well-behaved}} \Pr[\mathrm{suc}_3 \vee \mathrm{suc}_4 \mid f \wedge \mathrm{suc}_2]. \end{aligned} \tag{5}$$

Note that the probabilities in the equation above are determined by $f$ and the query number $q$ in the corresponding step. For convenience, let

$$P_1^q := \min_{f:\text{well-behaved}} \Pr[\mathrm{suc}_2 \mid f], \tag{6}$$

$$P_2^q := \min_{f:\text{well-behaved}} \Pr[\mathrm{suc}_3 \vee \mathrm{suc}_4 \mid f \wedge \mathrm{suc}_2]. \tag{7}$$

We are going to show, for sufficiently large $q$, that $P_1^q$ and $P_2^q$ have lower bounds asymptotically to 1. Which, in turn from (5), implies that promised $f$ is well-behaved, Algorithm 2 finds the collision with bounded error.

**Estimations of $P_1^q$ and $P_2^q$.** We show the following two results.

**Theorem 3.** *For any constant $c > 1$ and any well-behaved random function $f \leftarrow D^M$, Algorithm 2 succeeds in Step 2 with the probability at least $1 - \exp(-\frac{t}{2})$ after making at most $21.6t\sqrt{c}/\sqrt{n_{k_0} p(S_{k_0})}$ queries.*

**Theorem 4.** *Suppose it has successfully obtained $t$ pairs in Step 2 of Collision-Finding Algorithm, then for any constant $c > 1$ and any well-behaved random function $f \leftarrow D^M$, the algorithm, which makes at most $32.4\sqrt{c}/\sqrt{t\,p(S_{k_0})}$ queries, will find a collision with the probability at least $\frac{1}{2} \cdot (3/4 - e^{-t/2})$.*

According to the two conclusions, the algorithm will find a collision with high probability with at most $O(\max(t/\sqrt{n_{k_0}p(S_{k_0})}, 1/\sqrt{tp(S_{k_0})}))$ queries. It's easy to see that when $t := O(\sqrt[3]{n_{k_0}})$, namely $\frac{t}{\sqrt{n_{k_0}p(S_{k_0})}} = \Theta(\frac{1}{\sqrt{t\,p(S_{k_0})}})$, the order of magnitude of the number of queries reaches the minimum, which will be

$$O(\frac{t}{\sqrt{n_{k_0}p(S_{k_0})}}) = O(\frac{1}{\sqrt{t\,p(S_{k_0})}}) = O((n_{k_0}p^3(S_{k_0}))^{-1/6}) = O(\gamma^{1/6}(c)) \qquad (8)$$

In other words, for $t := \sqrt[3]{n_{k_0}}$, the algorithm with $O(\gamma^{1/6}(c))$ queries may successfully find a collision with probability $\Omega(1)$.

We also obtain the following straightforward conclusion.

**Lemma 5.** *For any well-behaved random function $f$, the modified BBHT algorithm that makes at most $q_1 := 5.4\sqrt{c}/\sqrt{n_{k_0}p(S_{k_0})}$ queries will find a pre-image of 1 for $F_1$ with probability at least $1/2$.*

We now give the proof of Theorem 3 by combining the Höffding's inequality with the proposition.

***Proof of Theorem 3.*** Let the modified BBHT algorithm be repeated in Step 2 independently $4t$ times, then the total query number is at most $q$, where $q = 4\,t\,q_1 = 21.6\,t \cdot \sqrt{c}/\sqrt{n_{k_0}p(S_{k_0})}$.

Let random indicator $X_i$ be 1 iff the $i$'th run of the modified BBHT successfully gets an element in $L$, and $|L| = \sum_{i=1}^{4\,t} X_i$. The expected value $\mathrm{E}(|L|) \geq 2\,t$. By Höffding's inequality,

$$P_1^q = 1 - \Pr[|L| < t] \geq 1 - \Pr[\mathrm{E}[|L|] - |L| \geq t] \geq 1 - \exp(-\frac{t}{2}).$$

Which ends the proof of Theorem 3. □

The remaining part of this subsection is to prove Theorem 4. We have to make some preparations.

We classify the list $L := \{(x_i, y_i) \mid i = 1, \ldots, t\}$ obtained in Step 2 into four cases. We call a pair $(x, y)$ *white* if $|f^{-1}(y)| \geq Mp(S_{k_0})/6c$, for convenience:

- Case 1: There exists $i, j, i \neq j$ such that $(x_i, y_i) = (x_j, y_j) \in L$.
- Case 2: For any $i \neq j$, $x_i \neq x_j$ in $L$ and there exist $i, j$ such that $y_i = y_j$. Hence, list $L$ contains a collision in this case.
- Case 3: For any $i, j, i \neq j$, it holds $x_i \neq x_j$, $y_i \neq y_j$ in $L$, and the total number of white pairs is at least $t/4$.
- Case 4: For any $i, j, i \neq j$, it holds $x_i \neq x_j$, $y_i \neq y_j$ in $L$, and the total number of white pairs is less than $t/4$.

Before to analyze the happening possibility of each case, we investigate each pair $(x, y)$ in $L$ getting in the algorithm. Since each $x$ there is the output of modified BBHT algorithm, $x$ is hence uniformly sampled from the solution space $T_f$ of $F_1$. Denote $P_i$ as the probability that $L$ is in case $i$, for $i = 1, 2, 3, 4$. Then we discuss case by case as follows.

Case 1: We show that $L$ is in case 1 with small probability $P_1$.

**Lemma 6.** *For any well-behaved random function $f$ and $t < \sqrt{n_{k_0}}$, we have $P_1 < 1/4$.*

*Proof of Lemma 6.* Since each $x_i$ is uniform from $T_f$, we have:

$$P_1 = 1 - \prod_{i=0}^{t-1} \frac{|T_f| - i}{|T_f|} \le 1 - \prod_{i=0}^{t-1} \Big(1 - \frac{3\,c\,i}{2Mn_{k_0}p(S_{k_0})}\Big)$$
$$< 1 - \exp\Big(-\sum_{i=1}^{t} \frac{3\,c\,i}{2Mn_{k_0}p(S_{k_0})}\Big) < \frac{3\,c^2\,(t+1)^2}{4Mn_{k_0}p(S_{k_0})} < 1/4.$$

Where $M > 12\,c^2/p(S_{k_0})$ is used in the last inequality above. □

Case 2: It's easy to see that in this case, our algorithm will output a collision with probability 1.
Case 3: In this case, $y_i \neq y_j$ for any $i, j$, since $M > 12\,c^2/p(S_{k_0})$ and the number of white pairs is at least $t/4$, the total number of solutions for $F_2$ is at least

$$\frac{t}{4} \cdot \Big(\frac{M\,p(S_{k_0})}{6c} - 1\Big) + \frac{3t}{4} \cdot 0 > \frac{1}{2} \cdot \frac{M\,t\,p(S_{k_0})}{24c}.$$

By Theorem 1, we have in this case:

**Lemma 7.** *For any well-behaved random function $f$, Algorithm 2, making at most $q_2 := 32.4\sqrt{c}/\sqrt{t\,p(S_{k_0})}$ queries, will find a pre-image of 1 for $F_2$ with error at most $1/2$.*

Case 4: We show that $P_4$ is negligible in $t$ as follows.

**Lemma 8.** *Suppose $f \leftarrow D^M$ is well-behaved and $L$ contains $t$ pairs in Step 2, then the probability that $L$ contains at most $t/4$ white pairs is at most $e^{-t/2}$.*

*Proof of Lemma 8.* Let's calculate the probability of getting a white pair from sampling in an experiment. By the meaning of white pair, we know:

$$\Pr[(x_i, y_i) \text{ is white}] \ge \frac{|T_f| - (n_{k_0} - 1) \cdot \frac{M\,p(S_{k_0})}{6c}}{|T_f|} > \frac{3}{4}.$$

Let $\ell_{\mathrm{w}}$ be the number of white pairs in $L$. Since each pair is obtained independently at random, by repeating $t$ times, the expected value of $\ell_{\mathrm{w}}$ is at least $3t/4$. Again, by Höffding's inequality, we have:

$$\begin{aligned} P_4 &\le \Pr[L \text{ contains at most } t/4 \text{ white pairs}] \\ &\le \Pr[\mathrm{E}[\ell_{\mathrm{w}}] - \ell_{\mathrm{w}} > t/2] \\ &\le \exp\Big(-\frac{2 \cdot \frac{t^2}{4}}{t}\Big) = e^{-t/2}. \end{aligned}$$

That ends the proof of Lemma 8. □

To combine the results discussed as above, we finally come to the proof of Theorem 4, as follows.

*Proof of Theorem 4.* Combining with the propositions above, we have, within at most $32.4\sqrt{c}/\sqrt{t\,p(S_{k_0})}$ queries, the success probability in Step 3 or 4 is at least

$$\begin{aligned}
P_2^q &= \sum_{i=1}^{4} \Pr[L \text{ is in Case } i] \cdot \Pr[\text{suc}_3 \vee \text{suc}_4 \mid f \wedge L \text{ is in Case } i] \\
&> 1 \cdot P_2 + \frac{1}{2} \cdot P_3 > \frac{1}{2} \cdot (P_2 + P_3) \\
&> \frac{1}{2} \cdot (3/4 - e^{-t/2}).
\end{aligned}$$

As desired in Theorem 4. □

Overall, the conclusions of Equations (4), (5), (6), (7), together with Theorem 3 and Theorem 4 will finally give the result in Theorem 2.

# 4 The Lower Bound

In this section, we are going to exploit a query complexity lower bound, during which the compressed oracle technique is adapted. For this purpose, we firstly introduce Zhandry's compressed oracle in Section 4.1. We then turn the collision-finding problem with respect to non-uniform functions into another problem with uniform functions, for which we are able to use compressed oracle technique to give a lower bound to the transformed problem as shown in Theorem 6 in Section 4.3. The relation of their success probabilities for two problems is then shown in Section 4.2. With Corollary 1 (a variant of Theorem 6), we explore the lower bound in Section 4.4 and get the main result in Theorem 7. In last subsection we discuss the parameter $\delta_c$ appeared in Theorem 7, which leads to an almost tight lower bound.

## 4.1 Zhandry's Compressed Oracle

There are two models of oracles in quantum computing called, respectively, the standard oracle and phase oracle. These two are widely used in quantum computation in the black-box setting. By using the Hadamard transformation, these two oracles have been shown to be completely equivalent, so only phase oracle will be introduced here.

