Quantum Multi-Collision Lower Bounds for Arbitrary Bounded Types

Tianci Peng* Shiqi Hou† Xiaoyang Dong‡

Abstract

We study quantum collision finding for functions of prescribed exact type. For fixed constants $k \geq r \geq 2$ and $N = kn$, consider functions $f : [N] \to \mathbb{Z}_q$ of type (k^n), with exactly n distinct output values and k preimages for each value. We prove that the worst-case bounded-error quantum query complexity of finding r-collisions in ℓ distinct equality classes is

$$\Theta_{k,r}\left(\ell^{2^{r-1}/(2^r-1)} N^{(2^{r-1}-1)/(2^r-1)}\right)$$

for $1 \leq \ell \leq N^{2^{-r}}$. The result is uniform over all feasible alphabet sizes $q \geq n$.

Our proof uses the compressed permutation oracle (CPO). We establish a recurrence for intermediate collision counts and control its accumulated error to obtain database tails exponentially small in ℓ. The improved CPO soundness estimate then converts these database bounds into function-query lower bounds.

Moreover, we extend the analysis to arbitrary exact types with bounded class sizes, without requiring any particular class size to occur linearly often. The resulting lower bounds are tight when forward permutation access is additionally available, although they need not be tight for the original function-query problem. Finally, a sampling reduction relates our collision bounds to sparse k-distinctness instances. A separate analysis using Belovs's transfer framework recovers the same lower-bound exponent, establishing a concrete connection between the two approaches.

1 Introduction

Quantum query complexity provides a framework for understanding the power and limitations of quantum algorithms when access to the input is restricted to oracle queries. In the worst-case setting, an algorithm must succeed on every input satisfying the promise. The polynomial method [5] and the adversary method [2, 23, 25] are central tools for proving quantum query lower bounds. These approaches nevertheless face distinct limitations. Approximate-degree bounds need not be tight. The general adversary bound characterizes bounded-error quantum query complexity for function evaluation up to constant factors [30, 25], but constructing suitable adversary matrices and bounding their norms can be technically demanding.

In the average-case setting, an algorithm's success probability is averaged over a prescribed input distribution. A query lower bound for such a distribution also gives a worst-case lower bound whenever its support consists of valid inputs. The structure of the distribution offers an

*Tsinghua University, Beijing, P. R. China. Email: ptc@mail.tsinghua.edu.cn.

†Tsinghua University, Beijing, P. R. China. Email: seventeenhsq@gmail.com.

‡Corresponding author. Affiliations: Tsinghua University, Beijing, P. R. China; Zhongguancun Laboratory, Beijing, P. R. China; State Key Laboratory of Cryptography and Digital Economy Security, Tsinghua University, Beijing, P. R. China. Email: xiaoyangdong@tsinghua.edu.cn.

additional way to analyze the information acquired through queries. A particularly useful approach is Zhandry's compressed-oracle technique [34], developed in the quantum random-oracle model. The technique can be viewed as a quantum analogue of lazy sampling. Instead of maintaining the entire truth table of a random function, it represents the oracle by a superposition of small, orthogonal databases. Each query changes the database by a bounded amount, and successful search can be related to the amplitude of databases containing the required witnesses. A query lower bound can therefore be obtained by analyzing how quickly these witnesses are formed. The technique has been successfully applied to quantum query lower bounds [15, 19], online extraction [16], and tighter chosen-ciphertext security proofs for key encapsulation mechanisms in the quantum random-oracle model [8, 18].

However, the original construction exploits the independence of function values at different inputs. Extending it to correlated input distributions is substantially more difficult. For a random permutation, for example, injectivity couples the values across the entire domain: the possible output at one input depends on the outputs already assigned elsewhere. These dependencies make it difficult to extend compressed-oracle techniques to random permutations, motivating the development of alternative oracle representations [32, 33, 28, 13].

Collision finding for random functions with prescribed multiplicities is a basic example of this difficulty. Fix a constant $k \geq 2$, and consider a uniformly random function of exact type (k^n): its domain has size $N = kn$, and each of its n image values has exactly k preimages. For $2 \leq r \leq k$, an r-collision consists of r distinct inputs with the same function value. We study the task of finding ℓ such collisions in ℓ distinct equality classes. These problems connect a central benchmark in quantum algorithms with the study of collision resistance. They provide a structured setting in which collisions are abundant, while their exact multiplicities impose dependencies between function values.

The study of quantum collision finding dates back to the algorithm of Brassard, Høyer, and Tapp [12]. The BHT algorithm finds a 2-collision in an arbitrary k-to-one function using $O((N/k)^{1/3})$ expected queries, improving on the classical birthday bound. Higher-order collision algorithms were developed by Hosoyamada, Sasaki, and Xagawa [22], with improved bounds for random functions obtained by Hosoyamada et al. [21]. Liu and Zhandry [26] showed that, for every fixed $r \geq 2$, an r-collision in an r-to-one function with n image values can be found using $O_r(n^{(2^{r-1}-1)/(2^r-1)})$ queries.

For finding many 2-collisions in random functions, Bonnetain et al. [9] introduced reusable quantum walks, extending the range in which the query lower bound is attained. Subsequent work by Bonnetain et al. [10] addresses the remaining parameter range under a heuristic on collision statistics, with a query bound matching the lower bound up to factors polynomial in the input and output bit lengths.

For independently sampled random functions, these algorithmic developments are accompanied by matching lower bounds. Liu and Zhandry [26] used the compressed-oracle technique to establish tight lower bounds for higher-order collision finding. Compressed-oracle arguments also yield lower bounds for finding many 2-collisions, including the strong success-probability bounds of Hamoudi and Magniez [20]. Magniez and Zur [27] study time–space lower bounds for finding a single 2-collision in a uniformly random function under a label-symmetry restriction on the algorithm.

However, these compressed-oracle arguments do not apply to uniformly random functions of exact type (k^n). Once some function values are fixed, they constrain the remaining multiplicities. The resulting distribution lacks the independence used by the original compression. Moreover, conditioning an independently sampled random function on having the desired exact type does not transfer its lower bounds without a separate argument. Thus the tight lower bounds for independent random-function inputs leave the corresponding exact-type problems unresolved.

For exact-multiplicity inputs, the established polynomial and adversary arguments only give tight lower bounds for finding a single 2-collision. Aaronson and Shi [1] proved a matching lower bound using the polynomial method, and Ambainis [3] subsequently extended polynomial lower bounds to smaller output ranges. Belovs and Rosmanis [7] later obtained a tight general adversary bound for the collision problem under a sufficiently large output-alphabet assumption. These proofs use general-purpose techniques developed for worst-case query complexity.

The existing lower-bound arguments for a single 2-collision do not directly extend to tight bounds for multiple targets or higher-order collisions. For uniformly random functions of exact type (k^n), matching lower bounds have remained open in two basic directions: the dependence on a growing number ℓ of requested 2-collisions, and the complexity of finding a single r-collision for $r \geq 3$.

1.1 Summary of Our Contributions

We study both directions described above: finding a growing number of 2-collisions and finding higher-order collisions under an exact-multiplicity promise. Our results treat these two directions simultaneously.

Tight bounds for pure types. We call an exact type of the form (k^n) a *pure type*. For $q \geq n$, a function $f : [kn] \to \mathbb{Z}_q$ of this type takes exactly n distinct values, each with exactly k preimages. Our main result determines the quantum query complexity of finding multiple collisions of any fixed order in such functions.

Theorem 1.1 (Tight collision bounds for pure types)**.** *Fix constants $k \geq r \geq 2$, and let $N = kn$. For sufficiently large N, every alphabet size $q \geq n$, and every integer $1 \leq \ell \leq N^{2^{-r}}$, the quantum query complexity of finding r-collisions in ℓ distinct equality classes of a function of exact type (k^n) is*

$$\Theta_{k,r}\left(\ell^{\,2^{r-1}/(2^r-1)} N^{(2^{r-1}-1)/(2^r-1)}\right).$$

The complexity is measured in the worst case, with success probability at least $2/3$ on every function of the prescribed type. The implicit constants are independent of q.

We prove the lower bound under the uniform distribution on functions of exact type (k^n), which also implies the stated worst-case lower bound. The matching upper bound follows from a multi-target extension of the construction of Liu and Zhandry [26]. In particular, the theorem gives $\Theta_k(\ell^{2/3}N^{1/3})$ queries for ℓ distinct 2-collisions and $\Theta_{k,r}(N^{(2^{r-1}-1)/(2^r-1)})$ queries for a single r-collision. It therefore resolves both exact-type questions in the stated range of target counts. The result includes the smallest feasible alphabet $q = n$, without an additional large-alphabet assumption.

