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 γ 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

Quantum computation has brought threats to classical cryptography since Shor's seminal article [30]. 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 for gaining advantages that might not be possible in traditional computations. As a result, many

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

classical public key encryption systems, including Diffie-Hellman protocol [14] and RSA cryptosystem [29], are broken by Shor's factoring and discrete-log algorithms [30]. The Grover's algorithm [21] allows greatly improves 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 [7, 15, 22, 25].

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 the limitations of quantum computing by considering the number of queries in the black-box model.

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

Collision resistance is one of the 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 [9, 10, 17].

It has been well studied on the complexity of finding a collision in the quantum setting. In view of the upper bounds, Brassard et al. [12] showed that $O(N^{1/3})$ queries are sufficient to find a collision in a two-to-one function. Ambainis [4] 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 [37] proposed an algorithm that takes $O(N^{1/3})$ quantum queries to find a collision in a uniformly random function. Targhi et al. [16] and Balogh et al. [8] proved, separately, that $O(2^{\beta/3})$ quantum queries are sufficient to find a collision in a non-uniform random function, where β 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. The result was further extended to the case of small range by Kutin [26] and Ambainis [5] independently. Yuen [35] proved that $\Omega(N^{1/5}/\mathtt{poly}\log N)$ quantum queries are necessary to find a collision in a uniformly random function $f : [N] \to [N]$. Zhandry [37] 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 [31], 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 $\Omega(2^{\mathtt{k}/5})$ [16]. Balogh et al. [8] 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 $[N]$, $\Theta(N^{1/3})$ quantum queries are both necessary and

sufficient to find a collision [37], and thus give the tight upper and lower bounds in uniform case.

In the research of the 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] [37]	uniform	$O(N^{1/3})$	$\Omega(N^{1/3})$
[TTU16] [31]	non-uniform		$\Omega(2^{k/9})$
[EU18] [16]	non-uniform	$O(\min\{2^{\beta/3}, 2^{k/2}\})$	$\Omega(\max\{2^{\beta/9}, 2^{k/5}\})$
[BES18] [8]	non-uniform	$O(\min\{2^{\beta/3}, 2^{k/2}\})$	$\Omega(\max\{2^{\beta/6}, 2^{k/3}\})$

Existing algorithms in this setting, for example in [8], 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 the 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 [18] and further discussions in [6,32].

This work aims to work out (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 first propose two collision-finding algorithms and analyze their 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 by a newly proposed collision parameter, and finally, result in tighter bounds.

In [16], 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 $O(2^{k/2})$ and the lower bound $\Omega(2^{k/3})$ are both the best possible ones. The gap between the upper and lower bounds there hints to us that min-entropy might not be good enough in describing the query complexity of the collision-finding problem.

This leads us to seek finer parameter that may reflect more properties of underlying distributions. In this work, we propose a new collision parameter γ 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 [19], 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, later in Proposition 6, we show that the constant c does not affect the magnitude of query complexity, and hence we may mention γ now and then rather than $\gamma(c)$.

In terms of this parameter, we can compose two algorithms that succeed 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 γ is an appropriate one in describing the query complexity of collision-finding.

It should be noted that our work is NOT a complete improvement on previous work. Previous work only used minimum entropy and collision entropy as indicators of the complexity of this problem. Our contribution is that we define a new entropy(i.e. $\log \gamma$), and according to this new entropy, we can give an almost tight answer to quantum query complexities of the collision-finding problem with non-uniform oracles. Noting that the classical version of this problem is still an open question(the best lower bound and upper bound are respectively $\Omega(2^{\mathtt{k}/2})$ and $O(2^{\beta/2})$), we hope that this new entropy we propose can be applied to future work.

The proposed parameter we propose, as the collision-entropy does (cf. Definition 1), will require more information about the distribution to calculate 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 take the min-entropy as input. This is based on the observation that the key to improving the acceleration in an 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 properties 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 [6,32]. 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 knows the underlying structure of the encryption algorithm (for example, the adversary knows in advance that f is the ElGamal encryption algorithm, which means the construction of this scheme is based on DDH assumptions). This 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 $y \in \{0,1\}^n$, to calculate $\Pr[y = f(\delta, H(\delta, c)) : c \leftarrow \mathtt{coin}]$).

That implies the adversary is capable and likely to know all the information about the 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 the full version of [6]), which will result in a tighter reducible bound in the CCA security of FO transformation. The KEM variants of FO transformation are widely used in the NIST Round-1 to Round-4 KEM submissions, and thus our results provide more accurate quantum security indicators for NIST KEM submissions.

1.2 Technical Overview

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

The parameter γ and its properties. A novel parameter γ is proposed in this work, aiming at accurately describing the quantum computational complexity of the 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 a 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 to as the collision domain in this work. Given non-uniform distribution D with weight sequence $\{p_1, \ldots, p_N\}$, assuming without loss 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 6), 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 the 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 the 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 3 and 4), with the min-entropy $\mathtt{k}$ and collision-entropy β used in existing algorithms and show that both the upper bound and lower bound in this work are at least as good as the

best prior result in [8]. To indicate the 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, compared to previous algorithms which mainly take min-entropy or collision-entropy as inputs. To make the computation of the collision domain simple in the application, we show a result in Proposition 5 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 6, 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. Here we give two collision-finding algorithms. In general, the two algorithms are consistent in terms of query complexity, but there are some differences in detail. The first algorithm is the quantum walk algorithm, which requires quantum memory but can successfully find collision even if there is only one collision in the collision domain. The second algorithm is a BHT-type algorithm, this algorithm only requires classical memory, but it has certain requirements for the input random function domain $[M]$. Informally, this algorithm works if there is enough collision in the collision domain.

Our first algorithm can be seen as a variation of Ambainis's element distinctness algorithm [4]. Johnson graph is a standard graph on which we perform quantum random walk (QRW). Including the element distinctness problem we mentioned, many problems with the form: given $x_1, \ldots, x_n \in \Omega$ and a relation $R \subset \Omega^k$, find $x_{i_1}, \ldots, x_{i_k}$ such that $(x_{i_1}, \ldots, x_{i_k}) \in R$, can be solved using the quantum walk over the Johnson graph $J(n, r)$ (here r is the number of quantum registers required) [13], [20], [2].

The point of our optimization is to modify the graph that performs the quantum walk. Since we can calculate the collision domain, when we want to find a collision in a random function $f : [M] \to [N]$, we do not use the general Johnson graph $J(M, r)$ for QRW like Ambainis's algorithm, but restrict QRW to the subgraph of $J(M, r)$ by some additional operations. With this modification, the query complexity of our algorithm can be reduced.

Our second algorithm is intuitively designed as follows. First, to use Grover's algorithm to find a list L of elements in collision domain S_{k_0}, and then to adopt a standard collision algorithm to find a collision in L. It uses an 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 BBHT algorithm must be made to be used in our algorithm. BBHT algorithm in [11] 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 modified BBHT algorithm here is actually a de-randomized version of BBHT algorithm.

Finally, we show that our collision-finding algorithms make $O(\gamma^{1/6}(c))$ queries succeed 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 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 first 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 calculate 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 [38] is with respect to uniform random functions. Therefore we transform the problem into another one (Problem 2), a problem with respect to uniform distributions. In such a way, the compressed oracle technique in the quantum random oracle model (QROM) proposed by Zhandry is successfully exploited.

The transformation technique in this work might be of independent interest 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 as the corresponding upper bound of the success probability after q quantum queries by the 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 3). We hence obtain $\Omega(\gamma^{1/6}(c)/\sqrt{\log \gamma(c)})$ as a quantum query complexity lower bound for the 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, our collision-finding algorithms and their correctness are presented. The quantum query complexity of the algorithm is also analyzed. In Section 4, a lower bound to the 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 prior works and will be mentioned later.

Definition 1 (Min-Entropy& Collision-Entropy [8]). *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 β. 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 the 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 the 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 satisfy $p_1 \geq p_2 \geq \cdots \geq p_N$ 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 $\{p_i \mid i \in S_2\}$, 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*

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

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 collision parameter $\gamma(c)$ with respect to c-partition in the same notations as above, as follows.

Definition 3 (Collision Parameter & Collision Domain). *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 to 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" that appeared in current literature. It heavily depends on the distribution D in evaluations. This is similar to "collision-entropy" in prior work, which 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 $2^{\mathtt{k}/2}$ in magnitude, the latter is proved optimal in that case.

In the general case (namely in arbitrary non-uniform distribution), compared with existing collision variables, the collision parameter proposed here will give more concise characterizations for both upper and lower query bounds to the 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 [21], Grover proposed a quantum algorithm for the database search problem, demonstrating the powerful acceleration effect of quantum computing on this issue.

Lemma 1 (Grover's Algorithm [21]). *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. [11] proposed a generalized algorithm 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 to as BBHT algorithm.

Lemma 2 (BBHT [11]). *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 $\Theta(\sqrt{N/t})$ expected queries.*

BBHT algorithm starts with uniform superposition state $|\psi_0\rangle = \sum_{i=0}^{N-1} \frac{1}{\sqrt{N}}|i\rangle$ 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 $\alpha_q = \Theta(1)$ when $q = \Theta(\sqrt{N/t})$, namely the algorithm may find the pre-image of 1 with constant probability.

2.3 Quantum random walk

In this subsection, we give a brief introduction to quantum random walk(QRW). To facilitate understanding, we will focus on the special case of QRW over Johnson graphs, since it's sufficient to obtain our results. There are several variants of QRW and our presentation is mostly based on [28] and [24].

We start with a general graph. Suppose we have a connected unweighted undirected simple graph $G = (V, E)$, and now we have a subset of marked vertices $M \subset V$, our goal is to find a vertex $v \in M$.

The above problems can be solved using a quantum random walk algorithm. Intuitively, a one-step classical random walk represents the process of randomly moving from a vertex in the graph to its neighbors. The quantum random walk differs from the classical case in that its initial input can be a superposition state $\frac{1}{\sqrt{|V|}}\sum_{v\in V}|v\rangle$ rather than a vertex v_0. We denote $N_G(u) := \{v \in V | \{u, v\} \in E\}$ the set of neighbors, then the cost of a quantum random walking algorithm can be divided into three categories:

Setup Cost S: it's the cost of constructing

$$|\pi\rangle := \sum_{u\in V} \sqrt{\pi_u}|u\rangle_L|0\rangle_R|D(u)\rangle_d$$

(here $D(u)$ is a data structure used to assist in the check process and $\pi := \{\pi_u\}$ is the stationary distribution of the transition matrix corresponding to our quantum random walk).