Let $\mathcal{A}^{\mathcal{O}}$ be a $q$-query quantum algorithm which is given oracle access to a function $\mathcal{O}$ from $N^M$. Let $|\psi_{start}\rangle$ be the input state for $\mathcal{A}^{\mathcal{O}}$ and $|\psi_{end}\rangle$ the output state before the final measurement. A quantum computation performed by $\mathcal{A}$ with $q$ queries is generally described as the product of a series of unitary transformations, in the following form:

$$|\psi_{end}\rangle = U_q\mathcal{O}\ldots U_1\mathcal{O}U_0|\psi_{start}\rangle.$$

Where $U_0,\ldots,U_q$ are some unitary operators independent of the input $x$. Let $|\psi^i\rangle$ denote the state before the $i$'th query, which is

$$|\psi^i\rangle = \sum_{x,y,z} a^i_{x,y,z}|x,y,z\rangle.$$

Here $x$ denotes the input register, $y$ is the output register and $z$ is some auxiliary bits.

**A Phase oracle** is a unitary transformation as follows:

$$\sum_{x,y,z} a_{x,y,z}|x,y,z\rangle \rightarrow \sum_{x,y,z}(-1)^{y\cdot f(x)}a_{x,y,z}|x,y,z\rangle$$

for any random function $f \rightarrow N^M$.

Zhandry discovered that $\mathcal{O}$ could be written in another form, in which $f$ would be written as a truth table $|f\rangle = |f(0), f(1)\ldots f(M-1)\rangle$, and a query to $\mathcal{O}$ should be considered as a quantum entanglement as follows:

$$\sum_{x,y,z} a_{x,y,z}|x,y,z\rangle \otimes \sum_f \frac{1}{\sqrt{N^M}}|f\rangle \rightarrow \sum_{x,y,z}(-1)^{y\cdot f(x)}a_{x,y,z}|x,y,z\rangle \otimes \sum_f \frac{1}{\sqrt{N^M}}|f\rangle.$$

**Compressed phase oracle:** In the model of compressed phase oracle [31], the superposition state $\sum|f\rangle$ in phase oracle above will be replaced by a database $\mathbf{D}$ which is initialized as $\mathbf{D} := \emptyset$. When the adversary makes a query to the compressed phase oracle $\mathcal{O}$ on $|x,y,z,\mathbf{D}\rangle$, it performs in sequence of the following steps:

1. If there is no $(x,y^*) \in \mathbf{D}$, it performs the map:

$$|x,y,z\rangle \otimes |\mathbf{D}\rangle \rightarrow |x,y,z\rangle \otimes \frac{1}{\sqrt{2^n}}\sum_w |\mathbf{D}\cup(x,w)\rangle.$$

If there is $(x,y_0) \in \mathbf{D}$, then check whether binary pairs in $\mathbf{D}$ should be deleted. More specifically, it performs the map:

$$\frac{1}{\sqrt{2^n}}\sum_{y_0}(-1)^{z'\cdot y_0}|\mathbf{D}\cup(x,y_0)\rangle \rightarrow \begin{cases} \frac{1}{\sqrt{2^n}}\sum_{y_0}(-1)^{z'\cdot y_0}|\mathbf{D}\cup(x,y_0)\rangle, & \text{if } z' \neq 0; \\ |\mathbf{D}\rangle, & \text{if } z' = 0. \end{cases}$$

2. Perform the following unitary transformation:

$$|x,y,z\rangle \otimes |\mathbf{D}\cup(x,w)\rangle \rightarrow (-1)^{y\cdot w}|x,y,z\rangle \otimes |\mathbf{D}\cup(x,w)\rangle.$$

3. Perform the Step 1 again.

Zhandry proved that the two random oracles are equivalent. That is, for any adversary $\mathcal{A}$, phase oracle and compressed phase oracle are perfectly indistinguishable. Moreover, under the compressed oracle model, the database attached is very likely to record the information obtained by the adversary. The details, verbatim quoted from the original paper, are as follows:

**Lemma 9 (Lemma 5 in [31]).** *Consider a quantum algorithm $\mathcal{A}$ making queries to a random oracle $H$ and outputting tuples $(x_1 \ldots x_k, y_1, \ldots y_k, z)$. Let $R$ be a collection 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 $A$ with the compressed phase oracle, and suppose the database $\mathbf{D}$ is measured after $\mathcal{A}$ produces its output. Let $p'$ be the probability that (1) the tuple is in $R$, and (2) $\mathbf{D}(x_i) = y_i$ for all $i$ (and in particular $\mathbf{D}(x_i) \neq \perp$). Then $\sqrt{p} < \sqrt{p'} + \sqrt{k/2^n}$.*

This indicates compressed oracle's record reliability to ensure that if $\mathcal{A}$ can find a solution to a problem with a non-negligible probability, it can also be found in $\mathbf{D}$ with a non-negligible probability (provided that $k$ is small enough).

In the collision-finding problem, the output of any quantum algorithm will be a binary pair, namely $k = 2$. This hints us a new way to seek about lower bound. Although the density matrices [2] and the polynomials [29] approaches are usually adapted to derive the quantum lower bound proofs. Zhandry's technique allows to turn attentions to the changes in $\mathbf{D}$ after each query (notice any unitary operators $U_j$ can not change the database $\mathbf{D}$), which would be, in some cases, more intuitive and convenient [20, 23].

### 4.2 Transformation from Non-Uniform Case to Uniform Case

Since the compressed oracle is equivalent to the phase oracle only for the uniform random functions and it might not be easy directly to apply in non-uniform setting, we therefore have to turn the collision-finding problem in non-uniform setting into another problem in uniform setting with a larger range.

For a given random distribution $D$ over $[N]$, we have $c$-partition defined as in Definition 2 and Definition 3. With the same notations as in Definitions, we pose the following problem.

*Problem 1.* For any $r \in [\ell]$, let $f : [M] \to [N]$ be a function chosen according to non-uniform distribution $D^M$, the problem is to find a collision $(x, x')$ such that $f(x) = f(x') \in S_r$.

The problem is the same as the general collision-finding problem except that a constraint $f(x) \in S_r$ is posed. It is easy to see that if an adversary $\mathcal{A}$ successfully finds a collision in general, then Problem 1 should be solved for some $S_r$. Hence, the success probability of $\mathcal{A}$ is bounded by the SUM of the upper bounds of the success probabilities for solving Problem 1 over all $S_r$ (as shown in Inequality (26) in Section 4.4). In this way, we will show the number of queries necessary for collision-finding.

In order to estimate the success probabilities to Problem 1, we turn to the following corresponding problem.

*Problem 2.* For the $r \in [\ell]$, let $g : [M] \to [KN]$ be a function chosen from $(KN)^M$ uniformly at random, where $K$ is a large integer (cf. Theorem 5 for the possible values of $K$). To define $s_i \subseteq [KN]$ as follows $(i = 1, \ldots, N+1)$.

$$s_i := \begin{cases} \big[1 \mathinner{\ldotp\ldotp} \lfloor KNp_1 \rfloor\big], & \text{for } i = 1; \\ \big[1 + \sum\limits_{j=1}^{i-1} \lfloor KNp_j \rfloor \mathinner{\ldotp\ldotp} \sum\limits_{j=1}^{i} \lfloor KNp_j \rfloor\big], & \text{for } i \in [2, N]; \\ [KN] \setminus (\bigcup\limits_{i=1}^{N} s_i), & \text{for } i = N+1. \end{cases}$$

The problem is to find two distinct inputs $x, x'$ such that $g(x), g(x') \in s_k$ for some $k \in S_r$.

The following result reveals the relation between these two problems above:

**Theorem 5.** *If there exists a $q$-query quantum algorithm $\mathcal{A}$ that solves Problem 1 with probability $\mathbf{P}_1$, then there exists a $q$-query quantum algorithm $\mathcal{B}$ that solves Problem 2 with probability $\mathbf{P}_2$ such that*

$$\mathbf{P}_1 - \mathbf{P}_2 \leq \frac{2\,q^2}{K}.$$

It tells us that, for large enough $K$, if there exists an algorithm $\mathcal{A}$ solving Problem 1 with success probability $\Omega(1)$, then there will have an algorithm $\mathcal{B}$ solving Problem 2 successfully with probability $\Omega(1)$.

In other words, if we get an upper bound of success probability with $q$ queries solving Problem 2, we will get an upper bound of success probability with $q$ queries solving Problem 1. That is, we use Problem 2 to functions with uniform distributions to simulate the problem 1 to functions with non-uniform distributions. Moreover, the larger the $K$ is, the better the simulation does.

*Proof of Theorem 5.* Firstly, assume that all weights $p_i$ of distribution $D$ are rational numbers for all $i \in [N]$. There are large enough $K$ such that all $KNp_i$ are positive integers. To set $K$ as one of such kind of integers in this case.

For any $g \leftarrow (KN)^M$, to define the function $h_g$ such that $h_g(x) := y$ iff $g(x) \in s_y$ for any $x \in [M]$ and $y \in [N]$.

In this way, since for any $i \in [N]$, $KNp_i$ is a positive integer, thus for any $x \in [M]$, $y \in [N]$,

$$\Pr_{g \leftarrow (KN)^M}[h_g(x) = y] = \Pr_{g \leftarrow (KN)^M}[g(x) \in s_y] = \frac{\lfloor KNp_y \rfloor}{KN} = \frac{KNp_y}{KN} = p_y. \quad (9)$$

That means, if $\mathcal{A}$ only makes oracle access to $h_g$, the function $h_g$ defined is a non-uniform random function according to $D^M$. Since Problem 2 is to find a collision $(x, x')$ on $h_g$ and $h_g(x) \in S_r$. We get, in this case, Problem 2 is equivalent to Problem 1.

We now turn to the case when some $p_i$ are irrational numbers. Intuitively from (9), as long as $K$ large enough, $h_g$ previously defined for $g \leftarrow (KN)^M$ and

$f \leftarrow D^M$ are tending to be equivalent, which means that any quantum algorithm will take great cost to distinguish between them (and the cost increases with $K$), the success probabilities in two problems are almost equal. Formally, we make the following calculus.

For any $g \leftarrow (KN)^M$, to set function $h'_g$ as follows: For any $x \in [M]$,

$$h'_g(x) := \begin{cases} y, & \text{if } g(x) \in s_y \text{ and } y \in [N]\,; \\ z, & \text{if } g(x) \in s_{N+1}, \text{ and } z \leftarrow D'\,. \end{cases}$$

Where $D'(y) := (KNp_y - \lfloor KNp_y \rfloor)/|s_{N+1}|$ for any $y \in [N]$.

We see that for any $x \in [M]$, $y \in [N]$, with respect to $g \leftarrow (KN)^M$, we have

$$\begin{aligned} &\Pr[h'_g(x) = y] \\ &\quad = 1 \cdot \Pr[g(x) \in s_y] + \Pr[g(x) \in s_N + 1] \cdot \Pr[h'_g(x) = y \mid g(x) \in s_{N+1}] \\ &\quad = \frac{\lfloor KNp_y \rfloor}{KN} + \frac{KNp_y - \lfloor KNp_y \rfloor}{|s_{N+1}|} \cdot \frac{|s_{N+1}|}{KN} = \frac{KNp_y}{KN} = p_y. \end{aligned}$$

In other words, $h'_g$ is exactly a non-uniform random function according to $D^M$.