Our lower bounds use the compressed permutation oracle (CPO). Carolan [13] established a sound compressed permutation oracle, extending the database approach of Zhandry [34] to random permutations. More recently, Carolan and Majenz [14] used representation theory to improve its soundness analysis from the $N^{1/12}$ query scale to the square-root scale. The resulting state-comparison error is $O(T/\sqrt{N}+T^2/N)$, and hence $O(T/\sqrt{N})$ for $T = O(\sqrt{N})$. Kim [24] also obtains an $O(T/\sqrt{N})$ soundness bound by comparing exact permutation states and compressed queries within the database space. Our proofs use the state-comparison and output-witness estimates of Carolan and Majenz. These estimates suffice to convert our database estimates into the tight lower bounds above. The restriction $\ell \leq N^{2^{-r}}$ comes from this conversion: our database threshold estimates apply to larger values of ℓ, but the corresponding query scale can exceed $\sqrt{N}$. Extending

CPO soundness to larger query counts in our setting would enlarge the admissible range of ℓ in our results.

Our analysis establishes local transition and commutator bounds for collision counters in a permutation database. These give a recurrence that retains the contribution of large intermediate counts, together with truncation estimates that control the resulting leakage. Thus CPO supports recursive control of intermediate collisions and multiple targets despite the dependence imposed by the exact-type promise.

Lower bounds for arbitrary exact types. We extend the analysis to every exact type $\lambda = (1^{n_1}, \ldots, k^{n_k})$ with fixed maximum block size k. Let $N = \sum_{i=1}^{k} i n_i$ and $n_* = \sum_{i=r}^{k} n_i$, the number of equality classes that can contain an r-collision. For $1 \le \ell \le n_*^{2^{-r}}$, Theorem 4.7 gives the lower bound

$$\Omega_{k,r}\left(\frac{\sqrt{N}\,\ell^{1/(2-2^{1-r})}}{n_*^{1/(2^{r+1}-2)}}\right).$$

The bound holds for every feasible alphabet size $q \ge \sum_{i=1}^{k} n_i$. No intermediate class size is required to occur linearly many times.

For arbitrary types, this function-query lower bound need not be tight. We do obtain a matching upper bound when the algorithm is additionally given forward permutation access in the public-bucket representation (Corollary 4.9). Such access reveals information about bucket membership that is not supplied by the original function oracle. Thus, tightness in this enlarged model does not establish tightness for arbitrary exact-type functions.

Connections with Belovs's framework. We establish concrete connections between our analysis and the transfer framework of Belovs [6]. First, the exact isometric lifting in Section 2.5 relates the exact-type transfer representation to the permutation and label registers used by our construction. Our good-database events define an alternative notion of high knowledge, although their orthogonal projections cannot be directly identified with Belovs's high-knowledge components.

Second, a sampling reduction transfers our pure-type lower bound to a family of sparse k-distinctness inputs. In the large-alphabet regime, Proposition 5.4 recovers the same exponent from Belovs's transfer estimates, uniformly over the fixed profiles in this family. The reduction and the profile analysis therefore provide two routes to the same lower-bound scale.

Finally, we examine the type $(k+1, k^n)$, an open case identified by Belovs. Our comparison isolates the complementary strengths of the two approaches: the database recurrence handles missing intermediate block sizes, whereas the transfer framework retains the original function-query interface. We formulate a possible connection between collision moments and progress toward a hidden exceptional block, and identify the additional estimates needed to turn this connection into a stronger lower bound.

1.2 Technique Overview

The main difficulty is that an exact-type promise correlates the function values. Consequently, the independent-coordinate structure underlying the compressed random oracle is unavailable. We instead represent the function through a random permutation and analyze collision events in its compressed database.

Representing exact types by permutations. Fix public buckets $Q_1, \ldots, Q_h$ partitioning $[N]$ with the prescribed sizes, and let $b(y)$ denote the bucket containing y. Choose a uniformly random

permutation π and an independent uniformly random injection $a : [h] \to \mathbb{Z}_q$. Then

$$f(x) = a(b(\pi(x)))$$

is uniform among functions of the prescribed exact type. The injectivity of a ensures that equal function values correspond exactly to equal bucket indices. This representation is valid for every $q \geq h$.

A function query can be simulated by two forward permutation swap queries: compute $\pi(x)$, apply the bucket label, and uncompute $\pi(x)$. We then replace the permutation oracle by its compressed version. The database analysis applies uniformly over the injective label table, so conditioning on this table introduces no dependence on the alphabet size.

A hierarchy of collision events. For a compressed database D, let $C_j(D)$ count the buckets containing at least j recorded permutation images. If ζ_s is the compressed state after s queries, write $a_{j,m}(s) = \|\mathbf{1}_{\{C_j \geq m\}}\zeta_s\|$. Thus the target event is $C_r \geq \ell$, while the lower counters describe partial progress toward it.

The key local estimate bounds the amplitude that one query can move across a counting threshold. A transition from $C_j = \ell - 1$ to $C_j \geq \ell$ must extend a bucket containing exactly $j - 1$ recorded images. If $C_{j-1} \leq L$, there are at most $(k - j + 1)(L - \ell + 1)$ available images that can cause such a transition. The local compressed-query operator converts this count into the square-root factor in Theorem 3.3:

$$a_{j,\ell}(s+1) \leq a_{j,\ell}(s) + \sqrt{\frac{8(k-j+1)(L-\ell+1)}{N-s}}\, a_{j,\ell-1}(s) + a_{j-1,L+1}(s).$$

The final term accounts for sources with too many intermediate collisions; it cannot simply be discarded.

The levels in this recurrence describe partial occupancy of buckets, rather than actual equality classes of size j. A k-to-one function therefore supports every intermediate level $1, \ldots, k$, even though its input type contains no smaller classes.

Controlling intermediate progress. First-moment estimates determine the constant-success query scale, including its dependence on the number of targets. To obtain exponentially small database tails below this scale, we introduce truncation thresholds for the intermediate counters and retain the contribution of every threshold violation. Orthogonality of the database events permits a comparison with a contractive evolution that enforces these cutoffs.

For arbitrary types, we count only eligible buckets, those of size at least r, at every intermediate level. The hierarchy then starts from the constant counter $C_0^* = n_*$. Using the same eligible set throughout prevents smaller buckets from inflating the bounds on intermediate progress toward an r-collision.

We impose the intermediate cutoffs simultaneously. Keeping the dependence on the query step allows the leakage at each layer to be summed through the identity

$$\sum_{s=0}^{T-1} \binom{s}{L} = \binom{T}{L+1}.$$

This yields the finite tail bound in Theorem 4.4, without an additional factor of T multiplying a uniform leakage estimate. Choosing the cutoffs appropriately gives database amplitudes exponentially

small in ℓ below the threshold

$$\frac{\sqrt{N}\,\ell^{1/(2-2^{1-r})}}{n_*^{1/(2^{r+1}-2)}}.$$

For pure types, $n_* = N/k$, recovering the scale in Theorem 1.1.

Returning to function-query algorithms. To connect a successful output with the database event, we append queries at the reported inputs and apply the output-witness estimate. The soundness comparison is applied once to the final state, giving the interface in (4.26). At a sufficiently small constant multiple of the claimed query threshold, the database amplitude and the simulation error are both small enough to rule out constant success probability. The stated range of ℓ ensures that this threshold is at most $\sqrt{N}$.

For the matching pure-type upper bound, we use the construction of Liu and Zhandry [26] with final list size ℓ. Rebalancing the intermediate list sizes gives the claimed dependence on ℓ. All buckets have the same size, so this construction uses only function queries and works on every input of the prescribed pure type.

Sampling and the transfer framework. The connection with k-distinctness comes from restricting a random k-to-one function on N inputs to a uniformly sampled set of $m = \Theta(N^{1-1/k})$ inputs. At this scale, the number of sampled k-blocks has a Poisson limit with constant mean, while the number of sampled j-blocks concentrates at $\Theta(N^{1-j/k})$ for $j < k$. The event of having a unique k-block and typical lower-layer counts occurs with probability $\Omega(1)$. Conditioned on its type, the restricted function is uniform within that exact type.

Each query to the restricted function requires one query to the original function. The reduction therefore transfers our single-target bound to the resulting sparse distribution, with exponent $k(2^{k-1}-1)/((k-1)(2^k-1))$ in terms of m. Tracking the individual layer sizes in Belovs's query-gain estimates yields the same exponent for the corresponding fixed profiles under the large-alphabet assumption in Proposition 5.4.

This agreement does not identify the two progress representations. Our counters are orthogonal projections in the compressed database, whereas Belovs's knowledge decomposition is defined before transfer. Moreover, our analysis allows independent forward permutation queries, which can reveal the location of an exceptional public bucket. The discussion in Section 6 examines how recursive database control might be combined with estimates that preserve the restricted function-query interface.