Update Cost U: it's the cost of perform one-step QRW. Specifically, it contains the cost of implementing the following two operators:

$$U_P : |u\rangle_L|0\rangle_R \to |u\rangle_L \sum_{v\in V} \frac{1}{\sqrt{|N_G(u)|}}|v\rangle_R$$

and

$$U_D : |u\rangle_L|v\rangle_R|D(u)\rangle_d \to |v\rangle_L|u\rangle_R|D(v)\rangle_d.$$

Check Cost C: it's the cost of checking whether a vertex v is marked with the assistance of data structure, namely the cost of constructing:

$$|u\rangle_L|D(u)\rangle_d \to (-1)^{F(u)}|u\rangle_L|D(u)\rangle_d.$$

Here $F(u) = 1$ if $u \in M$, otherwise $F(u) = 0$.

Suppose there is $\epsilon > 0$ that satisfies $\frac{|M|}{|V|} \geq \epsilon$ and δ be the spectral gap of G, then:

Lemma 3 ([28]). *If the fraction of marked vertices is at least ϵ, there is a QRW algorithm to find a marked vertex with constant probability. The cost incurred is*

$$O(\mathbf{S} + \frac{1}{\sqrt{\epsilon}}(\frac{1}{\sqrt{\delta}} \cdot \mathbf{U} + \mathbf{C})).$$

If we restrict ourselves to QRW over Johnson graphs, it will be more convenient in practical applications. The definition of Johnson graphs is as follows:

Definition 4. *The Johnson graph $J(n, r)$ is an (undirected) graph in which the vertex set is the set of all subsets of r-elements of $[n]$. $(u, v) \in E$ iff $u := \{u_1, \ldots, u_r\}$ and $v := \{v_1, \ldots, v_r\}$ satisfy that $|u \cap v| = r - 1$.*

The QRW over Johnson graphs has a nice property: the initial state $|\pi\rangle$ mentioned above can be constructed as the superposition state

$$\sum_{\substack{|u|=r, u_i \in [n] \\ u_i \neq u_j}} \frac{1}{\sqrt{C(n,r)}}|u\rangle_L|0\rangle_R|D(u)\rangle_d,$$

this enables $|u\rangle_L = \bigotimes_{i\in[r]} |u_i\rangle_L$ to be prepared efficiently in the setup process. In addition, the spectral gap δ of $J(n, r)$ satisfies $\delta \approx \frac{1}{r}$, so later on in the discussion of algorithmic complexity, we're left with the analysis of $\mathbf{S}$, $\mathbf{U}$, $\mathbf{C}$ and ϵ.

2.4 Some Probabilistic Inequalities

The following are some probabilistic inequalities used in subsequent sections.

Lemma 4 (Höffding's Inequality). *Let $X_1, \ldots, X_n$ 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,*

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

and

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

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 5 (Bernstein's Inequality). *Let $X_1, \ldots, X_n$ 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,*

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

and

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

3 Collision-finding Algorithms and their Query Complexity

3.1 Warming up

Balogh, Eaton and Song [8] proposed a collision-finding algorithm (referred as BES algorithm) based on the collision-entropy β. It applies Ambainis's quantum walk algorithm [4] 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 $\perp$

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 settings. 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 in 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 avoid this advantage. The query complexity of a non-uniform function would be heavily dependent on $p_1, \ldots, p_N$. That is the main ingredient in comparing a non-uniform distribution with a uniform distribution.

Let's try to figure out the obstacle in the 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 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 on collision-finding. An effective approach might be to use 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 a certain extent.

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

In view of current algorithms in the literature, the BES algorithm introduced above sets $[N]$ as the search domain, while the others set the search domain to $S = \{1\}$ using Grover's 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 Quautum Random Walk Algorithm

In this subsection, we propose our quantum random walk algorithm for collision-finding in non-uniform random functions. As mentioned in the previous overview, compared with directly using Ambainis's algorithm to solve the problem, the main improvement we have made is the modification of the graph that performs the quantum walk.

Algorithm 1 QRW Algorithm for Collision-Finding in Non-uniform Functions

Input: Collision domain S_{k_0} and a function $f : [M] \to [N]$ with $f \leftarrow D^M$.
Output: A collision (x_1, x_2) or $\perp$.

1: Choose $M' \subset M$ arbitrarily such that $|M'| = \frac{12c^2}{\sqrt{n_{k_0} p^2(S_{k_0})}}$.
2: Run $\mathbf{Init}(S_{k_0}, f, M')$ to generate the initial state

$$|\pi_0\rangle = \sum_{\substack{u_i \in |M_f| \\ u_i \neq u_j \text{ if } i \neq j \\ f(u_i) \in S_{k_0}}} \frac{1}{\sqrt{C(|M_f|, r)}} \bigotimes_{i \in [r]} |u_i\rangle_L |0\rangle_R \bigotimes_{i \in [r]} |f(u_i)\rangle_d$$

(Here M_f be the set of all $x \in M'$ such that $f(x) \in S_{k_0}$).
3: $O(1/\sqrt{\epsilon})$ times repeat:
(a) Perform the transformation $|u\rangle_L |v\rangle_R |D(u)\rangle_d \to -|u\rangle_L |v\rangle_R |D(u)\rangle_d$ for $u := (u_1, \ldots, u_r)$ satisfy that there exist $f(u_{i_1}) = f(u_{i_2})$ for some $u_{i_1}, u_{i_2} \in [M_f]$ (it's easy to see that a vertex u is marked iff there is a collision in $|u\rangle_L$).
(b) Perform $O(1/\sqrt{\delta})$ steps of the quantum random walk over the Johnson graph $J(|M_f|, r)$.
4: Measure $|D(u)\rangle_d$, if there exist $f(u_{i_1}) = f(u_{i_2})$ for some $u_{i_1}, u_{i_2} \in [M_f]$ then output (u_{i_1}, u_{i_2}) which can be find in $|u\rangle_L$, otherwise $\perp$.

Now we give our QRW algorithm, all the parameters that occur in the algorithm are analyzed in the next subsection. The key point of how to modify is the property of non-uniform. Since we can calculate the collision domain S_{k_0}, namely if we find a collision (x_1, x_2), it has a maximum probability satisfying $f(x_1) = f(x_2) \in S_{k_0}$ compared to other $S_i \subset [n]$. So our idea is to focus only on vertices $u := (u_1, \ldots, u_r)$ which satisfy $f(u_i) \in S_{k_0}$ for all $i \in [r]$, and quantum random walk over the subgraph induced on the Johnson graph $J(M, r)$ by the set of these vertices. As far as we know, the idea of limiting quantum walks to subgraphs has not appeared in previous quantum walk algorithms, we hope our idea can be applied to other problems.

There are two main obstacles to this idea. There is some convenience in performing the quantum walk directly on $J(M, r)$. First, the uniform superposition state

$$\sum_{\substack{|u|=r, u_i \in [M] \\ u_i \neq u_j}} \frac{1}{\sqrt{C(M, r)}} |u\rangle_L$$

can be prepared efficiently without any query to f by default, so $\mathbf{S}$ is actually the cost of constructing the data structure $|D(u)\rangle_d$. Second, when performing QRW on $J(M,r)$, the query complexity of implementing U_P is also 0 by default. So the previous quantum walk algorithm only needs to focus on changes in the data structure $D(u)$, and it's easy to get $\mathbf{S}=r$ and $\mathbf{U}=2$ (only need to perform $|u_i\rangle_L|0\rangle_d \to |u_i\rangle_L|f(u_i)\rangle_d$ for all $i\in[r]$ to get $D(u)$ and only an un-compute and a compute operation is required to implement U_D).

But if we constrain the quantum walk, what we have mentioned above needs to be considered carefully. We can't get to the initial state $\sum \alpha_u|u\rangle_L$ without any queries, since we can't determine whether $f(u_i)\in S_{k_0}$ for any $i\in[r]$. Similarly, we need additional queries to check whether $v\in N_G(u)$ meets our additional constraints. In the next subsection, we show how to solve these two problems with additional queries.

3.3 Analysis of Algorithm 1

Now we analyze the query complexity of our algorithm in the previous subsection. First of all, we estimate $|M_f|$, which will help us with some of our proofs later on. Let f is well-behaved if $6c\sqrt{n_{k_0}} < |M_f| < 18c^2\sqrt{n_{k_0}}$.

Lemma 6. *For any c-partition with constant $c>1$, f is well-behaved with constant probability.*

Proof. Since for any $x\in[M]$, $f(x)\in S_{k_0}$ with probability $P=\sum\limits_{i\in S_{k_0}} p_i$, by the binomial distribution it's easy to see that $\mathrm{E}(|M_f|)=M'P$ and $\mathrm{Var}(|M_f|)=M'P(1-P)<M'P$. According to the definition $|M'|=\frac{12c^2}{\sqrt{n_{k_0}p^2(S_{k_0})}}$ and $n_{k_0}p(S_{k_0})/c\le\sum\limits_{i\in S_{k_0}} p_i\le n_{k_0}p(S_{k_0})$, we have:

$$12c\sqrt{n_{k_0}}\le \mathrm{E}(|M_f|)\le 12c^2\sqrt{n_{k_0}} \quad \text{and} \quad \mathrm{Var}(|M_f|)<12c^2\sqrt{n_{k_0}}$$

Therefore, by Bernstein's inequality, we have:

$$\begin{aligned}\Pr_{f\leftarrow D^M}[(|M_f|\le 6c\sqrt{n_{k_0}})\vee(|M_f|\ge 18c^2\sqrt{n_{k_0}})] &\le \Pr_f\left[|\,\mathrm{E}[M_f]-|M_f||\ge 6c\sqrt{n_{k_0}}\right]\\ &\le 2\,\exp\Big(-\frac{18c^2\sqrt{n_{k_0}}}{12c^2+2c}\Big)<\ 2e^{-9/7}.\end{aligned}$$

The last inequality holds since $c>1$ and $n_{k_0}\ge 1$, so f is well-behaved with probability at least $3/4$. □

With this preparation, now we give estimates of the parameters in the quantum random walk.

Check cost C: it's easy to see that $\mathbf{C}=0$.