Suppose $\mathcal{B}$ wants to solve Problem 2 for a uniform random function $g$, then $\mathcal{B}$ can produce a function $h'_g$ by using the above method and only give oracle access to $\mathcal{A}$. Let $\mathbf{P}_1$ be the probability that $\mathcal{A}$ finds a solution of Problem 1: $(x_1, x_2)$, and $\mathbf{P}_2$ be the probability that $(x_1, x_2)$ is also a solution of Problem 2. We have

$$\begin{aligned} \mathbf{P}_1 &:= \Pr[(x_1, x_2), x_1 \neq x_2, h(x_1) = h(x_2) \in S_r \;:\; h \leftarrow D^M, (x_1, x_2) \leftarrow \mathcal{A}^h] \\ &= \Pr[(x_1, x_2), x_1 \neq x_2, h'_g(x_1) = h'_g(x_2) \in S_r \;:\; g \leftarrow (KN)^M, (x_1, x_2) \leftarrow \mathcal{A}^{h'_g}] \\ &< \underset{g \leftarrow (KN)^M}{\Pr}[(x_1, x_2), x_1 \neq x_2, g(x_1) = g(x_2) \in s_i, i \in S_r \;:\; (x_1, x_2) \leftarrow \mathcal{B}^g] \\ &\qquad + \Pr[(x_1, x_2), \exists x_i \texttt{ s.t. } g(x_i) \in s_{N+1} \;:\; g \leftarrow (KN)^M, (x_1, x_2) \leftarrow \mathcal{B}^g] \\ &\leq \mathbf{P}_2 + 2\Pr[g(x) \in s_{N+1} : g \leftarrow (KN)^M, x \leftarrow \mathcal{C}^g] = \mathbf{P}_2 + 2\Pr[\text{find}]. \end{aligned} \tag{10}$$

Where $\mathcal{C}$ be any algorithm inverting $g$ and

$$\Pr[\text{find}] := \Pr[g(x) \in s_{N+1} : g \leftarrow (KN)^M,\ x \leftarrow \mathcal{C}^g].$$

It should be noted from (10) that the gap between $\mathbf{P}_1$ and $\mathbf{P}_2$ is not excess two times the success probability of the database search problem for uniformly random function. Since $\frac{|s_{N+1}|}{KN} < \frac{1}{K}$, according to lower bound of database search problem [31], after $q$ queries, we have

$$\Pr[\text{find}] \leq \frac{q^2}{K}.$$

As desired. □

The result in Theorem 5 allows us, when considering the collision-finding problem, to focus on Problem 2, which is with respect to uniform random functions. That also makes the compressed oracle technology useful for our purpose.

### 4.3 Lower Bound for Problem 2

The main result in this subsection is following theorem.

**Theorem 6.** *Given $r > 0$ and distribution $D$ as in last section, for any quantum algorithm, $\Omega(\gamma_r^{1/6}(c))$ quantum queries are necessary to solve Problem 2 with constant probability.*

Let $S_r' := \bigcup_{j \in S_r} s_j$ and $s_j$ defined in last section, then we have the following lemma. The proof uses the compressed oracle technique, and mainly adopts the ideas from [23] and [31].

**Lemma 10.** *For any quantum algorithm that makes $q$ queries to compressed random oracle $\mathcal{O}$, the probability that $\mathbf{D}$ contains at least $i$ pre-images of $S_r'$ after the $q$-th query is at most $\left(eq\sqrt{n_r p(S_r)}/i\right)^i$.*

*Proof of Lemma 10.* Let's calculate the probability of getting $x$ that satisfies $g(x) \in S_r'$ after $q$ quantum queries, where $S_r' := \bigcup_{j \in S_r} s_j$.

For this purpose, we use the same idea as in [31] to classify the basic states. Suppose just before $q$'th query the joint state as

$$|\psi\rangle = \sum_{x,y,z,\mathbf{D}} \alpha_{x,y,z,\mathbf{D}} |x, y, z\rangle \otimes |\mathbf{D}\rangle,$$

then we can divide the basic states into four kinds $P, Q, R, S$. Where

1. $P$ be the projection onto the span of all basic states $|x, y, z\rangle \otimes |\mathbf{D}\rangle$ with $(x', y_0) \in \mathbf{D}$ and $y_0 \in S_r'$. (In this way, $\|P|\psi\rangle\|^2$ is just the probability that $D$ contains at least one pre-image of $S_r'$)
2. $Q$ be the projection onto the span of all states $|x, y, z\rangle \otimes |\mathbf{D}\rangle$ such that (a) there is no $(x', y_0) \in \mathbf{D}, y_0 \in S_r'$, and (b) there is no $(x, y') \in \mathbf{D}$, and (c) $y \neq 0$.
3. $R$ be the projection onto the span of all states $|x, y, z\rangle \otimes |\mathbf{D}\rangle$ satisfying that (a) there is $(x, y') \in \mathbf{D}, y \in [KN]/S_r'$, and (b) there is not $(x', y_0) \in \mathbf{D}, y_0 \in S_r'$ and (c) $y \neq 0$.
4. $T$ be the projection onto the span of all states $|x, y, z\rangle \otimes |\mathbf{D}\rangle$ such that (a) $\mathbf{D}$ does not contain $(x', y_0)$ and (b) $y = 0$.

According to the classification above, it is easy to see that

$$\|P\mathcal{O}P|\psi\rangle\| \leq \|P|\psi\rangle\|, \text{ and } \|P\mathcal{O}T|\psi\rangle\| = 0\,. \tag{11}$$

For convenience, we call a pair $(x, y)$ *good* iff $y \in S_r'$ for the rest of our proof. For a basic states $|x, y, z\rangle \otimes |\mathbf{D}\rangle$, there is an extra good pair in $\mathbf{D}$ after a query iff it's in the support of $Q$ or $R$.

Consider $Q$. From the definitions of $P, Q$, we see that

$$\begin{aligned} P\mathcal{O}Q|\psi\rangle &= P \cdot \sum_{x,y\neq 0,z,\mathbf{D},w} \alpha_{x,y,z,\mathbf{D}}|x,y,z\rangle \otimes \frac{(-1)^{y\cdot w}}{\sqrt{KN}}|\mathbf{D}\cup(x,w)\rangle \\ &= \sum_{x,y\neq 0,z,\mathbf{D}} \sum_{w\in S'_r} (\frac{(-1)^{y\cdot w}}{\sqrt{KN}}\alpha_{x,y,z,\mathbf{D}})|x,y,z\rangle \otimes |\mathbf{D}\cup(x,w)\rangle. \end{aligned}$$

Therefore, $\|P\mathcal{O}Q|\psi\rangle\|^2 \leq \frac{|S'_r|}{KN}\|Q|\psi\rangle\|^2$. Where

$$|S'_r| = |s_{n_1+\cdots+n_{r-1}+1}| + \cdots + |s_{n_1+\cdots+n_{r-1}+n_r}| \leq n_r KNp(S_r).$$

That is,

$$\|P\mathcal{O}Q|\psi\rangle\| \leq \sqrt{n_r p(S_r)}\|Q|\psi\rangle\|\,. \tag{12}$$

On the other hand, there is:

$$\begin{aligned} P\mathcal{O}R|\psi\rangle &= P\mathcal{O}\sum_{x,y,z,\mathbf{D},y'} \alpha_{x,y,z,\mathbf{D},y'}|x,y,z\rangle \otimes |\mathbf{D}\cup(x,y')\rangle \\ &= \sum_{x,y,z,\mathbf{D}} \Big(\sum_{w\in S'_r}\sum_{y'} -\frac{(-1)^{y\cdot(w\oplus y')}}{KN}\alpha_{x,y,z,\mathbf{D},y'}\Big)|x,y,z\rangle \otimes |\mathbf{D}\cup(x,w)\rangle \end{aligned}$$

Then from the Cauchy-Schwartz inequality,

$$\|P\mathcal{O}R|\psi\rangle\|^2 \leq \frac{K^2N^2 n_r p(S_r)}{K^2N^2} \cdot \sum_{x,y,z,\mathbf{D},y'} \|\alpha_{x,y,z,\mathbf{D},y'}\|^2 = n_r p(S_r)\|R|\psi\rangle\|^2.$$

Therefore,

$$\|P\mathcal{O}R|\psi\rangle\| \leq \sqrt{n_r p(S_r)}\|R|\psi\rangle\|\,. \tag{13}$$

In conclusion, we have: $\|P\mathcal{O}|\psi\rangle\| \leq \|P|\psi\rangle\| + \sqrt{n_r p(S_r)}$, that is, after $q$ queries, $\mathbf{D}$ contains a pre-image of $S'_r$ with probability at most $O(q^2 n_r p(S_r))$.

Using the similar idea from [23], one may extend the conclusion above to the case that $\mathbf{D}$ contains at least $i$ pre-image of $S'_r$. We can think of the pre-images numbers of $S'_r$ in $\mathbf{D}$ and $i$ as a counter, which is initially set to $i = 0$. Each query is going to turn a binary pair in $\mathbf{D}$ that doesn't meet the criteria into one that meets the requirements with probability $O(\sqrt{n_r p(S_r)})$, and then to increment the number in the counter by 1.

More formally, we define $P_i$ be the projection onto the span of all states $|x,y,z\rangle \otimes |\mathbf{D}\rangle$ with $\mathbf{D}$ containing at least $i$ pre-image of $S'_r$. Then we get

$$\|P_i\mathcal{O}|\psi\rangle\| \leq \|P_i|\psi\rangle\| + \sqrt{n_r p(S_r)}\|P'_{i-1}|\psi\rangle\| \leq \|P_i|\psi\rangle\| + \sqrt{n_r p(S_r)}\|P_{i-1}|\psi\rangle\|$$

for any $i \geq 1$. The notation $P'_{i-1}$ indicates that the projection onto the span of all states $|x,y,z\rangle \otimes |\mathbf{D}\rangle$ with $\mathbf{D}$ containing exact $i-1$ pre-image of $S'_r$.

It's obvious that $\|P'_{i-1}|\psi\rangle\| \le \|P_{i-1}|\psi\rangle\|$ and $\|P'_j|\psi_{start}\rangle\| = 0$ for any integer $j$. After $q$ quantum queries that $\mathbf{D}$ contains at least $i$ pre-image iff the counter has changed at least $i$ times. Then after $q$ quantum queries, it holds that, by Stirling's approximation,

$$\begin{aligned}\|P_i|\psi_{end}\rangle\| &\le \|P_i|\psi_{start}\rangle\| + C(q,i)\cdot\left(\sqrt{n_r p(S_r)}\right)^i \\ &< (q\sqrt{n_r p(S_r)})^i/i! < \left(e\,q\sqrt{n_r p(S_r)}/i\right)^i. \end{aligned} \tag{14}$$

Hence we get the conclusion of Lemma 10. □

By the similar ideas as above, we can also get the following result.