Organization. After introducing the necessary preliminaries, we establish the collision bounds for pure types in Section 3 and then extend the analysis to arbitrary exact types in Section 4. We next present the sampling reduction to sparse k-distinctness instances and the corresponding analysis within Belovs's framework in Section 5. Finally, in Section 6 we examine the type $(k+1, k^n)$ and discuss the additional estimates needed to combine the strengths of the two approaches.

2 Preliminaries

For a positive integer N, write $[N] = \{1, \dots, N\}$. Unless otherwise stated, all asymptotic statements and limits are taken as $N \to \infty$. For nonnegative integers a and b, let

$$(a)_b = a(a-1)\cdots(a-b+1)$$

denote the falling factorial, with $(a)_0 = 1$ and $(a)_b = 0$ when $b > a$.

All Hilbert spaces are finite-dimensional. We use $\|v\|$ for the Euclidean norm of a vector and $\|A\|$ for the induced operator norm. The identity operator is denoted by I, and $\mathbf{1}_E$ is the indicator of an event E. For an orthogonal projection P, write $P^\perp = I - P$ for the projection onto the orthogonal complement of its range. For self-adjoint operators A and B, we write $A \preceq B$ if $B - A$ is positive semidefinite. In particular, for orthogonal projections P_1 and P_2, $P_1 \preceq P_2$ means that P_1 is a subprojection of P_2, equivalently, $\mathrm{ran}(P_1) \subseteq \mathrm{ran}(P_2)$.

We identify an alphabet of size q with $\mathbb{Z}_q$ when specifying a query oracle. A function $f : [N] \to \mathbb{Z}_q$ has exact type $\lambda = (1^{n_1}, \ldots, k^{n_k})$ if its equality classes form a partition of $[N]$ containing exactly n_i blocks of size i for each $i \in [k]$. Thus

$$N = \sum_{i=1}^{k} i n_i, \qquad h = \sum_{i=1}^{k} n_i.$$

Such a type is realizable precisely when $q \geq h$. Our main collision lower bounds impose no further alphabet-size condition. We write $\mu(f)$ for this equality partition, abbreviating it by μ when f is clear from context. In particular, μ is uniquely determined by f, and distinct blocks of μ have distinct function values.

2.1 Quantum queries and purified input registers

Let $f : [N] \to \mathbb{Z}_q$ be the input function. The algorithm's register is $\mathcal{A} = \mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Z}$, where $\mathcal{X}$ holds a query input $x \in [N]$, $\mathcal{Y}$ holds a phase label $c \in \mathbb{Z}_q$, and $\mathcal{Z}$ contains the output and auxiliary registers. A query acts as

$$O_f|x, c, z\rangle = \omega_q^{cf(x)}|x, c, z\rangle, \qquad \omega_q = e^{2\pi i/q}. \tag{2.1}$$

By purifying internal randomness and deferring measurements, a T-query algorithm can be written as

$$U_T O_f U_{T-1} O_f \cdots U_1 O_f U_0 |\varphi_0\rangle,$$

followed by a projective measurement of its output register. The normalized initial state $|\varphi_0\rangle$ and all unitaries U_t are independent of f. Query complexity counts oracle calls and places no restriction on these input-independent operations.

We write $|\varphi_{t,f}\rangle$ for the state immediately after the t-th query and $|\varphi'_{t,f}\rangle$ for the state after the subsequent internal unitary. With $|\varphi_{0,f}\rangle = |\varphi_0\rangle$, the evolution satisfies

$$\begin{aligned} |\varphi'_{t,f}\rangle &= U_t|\varphi_{t,f}\rangle \quad (0 \leq t \leq T), \\ |\varphi_{t+1,f}\rangle &= O_f|\varphi'_{t,f}\rangle \quad (0 \leq t < T). \end{aligned}$$

Thus $|\varphi'_{t,f}\rangle$ is immediately before query $t+1$ when $t < T$, while $|\varphi'_{T,f}\rangle$ is the final state on which the output measurement is performed.

For distributional lower bounds, let $\mathcal{D}$ be a probability distribution on a finite family Ω of input functions. The function is sampled once and remains fixed throughout the computation. We purify this choice by introducing an inaccessible function register $\mathcal{F}$ with orthonormal basis $\{|f\rangle : f \in \Omega\}$, initialized in

$$|\mathcal{D}\rangle_{\mathcal{F}} = \sum_{f \in \Omega} \sqrt{\mathcal{D}(f)}\, |f\rangle.$$

The algorithm's internal unitaries act as $U_t \otimes I_{\mathcal{F}}$, and the joint query operator is

$$\mathcal{O} = \sum_{f \in \Omega} O_f \otimes |f\rangle\langle f|.$$

In particular, $\mathcal{O}|x, c, z\rangle|f\rangle = \omega_q^{cf(x)}|x, c, z\rangle|f\rangle$: the function register coherently controls the query without changing its computational-basis label.

The corresponding joint states are

$$\begin{aligned} |\psi_t^{\mathcal{D}}\rangle &= \sum_{f \in \Omega} \sqrt{\mathcal{D}(f)}\, |\varphi_{t,f}\rangle \otimes |f\rangle, \\ |\psi_t^{\mathcal{D}\prime}\rangle &= \sum_{f \in \Omega} \sqrt{\mathcal{D}(f)}\, |\varphi'_{t,f}\rangle \otimes |f\rangle. \end{aligned} \tag{2.2}$$

Both are normalized. Tracing out $\mathcal{F}$ gives

$$\mathrm{Tr}_{\mathcal{F}}(|\psi_t^{\mathcal{D}\prime}\rangle\langle\psi_t^{\mathcal{D}\prime}|) = \sum_{f \in \Omega} \mathcal{D}(f)\, |\varphi'_{t,f}\rangle\langle\varphi'_{t,f}|.$$

Hence the purification preserves all measurement statistics accessible to the algorithm. Internal unitaries preserve the reduced state of $\mathcal{F}$, so only oracle queries can change it. This representation allows us to track the algorithm's knowledge through the inaccessible input register.

A search problem is specified by a relation $\mathcal{R} \subseteq \Omega \times \mathsf{Out}$, where Out is the set of possible outputs. For output projectors $\{\Pi_\rho\}_{\rho \in \mathsf{Out}}$, the average success probability is

$$p = \sum_{f \in \Omega} \mathcal{D}(f) \sum_{\rho : (f,\rho) \in \mathcal{R}} \|\Pi_\rho \varphi'_{T,f}\|^2. \tag{2.3}$$

A query lower bound under $\mathcal{D}$ also applies to algorithms required to achieve the same success probability on every input in its support.

2.2 Zhandry's compressed oracle

Zhandry's compressed oracle represents queries to a uniformly random function by local updates to a quantum database [34]. Here $\Omega = \mathbb{Z}_q^N$ and $\mathcal{D}(f) = q^{-N}$, so the initial function state factors into N uniform single-coordinate states. For each coordinate, define the Fourier basis

$$|\widehat{a}\rangle = \frac{1}{\sqrt{q}} \sum_{y \in \mathbb{Z}_q} \omega_q^{ay} |y\rangle.$$

A phase query at x with label c changes the function-table Fourier label σ to $\sigma + c\mathbf{e}_x$. Thus, after t queries, only vectors with $|\operatorname{supp}(\sigma)| \le t$ can occur. The compressed representation turns this sparsity into a database of local records.

Definition 2.1 (Compressed function oracle [34])**.** Adjoin a state $|\perp\rangle$ orthogonal to the value states at each coordinate. Let C_x swap $|\perp\rangle$ and $|\widehat{0}\rangle$ at coordinate x, fixing their orthogonal complement. Extend the local phase operator by

$$P_{x,c}|y\rangle = \omega_q^{cy}|y\rangle \quad (y \in \mathbb{Z}_q), \qquad P_{x,c}|\perp\rangle = |\perp\rangle.$$

The compressed query at (x, c) is

$$O_{x,c}^{\mathrm{comp}} = C_x P_{x,c} C_x. \tag{2.4}$$

The database basis consists of partial assignments $D : [N] \rightharpoonup \mathbb{Z}_q$, with $\perp$ at unrecorded positions, and the database is initialized in $|\varnothing\rangle$.

This database representation has four closely related properties that make it useful for proving quantum query lower bounds.

Perfect simulation. The compressed oracle perfectly simulates the phase oracle, preserving the algorithm's behavior while making information about its queries explicit in a database register. Formally, the original and compressed joint states satisfy the following correspondence.