The spectral gap δ: from the property of Johnson graph, we have $\delta \approx r$ [34].

The fraction of marked vertices ϵ: we derive the following theorem:

Theorem 1. *If f is well-behaved, $\epsilon = \Omega(\frac{n_{k_0}}{r^2})$ with constant probability.*

The proof is given in Appendix A.1.

The Setup Cost: according to the following theorem, we have $\mathbf{S} = \Theta(\frac{r}{\sqrt{n_{k_0}p(S_{k_0})}})$.

Theorem 2. *If f is well-behaved, there exists an algorithm that constructs the initial state with $\Theta(\frac{r}{\sqrt{n_{k_0}p(S_{k_0})}})$ expected queries.*

The proof is given in Appendix A.2.

The Update Cost: according to the following theorem, we have $\mathbf{U} = U_P + U_D = \Theta(\frac{1}{\sqrt{n_{k_0}p(S_{k_0})}}) + 2 = \Theta(\frac{1}{\sqrt{n_{k_0}p(S_{k_0})}})$ since $n_{k_0}p(S_{k_0}) < c$ by c-partition.

Theorem 3. *If f is well-behaved, the expected number of queries required to implement U_P is $\Theta(\frac{1}{\sqrt{n_{k_0}p(S_{k_0})}})$.*

The proof is given in Appendix A.3.

Put it together, according to Lemma 3, Algorithm 1 succeeds with constant probability. The expected number of queries required is

$$\mathrm{E}(Q) := O(\frac{r}{\sqrt{n_{k_0}p(S_{k_0})}} + \sqrt{\frac{n_{k_0}}{r^2}} \cdot \sqrt{r} \cdot \frac{1}{\sqrt{n_{k_0}p(S_{k_0})}}) = O(\frac{r}{\sqrt{n_{k_0}p(S_{k_0})}} + \frac{1}{\sqrt{rp(S_{k_0})}})$$

Let $r = \Theta(\sqrt[3]{n_{k_0}})$, we have $\mathrm{E}(Q) = O(\gamma^{1/6}(c))$, namely there exists an algorithm find a collision in the non-uniform random function f with constant probability, it requires $O(\gamma^{1/6}(c))$ queries and $O(r) = O(\sqrt[3]{n_{k_0}})$ qubits of memory.

3.4 A BHT-type Collision-Finding Algorithm

Let's start by making some minor modification to BBHT algorithm. In our collision-finding algorithm, the modified 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 [11] provides only the expected number of queries when the algorithm succeeds. Actually, the modification can be seen as a de-randomized version of BBHT algorithm. The modified algorithm is presented in Appendix B.1 and the success probability of the modified BBHT algorithm is concluded as follows.

Theorem 4. *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 is also given in Appendix B.1.

With the preparation, we now propose a new collision-finding algorithm for random functions. The algorithm is presented as Algorithm 2.

Algorithm 2 Collision-Finding Algorithm in Non-uniform Functions

Input: Collision domain S_{k_0} and a function $f : [M] \to [N]$ with $f \leftarrow D^M$.
Output: A collision (x_1, x_2) or $\perp$.

1: Construct a function $F_1 : [M] \to \{0, 1\}$ such that $F_1(x) = 1$ iff $f(x) \in S_{k_0}$. Let L be a dynamic constructed set which initially is emptyset $L = \emptyset$.
2: Run Algorithm 4 with F_1 and q_1 to search for x such that $F_1(x) = 1$. Query $y := f(x)$ and check whether $y \in S_{k_0}$. 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 the Algorithm halts with $|L| < t$ and outputs $\perp$.
3: 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.
4: 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 $x_1 \in [M]$.
5: 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 7), $q_1 = O(1/\sqrt{n_{k_0} p(S_{k_0})})$ (Theorem 6) and $q_2 = O(1/\sqrt{t\, p(S_{k_0})})$ (Theorem 7).

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

Theorem 5. *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 3.5 is devoted to the proof of Theorem 5.

3.5 Proof of Theorem 5

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 $p(S_i)/c \leq p_j \leq p(S_i)$ for arbitrary $j \in S_i$.

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

Proposition 1. *For any constant $c > 1$, under the condition of Theorem 5, the probability which the random function f is well-behaved is at least 2/5.*

The proof is given in Appendix B.2.

Let suc denote the event that Algorithm 2 successfully finds a collision. According to Proposition 1, 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 1.

Notice that the key to the success of the algorithm is whether step 2 can successfully find t pairs (to denote as suc_2) and whether step 3 or step 4 can find a collision pair (to denote as suc_3 and suc_4 respectively). Namely our algorithm succeeds iff suc_2 happens and one of suc_3, 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 6. *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.*

The proof is given in Appendix B.3

Theorem 7. *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 $\frac{1}{2}\cdot(3/4 - e^{-t/2})$.*

The proof is given in Appendix B.4

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

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

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

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 $\Omega(1)$.

4 The Lower Bound

In this section, we are going to exploit a query complexity lower bound, during which the compressed oracle technique is adopted. For this purpose, we first 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 9 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 9), we explore the lower bound in Section 4.4 and get the main result in Theorem 12. In the last subsection, we discuss the parameter δ_c that appeared in Theorem 12, 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 the 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 $|\psi_0\rangle$ be the input state for $\mathcal{A}^{\mathcal{O}}$ and $|\psi_i\rangle$ the output state before the i'th measurement. A quantum computation performed by $\mathcal{A}$ with i queries is generally described as the product of a series of unitary transformations, in the following form:

$$|\psi_i\rangle = U_i\mathcal{O}\dots U_1\mathcal{O}U_0|\psi_0\rangle.$$

Where $U_0, \dots, U_i$ are some unitary operators independent of the input x.

A Phase oracle is a unitary transformation as follows:

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

For any random function $f \leftarrow N^M$. Here x denotes the input register, y is the output register and z is some auxiliary bits.

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

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

Compressed phase oracle: In the model of compressed phase oracle [38], the superposition state $\sum |f\rangle$ in phase oracle above will be replaced by a database $\mathbf{D}$ which records q binary and initialize as $\mathbf{D} := \{(0,\perp),...,(0,\perp)\}$(for convenience, for any $x \in [M]$, if there is (x,y) in $\mathbf{D}$, we denote $\mathbf{D}(x) := y$, otherwise we denote $D(x) = \perp$). When the adversary makes a query to the compressed phase oracle $\mathcal{O}$ on $|x,y,z,\mathbf{D}\rangle$, a binary pair in D is modified accordingly. More specifically, it performs in a sequence of the following steps:

1. If $\mathbf{D}(x) = \perp$, we denote $\mathbf{D}' \cup (0,\perp) = \mathbf{D}$, then it performs the map:

$$|x,y,z\rangle \otimes |\mathbf{D}' \cup (0,\perp)\rangle \to |x,y,z\rangle \otimes \frac{1}{\sqrt{2^n}} \sum_{w} |\mathbf{D}' \cup (x,w)\rangle.$$

   If $\mathbf{D}(x) = y_0$, then check whether binary pairs in $\mathbf{D}$ should be deleted. More specifically, if we denote $\mathbf{D}' \cup (x,y_0) = \mathbf{D}$, it performs the map:

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

2. If $\mathbf{D}(x) \neq \perp$, perform the following unitary transformation:

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

   If $\mathbf{D}(x) = \perp$, this step performs the identity transformation.
3. Perform 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 7 (Lemma 5 in [38]). *Consider a quantum algorithm $\mathcal{A}$ making queries to a random oracle H and outputting tuples $(x_1 \dots x_k, y_1, \dots y_k, z)$. 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 lower bound. Although the quantum adversary method [3] and the polynomials [36] approaches are usually adapted to derive the quantum lower bound proofs. Zhandry's technique allows us to turn attention to the changes in $\mathbf{D}$ after each query (notice any unitary operators U_i can not change the database $\mathbf{D}$), which would be, in some cases, more intuitive and convenient [23, 27].

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 the non-uniform setting, therefore we have to turn the collision-finding problem in the non-uniform setting into another problem in the 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 overall S_r (as shown in (8) 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 for 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 8 for the

possible values of K). To define $s_i \subseteq [KN]$ as follows ($i = 1, \ldots, N+1$).

$$s_i := \begin{cases} \left[1 \,..\, \lfloor KNp_1 \rfloor\right], & \text{for } i = 1; \\ \left[1 + \sum_{j=1}^{i-1} \lfloor KNp_j \rfloor \,..\, \sum_{j=1}^{i} \lfloor KNp_j \rfloor\right], & \text{for } i \in [2, N]; \\ [KN] \setminus (\bigcup_{i=1}^{N} s_i), & \text{for } i = N+1. \end{cases}$$

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 relationship between these two problems above:

Theorem 8. *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*

$$\mathbf{P}_1 - \mathbf{P}_2 < O(\frac{q^2}{K}).$$

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.

The proof is given in Appendix C.1

The result in Theorem 8 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 the following theorem.

Theorem 9. *Given $r > 0$ and distribution D as in the last section, for any quantum algorithm, $\Omega(\gamma_r^{1/6}(c))$ quantum queries are necessary to solve Problem 2 with constant probability.*

We give the following conclusions without proofs. The proofs of them mainly adopt the ideas from [27] and [38] and are given in Appendix C due to the lack of space. Let $S'_r := \bigcup_{j \in S_r} s_j$ and s_j defined in the last section, then we have:

Theorem 10. *For any quantum algorithm that makes at most q queries to the compressed random oracle $\mathcal{O}$, the amplitude that $\mathbf{D}$ contains at least j pre-images of S'_r after the i'th query is at most $\left(3ei\sqrt{n_r p(S_r)}/j\right)^j$.*

Denote P' be the projection onto the span of all states that $\exists x_1, x_2$ satisfying $\mathbf{D}(x_1), \mathbf{D}(x_2) \in s_k, s_k \subseteq S'_r$(namely $\mathbf{D}$ contains a solution of Problem 2). In addition, we call a pair (x, y) *good* iff $y \in S'_r$ and denote $\mathbf{\Phi}_j$ as the set of databases $\mathbf{D}$ satisfying that $\mathbf{D}$ containing exactly j good pair. Then we have:

Theorem 11. *For any quantum algorithm making queries to the compressed oracle $\mathcal{O}$, Suppose that just before i'th queries the joint state is*

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

then the following inequality holds:

$$\||P'|\psi_{i+1}\rangle\| \leq \||P'|\psi_i\rangle\| + \Big(\sum_{j=1}^{n_r} \big(6j\, p(S_r) \sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j} |\alpha^2_{i,x,y,z,\mathbf{D}}|\big)\Big)^{1/2}.$$

Noted that in the formula above, the part

$$\sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j} |\alpha^2_{i,x,y,z,\mathbf{D}}|$$

is the square of the amplitude that $\mathbf{D}$ contains exactly j pre-images of S'_r after the i'th query, while

$$\sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j, j\geq k_0} |\alpha^2_{i,x,y,z,\mathbf{D}}|$$

is exactly the square of the amplitude that $\mathbf{D}$ contains at least k_0 pre-images of S'_r after the i'th query, according to Theorem 10, we can get an upper bound for it.