**Lemma 11.** *For any quantum algorithm making queries to compressed $\mathcal{O}$, then after one query to a basic states $|x,y,z\rangle\otimes|\mathbf{D}\rangle$, the amplitude on $\mathbf{D}$ containing a solution of Problem 2 can only increase by $O(\sqrt{i\,p(S_r)})$ , where $i$ is the number of good pairs in database $\mathbf{D}$.*

*Proof of Lemma 11.* We only need to make minor changes to the partition projection for our purpose. To divide the basic states into the following kinds $P', Q', R', T'$ as follows.

1. $P'$ be the projection onto the span of all states that $\exists(x_1,y_1),(x_2,y_2)\in\mathbf{D}$ satisfying $y_1,y_2\in s_k, s_k\subseteq S'_r$(namely $\mathbf{D}$ contains a solution of Problem 2).
2. $Q'$ be the projection onto the span of all states $|x,y,z\rangle\otimes|\mathbf{D}\rangle$ satisfying that (a) $\neg\exists(x_1,y_1),(x_2,y_2)\in\mathbf{D}$ such that $y_1,y_2\in s_k, s_k\subseteq S'_r$. (b) $\neg\exists(x,y')\in\mathbf{D}$, and (c) $y\ne 0$.
3. $R'$ be the projection onto the span of all state $|x,y,z\rangle\otimes|\mathbf{D}\rangle$ satisfies that (a) $\neg\exists(x_1,y_1),(x_2,y_2)\in\mathbf{D}$ such that $y_1,y_2\in s_k, s_k\subseteq S'_r$. (b) $\exists(x,y')\in\mathbf{D}$ whether $y'$ is a member of $S'_r$ or not, and (c) $y\ne 0$.
4. $T'$ be the projection onto the span of all states $|x,y,z\rangle\otimes|\mathbf{D}\rangle$ satisfying that $\neg\exists(x_1,y_1),(x_2,y_2)\in\mathbf{D}$ such that $y_1,y_2\in s_k, s_k\subseteq S'_r$ and $y=0$.

Any basic states must be contained in one of the support of $P', Q', R', T'$. Similarly it's obvious that

$$\|P'\mathcal{O}|x,y,z\rangle\otimes|\mathbf{D}\rangle\| \le \|P'|x,y,z\rangle\otimes|\mathbf{D}\rangle\|$$

for basic state $|x,y,z\rangle\otimes|\mathbf{D}\rangle$ in the support of $P'$ and

$$\|P'\mathcal{O}|x,y,z\rangle\otimes|\mathbf{D}\rangle\| = 0$$

for basic state $|x,y,z\rangle\otimes|\mathbf{D}\rangle$ in the support of $T'$, which means that the Lemma 11 holds in these cases.

Since the proofs for the remaining two cases are similar and the results are the same, only one of them is proved here. For basic state $|x,y,z\rangle\otimes|\mathbf{D}\rangle$ in the support of $Q'$, after $q$'th queries, a new binary pair will be added to the database, so we have

$$P'\mathcal{O}|x,y,z\rangle\otimes|\mathbf{D}\rangle = |x,y,z\rangle\otimes\sum_{w\in\mathcal{S}}\frac{(-1)^{y\cdot w}}{\sqrt{KN}}|\mathbf{D}\cup(x,w)\rangle. \tag{15}$$

Where $\mathcal{S}$ is a union of $s_{k_j}$ which satisfies that $\exists (x,y) \in \mathbf{D}, y \in s_{k_j}$. We denote $\mathbb{K}$ as the collection of all such $k_j$ and $\mathbb{K} \subseteq S_r$. Then $\mathcal{S} = \bigcup\limits_{k \in \mathbb{K}} s_k$. The cardinality of $\mathcal{S}$ is determined by the composition of $\mathbf{D}$ in the basic state. Since $\mathbf{D}$ contains exactly $i$ good pairs just before $q$'th query, namely $|\mathbb{K}| = i$, and

$$|\mathcal{S}| = \sum_{k \in \mathbb{K}} \lfloor KNp_k \rfloor \leq i\, KNp(S_r). \tag{16}$$

From (15) and (16) we get

$$\|P'\mathcal{O}|x,y,z\rangle \otimes |\mathbf{D}\rangle\|^2 \leq \frac{|\mathcal{S}|}{KN} \leq i\, p(S_r).$$

In other words, we have

$$\|P'\mathcal{O}|x,y,z\rangle \otimes |\mathbf{D}\rangle\| \leq \sqrt{i\, p(S_r)}\,. \tag{17}$$

In conclusion, for any quantum algorithm that make queries to compress random oracle $\mathcal{O}$, then after one query to a basic states $|x,y,z\rangle \otimes |\mathbf{D}\rangle$, the amplitude on $\mathbf{D}$ containing a solution of Problem 2 can only increase by $\sqrt{ip(S_r)}$ , where $i$ is the number of good pairs in database $\mathbf{D}$. That concludes Lemma 11. □

To denote $\mathbf{\Phi}_i$ as the set of databases containing exactly $i$ good pairs. The discussions above allow us to get

$$\begin{aligned}
\|P'\mathcal{O}|\psi\rangle\| &\leq \|P'|\psi\rangle\| + \|P'\mathcal{O}\cdot(I-P')|\psi\rangle\| \\
&\leq \|P'|\psi\rangle\| + \Big\| \sum_{i=1}^{n_r} \Big( \sqrt{i\, p(S_r)} \sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_i} \alpha_{x,y,z,\mathbf{D}} |x,y,z\rangle \otimes |\mathbf{D}\rangle \Big) \Big\| \\
&\leq \|P'|\psi\rangle\| + \Big( \sum_{i=1}^{n_r} \big( i\, p(S_r) \sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_i} \alpha^2_{x,y,z,\mathbf{D}} \big) \Big)^{1/2}.
\end{aligned} \tag{18}$$

With these preparations, we finally come to the proof of Theorem 6.

*Proof of Theorem 6.* Let's start when $n_r = \Theta(1)$, namely there exists a constant $C > 0$ such that $n_r < C$.

In this case, we denote $\mathbf{\Phi'}$ the set of databases which contains at least one good pair just before $q$-th query, then:

$$\begin{aligned}
\Big( \sum_{i=1}^{n_r} \big( i\, p(S_r) \sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_i} \alpha^2_{x,y,z,\mathbf{D}} \big) \Big)^{1/2} &\leq \Big( C\, p(S_r) \sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi'}} \alpha^2_{x,y,z,\mathbf{D}} \Big)^{1/2} \\
&\leq \Big( C\, p(S_r) \cdot e\, q \sqrt{C\, p(S_r)} \Big)^{1/2} \\
&\leq e^{1/2} C^{3/4} q^{1/2} p^{3/4}(S_r).
\end{aligned}$$

Together with (18), we get

$$\|P'\mathcal{O}|\psi\rangle\| \leq \|P'|\psi\rangle\| + e^{1/2} C^{3/4} q^{1/2} p^{3/4}(S_r).$$

So after $q$ queries, the success probability $\mathbf{P}_2$ in solving Problem 2, as defined in Theorem 5, will be

$$\begin{aligned}\sqrt{\mathbf{P}_2} = \|P'|\psi_{end}\rangle\| &\leq \|P'|\psi_{start}\rangle\| + \sum_{i=1}^{q} e^{1/2} C^{3/4} q^{1/2} p^{3/4}(S_r) \\ &< e^{1/2} C^{3/4} q^{3/2} p^{3/4}(S_r).\end{aligned}$$

So in this case, a quantum algorithm with $q$ queries can solve Problem 2 with probability at most $O(q^3 p^{3/2}(S_r))$. In other words, $\Omega((n_r p^3(S_r))^{-1/6})$ quantum queries are necessary to solve Problem 2 with constant probability for any quantum algorithm.

Now, we consider the remaining case when $n_r$ is not a constant. That is, it holds that $1/n_r = o(1)$.

Let $j_i := \max\{2e \cdot i\sqrt{n_r p(S_r)}, n_r^{1/4}\}$ for any $i \in [q]$. We have

$$\begin{aligned}&\Big(\sum_{i=1}^{n_r} \big(i\, p(S_r) \sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_i} \alpha^2_{x,y,z,\mathbf{D}}\big)\Big)^{1/2} \\ &\quad \leq \sqrt{\sum_{i=1}^{j_q-1} \Big(i\, p(S_r) \sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_i} \alpha^2_{x,y,z,\mathbf{D}}\Big)} + \sqrt{\sum_{i=j_q}^{n_r} \Big(i\, p(S_r) \sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_i} \alpha^2_{x,y,z,\mathbf{D}}\Big)} \\ &\quad \leq \sqrt{j_q p(S_r) \sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_i, i<j_q} \alpha^2_{x,y,z,\mathbf{D}}} + \sqrt{n_r p(S_r) \sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_i, i\geq j_q} \alpha^2_{x,y,z,\mathbf{D}}} \\ &\quad \leq \sqrt{j_q p(S_r)\cdot 1} + \sqrt{n_r p(S_r)} \cdot (\frac{e\, q\sqrt{n_r p(S_r)}}{j_q})^{j_q}.\end{aligned}$$

Again with (18), we get, in this case

$$\|P'O|\psi\rangle\| \leq \|P'|\psi\rangle\| + \sqrt{j_q p(S_r)} + \sqrt{n_r p(S_r)} \cdot (\frac{e\, q\sqrt{n_r p(S_r)}}{j_q})^{j_q}. \tag{19}$$

Since $j_i$ is an increasing sequence, it has

$$\begin{aligned}\|P'|\psi_{end}\rangle\| &\leq \sum_{i=1}^{q} \Big(\sqrt{j_i p(S_r)\cdot 1} + \sqrt{n_r p(S_r)} \cdot (\frac{e\, i\sqrt{n_r p(S_r)}}{j_i})^{j_i}\Big) \\ &= \sum_{i=1}^{q} \sqrt{j_i p(S_r)\cdot 1} + \sum_{i=1}^{q} \sqrt{n_r p(S_r)} \cdot \Big(\frac{e\, i\sqrt{n_r p(S_r)}}{j_i}\Big)^{j_i} \\ &\leq \sqrt{j_q p(S_r)} + \sum_{i=1}^{q} \sqrt{n_r p(S_r)} \cdot \Big(\frac{e\, i\sqrt{n_r p(S_r)}}{j_i}\Big)^{j_i}.\end{aligned} \tag{20}$$