Lemma 2.2 (Exact function compression [34])**.** *Let $C = \bigotimes_x C_x$, and let $|\psi_t^{\mathrm{u}}\rangle$ be the purified joint state for a uniformly random function. The corresponding compressed joint state satisfies*

$$|\zeta_t\rangle = (I_{\mathcal{A}} \otimes C)|\psi_t^{\mathrm{u}}\rangle$$

and is supported on databases with at most t records. The same identity holds for the primed states. In particular, the compressed oracle perfectly simulates the phase oracle: the algorithm's reduced state is preserved exactly.

Reliable recording. The database also provides a reliable record of information acquired by the algorithm. Zhandry's recording lemma relates the probability p of a correct output to the probability p_{db} that the final database contains a suitable witness for success [34, 26]. This comparison has an error depending on the output size and the alphabet size, even though oracle simulation itself is exact. It allows a lower-bound argument to study the formation of database witnesses instead of analyzing the final output directly.

Local database conditions. The explicit records make it possible to specify a problem-dependent notion of a *good* database. A set $\mathcal{B}$ of databases defines the orthogonal projector

$$P_{\mathcal{B}} = \sum_{D \in \mathcal{B}} |D\rangle\langle D|,$$

with the identity on the algorithm's registers understood. The corresponding truncated state is $P_{\mathcal{B}}|\zeta_t\rangle$. For example, databases containing an r-collision are recognized by the condition $|D^{-1}(y)| \geq r$ for some y. When $\mathcal{B}$ is the chosen witness property,

$$p_{\mathrm{db}} = \|P_{\mathcal{B}}\zeta_T'\|^2.$$

Thus the relevant component is specified directly from the database records. This projection is used only in the analysis and does not require an intermediate measurement.

Bounded growth. Since a compressed query acts only on the record at its queried input, one can analyze transitions into $\mathcal{B}$ through local database updates. Bounds on these transition amplitudes control how much $\|P_{\mathcal{B}}\zeta_t\|$ can increase in a single query. Accumulating the bounds over the computation then limits how quickly the algorithm can produce a database witness. The quantitative estimate depends on the chosen property; the bound on the number of records alone does not supply this amplitude bound.

These four properties also guide the CPO framework: soundness ensures faithful oracle simulation; the fundamental lemma connects algorithmic success to database information; the local database construction makes the desired truncations explicit; and bounded growth controls their evolution under queries. Realizing analogous properties for permutations requires a more involved construction, since injectivity couples the values at different inputs and prevents the independent-coordinate construction above from applying directly. In particular, the CPO soundness estimate gives an approximate simulation in place of the exact equivalence available for random functions.

2.3 Compressed permutation oracles

We use Carolan's compressed permutation oracle [13], together with the improved soundness analysis of Carolan and Majenz [14]. Its orthogonal database basis is indexed by partial injections, allowing witness conditions to be specified directly from the recorded input-output pairs. This framework provides counterparts of the four properties discussed above, with approximate simulation replacing perfect simulation. We recall the construction and the estimates needed below.

Function queries use the phase-query convention introduced above. For permutation queries, we use an auxiliary response register $\mathcal{W}$ with orthonormal basis

$$\{|\perp\rangle\} \cup \{|y\rangle : y \in [N]\}.$$

When writing a permutation query, we display only the input register and $\mathcal{W}$, suppressing all other algorithm registers. We keep $\mathcal{W}$ separate from $\mathcal{A}$, which contains all remaining algorithm-accessible registers. All query counts in this subsection refer to forward permutation swap queries.

Definition 2.3 (Compressed permutation oracle [14, Sections 3.1 and 3.3])**.** For each $y \in [N]$, let S_y be the unitary on $\mathcal{W}$ that swaps $|\perp\rangle$ and $|y\rangle$ and fixes every other basis state.

The database register $\mathcal{L}$ has orthonormal basis

$$\{|D\rangle : D : [N] \rightharpoonup [N] \text{ is a partial injection}\}$$

and is initialized in $|\varnothing\rangle$. Write $|D| = |\operatorname{dom}(D)|$. For a fixed input x, deleting its record partitions this basis into mutually orthogonal subspaces

$$\operatorname{span}\Big(\{|H\rangle\} \cup \{|H[x \mapsto y]\rangle : y \notin \operatorname{im}(H)\}\Big), \qquad x \notin \operatorname{dom}(H).$$

On each such subspace, C_x swaps $|H\rangle$ with

$$|+_{x,H}\rangle = \frac{1}{\sqrt{N - |H|}} \sum_{y \notin \operatorname{im}(H)} |H[x \mapsto y]\rangle$$

and fixes the orthogonal complement of their span.

Let $Q_{x,y}$ be the orthogonal projector on $\mathcal{L}$ onto databases with $D(x) = y$, and let $Q_{x,\perp}$ project onto those with $x \notin \operatorname{dom}(D)$. Define operators on $\mathcal{W} \otimes \mathcal{L}$ by

$$\begin{aligned} \mathsf{U}_x &= I_{\mathcal{W}} \otimes Q_{x,\perp} + \sum_{y \in [N]} S_y \otimes Q_{x,y}, \\ \mathsf{Q}_x &= (I_{\mathcal{W}} \otimes \mathsf{C}_x)\mathsf{U}_x(I_{\mathcal{W}} \otimes \mathsf{C}_x). \end{aligned} \tag{2.5}$$

The full forward compressed permutation oracle is

$$\mathsf{cpO} = \sum_{x\in[N]} |x\rangle\langle x| \otimes \mathsf{Q}_x,$$

where identities on the remaining algorithm registers are suppressed. Thus Q_x describes the query at a fixed input, while cpO allows queries in superposition over inputs.

For a fixed input x, the query preserves each subspace indexed by H, so it changes database size by at most one. The state $|\zeta_s\rangle$ after s compressed queries is therefore supported on $|D| \le s$. Intermediate algorithmic operations act only on $\mathcal{A}\otimes\mathcal{W}$.

For a fixed permutation π, the real forward swap oracle acts as

$$P_\pi|x,w\rangle = |x\rangle \otimes S_{\pi(x)}|w\rangle, \qquad w \in [N]\cup\{\perp\}.$$

Its purified counterpart is

$$\mathsf{PO} = \sum_{\pi\in S_N} P_\pi \otimes |\pi\rangle\langle\pi|.$$

The permutation register is initialized in $|S_N\rangle$ and is inaccessible to the algorithm. The soundness lemma below compares computations using PO with those using cpO.

Lemma 2.4 (CPO soundness [14, Theorem 4.2])**.** *For $N \ge 3$, there is an isometry $\widetilde{V} : \mathbb{C}^{S_N} \to \mathcal{L}$ satisfying*

$$\widetilde{V}|S_N\rangle = |\varnothing\rangle, \qquad |S_N\rangle = \frac{1}{\sqrt{N!}} \sum_{\pi\in S_N} |\pi\rangle.$$

Run the same algorithm with purified random-permutation swap queries and with compressed queries, obtaining final joint states $|\psi_T^{\mathrm{perm}}\rangle$ and $|\zeta_T\rangle$, respectively. The initial algorithm state is the same in both experiments and independent of the oracle register. For $3T-1 \le N$,

$$\big\||\zeta_T\rangle - (I_{\mathcal{A}\otimes\mathcal{W}}\otimes\widetilde{V})|\psi_T^{\mathrm{perm}}\rangle\big\| \le \delta_N(T) := \frac{60T}{\sqrt{N-T+1}} + \frac{T(T+1)}{2(N-2T+1)}. \tag{2.6}$$

This holds for arbitrary auxiliary registers and intermediate unitaries on the accessible space.

Using the improved compression isometry $\widetilde{V}$ of [14], the improved soundness analysis guarantees vanishing simulation error for $T = o(\sqrt{N})$, extending the $T = o(N^{1/12})$ range guaranteed by Carolan's original analysis [13, Theorem 5.19]. The following lemma bounds the square root of the real witness success probability by the corresponding compressed-database amplitude, with additive witness and simulation errors.

Lemma 2.5 (Output witnesses [14, Lemma 5.1])**.** *Let $\mathcal{B}$ be an increasing property of partial injections: $D\in\mathcal{B}$ and $D\subseteq D'$ imply $D'\in\mathcal{B}$. An algorithm making T forward swap queries outputs at most L input-output pairs. Let p be the probability that the pairs agree with the real permutation and, after removing duplicates, form a partial injection in $\mathcal{B}$. Let p_{db} be the probability that the final database after the same T compressed queries belongs to $\mathcal{B}$. If $N\ge 3$ and $3(T+L)-1\le N$, then*

$$\sqrt{p} \le \sqrt{p_{\mathrm{db}}} + L\sqrt{\frac{2}{N-T}} + \delta_N(T+L). \tag{2.7}$$

2.4 From exact-type functions to compressed databases

Fix an exact type $\lambda = (1^{n_1}, \ldots, k^{n_k})$, where

$$N = \sum_{i=1}^{k} i n_i \geq 3, \qquad h = \sum_{i=1}^{k} n_i,$$

and an alphabet size $q \geq h$. Let $\mathcal{F}_{\lambda,q}$ be the set of functions $[N] \to \mathbb{Z}_q$ of exact type λ. Fix public buckets $Q_1, \ldots, Q_h$ partitioning $[N]$ with size multiset λ, and let $b(y)$ denote the index of the bucket containing y.