With these preparations, we finally come to the proof of Theorem 9, which we give in the Appendix C.4.

As a variant of Theorem 9, the following result is obtained directly from the proof of Theorem 9.

Corollary 1. *For any quantum algorithm 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 the collision-finding problem with respect to non-uniform distributions.

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

$$\begin{aligned}
&\Pr_{f \leftarrow D^M}\left[f(x) = f(x'), x \neq x' : (x, x') \leftarrow \mathcal{A}^f\right] \\
&= \Pr_{f \leftarrow D^M}[f(x) = f(x'), x \neq x', f(x) \in \bigcup_{r=1}^{\ell} S_r : (x, x') \leftarrow \mathcal{A}^f] \qquad (8) \\
&\leq \sum_{r=1}^{\ell} \Pr_{f \leftarrow D^M}[f(x) = f(x'), x \neq x', f(x) \in S_r : (x, x') \leftarrow \mathcal{A}^f]
\end{aligned}$$

According to Theorem 8, there is an algorithm $\mathcal{B}$ such that

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

which in turn implies from (8)

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

By Corollary 1, we have

$$\begin{aligned}
&\Pr[g(x), g(x') \in s_j, x \neq x', s_j \subset S'_r : (x, x') \leftarrow \mathcal{B}^g, g \leftarrow (KN)^M] \\
&\qquad \leq O(\max\{q^3 n_r^{1/2} p^{3/2}(S_r), q^2 n_r^{1/3} p(S_r)\}) \qquad (9)
\end{aligned}$$

for any $i \in [\ell]$. To combine (8) and (9), we have

$$\begin{aligned}
&\Pr[f(x) = f(x'), x \neq x' : (x, x') \leftarrow A^f, f \leftarrow D^M] \\
&\quad \leq O(\frac{\ell q^2}{K}) + \sum_{r=1}^{\ell} O\big(\max\{q^3 n_r^{1/2} p^{3/2}(S_r), q^2 n_r^{1/3} p(S_r)\}\big) \qquad (10) \\
&\quad \leq O\Big(\sum_{r=1}^{\ell} \max\{q^3 n_r^{1/2} p^{3/2}(S_r), q^2 n_r^{1/3} p(S_r)\}\Big).
\end{aligned}$$

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

By the upper bound for q in Theorem 5, we assume $q < \gamma^{1/6}(c) \leq \gamma_r^{1/6}(c)$. 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 (10), we have:

$$\Pr_{f \leftarrow D^M}[f(x) = f(x'), x \neq x' : (x, x') \leftarrow A^f] \leq O(q^2 \sum_{r=1}^{\ell} n_r^{1/3} p(S_r))$$
$$= O(q^2 \cdot \delta_c \max_r\{(n_r p^3(S_r))^{1/3}\}) = O(\delta_c q^2 \gamma^{-1/3}(c)). \tag{11}$$

Where

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

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

Theorem 12. *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.*

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 [37].

We let readers convince themselves that the upper bound and the lower bound here also meets on flat-$\mathtt{k}$-distribution and δ-$\mathtt{k}$ distribution (cf. [8]), respectively. Here we refer to these two distributions to illustrate our algorithm's optimality. It 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 δ_c to derive a concise lower bound.

4.5 Estimation of Upper Bound of δ_c

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

Proposition 2. *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 $a_1 = 1, a_i \leq 1$ for any $i \in [\ell]$ and obviously

$$\sum_{i=1}^{\ell} a_i = \sum_{i=1}^{\ell} b_i = \frac{\sum_{r=1}^{\ell} \gamma_r^{-1/3}(c)}{\gamma^{-1/3}(c)} = \delta_c \tag{12}$$

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 3 that $\gamma^{1/6}(c) \le 2^{\mathtt{k}/2}$ for any constant $c > 1$. It implies, for any $k \in [\ell]$

$$p(S_{i_k}) \ge \frac{1}{\sqrt{c}} \cdot a_k^{3/2} \cdot 2^{-3\,\mathtt{k}/2}. \tag{13}$$

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 $j \le \lfloor \frac{\ell}{\lfloor \delta_c/2 \rfloor} \rfloor$ that satisfies $a_{j\lfloor \delta_c/2 \rfloor} \ge 1/(j+1)^4$. Otherwise, if $a_{j\lfloor \delta_c/2 \rfloor} < 1/(j+1)^4$ for all $j \le \lfloor \frac{\ell}{\lfloor \delta_c/2 \rfloor} \rfloor$, then by (12),

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

This is a contradiction. Let j_0 be the smallest $j \le \lfloor \frac{\ell}{\lfloor \delta_c/2 \rfloor} \rfloor$ that satisfies the inequality $a_{j\lfloor \delta_c/2 \rfloor} \ge 1/(j+1)^4$. Together with (13), it implies, for all $k \in \left[j_0 \lfloor \frac{\delta_c}{2} \rfloor\right]$

$$\begin{aligned}
p(S_{i_k}) &\ge \frac{1}{\sqrt{c}} \cdot a_k^{3/2} \cdot 2^{-3\mathtt{k}/2} \\
&\ge \frac{1}{\sqrt{c}} \cdot a_{j_0\lfloor \frac{\delta_c}{2} \rfloor}^{3/2} \cdot 2^{-3\mathtt{k}/2} \ge \frac{1}{\sqrt{c} \cdot (j_0+1)^6 \cdot 2^{3\mathtt{k}/2}}.
\end{aligned} \tag{14}$$

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

$$1/2^{\mathtt{k}} = p_1 \ge p(S_{i_k}) \tag{15}$$

According to (14) and (15), we get, by definition of c-partition,

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

Namely, we have

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

As desired. □

Together with Theorem 12, this conclusion 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 $c > 1$. In combination with Proposition 3, which we will prove in the next section, we can get our final answer to the lower bound: $\Omega(\gamma^{1/6}(c)/\sqrt{\log\gamma(c)})$. Compared with the upper bound $O(\gamma^{1/6}(c))$ in Theorem 5, it is nearly a tight lower bound.

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

In Definition 2, we have introduced a new parameter $\gamma(c)$. In this section, we will give some properties of $\gamma(c)$. The proofs of them appear in Appendix D. We will present the relations of γ with the min-entropy $\mathtt{k}$ and collision-entropy β, 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, 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 $k \in [\ell]$:

$$n_k p(S_k) \leq \sum_{i=1}^{\ell} n_i p(S_i) = c \sum_{i=1}^{\ell} \frac{n_i p(S_i)}{c} \leq c \sum_{i=1}^{N} p_i = c. \tag{16}$$

Now we show a relation between parameter γ and min-entropy $\mathtt{k}$.

Proposition 3. *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.*

The results in Proposition 2, Proposition 3 and Theorem 12 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.*

We show the relation between γ and collision-entropy β as follows.

Proposition 4. *For any non-uniform distribution D and any constant $c > 1$, we have $\frac{1}{c} \cdot 2^{\beta} \leq \gamma(c) < \frac{16c^3}{(c-1)^2} \cdot 2^{2\beta}$.*

From Table 1, we see Proposition 3 and 4 show that, in general non-uniform distribution, the upper bound and lower bound here in the parameter γ are always at least as good as the best prior results.

In Appendix D.3 , we present an example to show that the results here can be better in general.

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

Proposition 5. *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.

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

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

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. Shyan Akmal and Ce Jin. Near-optimal quantum algorithms for string problems, 10 2021.
3. 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.
4. Andris Ambainis. Quantum walk algorithm for element distinctness. In *FOCS 2004*, pages 22–31. IEEE Computer Society, 2004.
5. Andris Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory Comput.*, 1(1):37–46, 2005.
6. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In *CRYPTO 2019*, pages 269–295. Springer, 2019.
7. 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.
8. Marko Balogh, Edward Eaton, and Fang Song. Quantum collision-finding in non-uniform random functions. In *PQCrypto 2018*, pages 467–486. Springer, 2018.
9. Itay Berman, Akshay Degwekar, Ron D. Rothblum, and Prashant Nalini Vasudevan. Multi-collision resistant hash functions and their applications. In *Advances in Cryptology - EUROCRYPT 2018*, pages 133–161, 2018.
10. Alexandra Boldyreva and Virendra Kumar. A new pseudorandom generator from collision-resistant hash functions. In *CT-RSA 2012*, volume 7178 of *Lecture Notes in Computer Science*, pages 187–202. Springer, 2012.

11. 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.
12. Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. *SIGACT News*, 28(2):14–19, 1997.
13. André Chailloux and Johanna Loyer. Lattice sieving via quantum random walks. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021*, pages 63–91. Springer, 2021.
14. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 22(6):644–654, 1976.
15. Xiaoyang Dong, Bingyou Dong, and Xiaoyun Wang. Quantum attacks on some feistel block ciphers. *Des. Codes Cryptogr.*, 88(6):1179–1203, 2020.
16. 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.
17. 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.
18. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO 99*, pages 537–554. Springer, 1999.
19. Benjamin Fuller, Leonid Reyzin, and Adam D. Smith. When are fuzzy extractors possible? In *ASIACRYPT 2016*, pages 277–306, 2016.
20. Francois Le Gall. Improved quantum algorithm for triangle finding via combinatorial arguments. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 216–225, 2014.
21. 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.
22. Sean Hallgren, Adam D. Smith, and Fang Song. Classical cryptographic protocols in a quantum world. *CoRR*, abs/1507.01625, 2015.
23. Akinori Hosoyamada and Tetsu Iwata. 4-round luby-rackoff construction is a qprp. In *ASIACRYPT 2019*, pages 145–174. Springer, 2019.
24. Stacey Jeffery, Robin Kothari, and Frederic Magniez. Nested quantum walks with quantum data structures. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, jan 2013.
25. 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.
26. Samuel Kutin. Quantum lower bound for the collision problem with small range. *Theory Comput.*, 1(1):29–36, 2005.
27. Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT*, pages 189–218. Springer, 2019.
28. Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. *SIAM Journal on Computing*, 40(1):142–164, 2011.
29. 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.

30. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.
31. 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.
32. Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the fujisaki-okamoto and OAEP transforms. In *TCC 2016-B*, pages 192–216, 2016.
33. Michael J. Wiener. Bounds on birthday attack times. Cryptology ePrint Archive, Paper 2005/318, 2005. https://eprint.iacr.org/2005/318.
34. Ronald Wolf. Quantum computing: Lecture notes, 07 2019.
35. Henry Yuen. A quantum lower bound for distinguishing random functions from random permutations. *Quantum Inf. Comput.*, 14(13-14):1089–1097, 2014.
36. Mark Zhandry. How to construct quantum random functions. In *FOCS 2012*, pages 679–687. IEEE Computer Society, 2012.
37. Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Information & Computation*, 15(7&8):557–567, 2015.
38. 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.

A The Analysis of Algorithm 1

A.1 Proof of Theorem 1

First, we show if f is well-behaved, there exists $u_1, u_2 \in M_f$ such that $f(u_1) = f(u_2)$ with constant probability(namely there exists a marked vertex). Note that our goal is to find a collisions in M_f, and for any $x \in M_f$ and $y \in S_0$, we have:

$$\Pr_{f \leftarrow D^M}[f(x) = y | x \in M_f] = \frac{\Pr_f[f(x) = y \vee x \in M_f]}{\Pr_f[x \in M_f]} = p_y / \sum_{i \in S_0} p_i$$

.

The other way to think about it, premise that f is well-behaved, since $|M_f| > 6c\sqrt{n_{k_0}}$, namely we can get at least $6c\sqrt{n_{k_0}}$ times to samples from a distribution D', and $D'(y) = p_y / \sum_{i \in S_0} p_i$. Here we can use the following lemma in [33].

Lemma 8. *Let X_D be the random variable for the number of times we sample from the distribution D until we find a collision. then*

$$\Pr[X_D > k] \leq (1 + \frac{k-1}{2^{\beta/2}}) e^{-\frac{k-1}{2^{\beta/2}}},$$

here β is the collision-entropy of D

For a general non-uniform random function, the collision-entropy β is hard to estimate. But according to the definition of c-partition: $p(S_0) \geq p_i > \frac{p(S_0)}{c}$ for any $i \in S_0$, let β'_D be the collision-entropy of D', then we have:

$$\frac{n}{c^2} < 2^{\beta'_D} = \frac{1}{\sum_{y \in S_0} (D'(y))^2} < c^2 n$$

Let $k = 6c\sqrt{n_{k_0}}$, then the probability of finding a collision (u_1, u_2) in M_f is at least

$$1 - \Pr[X'_D > 6c\sqrt{n_{k_0}}] \geq 1 - 7e^{-6} > 0.98.$$

Now let's restrict the condition of a marked vertex from finding a collision in $|u\rangle_L$ to finding (u_1, u_2) in $|u\rangle_L$, which of course reduces the fraction of marked vertices ϵ. To combine the property of well-behave f that $|M_f| < 18c^2\sqrt{n_{k_0}}$, we have:

$$\epsilon \geq \Pr_{u\in V}[u_1, u_2 \in u] = \frac{C(|M_f| - 2, r - 2)}{C(|M_f|, r)} = \frac{r(r-1)}{|M_f|(|M_f| - 1)} > \Theta(\frac{r^2}{n_{k_0}}).$$

(The last inequality holds since we have $r > 1$ by defalt.) □

A.2 Proof of Theorem 2

Algorithm 3 Init(S_{k_0}, f, M')

Input: Collision domain S_{k_0}, $f : [M] \to [N]$ with $f \leftarrow D^M$ and $M' \subset M$.
Output: A state $|\pi_0\rangle$ or $\perp$.

1: Repeat the above operate until get r $|\sigma^*\rangle$:
   (a) Generate the uniform state $|\sigma_0\rangle = \sum_{x\in M'} \frac{1}{\sqrt{M'}}|x\rangle_L$.
   (b) Run BBHT algorithm with $q_1 = \Theta(\sqrt{|M'|/|M_f|})$ queries to f for $|\sigma_0\rangle$, add a new register and perform the unitary transformation: $|x\rangle_L|0\rangle \to |x\rangle_L|1\rangle$ for $f(x) \in S_0$. Measure the second register and get the state in the first register $|\sigma^*\rangle = \sum_{x\in M_f} \frac{1}{\sqrt{|M_f|}}|x\rangle_L$ if the result is 1, otherwise discard it.
2: Take $|\sigma^*\rangle_L^{\otimes r} := \otimes_{i\in[r]} \sum_{u_i\in M_f} \frac{1}{\sqrt{|M_f|}}|u_i\rangle_L$ as input and pass the following test to get

$$\sum_{\substack{|u|=r, u_i\in M_f \\ u_i\neq u_j}} \frac{1}{\sqrt{C(M,r)}} \bigotimes_{i\in[r]} |u_i\rangle_L$$

.
   (a) Construct a function $F : (M_f)^r \to \{0, 1\}$ such that $F(u_1, \ldots, u_r) = 1$ iff $u_i \neq u_j$ for any $i \neq j$. Perform BBHT algorithm with F and an appropriate $q_2 = \Theta(\sqrt{\frac{C(|M_f|,r)}{|M_f|^r}})$ to amplify the amplitude of $|u\rangle$ such that $F(u) = 1$.
   (b) add a new register and perform the unitary transformation: $|u\rangle_L|0\rangle \to |u\rangle_L|F(u)\rangle$. Measure the second register and get the state in the first register if the result is 1, otherwise $\perp$.
3: add new registers $|0\rangle_R|0\rangle_d$ and perform the following transformation to get the initial state $|\pi_0\rangle$:

$$\sum_{\substack{|u|=r, u_i\in[M] \\ u_i\neq u_j}} \frac{1}{\sqrt{C(M,r)}}|u\rangle_L|0\rangle_R|0\rangle_d \to \sum_{\substack{|u|=r, u_i\in[M] \\ u_i\neq u_j}} \frac{1}{\sqrt{C(M,r)}}|u\rangle_L|0\rangle_R|f(u)\rangle_d$$

Now we give the algorithm $\mathbf{Init}(S_{k_0}, f, M')$.

Noted that in terms of query complexity, the amplitude amplification in step 2 of Algorithm 3 can be completed without any queries. Therefore, we assume by default that step 2 always succeeds with constant probability, and we only consider the number of queries required for step 1 and step 3 (But if we consider the time complexity, the cost of step 2 needs to be analyzed).

Assuming f is well-behaved, namely $|M_f| = \Theta(\sqrt{n_{k_0}})$. According to BBHT algorithm, the expected queries of getting a state $|\sigma^*\rangle$ is $\Theta(\sqrt{1/n_{k_0}p(S_{k_0})})$ queries, namely the expected queries in step 1 is $\Theta(r/\sqrt{n_{k_0}p(S_{k_0})})$. The number of queries required in step 3 is r, so the expected queries of generating $|\pi_0\rangle$ is $\Theta(r/\sqrt{n_{k_0}p(S_{k_0})}) + r = \Theta(r/\sqrt{n_{k_0}p(S_{k_0})})$ since $n_{k_0}p(S_{k_0}) < c$ by the definition of c-partition. □

A.3 Proof of Theorem 3

From the original method of implementing U_P [4], we can intuitively see the changes brought by our modification to the update cost:

The original method of implementing U_P is mainly as follows:

1. Add a new register and generate the state $\sum_{i\in[r]} \frac{1}{\sqrt{r}}|i\rangle_r$
2. For any $i \in [r]$, perform the unitary that acts as

$$|u_i\rangle_L|0\rangle_R|j\rangle_r \to \begin{cases} |u_i\rangle_L|u_i\rangle_R|j\rangle_r, & \text{if } i \neq j\,; \\ |u_i\rangle_L|0\rangle_R|j\rangle_r, & \text{if } i = j\,, \end{cases}$$

   (That is, construct $v := (v_1, \ldots, v_r)$ such that $v_i = u_i$ for any $i \neq j$).
3. Generate the state $|\eta\rangle = \sum_{x\neq u_i} \frac{1}{\sqrt{M-r}}|x\rangle$, for the jth register of $|v\rangle_R$ (namely the only register that is $|0\rangle_R$), perform transformations: $|0\rangle_R|\eta\rangle \to |\eta\rangle_R|\eta\rangle \to |\eta\rangle_R|0\rangle$ (to add a random v_j such that $(u, v) \in E$ over the Johnson graph $J(M, r)$).

Noted that all of the above operations can be completed without any query to f, so its cost is 0. Within our constraints, step one and step two remain unchanged, and what we need to think about is step 3, we need to guarantee not only $v_j \neq u_i$ for any $i \in [r]$ but also $v_j \in M_f$ to make sure that $v := (v_1, \ldots, v_r)$ is a "legitimate" vertex on $J(|M_f|, r)$.

The approach to solving this problem is also natural. Similarly, we use BBHT algorithm to get the quantum states

$$|\sigma^*\rangle = \sum_{x\in M_f} \frac{1}{\sqrt{|M_f|}}|x\rangle,$$

then we just take $|u\rangle_L \otimes |\sigma^*\rangle$ as input and run the test in step 2 of Algorithm 3 to get

$$|\eta^*\rangle = \sum_{\substack{x\in M_f \\ x\neq u_i}} \frac{1}{\sqrt{|M_f| - r}}|x\rangle.$$

Therefore, the expected number of queries of implementing U_P is $\Theta(\frac{1}{\sqrt{n_{k_0} p(S_{k_0})}})$ (the expected queries required to construct the state $|\sigma^*\rangle$). □

B The Analysis of Algorithm 2

B.1 Proof of Theorem 4

We give the modified algorithm as follows:

Algorithm 4 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 $\perp$.
1: Do $\texttt{Expt}_0$: Choose $x_0 \leftarrow [N]$ uniformly at random
2: 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.
3: 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. For the above algorithm, the total number of queries Q is

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

The proof of Theorem 4 will use the following fact, and the proof is easy and hence omitted.