According to the definition of $j_i$, it holds that

$$\sqrt{n_r p(S_r)} \cdot \Big(\frac{e\, i\sqrt{n_r p(S_r)}}{j_i}\Big)^{j_i} \leq \sqrt{n_r p(S_r)} \cdot (\frac{1}{2})^{n_r^{1/4}},$$

and hence

$$\sum_{i=1}^{q} \sqrt{n_r p(S_r)} \cdot \Big(\frac{e\, i \sqrt{n_r p(S_r)}}{j_i}\Big)^{j_i} \leq q\sqrt{n_r p(S_r)} \cdot \Big(\frac{1}{2}\Big)^{n_r^{1/4}}. \tag{21}$$

Since one may, in order for solving Problem 2, use Grover's algorithm twice to find two distinct inputs $x, x'$ such that $g(x), g(x') \in s_i$ for $i \in S_r$. In other words, there is a ready upper bound $O(p^{-1/2}(S_r))$ for Problem 2, so we assume $q = O(p^{-1/2}(S_r))$. These, together with (20) and (21), give

$$\begin{aligned}\sqrt{\mathbf{P}_2} = \|P'|\psi_{end}\rangle\| &\leq q\sqrt{j_q p(S_r)} + \sum_{i=1}^{q} \sqrt{n_r p(S_r)} \cdot \Big(\frac{e\, i\sqrt{n_r p(S_r)}}{j_i}\Big)^{j_i} \\ &\leq q\sqrt{j_q p(S_r)} + q\sqrt{n_r p(S_r)} \cdot \Big(\frac{1}{2}\Big)^{n_r^{1/4}} \\ &\leq q\sqrt{j_q p(S_r)} + n_r^{1/2} \cdot \Big(\frac{1}{2}\Big)^{n_r^{1/4}}. \end{aligned} \tag{22}$$

From $j_q = \max\{2e \cdot q\sqrt{n_r p(S_r)}, n_r^{1/4}\} \geq n_r^{1/4}$, it holds that

$$\sqrt{\frac{j_q}{n_r}} \geq n_r^{-3/8} = \Omega(2^{-n_r^{1/4}}).$$

That is, $1/2^{n_r^{1/4}} = O(\sqrt{j_q/n_r})$. So we have

$$\begin{aligned}\sqrt{\mathbf{P}_2} &\leq q\sqrt{j_q p(S_r)} + q\sqrt{n_r p(S_r)} \cdot \Big(\frac{1}{2}\Big)^{n_r^{1/4}} \\ &= q\sqrt{n_r p(S_r)} \cdot \sqrt{\frac{j_q}{n_r}} + q\sqrt{n_r p(S_r)} \cdot \Big(\frac{1}{2}\Big)^{n_r^{1/4}} \\ &\leq q\sqrt{n_r p(S_r)} \cdot \Big(\sqrt{\frac{j_q}{n_r}} + \Big(\frac{1}{2}\Big)^{n_r^{1/4}}\Big) \\ &\leq q\sqrt{n_r p(S_r)} \cdot (1 + O(1))\sqrt{\frac{j_q}{n_r}} \\ &= O\Big(q\sqrt{j_q p(S_r)}\Big). \end{aligned} \tag{23}$$

Next we consider two cases according to the value of $j_q$.

When $j_q = n_r^{1/4}$, from (23), we have

$$\mathbf{P}_2 = \|P'|\psi_{end}\rangle\|^2 \leq O\Big(q^2 j_q p(S_r)\Big) = O\Big(q^2 n_r^{1/4} p(S_r)\Big). \tag{24}$$

However, when $j_q = n_r^{1/4}$, we have

$$q < \frac{n_r^{1/4}}{2e\sqrt{n_r p(S_r)}}.$$

These, together with (24) and $n_r = \omega(1)$, give

$$\mathbf{P}_2 = O(n_r^{-1/4}) = o(1).$$

When $j_q = 2e \cdot q\sqrt{n_r p(S_r)}$, we have

$$\mathbf{P}_2 = O\Big(q^3 n_r^{1/2} p^{3/2}(S_r)\Big). \tag{25}$$

It is not hard to see that the Equations (24) and (25) and the discussions above indicate that if the success probability for a collision-finding quantum algorithm with $q$ queries is a constant, it should be

$$q = \Omega((n_r p^3(S_r))^{-1/6}) = \Omega(\gamma_r^{1/6})\,.$$

Which is the conclusion of Theorem 6. □

As a variant of Theorem 6, the following result is obtained directly from Equations (24) and (25).

**Corollary 1.** *For any quantum algorithm by making $q$ queries, the success probability in solving Problem 2 is at most $O(\max\{q^3 n_r^{1/2} p^{3/2}(S_r), q^2 n_r^{1/4} p(S_r)\})$.*

In the next section, we will work out the lower bound for collision-finding in the non-uniform random functions by this corollary.

### 4.4 The Lower Bound for Collision-Finding

We now explore a quantum query lower bound for collision-finding problem with respect to non-uniform distributions.

Recall the definition that $S'_r := \bigcup_{j \in S_r} s_j$ as in last section, we get

$$\begin{aligned}
&\Pr_{f \leftarrow D^M}\left[f(x) = f(x'), x \neq x' : (x, x') \leftarrow \mathcal{A}^f\right] \\
&\quad = \Pr_{f \leftarrow D^M}[f(x) = f(x'), x \neq x', f(x) \in \bigcup_{r=1}^{\ell} S_r : (x, x') \leftarrow \mathcal{A}^f] \qquad (26)\\
&\quad \leq \sum_{r=1}^{\ell} \Pr_{f \leftarrow D^M}[f(x) = f(x'), x \neq x', f(x) \in S_r : (x, x') \leftarrow \mathcal{A}^f]
\end{aligned}$$

According to Theorem 5, there is an algorithm $\mathcal{B}$ such that

$$\begin{aligned}
&\Pr_{f \leftarrow D^M}[f(x) = f(x'), x \neq x', f(x) \in S_r : (x, x') \leftarrow \mathcal{A}^f] \\
&\quad \leq \Pr_{g \leftarrow (KN)^M}[g(x), g(x') \in s_j, x \neq x', s_j \subset S'_r : (x, x') \leftarrow \mathcal{B}^g] + q^2/K
\end{aligned}$$

which in turn implies from (26)

$$\begin{aligned}&\Pr_{f\leftarrow D^M}\left[f(x)=f(x'), x\neq x' : (x,x')\leftarrow \mathcal{A}^f\right]\\ &\quad\le \sum_{r=1}^{\ell}\Pr_{g\leftarrow (KN)^M}\left[g(x), g(x')\in s_j, x\neq x', s_j\subset S'_r : (x,x')\leftarrow \mathcal{B}^g\right]+\ell q^2/K.\end{aligned}$$

By Corollary 1, we have

$$\begin{aligned}&\Pr[g(x), g(x')\in s_j, x\neq x', s_j\subset S'_r : (x,x')\leftarrow \mathcal{B}^g, g\leftarrow (KN)^M]\\ &\qquad\le O(\max\{q^3 n_r^{1/2}p^{3/2}(S_r), q^2 n_r^{1/3}p(S_r)\})\end{aligned}\tag{27}$$

for any $i\in[\ell]$. To combine Inequalities (26) and (27), we have

$$\begin{aligned}&\Pr[f(x)=f(x'), x\neq x' : (x,x')\leftarrow A^f, f\leftarrow D^M]\\ &\quad\le \ell q^2/K+\sum_{r=1}^{\ell}O\big(\max\{q^3 n_r^{1/2}p^{3/2}(S_r), q^2 n_r^{1/3}p(S_r)\}\big)\qquad(28)\\ &\quad\le O\Big(\sum_{r=1}^{\ell}\max\{q^3 n_r^{1/2}p^{3/2}(S_r), q^2 n_r^{1/3}p(S_r)\}\Big).\end{aligned}$$

The last inequality holds since $\ell/K$ can be as small as required with large enough $K$. In fact, when distribution $D$ is given, $\ell$ is fixed.

By the upper bound for $q$ in Theorem 2, we assume $q<\gamma^{1/6}(c)\le \gamma_r^{1/6}(c)$. That implies

$$\max\{q^3 n_r^{1/2}p^{3/2}p(S_r), q^2 n_r^{1/3}p(S_r)\}=q^2 n_r^{1/3}p(S_r).$$

for any any $r\in[\ell]$. Hence from Inequality (28), we have:

$$\begin{aligned}\Pr_{f\leftarrow D^M}[f(x)=f(x'), x\neq x' : (x,x')\leftarrow A^f]&\le O(q^2\sum_{r=1}^{\ell}n_r^{1/3}p(S_r))\\ &=O(q^2\cdot\delta_c\max_r\{(n_r p^3(S_r))^{1/3}\})=O(\delta_c q^2\gamma^{-1/3}(c)).\end{aligned}\tag{29}$$

Where

$$\delta_c := \frac{\sum_{r=1}^{\ell}(n_r p^3(S_r))^{1/3}}{\max_r\{(n_r p^3(S_r))^{1/3}\}}=\frac{\sum_{r=1}^{\ell}\gamma_r^{-1/3}(c)}{\gamma^{-1/3}(c)}.$$

The Inequality (29) implies that if success probability for collision-finding is a constant, then $q=\Omega(\gamma^{1/6}(c)\cdot\delta_c^{-1/2})$. This gives the main conclusion of this section as follows.

**Theorem 7.** *For any quantum collision-finding algorithm with respect to a non-uniform distribution, $\Omega(\gamma^{1/6}(c)\cdot\delta_c^{-1/2})$ queries are necessary to find a collision with constant probability.*

That the lower bound obtained here is an almost tight one compared with the upper bound from Theorem 2.

Specially, when $D$ is a uniform distribution, for any constant $c > 1$, one can easily get $\delta_c = 1$ and $\gamma(c) = N^2$, and hence the upper bound and lower bound here meet as $\Theta(\gamma^{1/6}(c)) = \Theta(N^{1/3})$ as shown in [30].

We let readers convince themselves that the lower bound and the upper bound here also meets on flat-$\mathtt{k}$-distribution and $\delta$-$\mathtt{k}$ distribution (cf. [7]), respectively. The two distributions are referred there to illustrate the algorithms's optimality. These hint that the upper bound and lower bound obtained in this work possess more generality.

In the following section, we will estimate the value of $\delta_c$ to derive a concise lower bound.

### 4.5 Estimation of Upper Bound of $\delta_c$

We now estimate $\delta_c$ appeared in the lower bound in last section. We prove the following upper bound for $\delta_c$ mentioned above. That will lead a nearly tight lower bound.