We now explain how we use the CPO framework in our proof. Our reduction follows the chain

$$A^{O_f} \xrightarrow{\text{exact reduction}} B^{P_\pi} \xrightarrow{\text{CPO comparison}} B^{\mathsf{cpO}}.$$

Here B simulates the function-query algorithm A and appends the queries needed to produce an output witness. The first reduction preserves the success probability under the corresponding input distributions. The second step uses CPO soundness and the output-witness lemma to bound this probability in terms of a compressed-database amplitude.

Exact sampling. Independently sample a uniform permutation $\pi \in S_N$ and a uniform injective label assignment $a : [h] \hookrightarrow \mathbb{Z}_q$. Define

$$f_{\pi,a}(x) = a(b(\pi(x))). \tag{2.8}$$

Proposition 2.6 (Exact sampling of a prescribed type)**.** *The function $f_{\pi,a}$ is uniform over $\mathcal{F}_{\lambda,q}$. More precisely, every $f \in \mathcal{F}_{\lambda,q}$ has exactly*

$$c_\lambda = \prod_{i=1}^{k} n_i! \, (i!)^{n_i}$$

representations of the form (2.8)*. Consequently,*

$$|\mathcal{F}_{\lambda,q}| = \frac{N!(q)_h}{c_\lambda}.$$

Proof. Fix $f \in \mathcal{F}_{\lambda,q}$. For each i, match its n_i equality classes of size i to the public buckets of that size, and choose a bijection from each class to its assigned bucket. There are $n_i!(i!)^{n_i}$ choices. Together, these choices determine π. The function values then determine a uniquely, and a is injective because different equality classes have different values. Thus every f has exactly c_λ representations. Since all $N!(q)_h$ pairs (π, a) are equally likely, the induced function is uniform. □

Simulation of phase queries. One phase query to $f_{\pi,a}$ can be simulated exactly by two forward permutation swap queries. For a fixed label assignment a, define the diagonal unitary

$$D_a|c, w\rangle = \begin{cases} \omega_q^{c\, a(b(w))}|c, w\rangle, & w \in [N], \\ |c, \perp\rangle, & w = \perp. \end{cases}$$

It acts on the original phase-label register and the auxiliary response register $\mathcal{W}$, and requires no oracle query. Starting with $\mathcal{W}$ in $|\perp\rangle$, the simulator performs

$$\begin{aligned}|x,c,z\rangle|\perp\rangle &\xrightarrow{P_\pi} |x,c,z\rangle|\pi(x)\rangle \\ &\xrightarrow{D_a} \omega_q^{c\,a(b(\pi(x)))}|x,c,z\rangle|\pi(x)\rangle \\ &\xrightarrow{P_\pi} \omega_q^{c\,f_{\pi,a}(x)}|x,c,z\rangle|\perp\rangle.\end{aligned}$$

Both calls are forward queries: P_π is an involution, so uncomputing the response does not require access to π^{-1}.

The simulator stores a, while the original algorithm interacts only with the simulated function oracle. The independent randomness in a may be purified in an auxiliary register. The soundness and output-witness lemmas therefore apply to the joint experiment, including this register.

The database bounds below hold uniformly over arbitrary forward permutation-query algorithms with the buckets and label assignment available. This class contains the simulator above and grants more oracle access than the original function-query model. Inverse permutation access is excluded: by choosing r outputs in each of ℓ distinct buckets and querying their preimages, one could otherwise obtain the desired collisions in $r\ell$ queries.

The target database event. Fix $2 \le r \le k$ and $1 \le \ell \le \sum_{i=r}^{k} n_i$. Throughout, finding ℓ r-collisions means outputting ℓ groups of r distinct inputs such that each group has a common function value and different groups have different common values. Thus the groups belong to distinct equality classes.

For a partial injection D, define

$$C_r(D) = \big|\{b : |\operatorname{im}(D) \cap Q_b| \ge r\}\big|, \qquad P_{r,\ell} = \sum_{D:\, C_r(D)\ge\ell} |D\rangle\langle D|.$$

The property $\mathcal{B}_{r,\ell} = \{D : C_r(D) \ge \ell\}$ is increasing. For the joint state $|\zeta_s\rangle$ of a particular compressed-query algorithm, write

$$a_{r,\ell}(s) = \|P_{r,\ell}\zeta_s\|,$$

with identities on all other registers understood. The state includes all auxiliary registers, including the purification of the label assignment.

Theorem 2.7 (Conversion to real success probability)**.** *Suppose an algorithm makes t function phase queries and finds ℓ r-collisions in distinct equality classes with probability p, averaged over the uniform distribution on $\mathcal{F}_{\lambda,q}$. If $6t + 6r\ell - 1 \le N$, then there is an algorithm making at most $2t + r\ell$ forward compressed permutation queries such that*

$$\sqrt{p} \le a_{r,\ell}(2t + r\ell) + r\ell\sqrt{\frac{2}{N - 2t - r\ell}} + \delta_N(2t + 2r\ell). \tag{2.9}$$

Here δ_N is defined in (2.6)*, and $a_{r,\ell}(2t + r\ell)$ is the database amplitude of the compressed algorithm constructed below.*

Proof. Use Proposition 2.6 to sample the function, and simulate the original algorithm using $2t$ forward permutation swap queries. After obtaining its output, read $\pi(x)$ at each of the at most $r\ell$ reported inputs, using fresh blank response registers and retaining the resulting values. These additional reads do not change the distribution of the original output.

Check the originally specified grouping of the inputs. Each group must contain r distinct inputs whose permutation images lie in a single bucket, and different groups must correspond to different buckets. If the output is malformed or the check fails, output the empty list. Otherwise, output the resulting list of permutation input-output pairs. Because a is injective, this check succeeds exactly when the original output is successful.

The resulting real permutation-query algorithm uses at most $2t + r\ell$ queries and outputs at most $r\ell$ pairs. Every reported pair is correct. On success, the list witnesses $\mathcal{B}_{r,\ell}$; on failure, the empty list does not. Its real witness success probability is therefore exactly p. Checking the original grouping is essential: the combined list may contain ℓ bucket witnesses even when the original groups are not valid collisions.

Now run this entire algorithm with compressed queries in place of the permutation swap queries, and let $|\zeta_{2t+r\ell}\rangle$ be its final joint state. All auxiliary registers are retained in this experiment. In particular, the definition of the compressed algorithm does not require the auxiliary response register to return to its blank state after each simulated function query.

Apply Lemma 2.5 to this algorithm, with query count $2t + r\ell$, output-list length $r\ell$, and property $\mathcal{B}_{r,\ell}$. The resulting inequality is exactly (2.9). □

The output-witness lemma already includes the final simulation error $\delta_N(2t + 2r\ell)$, and the reduction holds for every feasible alphabet size $q \geq h$, with no additional dependence on q. Consequently, it remains to bound $a_{r,\ell}(s)$ uniformly over all forward compressed-query algorithms. The following analysis establishes such bounds and then substitutes them into (2.9). The admissible ranges of the resulting function-query lower bounds are determined by controlling both the database term and the additive error terms.

2.5 Relation to Belovs's framework

We now relate our orthogonal database representation to Belovs's transfer framework, which he used to establish a tight quantum query lower bound for k-distinctness [6]. The maps introduced here make this relationship explicit. The success-probability reduction above follows directly from the permutation simulator and the CPO output-witness lemma, independently of these maps.

The strict transfer map. Let $\mathcal{H}_{\mathrm{all}}$ be the function-register space with basis indexed by all functions $[N] \to \mathbb{Z}_q$, and let $\mathcal{H}_{\lambda,q}$ be a separate space with basis indexed by $\mathcal{F}_{\lambda,q}$. Define

$$\Upsilon_\lambda |f\rangle = \begin{cases} \sqrt{\dfrac{q^N}{|\mathcal{F}_{\lambda,q}|}}\,|f\rangle, & f \in \mathcal{F}_{\lambda,q}, \\ 0, & f \notin \mathcal{F}_{\lambda,q}. \end{cases}$$

Writing

$$|u_{\mathrm{all}}\rangle = q^{-N/2} \sum_{f:[N]\to\mathbb{Z}_q} |f\rangle, \qquad |u_{\lambda,q}\rangle = \frac{1}{\sqrt{|\mathcal{F}_{\lambda,q}|}} \sum_{f\in\mathcal{F}_{\lambda,q}} |f\rangle,$$

we have

$$\Upsilon_\lambda |u_{\mathrm{all}}\rangle = |u_{\lambda,q}\rangle.$$

Since the map preserves the function label, it intertwines function queries and algorithm-side unitaries. It therefore converts the purified uniform-function experiment into the purified exact-type experiment. It is generally not an isometry on $\mathcal{H}_{\mathrm{all}}$.