Proposition 7. *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 $b/3 \le a_j < b$.*

Proof of Theorem 4. From Lemma 2 and the note there, we see that $\texttt{Expt}_0$ succeeds with probability $\sin^2 \theta = t_0/N$. For any $i \ge 1$, the success probability of $\texttt{Expt}_i$ is $\sin^2 \left(2 \cdot 3^{i-1} + 1\right)\theta$.

For our purpose, let $a_0 := \theta, a_i := (2 \cdot 3^{i-1} + 1)\theta$ and $b := 3\pi/4$. We have $0 < a_0 = \theta \le \pi/2 < 3\pi/4 = b$, and $a_{i+1} = (2 \cdot 3^i + 1)\theta < 3(2 \cdot 3^{i-1} + 1)\theta = 3a_i$ for all i. Proposition 7 tells that there is a j such that $\pi/4 \le a_j < 3\pi/4$. Let j be the least such index. We then have $\sin^2 a_j \ge 1/2$. 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 $1/2$. Therefore, the probability of success of the algorithm is 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. □

B.2 Proof of Proposition 1

For any $j \in S_i$ and the collision domain S_{k_0}, we have, according to Definition 2 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 5) and the definition of well-behaved function, we get

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

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

B.3 Proof of Theorem 6

We also obtain the following straightforward conclusion.

Lemma 9. *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 6 by combining the Höffding's inequality with the lemma.

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 6. □

B.4 Proof of Theorem 7

The remaining part of this subsection is to prove Theorem 7. 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 analyzing 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 the 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 this case with small probability P_1.

Lemma 10. *For any well-behaved random function f and $t < \sqrt{n_{k_0}}$, we have $P_1 < 1/4$.*

Proof of Lemma 10. Since each x_i is uniform from T_f, we have:

$$P_1 = 1 - \prod_{i=0}^{t-1} \frac{|T_f| - i}{|T_f|} \leq 1 - \prod_{i=0}^{t-1} \Big(1 - \frac{3\,c\,i}{2Mn_{k_0}p(S_{k_0})}\Big)$$
$$\overset{(*)}{<} 1 - \exp\Big(-\sum_{i=0}^{t-1} \frac{3\,c\,i}{2Mn_{k_0}p(S_{k_0}) - 3\,c\,i}\Big) < \frac{3\,c^2\,t^2}{4Mn_{k_0}p(S_{k_0}) - 6\,c\,t} \overset{(**)}{<} 1/4.$$

Where the inequality $1 - t > \exp(-\frac{t}{1-t})$ is used in the inequality (*) above, $M > 12\,c^2/p(S_{k_0})$ and $t < \sqrt{n_{k_0}}$ is used in the 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 4, we have in this case:

Lemma 11. *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 12. *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 12. Let's calculate the probability of getting a white pair from sampling in an experiment. By the meaning of white pair, we know:

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

Let ℓ_{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 ℓ_{w} is at least $3t/4$. Again, by Höffding's inequality, we have:

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

That ends the proof of Lemma 12. □

To combine the results discussed 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

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

As desired in Theorem 7. □

C Analysis for the Lower Bound

C.1 Proof of Theorem 8

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 $g \leftarrow (KN)^M$, 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 $x \in [M]$, $y \in [N]$,

$$\Pr_{g \leftarrow (KN)^M}[h_g(x) = y] = \Pr_{g \leftarrow (KN)^M}[g(x) \in s_y] = \frac{\lfloor KNp_y \rfloor}{KN} = \frac{KNp_y}{KN} = p_y. \quad (17)$$

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 (17), 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 a 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

$$\begin{aligned}
\mathbf{P}_1 &:= \Pr[(x_1, x_2), x_1 \neq x_2, h(x_1) = h(x_2) \in S_r \,:\, h \leftarrow D^M, (x_1, x_2) \leftarrow \mathcal{A}^h] \\
&= \Pr[(x_1, x_2), x_1 \neq x_2, h'_g(x_1) = h'_g(x_2) \in S_r \,:\, g \leftarrow (KN)^M, (x_1, x_2) \leftarrow \mathcal{A}^{h'_g}] \\
&< \Pr_{g \leftarrow (KN)^M}[(x_1, x_2), x_1 \neq x_2, g(x_1) = g(x_2) \in s_i, i \in S_r \,:\, (x_1, x_2) \leftarrow \mathcal{B}^g] \\
&\quad + \Pr[(x_1, x_2), \exists x_i \ \texttt{s.t.}\ g(x_i) \in s_{N+1} \,:\, g \leftarrow (KN)^M, (x_1, x_2) \leftarrow \mathcal{B}^g] \\
&\leq \mathbf{P}_2 + 2\Pr[g(x) \in s_{N+1} : g \leftarrow (KN)^M, x \leftarrow \mathcal{C}^g] = \mathbf{P}_2 + 2\Pr[\text{find}], \qquad (18)
\end{aligned}$$

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 (18) that the gap between $\mathbf{P}_1$ and $\mathbf{P}_2$ doesn't exceed 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 [38], after q queries, we have

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

As desired. □

C.2 Proof of Theorem 10

Let's calculate the probability of getting x that satisfies $g(x) \in S'_r$ after i quantum queries, where $S'_r := \bigcup_{j \in S_r} s_j$.

For this purpose, we use the same idea as in [38] to classify the basic states. Suppose that just before i'th queries the joint state is

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

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

1. P is 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, $\|P|\psi_i\rangle\|^2$ is the probability that D contains at least one pre-image of S'_r just before i'th queries.)
2. Q is the projection onto the span of all states $|x,y,z\rangle \otimes |\mathbf{D}\rangle$ such that (a) there is no x' such that $\mathbf{D}(x') \in S'_r$, (b) $\mathbf{D}(x) = \perp$ and (c) $y \neq 0$.
3. R is the projection onto the span of all states $|x,y,z\rangle \otimes |\mathbf{D}\rangle$ satisfying that (a) $\mathbf{D}(x) \neq \perp$, $\mathbf{D}(x) \notin S'_r$, (b) there is no x' such that $\mathbf{D}(x') \in S'_r$ and (c) $y \neq 0$.
4. T is the projection onto the span of all states $|x,y,z\rangle \otimes |\mathbf{D}\rangle$ such that (a) there is no x' such that $\mathbf{D}(x') \in S'_r$ and (b) $y = 0$.

According to the classification above, it is easy to see that for any $i \in [q]$

$$\|P\mathcal{O}P|\psi_i\rangle\| \leq \|P|\psi_i\rangle\|, \text{ and } \|P\mathcal{O}T|\psi_i\rangle\| = 0\,. \tag{19}$$

In addition, recall that any unitary operators U_j can not change the database $\mathbf{D}$, so we have $\|P|\psi_0\rangle\| = \|P|\psi_1\rangle\| = 0$.

For convenience, we call a pair (x,y) *good* iff $y \in S'_r$ for the rest of our proof. For a basic state $|x,y,z\rangle \otimes |\mathbf{D}\rangle$, 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

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

Therefore, $\|P\mathcal{O}Q|\psi_i\rangle\|^2 \leq \frac{|S'_r|}{KN}\|Q|\psi_i\rangle\|^2$ for any $i \in [q]$, where

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

That is,

$$\|P\mathcal{O}Q|\psi_i\rangle\| \leq \sqrt{n_r p(S_r)}\|Q|\psi_i\rangle\|\,. \tag{20}$$

On the other hand, denote $\mathbf{D} := \mathbf{D}' \cup (x,y')$, then we have:

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

Note that for any y' and w, $(1-(-1)^{y\cdot y'}-(-1)^{y\cdot w})^2 \le 9$, so we have

$$\begin{aligned}
\|P\mathcal{O}R|\psi_i\rangle\|^2 &= \sum_{w\in S'_r} \sum_{\substack{x,y,z,\mathbf{D}' \\ |x,y,z\rangle\otimes|\mathbf{D}'\cup(x,y')\rangle\in R}} |\sum_{y'} \frac{1-(-1)^{y\cdot y'}-(-1)^{y\cdot w}}{KN}\alpha_{i,x,y,z,\mathbf{D}',y'}|^2 \\
&\overset{(*)}{\le} \sum_{w\in S'_r} \sum_{\substack{x,y,z,\mathbf{D}' \\ |x,y,z\rangle\otimes|\mathbf{D}'\cup(x,y')\rangle\in R}} \Big(\Big(\sum_{y'}\frac{9}{K^2N^2}\Big)\cdot\Big(\sum_{y'}|\alpha_{i,x,y,z,\mathbf{D}',y'}|^2\Big)\Big) \\
&\le \frac{9KNn_rp(S_r)\cdot KN}{K^2N^2}\cdot \sum_{\substack{x,y,z,\mathbf{D}',y' \\ |x,y,z\rangle\otimes|\mathbf{D}'\cup(x,y')\rangle\in R}} |\alpha_{i,x,y,z,\mathbf{D}',y'}|^2 \\
&= 9n_rp(S_r)\|R|\psi_i\rangle\|^2.
\end{aligned}$$

Here (*) holds due to the Cauchy–Schwarz inequality. Therefore, for any $i\in[q]$

$$\|P\mathcal{O}R|\psi_i\rangle\| \le 3\sqrt{n_rp(S_r)}\|R|\psi_i\rangle\|\,. \tag{21}$$

In conclusion, we have: $\|P|\psi_{i+1}\rangle\| = \|P\mathcal{O}|\psi_i\rangle\| \le \|P|\psi_i\rangle\| + 3\sqrt{n_rp(S_r)}$ for any $i\in[q]$, that is, after i queries, $\mathbf{D}$ contains a pre-image of S'_r with probability at most $O(i^2n_rp(S_r))$.

Using the similar idea from [27], one may extend the conclusion above to the case that $\mathbf{D}$ contains at least j pre-image of S'_r. Specifically, we define P_j to be the projection onto the span of all states $|x,y,z\rangle\otimes|\mathbf{D}\rangle$ with $\mathbf{D}$ containing at least j pre-image of S'_r. Then for any $i,j\in[q]$, we have

$$\|P_j|\psi_{i+1}\rangle\| \le \|P_j|\psi_i\rangle\| + 3\sqrt{n_rp(S_r)}\|P'_{j-1}|\psi_i\rangle\| \le \|P_j|\psi_i\rangle\| + 3\sqrt{n_rp(S_r)}\|P_{j-1}|\psi_i\rangle\|$$

. The notation P'_{j-1} indicates that the projection onto the span of all states $|x,y,z\rangle\otimes|\mathbf{D}\rangle$ with $\mathbf{D}$ containing exact $j-1$ pre-image of S'_r. It's obvious that $\|P'_{j-1}|\psi_i\rangle\| \le \|P_{j-1}|\psi_i\rangle\|$.