**Proposition 3.** *We have $\delta_c = O(\mathtt{k})$ for any non-uniform distribution $D$ and any constant $c > 1$. Where $\mathtt{k} = -\log p_1$ is the min-entropy of $D$.*

*Proof.* Let $b_i := \gamma_i^{-1/3}(c)/\gamma^{-1/3}(c)$ for all $i \in [\ell]$. We rearrange the list $b_1, \ldots, b_\ell$ in non-increasing order and denote it as $a_1, \ldots, a_\ell$ with $a_1 = 1, a_i \le 1$ for any $i \in [\ell]$ and obviously

$$\sum_{i=1}^{\ell} a_i = \sum_{i=1}^{\ell} b_i = \frac{\sum_{r=1}^{\ell} \gamma_r^{-1/3}(c)}{\gamma^{-1/3}(c)} = \delta_c \tag{30}$$

Let $a_k = \gamma_{i_k}^{-1/3}(c)/\gamma^{-1/3}(c)$, and hence $a_k^3 = n_{i_k} p^3(S_{i_k})/n_{k_0} p^3(S_{i_{k_0}})$, where $k_0$ is the collision domain index in Definition 3. By $c$-partition, it holds that $n_{i_k} p(S_{i_k}) \le c$. Hence we have

$$c\, p(S_{i_k})^2 \ge n_{i_k} p(S_{i_k})^3 = a_k^3 \gamma^{-1} \ge a_k^3 \cdot 2^{-3\,\mathtt{k}}.$$

The last inequality is by Proposition 4 that $\gamma^{1/6}(c) \le 2^{\mathtt{k}/2}$ for any constant $c > 1$. It implies, for any $k \in [\ell]$

$$p(S_{i_k}) \ge \frac{1}{\sqrt{c}} \cdot a_k^{3/2} \cdot 2^{-3\,\mathtt{k}/2}. \tag{31}$$

It is easy to see that $\delta_c \le \ell$, otherwise $\delta_c = \sum_{i=1}^{\ell} a_i \le \ell \cdot a_1 < \delta_c$ for decreasing sequence $a_i$ with $a_1 = 1$, which is impossible.

Next, we show there is a $j \le \lfloor \frac{l}{\lfloor \delta_c/2 \rfloor} \rfloor$ that satisfies $a_{j\lfloor \delta_c/2 \rfloor} \ge 1/(j+1)^4$. For otherwise, if $a_{j\lfloor \delta_c/2 \rfloor} < 1/(j+1)^4$ for all $j \le \lfloor \frac{l}{\lfloor \delta_c/2 \rfloor} \rfloor$, then by (30),

$$\begin{aligned}\delta_c &= \sum_{i=1}^{\ell} a_i = \sum_{i=1}^{\lfloor \frac{\delta_c}{2} \rfloor} a_i + \sum_{i=\lfloor \frac{\delta_c}{2} \rfloor+1}^{2\lfloor \frac{\delta_c}{2} \rfloor} a_i + \cdots + \sum_{i=\lfloor \frac{l}{\lfloor \delta_c/2 \rfloor} \rfloor \cdot \lfloor \frac{\delta_c}{2} \rfloor+1}^{\ell} a_i \\ &< \lfloor \frac{\delta_c}{2} \rfloor \cdot \Big(1 + \frac{1}{2^4} + \cdots + \frac{1}{(\lfloor \frac{2\ell}{\delta_c} \rfloor + 1)^4}\Big) < \lfloor \frac{\delta_c}{2} \rfloor \cdot \sum_{j=1}^{\infty} \frac{1}{j^4} < \frac{9}{8} \cdot \lfloor \delta_c/2 \rfloor < \delta_c.\end{aligned}$$

This is a contradiction. Let $j_0$ be the smallest $j \leq \lfloor \frac{l}{\lfloor \delta_c/2 \rfloor} \rfloor$ satisfying the inequality $a_{j_0 \lfloor \frac{\delta_c}{2} \rfloor} \geq 1/(j+1)^4$. Together with (31), it implies, for all $k \in \big[j_0 \lfloor \frac{\delta_c}{2} \rfloor\big]$

$$\begin{aligned}p(S_{i_k}) &\geq \frac{1}{\sqrt{c}} \cdot a_k^{3/2} \cdot 2^{-3\mathtt{k}/2} \\ &\geq \frac{1}{\sqrt{c}} \cdot a_{j_0 \lfloor \frac{\delta_c}{2} \rfloor}^{3/2} \cdot 2^{-3\mathtt{k}/2} \geq \frac{1}{\sqrt{c} \cdot (j_0+1)^6 \cdot 2^{3\mathtt{k}/2}}. \qquad (32)\end{aligned}$$

On the other hand, from the definition of min-entropy $\mathtt{k}$, for all $k \in \big[j_0 \lfloor \frac{\delta_c}{2} \rfloor\big]$,

$$1/2^{\mathtt{k}} = p_1 \geq p(S_{i_k}) \qquad (33)$$

According to (32) and (33), we get, by definition of $c$-partition,

$$\frac{1}{2^{\mathtt{k}}} \cdot \sqrt{c} \cdot (j_0+1)^6 \cdot 2^{\frac{3\mathtt{k}}{2}} \geq \frac{\max\{p(S_{i_k})\}_{k \in j_0 \lfloor \frac{\delta_c}{2} \rfloor}}{\min\{p(S_{i_k})\}_{k \in j_0 \lfloor \frac{\delta_c}{2} \rfloor}} \geq c^{j_0 \lfloor \frac{\delta_c}{2} \rfloor}.$$

Namely, we have

$$\delta_c \leq \frac{\mathtt{k} + 12 \log_2(j_0+1) + \log_2 c}{j_0 \log_2 c} + 2 = O(\mathtt{k}).$$

As desired. □

This conclusion together with Theorem 7 shows that $\Omega(\gamma^{1/6}(c)/\sqrt{\mathtt{k}})$ quantum queries are necessary to find a collision for any non-uniform random function and any constant $c > 1$. Combining with Proposition 4 which we will prove in the next section, we can get our final lower bound: $\Omega(\gamma^{1/6}(c)/\sqrt{\log \gamma(c)})$, compared with the upper bound $O(\gamma^{1/6}(c))$ in Theorem 2, that it is nearly a tight lower bound.

## 5 An Analysis on the Properties of $\gamma(c)$

In Definition 2, we have introduced the parameter $\gamma(c)$. In this section, we explore some properties of $\gamma(c)$. We will present the relations of $\gamma$ with the min-entropy $\mathtt{k}$ and collision-entropy $\beta$, respectively. An upper bound of $p(S_{k_0})$ is given so as to help to simplify the calculation of collision parameter and collision domain whenever using Algorithm 2. In the last, we show that the constant $c$ in partition does not affect the magnitude of query complexity in collision-finding.

First of all, we point out a fact that will be frequently used in the subsequent proofs. Since $\sum_{i=1}^{N} p_i = 1$, we have by $c$-partition that, for any $k \in [\ell]$:

$$n_k p(S_k) \leq \sum_{i=1}^{\ell} n_i p(S_i) = c \sum_{i=1}^{\ell} \frac{n_i p(S_i)}{c} \leq c \sum_{i=1}^{N} p_i = c. \tag{34}$$

We now show a relation between parameter $\gamma$ and min-entropy $\mathtt{k}$.

**Proposition 4.** *For any non-uniform distribution $D$ and any constant $c > 1$, it holds that $2^{2\,\mathtt{k}}/c \leq \gamma(c) \leq 2^{3\,\mathtt{k}}$. Where $\mathtt{k} = -\log p_1$ is the min-entropy.*

*Proof.* By definition of $\gamma(c)$, $n_i = |S_i|$ for all $i \in [\ell]$ and $n_1 \geq 1$, we get at once

$$\gamma^{-1}(c) = \max_{i \in [\ell]} \{n_i p^3(S_i)\} \geq n_1 p_1^3 \geq p_1^3 = 2^{-3\,\mathtt{k}}.$$

Which implies $\gamma(c) \leq 2^{3\mathtt{k}}$.

On the other hand, according to (34), we have

$$\gamma^{-1}(c) = n_{k_0} p^3(S_{k_0}) \leq c\, p^2(S_{k_0}) \leq c p_1^2 \leq c \cdot 2^{-2\,\mathtt{k}}.$$

That is $\gamma(c) \geq 2^{2\,\mathtt{k}}/c$. As desired in Proposition 4. □

The results in Propositions 3, Proposition 4 and in Theorem 7 claim the following conclusion.

**Corollary 2.** *The number $\Omega(\gamma^{1/6}(c)/\sqrt{\log \gamma(c)})$ of quantum queries are necessary for any algorithms of collision-finding in random functions.*

The relation between $\gamma$ and collision-entropy $\beta$ is shown as follow.

**Proposition 5.** *For any non-uniform distribution $D$ and any constant $c > 1$, we have $\frac{1}{c} \cdot 2^{\beta} \leq \gamma(c) < \frac{16c^3}{(c-1)^2} \cdot 2^{2\beta}$.*

*Proof.* Recall that $\gamma^{-1}(c) = \max_{i \in [\ell]} \{n_i p^3(S_i)\} = n_{k_0} p^3(S_{k_0})$. From (34), we know

$$\gamma^{-1}(c) \leq c\, p^2(S_{k_0}) \leq c \cdot \sum_{i=1}^{N} p_i^2 = c \cdot 2^{-\beta}.$$

That is $2^{\beta}/c \leq \gamma(c)$.

On the other side, since $p(S_i) > c\, p(S_{i+1})$ by $c$-partition for any $i \in [\ell - 1]$. We have

$$
\begin{aligned}
2^{-2\beta} &= \Big(\sum_{i=1}^{N} p_i^2\Big)^2 \le \Big(n_1 p^2(S_1) + \cdots + n_\ell p^2(S_\ell)\Big)^2 \\
&\le \Big(\sum_{j=1}^{k_0} n_{k_0} p^2(S_{k_0}) \cdot \big(p(S_{k_0})/p(S_j)\big) + \sum_{j=k_0+1}^{\ell} n_j p^2(S_j)\Big)^2 \\
&\le \Big(\sum_{j=1}^{k_0} \frac{1}{c^{k_0-j}} n_{k_0} p^2(S_{k_0}) + \sum_{j=k_0+1}^{\ell} n_j p^2(S_j)\Big)^2 \\
&< \Big(\frac{c}{c-1} \cdot n_{k_0} p^2(S_{k_0}) + \sum_{j=k_0+1}^{\ell} n_j p^2(S_j)\Big)^2 . \qquad (35)
\end{aligned}
$$

If let $\alpha > 0$ satisfy

$$
\sum_{j=k_0+1}^{\ell} n_j p^2(S_j) = \alpha \cdot n_{k_0} p^2(S_{k_0}). \qquad (36)
$$

Then, (35) becomes

$$
2^{-2\beta} < (\frac{c}{c-1} + \alpha)^2 \cdot n_{k_0} p(S_{k_0}) \cdot n_{k_0} p^3(S_{k_0}) \,. \qquad (37)
$$

While $\alpha$ can be bounded, for any non-uniform distribution $D$, as follows.

$$
\begin{aligned}
\alpha &= \frac{\sum\limits_{j=k_0+1}^{\ell} n_j p^2(S_j)}{n_{k_0} p^2(S_{k_0})} \le \sum_{j=k_0+1}^{\ell} \frac{p(S_{k_0})}{p(S_j)} \\
&\le \sum_{j=0}^{l-k_0} \frac{1}{c^j} \cdot \frac{p(S_{k_0})}{p(S_\ell)} < \frac{c}{c-1} \cdot \frac{p(S_{k_0})}{p(S_\ell)} < \frac{3\,c}{c-1} \cdot \frac{p(S_{k_0})}{p(S_\ell)} .
\end{aligned}
$$

Hence, we estimate $2^{-2\beta}$ respectively in the following three cases.