For an exact-type function f, its equality partition $\mu(f)$ is uniquely determined. Thus $|f\rangle$ may equivalently be written as $|\mu(f), f\rangle$; no additional choice of a hidden partition is required.

Lifting to the database representation. For $f \in \mathcal{F}_{\lambda,q}$, let

$$\mathcal{R}_f = \{(\pi, a) : \pi \in S_N,\ a : [h] \hookrightarrow \mathbb{Z}_q,\ a \circ b \circ \pi = f\}.$$

Proposition 2.6 gives $|\mathcal{R}_f| = c_\lambda$.

Definition 2.8 (The database isometry)**.** Define the lifting to permutation and label registers by

$$K_{\lambda,q}|f\rangle = \frac{1}{\sqrt{c_\lambda}} \sum_{(\pi,a)\in\mathcal{R}_f} |\pi, a\rangle.$$

Using the compression isometry $\widetilde{V}$ from Lemma 2.4, define

$$\begin{aligned} J &= (\widetilde{V} \otimes I_a) K_{\lambda,q}, \\ J|f\rangle &= \frac{1}{\sqrt{c_\lambda}} \sum_{(\pi,a)\in\mathcal{R}_f} (\widetilde{V}|\pi\rangle) \otimes |a\rangle. \end{aligned} \tag{2.10}$$

The label register is retained in both maps.

Lemma 2.9 (Exact isometric embedding)**.** *For every $q \geq h$,*

$$K_{\lambda,q}^\dagger K_{\lambda,q} = I, \qquad J^\dagger J = I.$$

Moreover,

$$J|u_{\lambda,q}\rangle = |\varnothing\rangle \otimes |u_{\mathrm{lab}}\rangle, \qquad |u_{\mathrm{lab}}\rangle = \frac{1}{\sqrt{(q)_h}} \sum_{a:[h]\hookrightarrow\mathbb{Z}_q} |a\rangle. \tag{2.11}$$

Proof. The sets $\mathcal{R}_f$ are disjoint and all have size c_λ, so $K_{\lambda,q}$ maps distinct basis vectors to orthonormal vectors. Using $|\mathcal{F}_{\lambda,q}|c_\lambda = N!(q)_h$, we obtain

$$K_{\lambda,q}|u_{\lambda,q}\rangle = |S_N\rangle \otimes |u_{\mathrm{lab}}\rangle.$$

The claims for J follow because $\widetilde{V}$ is an isometry and maps $|S_N\rangle$ to $|\varnothing\rangle$. □

Every pair in $\mathcal{R}_f$ induces the same function oracle O_f. Hence $K_{\lambda,q}$ preserves the function-query experiment exactly. Composing with $\widetilde{V}$ gives an exact isometric embedding of its states, but implementing the evolution with compressed queries remains approximate.

More precisely, let $|\psi_t^{\mathrm{tar}}\rangle$ be the purified exact-type function state after t phase queries, with the simulator's auxiliary registers adjoined in their blank states. Let $|\zeta_{2t}\rangle$ be the state obtained by replacing the simulator's swap calls with compressed queries. Suppressing identities on unaffected registers,

$$\left\|\zeta_{2t} - J\psi_t^{\mathrm{tar}}\right\| \leq \delta_N(2t), \qquad 6t - 1 \leq N. \tag{2.12}$$

Two different knowledge decompositions. We compare the source knowledge decomposition in Belovs's transfer framework with the orthogonal database decomposition for the exact-type

distribution and CPO embedding defined above. Let $\mathcal{M}_\lambda$ be the set of domain partitions of type λ. For $\mu \in \mathcal{M}_\lambda$, define

$$\Upsilon_{\lambda,\mu}|f\rangle = \begin{cases} \sqrt{\dfrac{q^N}{(q)_h}}\,|f\rangle, & \mu(f) = \mu, \\ 0, & \text{otherwise.} \end{cases}$$

Since $|\mathcal{F}_{\lambda,q}| = |\mathcal{M}_\lambda|(q)_h$,

$$\Upsilon_\lambda = \frac{1}{\sqrt{|\mathcal{M}_\lambda|}} \sum_{\mu \in \mathcal{M}_\lambda} \Upsilon_{\lambda,\mu}.$$

To compare the knowledge definitions for the same strict distribution, let

$$L_\mu^+ = \{S \subseteq [N] : |\{B \in \mu : |B \cap S| \geq r\}| \geq \ell\}.$$

In the source Fourier basis, define

$$\Pi_\mu^+ = \sum_{\substack{\sigma \in \mathbb{Z}_q^N \\ \mathrm{supp}(\sigma) \in L_\mu^+}} |\widehat{\sigma}\rangle\langle\widehat{\sigma}|, \qquad \Pi_\mu^- = I - \Pi_\mu^+.$$

The corresponding Belovs-style source decomposition is

$$\Upsilon_\lambda^\pm = \frac{1}{\sqrt{|\mathcal{M}_\lambda|}} \sum_{\mu \in \mathcal{M}_\lambda} \Upsilon_{\lambda,\mu}\Pi_\mu^\pm, \qquad \Upsilon_\lambda = \Upsilon_\lambda^+ + \Upsilon_\lambda^-. \tag{2.13}$$

Here the source Fourier support is classified before applying the transfer map.

Our database decomposition instead classifies the recorded pairs after transfer and embedding:

$$J\Upsilon_\lambda = P_{r,\ell}J\Upsilon_\lambda + (I - P_{r,\ell})J\Upsilon_\lambda.$$

Because $P_{r,\ell}$ is an orthogonal projector, these two components have orthogonal images. In particular,

$$\|J\Upsilon_\lambda v\|^2 = \|P_{r,\ell}J\Upsilon_\lambda v\|^2 + \|(I - P_{r,\ell})J\Upsilon_\lambda v\|^2.$$

By contrast, the images of Υ_λ^+ and Υ_λ^- need not be orthogonal. The isometry J preserves their inner products:

$$\langle J\Upsilon_\lambda^+ v, J\Upsilon_\lambda^- v\rangle = \langle \Upsilon_\lambda^+ v, \Upsilon_\lambda^- v\rangle.$$

Consequently, the database high-knowledge component cannot in general be identified with $J\Upsilon_\lambda^+$.

Remark 2.10 (The two components need not coincide). Take $N = 4$, $q = 2$, $\lambda = (2^2)$, $r = 2$, and $\ell = 1$. Let $\mathbf{0} = (0,0,0,0)$ and $\mathbf{1} = (1,1,1,1)$. Every exact-type function has exactly two values equal to 1, so

$$\Upsilon_\lambda|\widehat{\mathbf{1}}\rangle = \Upsilon_\lambda|\widehat{\mathbf{0}}\rangle = |u_{\lambda,q}\rangle.$$

The full support of $\mathbf{1}$ is high knowledge for every $\mu \in \mathcal{M}_\lambda$, whereas the empty support of $\mathbf{0}$ is low knowledge for every such μ. Using (2.11), we obtain

$$P_{2,1}J\Upsilon_\lambda|\widehat{\mathbf{1}}\rangle = 0, \qquad J\Upsilon_\lambda^+|\widehat{\mathbf{1}}\rangle = |\varnothing\rangle|u_{\mathrm{lab}}\rangle \neq 0.$$

Thus $P_{2,1}J\Upsilon_\lambda \neq J\Upsilon_\lambda^+$. Moreover, for $v = (|\widehat{\mathbf{0}}\rangle + |\widehat{\mathbf{1}}\rangle)/\sqrt{2}$, the inner product of the two transferred knowledge components is $1/2$.

Oracle access and the square-root obstruction. The lifting $K_{\lambda,q}$ only changes the representation of the original function-query algorithm. The stronger oracle model enters when we bound database growth uniformly over arbitrary forward permutation-query algorithms, rather than only over circuits composed of the two-swap function-query blocks.

This distinction matters for marked targets. For a fixed public bucket of constant size at least r, forward permutation access allows quantum search for r distinct preimages using $O(\sqrt{N})$ queries. A lower bound applying to every algorithm with this access cannot exceed that scale for this task. The additive CPO simulation error imposes a separate restriction on the range in which database bounds transfer to useful bounds for the original algorithm.

Belovs's function-query framework grants no separate oracle access to the hidden partition or to a representing permutation. It therefore does not inherit this particular public-bucket search obstruction. Retaining that restriction may permit stronger bounds, but neither the isometry J nor the database estimates above establish such an improvement.