To estimate $\|P_j|\psi_i\rangle\|$, we can think of the pre-images numbers of S'_r in $\mathbf{D}$, namely j as a counter, which is initially set to $j=0$. 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 $O(\sqrt{n_rp(S_r)})$, and then to increase the number in the counter by 1. After i quantum queries, $\mathbf{D}$ contains at least j pre-image **iff** the counter has changed at least j times. More formally, by Stirling's approximation,

we have

$$\begin{aligned}
\|P_j|\psi_{i+1}\rangle\| &\le \sum_{1\le k_1\le i} 3\sqrt{n_r p(S_r)}\|P_{j-1}|\psi_{k_1}\rangle\| \qquad \text{(remaining to change } j-1 \text{ times)} \\
&\le \sum_{1\le k_2<k_1\le i} \left(3\sqrt{n_r p(S_r)}\right)^2 \|P_{j-2}|\psi_{k_2}\rangle\| \\
&\le \cdots \le \sum_{1\le k_j<\cdots<k_1\le i} \left(3\sqrt{n_r p(S_r)}\right)^j \|P_0|\psi_{k_j}\rangle\| \\
&\le C(i,j)\cdot\left(3\sqrt{n_r p(S_r)}\right)^j < (3i\sqrt{n_r p(S_r)})^j/j! \\
&< \left(3e\, i\sqrt{n_r p(S_r)}/j\right)^j. \qquad (22)
\end{aligned}$$

Hence we get the conclusion of Lemma 10. □

C.3 Proof of Theorem 11

First, we prove the following lemma.

Lemma 13. *For any quantum algorithm making queries to the compressed oracle $\mathcal{O}$, then after one query to the superposition state*

$$|\phi_j\rangle := \sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j} \alpha_{x,y,z,\mathbf{D}}|x,y,z\rangle\otimes|\mathbf{D}\rangle$$

, the amplitude on $\mathbf{D}$ containing a solution of Problem 2 can only increase by $O(\sqrt{j\,p(S_r)})$.

The proof of Lemma 13. 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 $\exists x_1, x_2$ satisfying $\mathbf{D}(x_1), \mathbf{D}(x_2) \in s_k, s_k \subseteq S'_r$(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) $\neg\exists x_1, x_2$ such that $\mathbf{D}(x_1), \mathbf{D}(x_2) \in s_k, s_k \subseteq S'_r$, (b) $\mathbf{D}(x) = \perp$ and (c) $y \neq 0$.
3. R' be the projection onto the span of all state $|x,y,z\rangle\otimes|\mathbf{D}\rangle$ satisfies that (a) $\neg\exists x_1, x_2$ such that $\mathbf{D}(x_1), \mathbf{D}(x_2) \in s_k, s_k \subseteq S'_r$, (b) $\mathbf{D}(x) \neq \perp$ and (c) $y \neq 0$
4. T' be the projection onto the span of all states $|x,y,z\rangle\otimes|\mathbf{D}\rangle$ satisfying that $\neg\exists x_1, x_2$ such that $\mathbf{D}(x_1), \mathbf{D}(x_2) \in s_k, s_k \subseteq S'_r$, and $y = 0$.

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

$$\|P'\mathcal{O}P'|\phi_j\rangle\| \le \|P'|\phi_j\rangle\|, \qquad \|P'\mathcal{O}T'|\phi_j\rangle\| = 0,$$

and

$$\|P'|\phi_{j+1}\rangle\| = \|P'\mathcal{O}|\phi_j\rangle\| \leq \|P'|\phi_j\rangle\| + \|P'\mathcal{O}Q'|\phi_j\rangle\| + \|P'\mathcal{O}R'|\phi_j\rangle\|.$$

Consider Q' (For convenience, in the proof formula, the condition $|x,y,z\rangle \otimes |\mathbf{D}\rangle \in Q'$(or R') and $\mathbf{D} \in \mathbf{\Phi}_j$ will hold by default and will not be write down), in this case, a new binary pair will be added to the database, so we have

$$P'\mathcal{O}Q'|\phi_j\rangle = \sum_{x,y,z,\mathbf{D}} \sum_{w\in\mathcal{S}} \frac{(-1)^{y\cdot w}}{\sqrt{KN}} \alpha_{x,y,z,\mathbf{D}} |x,y,z\rangle \otimes |\mathbf{D} \cup (x,w)\rangle, \tag{23}$$

where $\mathcal{S}$ is a union of s_k which satisfies that $\exists x, \mathbf{D}(x) \in s_k$. We denote $\mathbb{K}$ as the collection of all such k and $\mathbb{K} \subseteq S_r$. Then $\mathcal{S} = \bigcup_{k\in\mathbb{K}} s_k$. The cardinality of $\mathcal{S}$ is determined by the composition of $\mathbf{D}$ in the basic state. Since $\mathbf{D}$ contains exactly j good pairs, namely $|\mathbb{K}| = j$, and

$$|\mathcal{S}| = \sum_{k\in\mathbb{K}} \lfloor KNp_k \rfloor \leq j\, KNp(S_r). \tag{24}$$

From (23) and (24) we get

$$\|P'\mathcal{O}Q'|\phi_j\rangle\|^2 \leq \frac{|\mathcal{S}|}{KN} \sum_{x,y,z,\mathbf{D}} |\alpha_{x,y,z,\mathbf{D}}|^2 \leq j\, p(S_r) \|Q'|\phi_j\rangle\|^2.$$

In other words, we have

$$\|P'\mathcal{O}Q'|\phi_j\rangle\| \leq \sqrt{j\, p(S_r)} \|Q'|\phi_j\rangle\|. \tag{25}$$

Consider R', similarly we have

$$\begin{aligned} P'\mathcal{O}R'|\phi_j\rangle &= P'\mathcal{O} \sum_{x,y,z,\mathbf{D}',y'} \alpha_{x,y,z,\mathbf{D}',y'} |x,y,z\rangle \otimes |\mathbf{D}' \cup (x,y')\rangle \\ \leq \sum_{w\in\mathcal{S}} \sum_{x,y,z,\mathbf{D}'} \sum_{y'} & \frac{1-(-1)^{y\cdot y'} - (-1)^{y\cdot w}}{KN} \alpha_{x,y,z,\mathbf{D}',y'} |x,y,z\rangle \otimes |\mathbf{D}' \cup (x,w)\rangle \end{aligned} \tag{26}$$

and

$$\begin{aligned} \|P'\mathcal{O}R'|\phi_j\rangle\|^2 &\leq \frac{9jKNn_r p(S_r)\cdot KN}{K^2N^2} \cdot \sum_{x,y,z,\mathbf{D}',y'} |\alpha_{x,y,z,\mathbf{D}',y'}|^2 \\ &= 9jn_r p(S_r) \|R'|\phi_j\rangle\|^2. \end{aligned} \tag{27}$$

The proof details of (26) and (27) can refer to the proof of Theorem 10, with the only difference being that the number of w's available is inconsistent.

In conclusion, for any quantum algorithm that makes queries to compressed random oracle $\mathcal{O}$, then after one query to $|\phi_j\rangle$, the amplitude on $\mathbf{D}$ containing a solution of Problem 2 can only increase by $O(\sqrt{jp(S_r)})$. That concludes Lemma 13. □

Suppose basic state $|x,y,z\rangle\otimes|\mathbf{D}\rangle, \mathbf{D}\in\mathbf{\Phi}_j$ is in the support of $I-P'$, namely, $\neg\exists x_1,x_2$ such that $\mathbf{D}(x_1),\mathbf{D}(x_2)\in s_k, s_k\subseteq S'_r$. Combined with the pigeon cage principle, it is easy to see that j is at most n_r. The discussions above allow us to get

$$
\begin{aligned}
&\|P'|\psi_{i+1}\rangle\| \leq \|P'|\psi_i\rangle\| + \|P'\mathcal{O}Q'|\psi_i\rangle\| + \|P'\mathcal{O}R'|\psi_i\rangle\| \\
&\quad = \|P'|\psi_i\rangle\| + \Big\| \sum_{j=1}^{n_r} \Big(P'\mathcal{O} \sum_{\substack{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j \\ |x,y,z\rangle\otimes|\mathbf{D}\rangle\in Q'}} \alpha_{i,x,y,z,\mathbf{D}}|x,y,z\rangle\otimes|\mathbf{D}\rangle \Big) \Big\| \\
&\quad + \Big\| \sum_{j=1}^{n_r} \Big(P'\mathcal{O} \sum_{\substack{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j \\ |x,y,z\rangle\otimes|\mathbf{D}\rangle\in R'}} \alpha_{i,x,y,z,\mathbf{D}}|x,y,z\rangle\otimes|\mathbf{D}\rangle \Big) \Big\| \\
&\quad \overset{(*)}{\leq} \|P'|\psi_i\rangle\| + \Big(\sum_{j=1}^{n_r} \big(3j\, p(S_r) \sum_{\substack{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j \\ |x,y,z\rangle\otimes|\mathbf{D}\rangle\in Q'}} |\alpha^2_{i,x,y,z,\mathbf{D}}|\big) \Big)^{1/2} \\
&\quad + \Big(\sum_{j=1}^{n_r} \big(3j\, p(S_r) \sum_{\substack{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j \\ |x,y,z\rangle\otimes|\mathbf{D}\rangle\in R'}} |\alpha^2_{i,x,y,z,\mathbf{D}}|\big) \Big)^{1/2}. \\
&\quad \overset{(**)}{\leq} \|P'|\psi_i\rangle\| + \Big(\sum_{j=1}^{n_r} \big(6j\, p(S_r) \sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j} |\alpha^2_{i,x,y,z,\mathbf{D}}|\big) \Big)^{1/2}. \qquad (28)
\end{aligned}
$$