- When $0 < \alpha < \frac{3c}{c-1}$, from (37) we have

$$
2^{-2\beta} < (\frac{c}{c-1} + \frac{3\,c}{c-1})^2 \cdot c \cdot n_{k_0} p^3(S_{k_0}) < \frac{16\,c^3}{(c-1)^2} \cdot \gamma^{-1}(c). \qquad (38)
$$

- When $\frac{3\,c}{c-1} \le \alpha < \frac{3\,c}{c-1} \cdot \frac{p(S_{k_0})}{p(S_{k_0+1})}$, since from (36) that

$$
\alpha \cdot n_{k_0} p^2(S_{k_0}) = \sum_{j=k_0+1}^{\ell} n_j p^2(S_j) \le (\sum_{j=k_0+1}^{\ell} n_j p(S_j)) \cdot p(S_j) \le c\, p(S_{k_0+1}) \,.
$$

Therefore

$$n_{k_0} p(S_{k_0}) \le c\, p(S_{k_0+1})/\alpha\, p(S_{k_0}) < 3\, c^2/\alpha^2 (c-1).$$

Again, from (37) for any constant $c > 1$,

$$\begin{aligned} 2^{-2\beta} &< (\alpha + \frac{c}{c-1})^2\ \frac{3\, c^2\, \gamma^{-1}(c)}{(c-1)\cdot \alpha^2} \\ &\le (\frac{4\,\alpha}{3})^2\ \frac{3\, c^2\, \gamma^{-1}(c)}{(c-1)\cdot \alpha^2} = \frac{16\, c^2\, \gamma^{-1}(c)}{3(c-1)}. \end{aligned} \tag{39}$$

– When $\frac{3\,c}{c-1} \cdot \frac{p(S_{k_0})}{p(S_{k_0+k})} \le \alpha < \frac{3\,c}{c-1} \cdot \frac{p(S_{k_0})}{p(S_{k_0+k+1})}$ holds for some $k \in [\ell - k_0 - 1]$. From (36), we know

$$\begin{aligned} \alpha \cdot n_{k_0} p^2(S_{k_0}) &= \sum_{j=k_0+1}^{\ell} n_j p^2(S_j) \\ &\le \sum_{j=k_0+1}^{k_0+k} n_j p^2(S_j) + (\sum_{j=k_0+k+1}^{\ell} n_j p(S_j)) \cdot p(S_{k_0+k+1}) \\ &\le n_{k_0} p^2(S_{k_0}) \cdot \Big( \sum_{j=k_0+1}^{k_0+k} \frac{p(S_{k_0})}{p(S_j)} \Big) + c\, p(S_{k_0+k+1})\,. \end{aligned} \tag{40}$$

by $\frac{3\,c}{c-1} \cdot \frac{p(S_{k_0})}{p(S_{k_0+k})} \le \alpha$, we get

$$\sum_{j=k_0+1}^{k_0+k} \frac{p(S_{k_0})}{p(S_j)} < \sum_{j=0}^{k-1} \frac{1}{c^j} \cdot \frac{p(S_{k_0})}{p(S_{k_0+k})} < \frac{c}{c-1} \cdot \frac{p(S_{k_0})}{p(S_{k_0+k})} \le \frac{\alpha}{3}. \tag{41}$$

To combine (40) and (41) gives

$$\alpha \cdot n_{k_0} p^2(S_{k_0}) \le n_{k_0} p^2(S_{k_0}) \cdot \frac{\alpha}{3} + c\, p(S_{k_0+k+1}).$$

Therefore, using $\alpha < \frac{3c}{c-1} \cdot \frac{p(S_{k_0})}{p(S_{k_0+k+1})}$, simple computation will get

$$n_{k_0} p(S_{k_0}) \le \frac{3\, c\, p(S_{k_0+k+1})}{2\alpha p(S_{k_0})} < \frac{9\, c^2}{2(c-1)\alpha^2}$$

which implies, still from (37) and $c > 1$, that

$$2^{-2\beta} < (\alpha + \frac{c}{c-1})^2 \cdot \frac{9c^2}{2(c-1)\alpha^2} \cdot \gamma^{-1}(c) < \frac{16\, c^3}{(c-1)^2} \cdot \gamma^{-1}(c)\,. \tag{42}$$

Equations (38), (39) and (42) together affirm that

$$\gamma(c) < \frac{16\, c^3}{(c-1)^2} \cdot 2^{2\beta}\,.$$

As desired. □

From Table 1, we see Proposition 4 and 5 show that, in general non-uniform distribution, the upper bound and lower bound here in parameter $\gamma$ are always at least as good as the best prior results.

In the following, we present an example to show that the results here can be better in general.

*Example 1.* Let $M, N > 0$ be two integers such that $N := 2^{2n} - 2^{7n/4} + 1$ for a large integer $n > 0$, a non-uniform distribution $D$ with weights defined as follows.

$$p_i := \begin{cases} 2^{-n}, & \text{for } i = 1; \\ 2^{-5n/4}, & \text{for } i \in [2, 1 + 2^n]; \\ 2^{-2n}, & \text{for } i \in [2 + 2^n, 2^{2n} - 2^{7n/4} + 1]. \end{cases}$$

The calculation show that the min-entropy is $\mathtt{k} = n$, and the collision entropy is $\beta \approx 3n/2$ for sufficiently large $n$. Namely, it will have the upper bound: $O(2^{n/2})$ and the lower bound: $\Omega(2^{n/3})$ in [7].

If, for sufficiently large $n$ and any constant $c > 1$, we divide the set $[N]$ into three parts with $n_1 = 1, n_2 = 2^n, n_3 \approx 2^{2n}$, we will have

$$\gamma^{1/6}(c) = \gamma_2^{1/6}(c) = (2^{-n})^{1/6} \cdot (2^{5n/4})^{1/2} = 2^{11n/24}.$$

In this case, our lower bound is

$$\Omega(\gamma^{1/6}/\sqrt{\log \gamma}) = \Omega(2^{11n/24}/n) > \max\{2^{\beta/6}, 2^{\mathtt{k}/3}\} = 2^{n/3}.$$

And the upper bound in this work is

$$O(\gamma^{1/6}) = O(2^{11n/24}) < \min\{2^{\beta/3}, 2^{\mathtt{k}/2}\} = 2^{n/2}.$$

That is, in this case, both the upper bound and the lower bound obtained in this work are better than the best prior bounds.

The following technique result will allow to simplify the calculation of collision domain in $c$-partition.

**Proposition 6.** *For any non-uniform distribution $D$, we have*

$$p(S_{k_0}) \geq \max\left\{ c^{-1/2} \cdot 2^{-\frac{3\mathtt{k}}{2}}, \frac{c^2 - 1}{c^2 N} \right\}.$$

*Where $p(S_{k_0})$ satisfies $n_{k_0} p^3(S_{k_0}) = \gamma^{-1}(c)$, and $\mathtt{k} = -\log p_1$ is the min-entropy.*

*Proof.* According to Definition 3, we have

$$n_{k_0} p^3(S_{k_0}) \geq n_1 p_1^3 \geq 2^{-3\mathtt{k}}.$$

On the other hand, according to (34) we know

$$n_{k_0} p^3(S_{k_0}) \leq n_{k_0} p(S_{k_0}) \cdot p^2(S_{k_0}) \leq c\, p^2(S_{k_0}).$$

Hence $c\,p^2(S_{k_0}) \geq 2^{-3\mathrm{k}}$, namely $p(S_{k_0}) \geq c^{-1/2} \cdot 2^{-\frac{3\mathrm{k}}{2}}$.

For other case, since

$$\begin{aligned} 1 = \sum_{i=1}^{N} p_i &\leq \sum_{i=1}^{k_0-1} n_i p(S_i) + \sum_{i=k_0}^{\ell} n_i p(S_i) \\ &\leq p(S_{k_0}) \cdot \sum_{i=k_0}^{\ell} n_i + \sum_{i=1}^{k_0-1} n_i p(S_i) \qquad (43) \\ &\leq p(S_{k_0}) \cdot N + \sum_{i=1}^{k_0-1} n_i p(S_i) \end{aligned}$$

Since for any $j < k_0$, it holds $c^{(k_0-j)} \cdot p(S_{k_0}) \leq p(S_j)$ by $c$-partition, hence

$$n_j p(S_j) \leq n_{k_0} p(S_{k_0}) \cdot (p(S_{k_0})/p(S_j))^2 \leq \frac{1}{c^{2(k_0-j)}} n_{k_0} p(S_{k_0}).$$

(43) becomes

$$\begin{aligned} 1 &\leq p(S_{k_0}) \cdot N + n_{k_0} p(S_{k_0}) \cdot \sum_{i=1}^{k_0-1} \frac{1}{c^{2i}} \\ &\leq N p(S_{k_0}) \cdot (1 + \sum_{i=1}^{\infty} \frac{1}{c^{2i}}) = \frac{c^2}{c^2-1} \cdot N p(S_{k_0}). \end{aligned}$$

Namely $p(S_{k_0}) \geq \frac{c^2-1}{c^2 N}$. That finishes the proof. □

The result above also implies that in calculation the collision parameter and the collision domain, one only needs seek the sets $S_i$ satisfying $p(S_i) = \Omega(2^{-3\mathrm{k}/2})$ or $p(S_i) = \Omega(N^{-1})$.

Moreover, in the upper bound proof, for any constant $c > 1$, it also allows to replace the condition of $M$ in Theorem 2 from $M > 12c^2/p(S_{k_0})$ to $M = \Omega(2^{3\mathrm{k}/2})$ or $M = \Omega(N)$.

The following proposition indicates that the choice of $c$ does not affect the order of magnitude of $\gamma(c)$ (as long as $c > 1$ is a constant).