3 Tight Multi-Collision Lower Bounds for (k^n)-Type Functions

In this section, we establish quantum query lower bounds for multi-collision finding in random functions of exact type (k^n). These bounds match the upper bounds obtained by extending the construction of Liu and Zhandry [26] to multiple targets, and are therefore tight. Our analysis also demonstrates that CPO retains a useful feature of Zhandry's compressed oracle: it supports recursive lower-bound arguments for complex search problems by tracking the growth of database witnesses through successive levels.

We first introduce some notation. Fix $2 \le r \le k$, where k, r are constants. Partition $[N]$ into public buckets $Q_1, \dots, Q_h$, and write $d_b = |Q_b| \le k$. The pure type (k^n) has $h = n$, all bucket sizes $d_b = k$, and $N = kn$. For the corresponding real function problem, the alphabet Σ has size $q \ge h$. All real-query bounds below are uniform over this full feasible range; no additional large-alphabet assumption is made. Write $o_b(D) = |\operatorname{im}(D) \cap Q_b|$ for the occupancy of bucket Q_b. We extend the database notation to every level $1 \le j \le k$:

$$C_j(D) = |\{b : o_b(D) \ge j\}|, \quad P_{j,m} := \sum_{D : C_j(D) \ge m} |D\rangle\langle D| = \mathbf{1}_{\{C_j \ge m\}}, \quad a_{j,m}(s) = \|P_{j,m}\zeta_s\|\,.$$

Here $m \ge 0$, so $P_{j,0} = I$. When the database is clear from the context, we write C_j for $C_j(D)$, as in the event $C_j \ge m$. All counters in this section range over the full set of buckets.

Define

$$\alpha_j = 2 - 2^{1-j}, \qquad \beta_j = 1 - 2^{1-j} = \alpha_j - 1. \tag{3.1}$$

For an integer query horizon $1 \le T < N$, put

$$d = N - T, \qquad \lambda_j = \frac{T^{\alpha_j}}{d^{\beta_j}}, \qquad c_j = \sqrt{8(k - j + 1)} \qquad (2 \le j \le k). \tag{3.2}$$

Then

$$\lambda_1 = T, \qquad \lambda_j = T\sqrt{\frac{\lambda_{j-1}}{d}}.$$

corresponding to the exponent recurrences

$$\alpha_j = 1 + \frac{\alpha_{j-1}}{2}, \qquad \beta_j = \frac{1 + \beta_{j-1}}{2}.$$

When $T \le N/2$, replacing d by N changes only a constant factor. For type (k^n) and fixed k, the same holds when replacing $N = kn$ by n.

For $1 \le \ell \le n$, define the query scale

$$\tau_{r,\ell} := \ell^{1/\alpha_r} N^{\beta_r/\alpha_r} = \ell^{1/(2-2^{1-r})} N^{(1-2^{1-r})/(2-2^{1-r})}. \tag{3.3}$$

The main result of this section is the following theorem.

Theorem 3.1 (Multi-collision lower bounds for type (k^n))**.** *Fix integers $2 \le r \le k$, and let $N = kn$. The following statements hold.*

1. *For every fixed $\theta \in (0,1)$, there is a constant $c_{k,r,\theta} > 0$ such that, for all sufficiently large N, every forward compressed-query algorithm satisfies*

$$a_{r,\ell}(T) \le \theta^{\ell} \qquad \text{whenever} \qquad 0 \le T \le c_{k,r,\theta}\tau_{r,\ell},$$

   uniformly over integers $1 \le \ell \le n$.

2. *For every constant $p_0 \in (0,1)$ and all sufficiently large N, any quantum algorithm that finds ℓ r-collisions in distinct equality classes with probability at least p_0 for a uniformly random function of exact type (k^n) requires*

$$t = \Omega_{k,r,p_0}(\tau_{r,\ell}) = \Omega_{k,r,p_0}\Big(\ell^{1/(2-2^{1-r})} N^{(1-2^{1-r})/(2-2^{1-r})}\Big)$$

   function queries, provided $1 \le \ell \le N^{2^{-r}}$. This bound holds uniformly over all $q \ge n$.

We prove the database bound in Theorem 3.8. Combining it with the success-to-database reduction gives the function-query lower bound, as shown in Corollary 3.10.

3.1 Local operator bounds and the good/bad source recurrence

In this subsection, we first establish local bounds for a single compressed query and derive a recurrence for $a_{r,\ell}(s)$.

Orthogonal databases and a single query. Fix an input x. For each partial injection H with $x \notin \mathrm{dom}(H)$, define

$$Y_H = [N] \setminus \mathrm{im}(H), \qquad m_H = N - |H|, \qquad H_y = H[x \mapsto y] \quad (y \in Y_H),$$

and let

$$\mathcal{F}_{x,H} = \mathrm{span}\{|H\rangle, |H_y\rangle : y \in Y_H\}.$$

Deleting the pair at input x from any database gives a unique H. Hence these subspaces are pairwise orthogonal and their direct sum is $\mathcal{L}$.

On $\mathcal{F}_{x,H}$, the operator C_x swaps

$$|H\rangle \quad \text{and} \quad |+_{x,H}\rangle = \frac{1}{\sqrt{m_H}} \sum_{y \in Y_H} |H_y\rangle,$$

and fixes their orthogonal complement. Consequently,

$$\begin{aligned}\mathsf{C}_x|H\rangle &= \frac{1}{\sqrt{m_H}}\sum_{y\in Y_H}|H_y\rangle,\\ \mathsf{C}_x|H_y\rangle &= |H_y\rangle - \frac{1}{m_H}\sum_{z\in Y_H}|H_z\rangle + \frac{1}{\sqrt{m_H}}|H\rangle \qquad (y\in Y_H).\end{aligned} \tag{3.4}$$

In particular, C_x is a self-adjoint unitary, with $\mathsf{C}_x^2 = I$. Recall that S_y swaps the answer states $|\perp\rangle$ and $|y\rangle$ and fixes all other answer states. Restricted to $\mathcal{W}\otimes\mathcal{F}_{x,H}$, the operators defining the compressed query are

$$\begin{aligned}\mathsf{U}_x &= I_{\mathcal{W}}\otimes|H\rangle\langle H| + \sum_{y\in Y_H} S_y\otimes|H_y\rangle\langle H_y|,\\ \mathsf{Q}_x &= (I_{\mathcal{W}}\otimes\mathsf{C}_x)\mathsf{U}_x(I_{\mathcal{W}}\otimes\mathsf{C}_x).\end{aligned}$$

We write

$$\mathsf{cpO} = \sum_x |x\rangle\langle x|\otimes\mathsf{Q}_x$$

for the full controlled query. Identities on other registers are implicit.

Each local query preserves $\mathcal{F}_{x,H}$ and therefore changes the database size by at most one. Since the database starts in $|\varnothing\rangle$, the state after s queries is supported on $|D|\le s$, and the algorithmic unitaries act trivially on the database register and therefore commute with all database counting operators and their spectral projections.

Now we establish the following local boundary-crossing and commutator bounds, which will be used to derive the query recurrences.

Lemma 3.2 (Local boundary crossing and commutator bounds)**.** *Fix a subspace $\mathcal{F}_{x,H}$. Let $E\subseteq Y_H$, put $b=|E|$, and define*

$$B = \sum_{y\in E}|H_y\rangle\langle H_y|.$$

On this subspace, allowing arbitrary auxiliary registers,

$$\|B\mathsf{Q}_x(I-B)\| \le \|[B,\mathsf{Q}_x]\| \le 2\sqrt{\frac{2b}{m_H}-\frac{b^2}{m_H^2}} \le 2\sqrt{\frac{2b}{m_H}}. \tag{3.5}$$

Proof. Consider $B\mathsf{C}_x(I-B)$ on the database register. Use the basis $\{|H_y\rangle : y\in E\}$ for its range and

$$\{|H\rangle\}\cup\{|H_z\rangle : z\in Y_H\setminus E\}$$

for its domain. By (3.4), this matrix has b identical rows, each equal to

$$(m_H^{-1/2}, -m_H^{-1}, \dots, -m_H^{-1}),$$

where $-m_H^{-1}$ occurs m_H-b times. Its rank is at most one, and hence

$$\|B\mathsf{C}_x(I-B)\|^2 = b\left(\frac{1}{m_H}+\frac{m_H-b}{m_H^2}\right) = \frac{2b}{m_H}-\frac{b^2}{m_H^2}.$$