The inequality (*) holds due to the fact that for any two states $|\phi_i\rangle$, $|\phi_j\rangle$ (defined as Lemma 11) and $i\neq j$, $P'\mathcal{O}Q'|\phi_i\rangle$ (resp. $P'\mathcal{O}R'|\phi_i\rangle$) must be orthogonal to $P'\mathcal{O}Q'|\phi_j\rangle$ (resp. $P'\mathcal{O}R'|\phi_j\rangle$) (It is easy to see from the proof of Lemma 11 in Appendix C). The inequality (**) holds due to the simple inequality $\sqrt{a}+\sqrt{b}\leq\sqrt{2(a+b)}$. □

C.4 Proof of Theorem 9

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 by $\mathbf{\Phi}'$ the set of databases that contains at least one good pair, then for any $i \in [q]$, we have:

$$\begin{aligned}\Big(\sum_{j=1}^{n_r}\big(6j\,p(S_r)\sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j}|\alpha^2_{i,x,y,z,\mathbf{D}}|\big)\Big)^{1/2} &\le \Big(6C\,p(S_r)\sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}'}|\alpha^2_{i,x,y,z,\mathbf{D}}|\Big)^{1/2}\\ &\le \Big(6C\,p(S_r)\cdot 3e\,i\sqrt{C\,p(S_r)}\Big)^{1/2}\\ &\le \sqrt{18e}C^{3/4}i^{1/2}p^{3/4}(S_r).\end{aligned}$$

Together with (28), we get

$$\|P'|\psi_{i+1}\rangle\| \le \|P'|\psi_i\rangle\| + \sqrt{18e}C^{3/4}i^{1/2}p^{3/4}(S_r).$$

So after q queries, the success probability $\mathbf{P}_2$ in solving Problem 2, as defined in Theorem 8, will be

$$\begin{aligned}\sqrt{\mathbf{P}_2} = \|P'|\psi_{q+1}\rangle\| &\le \|P'|\psi_0\rangle\| + \sum_{i=1}^{q}\sqrt{18e}C^{3/4}i^{1/2}p^{3/4}(S_r)\\ &< \sqrt{18e}C^{3/4}q^{3/2}p^{3/4}(S_r).\end{aligned}$$

So in this case, a quantum algorithm with q queries can solve Problem 2 with probability at most $O(q^3p^{3/2}(S_r))$. In other words, $\Omega((n_rp^3(S_r))^{-1/6})$ 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 $k_i := \max\{6e\cdot i\sqrt{n_rp(S_r)}, n_r^{1/4}\}$ for any $i \in [q]$. We have

$$\begin{aligned}&\Big(\sum_{j=1}^{n_r}\big(6j\,p(S_r)\sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j}|\alpha^2_{i,x,y,z,\mathbf{D}}|\big)\Big)^{1/2}\\ &\quad\le \sqrt{\sum_{j=1}^{k_i-1}\Big(6j\,p(S_r)\sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j}|\alpha^2_{i,x,y,z,\mathbf{D}}|\Big)} + \sqrt{\sum_{j=k_i}^{n_r}\Big(6j\,p(S_r)\sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j}|\alpha^2_{i,x,y,z,\mathbf{D}}|\Big)}\\ &\quad\le \sqrt{6k_ip(S_r)\sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j,j<k_i}|\alpha^2_{i,x,y,z,\mathbf{D}}|} + \sqrt{6n_rp(S_r)\sum_{x,y,z,\mathbf{D}\in\mathbf{\Phi}_j,j\ge k_i}|\alpha^2_{i,x,y,z,\mathbf{D}}|}\\ &\quad\le \sqrt{6k_ip(S_r)\cdot 1} + \sqrt{6n_rp(S_r)}\cdot\Big(\frac{3e\,i\sqrt{n_rp(S_r)}}{k_i}\Big)^{k_i}.\end{aligned}$$

Again with (28), we get, in this case

$$\|P'|\psi_{i+1}\rangle\| \le \|P'|\psi_i\rangle\| + \sqrt{6k_ip(S_r)} + \sqrt{6n_rp(S_r)}\cdot\Big(\frac{3e\,i\sqrt{n_rp(S_r)}}{k_i}\Big)^{k_i}. \quad (29)$$

Since k_i is an increasing sequence, it has

$$\begin{aligned}\|P'|\psi_{q+1}\rangle\| &\le \sum_{i=1}^{q}\Big(\sqrt{6k_i p(S_r)\cdot 1}+\sqrt{6n_r p(S_r)}\cdot\big(\frac{3e\,i\sqrt{n_r p(S_r)}}{k_i}\big)^{k_i}\Big)\\ &= \sum_{i=1}^{q}\sqrt{6k_i p(S_r)\cdot 1}+\sum_{i=1}^{q}\sqrt{6n_r p(S_r)}\cdot\Big(\frac{3e\,i\sqrt{n_r p(S_r)}}{k_i}\Big)^{k_i}\\ &\le q\sqrt{6k_q p(S_r)}+\sum_{i=1}^{q}\sqrt{6n_r p(S_r)}\cdot\Big(\frac{3e\,i\sqrt{n_r p(S_r)}}{k_i}\Big)^{k_i}. \end{aligned}\tag{30}$$

According to the definition of k_i, for any $i\in[q]$, it holds that

$$\sqrt{6n_r p(S_r)}\cdot\Big(\frac{3e\,i\sqrt{n_r p(S_r)}}{k_i}\Big)^{k_i}\le\sqrt{6n_r p(S_r)}\cdot\big(\frac{1}{2}\big)^{n_r^{1/4}},$$

and hence

$$\sum_{i=1}^{q}\sqrt{6n_r p(S_r)}\cdot\Big(\frac{3e\,i\sqrt{n_r p(S_r)}}{k_i}\Big)^{k_i}\le q\sqrt{6n_r p(S_r)}\cdot\Big(\frac{1}{2}\Big)^{n_r^{1/4}}.\tag{31}$$

Together with (30) and (31), give

$$\begin{aligned}\sqrt{\mathbf{P}_2}=\|P'|\psi_{q+1}\rangle\| &\le q\sqrt{6k_q p(S_r)}+\sum_{i=1}^{q}\sqrt{6n_r p(S_r)}\cdot\Big(\frac{3e\,i\sqrt{n_r p(S_r)}}{k_i}\Big)^{k_i}\\ &\le q\sqrt{6k_q p(S_r)}+q\sqrt{6n_r p(S_r)}\cdot\Big(\frac{1}{2}\Big)^{n_r^{1/4}}.\end{aligned}$$

From $k_q=\max\{6e\cdot q\sqrt{n_r p(S_r)},n_r^{1/4}\}\ge n_r^{1/4}$, it holds that

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

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

$$\begin{aligned}\sqrt{\mathbf{P}_2} &\le q\sqrt{6k_q p(S_r)}+q\sqrt{6n_r p(S_r)}\cdot\Big(\frac{1}{2}\Big)^{n_r^{1/4}}\\ &= q\sqrt{6n_r p(S_r)}\cdot\sqrt{\frac{k_q}{n_r}}+q\sqrt{6n_r p(S_r)}\cdot\Big(\frac{1}{2}\Big)^{n_r^{1/4}}\\ &\le q\sqrt{6n_r p(S_r)}\cdot\Big(\sqrt{\frac{k_q}{n_r}}+\Big(\frac{1}{2}\Big)^{n_r^{1/4}}\Big)\\ &\le q\sqrt{6n_r p(S_r)}\cdot(1+O(1))\sqrt{\frac{k_q}{n_r}}\\ &= O\Big(q\sqrt{k_q p(S_r)}\Big).\end{aligned}\tag{32}$$

Next we consider two cases according to the value of k_q.

When $k_q = n_r^{1/4}$, from (32), we have

$$\mathbf{P}_2 = \|P'|\psi_{q+1}\rangle\|^2 \leq O\Big(q^2 k_q p(S_r)\Big) = O\Big(q^2 n_r^{1/4} p(S_r)\Big). \tag{33}$$

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

$$q < \frac{n_r^{1/4}}{6e\sqrt{n_r p(S_r)}}.$$

These, together with (33) and $n_r = \omega(1)$, give

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

When $k_q = 6e \cdot q\sqrt{n_r p(S_r)}$, we have

$$\mathbf{P}_2 = O\Big(q^3 n_r^{1/2} p^{3/2}(S_r)\Big). \tag{34}$$

It is not hard to see that (33) and (34) 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 9. □

D A Brief Analysis on the Properties of $\gamma(c)$

D.1 Proof of Proposition 3

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 (16), 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 3. □

D.2 Proof of Proposition 4

Recall that $\gamma^{-1}(c) = \max_{i\in[\ell]}\{n_i p^3(S_i)\} = n_{k_0} p^3(S_{k_0})$. From (16), we know

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

That is $2^\beta/c \le \gamma(c)$.

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

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

Suppose that $\alpha > 0$ satisfies

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

Then, (35) becomes

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

Note that α can be bounded, for any non-uniform distribution D, as follows.

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

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

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

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

– 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

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

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 $c > 1$ and the condition $\frac{3c}{c-1} \le \alpha$,

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

– 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 $\frac{3\,c}{c-1} \cdot \frac{p(S_{k_0})}{p(S_{k_0+k})} \leq \alpha$ (It's easy to see in this case we also have $\frac{c}{c-1} \leq \frac{\alpha}{3} \cdot \frac{p(S_{k_0+k})}{p(S_{k_0})} < \frac{\alpha}{3}$), then

$$\begin{aligned} 2^{-2\beta} &< (\alpha + \frac{c}{c-1})^2 \cdot \frac{9c^2}{2(c-1)\alpha^2} \cdot \gamma^{-1}(c) \\ &< \frac{4\alpha}{3} \cdot \frac{9c^2}{2(c-1)\alpha^2} \cdot \gamma^{-1}(c) < \frac{16\,c^3}{(c-1)^2} \cdot \gamma^{-1}(c)\,. \end{aligned} \tag{42}$$

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

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

As desired. □

D.3 An Example for Separating β, $\mathtt{k}$ and $\gamma(c)$

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 shows 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 [8].

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

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

Therefore, in this case, the upper and lower bounds in this work are better than the best prior bounds.

D.4 Proof of Proposition 5

According to Definition 3, we have

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

From (16) 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 $c\,p^2(S_{k_0}) \geq 2^{-3\mathtt{k}}$, namely $p(S_{k_0}) \geq c^{-1/2} \cdot 2^{-\frac{3\mathtt{k}}{2}}$.

On the other hand, since

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

Note that 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. □

D.5 Proof of Proposition 6

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 $S_{i^*}^{(1)}$ 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. □