**Proposition 7.** *For any non-uniform distribution $D$ and constants $c_1, c_2$ satisfying $c_2 > c_1 > 1$, we have $\frac{c_1^3-1}{c_1^3 c_2^3} \cdot \gamma(c_1) < \gamma(c_2) \leq 2c_1^3 \cdot \gamma(c_1)$.*

*Proof.* Given two constants satisfying $c_2 > c_1 > 1$, we denote by $\{S_1^{(1)}, \ldots, S_\ell^{(1)}\}$ and $\{S_1^{(2)}, \ldots, S_{\ell'}^{(2)}\}$ the partition results, respectively, by the $c_1$-partition and $c_2$-partition of $[N]$ with respect to $D$. Similarly, we let $n_i^{(j)}$ be the size of $S_i^{(j)}$ and $p^{(j)}(S_i)$ the maximum one in $\{p_k, k \in S_i^{(j)}\}$. In addition, we also let $\bar{p}^{(j)}(S_i)$ be $\min\{p_k \mid k \in S_i^{(j)}\}$ in this section.

Assume that $S^{(1)}_{i^*}$ as the collision domain of $D$ in $c_1$-partition. Accordingly, $\gamma(c_1) := \gamma_{i^*}(c_1)$. By the definition of $c$-partition and that $c_2 > c_1 > 1$, there exists a $i_0 \in [\ell' - 1]$ that satisfies $S^{(2)}_{i_0} \cup S^{(2)}_{i_0+1} \supset S^{(1)}_{i^*}$ and $p^{(2)}(S_{i_0+1}) \geq \bar{p}^{(1)}(S_{i^*})$. That is, it takes at most two sets in $c_2$-partition to cover $S^{(1)}_{i^*}$.

Now we turn to estimate $\gamma_{i_0}(c_2)$ and $\gamma_{i_0+1}(c_2)$. Since that $S^{(2)}_{i_0} \cup S^{(2)}_{i_0+1} \supset S^{(1)}_{i^*}$, we have $n^{(2)}_{i_0} + n^{(2)}_{i_0+1} \geq n^{(1)}_{i^*}$, namely

$$\max\{n^{(2)}_{i_0}, n^{(2)}_{i_0+1}\} \geq n^{(1)}_{i^*}/2. \tag{44}$$

Moreover, for any non-uniform distribution $D$, we know

$$p^{(2)}(S_{i_0}) > p^{(2)}(S_{i_0+1}) \geq \bar{p}^{(1)}(S_{i^*}) > p^{(1)}(S_{i^*})/c_1. \tag{45}$$

Therefore, from (44) and (45) we can get:

$$\begin{aligned}\max\{\gamma^{-1}_{i_0}(c_2), \gamma^{-1}_{i_0+1}(c_2)\} &= \max\{n^{(2)}_{(i_0)}(p^{(2)}(S_{i_0}))^3, n^{(2)}_{i_0+1}(p^{(2)}(S_{i_0+1}))^3\}\\ &\geq \frac{n^{(1)}_{i^*}}{2} \cdot (\frac{p^{(1)}(S_{i^*})}{c_1})^3 = \frac{1}{2c_1^3} \cdot n^{(1)}_{i^*}(p^{(1)}(S_{i^*}))^3\\ &= \frac{1}{2c_1^3} \cdot \gamma^{-1}(c_1). \end{aligned}\tag{46}$$

It implies that $\gamma(c_2) \leq \min\{\gamma_{i_0}(c_2), \gamma_{i_0+1}(c_2)\} \leq 2c_1^3 \cdot \gamma(c_1)$.

For the other part of the proof, assume $\gamma(c_2) := \gamma_{j^*}(c_2)$ for some $j^*$. From the definition of $c$-partition, we can use at most $\lfloor \frac{\ln c_2}{\ln c_1} \rfloor + 2$ sets in $c_1$-partition to cover $S^{(2)}_{j^*}$. Suppose there is a $j_0$ that satisfies

$$\bigcup_{k=j_0-\lfloor \frac{\ln c_2}{\ln c_1} \rfloor - 1}^{j_0} S^{(1)}_k \supset S^{(2)}_{j^*} \qquad \text{and} \qquad p^{(1)}(S_{j_0}) \geq \bar{p}^{(2)}(S_{j^*}).$$

It implies

$$n^{(2)}_{j^*} \leq \sum_{k=0}^{\lfloor \frac{\ln c_2}{\ln c_1} \rfloor + 1} n^{(1)}_{j_0-k} \tag{47}$$

On the other hand, for any non-uniform distribution $D$, it holds that

$$p^{(1)}(S_{j_0}) \geq \bar{p}^{(2)}(S_{j^*}) > p^{(2)}(S_{j^*})/c_2\,. \tag{48}$$

Moreover, for any $i$, we have $p^{(1)}(S_{i-1}) \geq c_1 p^{(1)}(S_i)$, therefore

$$p^{(1)}(S_{j_0-k}) \geq c_1^k p^{(1)}(S_{j_0}) \tag{49}$$

for any $k \in \left[\lfloor \frac{\ln c_2}{\ln c_1} \rfloor + 1\right]$. In conclusion, combining (47), (48) and (49), we have:

$$
\begin{aligned}
\gamma^{-1}(c_2) &= \gamma_{j^*}^{-1}(c_2) = n_{j^*}^{(2)}(p^{(2)}(S_{j^*}))^3 \\
&\leq (p^{(2)}(S_{j^*}))^3 \cdot \sum_{k=0}^{\lfloor \frac{\ln c_2}{\ln c_1} \rfloor + 1} n_{j_0-k}^{(1)} \leq c_2^3 \sum_{k=0}^{\lfloor \frac{\ln c_2}{\ln c_1} \rfloor + 1} n_{j_0-k}^{(1)} \cdot (p^{(1)}(S_{j_0}))^3 \\
&\leq c_2^3 \sum_{k=0}^{\lfloor \frac{\ln c_2}{\ln c_1} \rfloor + 1} \frac{1}{c_1^{3k}} \cdot n_{j_0-k}^{(1)} \cdot (p^{(1)}(S_{j_0-k}))^3 = c_2^3 \sum_{k=0}^{\lfloor \frac{\ln c_2}{\ln c_1} \rfloor + 1} \frac{1}{c_1^{3k}} \cdot \gamma_{j_0-k}^{-1}(c_1) \\
&\leq c_2^3 \cdot \gamma^{-1}(c_1) \sum_{k=0}^{\lfloor \frac{\ln c_2}{\ln c_1} \rfloor + 1} \frac{1}{c_1^{3k}} \leq \frac{c_1^6 c_2^3 - 1}{c_1^6 - c_1^3} \cdot \gamma^{-1}(c_1) \\
&< \frac{c_1^3 c_2^3}{c_1^3 - 1} \cdot \gamma^{-1}(c_1). \qquad (50)
\end{aligned}
$$

Namely $\gamma(c_2) > \frac{c_1^3 - 1}{c_1^3 c_2^3} \cdot \gamma(c_1)$. That finishes the proof. □

## Acknowledgement

## References

1. Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, 2004.
2. Andris Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 636–643, 2000.
3. Andris Ambainis. Quantum walk algorithm for element distinctness. In *FOCS 2004*, pages 22–31. IEEE Computer Society, 2004.
4. Andris Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory Comput.*, 1(1):37–46, 2005.
5. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 269–295. Springer, 2019.
6. Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 474–483, 2014.
7. Marko Balogh, Edward Eaton, and Fang Song. Quantum collision-finding in non-uniform random functions. In *PQCrypto 2018*, pages 467–486. Springer, 2018.

8. Itay Berman, Akshay Degwekar, Ron D. Rothblum, and Prashant Nalini Vasudevan. Multi-collision resistant hash functions and their applications. In Jesper Buus Nielsen and Vincent Rijmen, editors, *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 II*, volume 10821 of *Lecture Notes in Computer Science*, pages 133–161. Springer, 2018.
9. Alexandra Boldyreva and Virendra Kumar. A new pseudorandom generator from collision-resistant hash functions. In Orr Dunkelman, editor, *Topics in Cryptology - CT-RSA 2012 - The Cryptographers' Track at the RSA Conference 2012, San Francisco, CA, USA, February 27 - March 2, 2012. Proceedings*, volume 7178 of *Lecture Notes in Computer Science*, pages 187–202. Springer, 2012.
10. Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics*, 46(4-5):493–505, 1998.
11. Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. *SIGACT News*, 28(2):14–19, 1997.
12. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 22(6):644–654, 1976.
13. Xiaoyang Dong, Bingyou Dong, and Xiaoyun Wang. Quantum attacks on some feistel block ciphers. *Des. Codes Cryptogr.*, 88(6):1179–1203, 2020.
14. Ehsan Ebrahimi and Dominique Unruh. Quantum collision-resistance of non-uniformly distributed functions: upper and lower bounds. *Quantum Inf. Comput.*, 18(15&16):1332–1349, 2018.
15. Marc Fischlin and Anja Lehmann. Security-amplifying combiners for collision-resistant hash functions. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, volume 4622 of *Lecture Notes in Computer Science*, pages 224–243. Springer, 2007.
16. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO 99*, pages 537–554. Springer, 1999.
17. Benjamin Fuller, Leonid Reyzin, and Adam D. Smith. When are fuzzy extractors possible? In *ASIACRYPT 2016*, pages 277–306, 2016.
18. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219, 1996.
19. Sean Hallgren, Adam D. Smith, and Fang Song. Classical cryptographic protocols in a quantum world. *CoRR*, abs/1507.01625, 2015.
20. Akinori Hosoyamada and Tetsu Iwata. 4-round luby-rackoff construction is a qprp. In *ASIACRYPT 2019*, pages 145–174. Springer, 2019.
21. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *CRYPTO 2016*, pages 207–237. Springer, 2016.
22. Samuel Kutin. Quantum lower bound for the collision problem with small range. *Theory Comput.*, 1(1):29–36, 2005.
23. Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT*, pages 189–218. Springer, 2019.
24. Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

25. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.
26. Ehsan Ebrahimi Targhi, Gelo Noel Tabia, and Dominique Unruh. Quantum collision-resistance of non-uniformly distributed functions. In Tsuyoshi Takagi, editor, *PQCrypto 2016*, pages 79–85. Springer, 2016.
27. Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the fujisaki-okamoto and OAEP transforms. In *TCC 2016-B*, pages 192–216, 2016.
28. Henry Yuen. A quantum lower bound for distinguishing random functions from random permutations. *Quantum Inf. Comput.*, 14(13-14):1089–1097, 2014.
29. Mark Zhandry. How to construct quantum random functions. In *FOCS 2012*, pages 679–687. IEEE Computer Society, 2012.
30. Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Information & Computation*, 15(7&8):557–567, 2015.
31. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Advances in Cryptology - CRYPTO 2019,August 18-22, 2019,*, pages 239–268, 2019.