Since C_x is self-adjoint, its block decomposition with respect to the ranges of B and $I-B$ gives

$$\|[B,\mathsf{C}_x]\| = \|B\mathsf{C}_x(I-B)\|.$$

Moreover, $[B, \mathsf{U}_x] = 0$. Suppressing identity operators on the answer register, we obtain

$$[B, \mathsf{Q}_x] = [B, \mathsf{C}_x]\mathsf{U}_x\mathsf{C}_x + \mathsf{C}_x\mathsf{U}_x[B, \mathsf{C}_x].$$

Unitarity and the triangle inequality therefore imply

$$\|[B, \mathsf{Q}_x]\| \le 2\,\|[B, \mathsf{C}_x]\| = 2\sqrt{\frac{2b}{m_H} - \frac{b^2}{m_H^2}}.$$

The first inequality in (3.5) follows from

$$B[B, \mathsf{Q}_x](I - B) = B\mathsf{Q}_x(I - B).$$

Tensoring the operators with identities on auxiliary registers does not change their norms, so the bounds allow arbitrary auxiliary entanglement. □

Exact boundary counts. Fix an input x and a partial injection H with $x \notin \mathrm{dom}(H)$. Recall that the subspace $\mathcal{F}_{x,H}$ is spanned by $|H\rangle$ and the states $|H_y\rangle$ for $y \in Y_H$, where $H_y = H[x \mapsto y]$. Thus every database D labeling a basis state in $\mathcal{F}_{x,H}$ satisfies

$$C_r(D) \in \{C_r(H),\, C_r(H) + 1\}.$$

Since Q_x preserves $\mathcal{F}_{x,H}$, a transition from $C_r < \ell$ to $C_r \ge \ell$ is possible only if

$$C_r(H) = \ell - 1.$$

Indeed, if $C_r(H) \le \ell - 2$, every database in the subspace has count below ℓ; if $C_r(H) \ge \ell$, every database in the subspace already has count at least ℓ.

The outputs that increase C_r are precisely the unused outputs in buckets satisfying $o_b(H) = r-1$. Writing $d_b = |Q_b|$, their number is

$$b_r(H) = \sum_{b:o_b(H)=r-1} (d_b - r + 1) \le (k - r + 1)\big(C_{r-1}(H) - C_r(H)\big). \tag{3.6}$$

For type (k^n), the inequality is an equality. The same estimate also applies when the bucket sizes are unequal but bounded by k. We next bound the growth of the target database component under a single query and derive the corresponding recurrence.

Theorem 3.3 (One-step transition bound and query recurrence)**.** *Let $2 \le r \le k$, let $\ell \ge 1$ and $L \ge \ell - 1$ be integers, and let $0 \le s < N$ be an integer. Define*

$$G = \mathbf{1}_{\{C_r=\ell-1,\ C_{r-1}\le L\}}, \qquad R = \mathbf{1}_{\{C_r=\ell-1,\ C_{r-1}\ge L+1\}}, \qquad \gamma = c_r\sqrt{\frac{L-\ell+1}{N-s}}.$$

For every vector v supported on databases of size at most s,

$$\begin{aligned}\|P_{r,\ell}U_{s+1}\mathsf{cpO}v\| &\le \|P_{r,\ell}v\| + \gamma\,\|Gv\| + \|Rv\| \\ &\le \|P_{r,\ell}v\| + \gamma\,\|P_{r,\ell-1}v\| + \|P_{r-1,L+1}v\|\,.\end{aligned} \tag{3.7}$$

The vector v need not be normalized.

In particular, since $\zeta_{s+1} = U_{s+1}\mathsf{cpO}\zeta_s$,

$$a_{r,\ell}(s+1) \le a_{r,\ell}(s) + \gamma a_{r,\ell-1}(s) + a_{r-1,L+1}(s). \tag{3.8}$$

Proof. Since U_{s+1} acts outside the database register, it commutes with $P_{r,\ell}$, and hence

$$\|P_{r,\ell}U_{s+1}\mathsf{cpO}v\| = \|U_{s+1}P_{r,\ell}\mathsf{cpO}v\| = \|P_{r,\ell}\mathsf{cpO}v\|.$$

It therefore suffices to analyze the compressed query.

A query changes C_r by at most one, so

$$P_{r,\ell}\mathsf{cpO}\mathbf{1}_{\{C_r\le\ell-2\}} = 0.$$

Consider a subspace $\mathcal{F}_{x,H}$ containing databases on both sides of the threshold $C_r = \ell$. Then $C_r(H) = \ell - 1$. If this subspace contains a database D in the support of G with $|D| \le s$, then $H \subseteq D$ gives

$$C_{r-1}(H) \le L, \qquad m_H = N - |H| \ge N - s.$$

By (3.6),

$$b_r(H) \le (k-r+1)\big(C_{r-1}(H) - C_r(H)\big) \le (k-r+1)(L-\ell+1).$$

Lemma 3.2 therefore bounds the boundary-crossing norm on this subspace by

$$2\sqrt{\frac{2b_r(H)}{m_H}} \le c_r\sqrt{\frac{L-\ell+1}{N-s}} = \gamma.$$

Subspaces containing databases on only one side of the threshold contribute zero. Taking the orthogonal direct sum over H and then over inputs x yields

$$\left\|P_{r,\ell}\mathsf{cpO}G\mathbf{1}_{\{|D|\le s\}}\right\| \le \gamma.$$

Using the decomposition of the identity into mutually orthogonal projections,

$$I = P_{r,\ell} + G + R + \mathbf{1}_{\{C_r\le\ell-2\}},$$

we obtain, for every v supported on $|D| \le s$,

$$\begin{aligned}\|P_{r,\ell}\mathsf{cpO}v\| &\le \|P_{r,\ell}\mathsf{cpO}P_{r,\ell}v\| + \|P_{r,\ell}\mathsf{cpO}Gv\| + \|P_{r,\ell}\mathsf{cpO}Rv\| \\ &\le \|P_{r,\ell}v\| + \gamma\|Gv\| + \|Rv\|.\end{aligned}$$

The first and third terms use unitarity of cpO and contractivity of the orthogonal projection $P_{r,\ell}$. Moreover, the projections G and R satisfy

$$G \preceq P_{r,\ell-1}, \qquad R \preceq P_{r-1,L+1}. \tag{3.9}$$

The corresponding norm inequalities give (3.7). Finally, substituting $v = \zeta_s$ gives (3.8). □

3.2 First-moment recurrence: single-target bounds without truncation

In this subsection, we use a first-moment recurrence to obtain single-collision bounds of every order without truncation. For each $j \in [k]$, define the diagonal counting operator

$$\widehat{C}_j := \sum_D C_j(D)\,|D\rangle\langle D| = \sum_{i=1}^{h} P_{j,i},$$

where $P_{j,i}$ projects onto databases satisfying $C_j(D) \geq i$. We write

$$m_j(s) := \langle \zeta_s, \widehat{C}_j \zeta_s \rangle = \sum_{i=1}^{h} a_{j,i}(s)^2.$$

Since the database has size at most s, we have $m_1(s) \leq s$. We bound these expectations directly in the compressed state. Formally, we have the following lemma.

Lemma 3.4 (Weighted increments for first moments and single-target norms)**.** *For every $2 \leq j \leq k$ and $s < N$,*

$$|m_j(s+1) - m_j(s)| \leq c_j \sqrt{\frac{m_{j-1}(s)}{N-s}}, \tag{3.10}$$

$$a_{j,1}(s+1) - a_{j,1}(s) \leq c_j \sqrt{\frac{m_{j-1}(s)}{N-s}}. \tag{3.11}$$

Proof. Let v be the normalized state immediately before the $(s+1)$st compressed query. The intervening algorithmic operations act outside the database register, so

$$\langle v, \widehat{C}_j v \rangle = m_j(s), \qquad \|P_{j,i} v\| = a_{j,i}(s).$$

Decompose v orthogonally according to the query input x and the database H obtained by deleting its record:

$$v = \sum_{x,H} v_{x,H}, \qquad w_{x,H} := \|v_{x,H}\|^2, \qquad \sum_{x,H} w_{x,H} = 1.$$

Every occupied subspace satisfies $|H| \leq s$.

On the subspace $\mathcal{F}_{x,H}$, the counting operator satisfies

$$\widehat{C}_j\Big|_{\mathcal{F}_{x,H}} = C_j(H) I + B_{j,H}, \tag{3.12}$$

where $B_{j,H}$ projects onto extensions of H that increase the level-j count by one. Here and below, identities on auxiliary registers are implicit. Because Q_x preserves this subspace, the constant term cancels under conjugation. Thus, with all operators restricted to the subspace,

$$\mathsf{Q}_x^\dagger \widehat{C}_j \mathsf{Q}_x - \widehat{C}_j = \mathsf{Q}_x^\dagger [B_{j,H}, \mathsf{Q}_x].$$

Equation (3.6) gives

$$b_j(H) \leq (k-j+1) C_{j-1}(H).$$

Consequently, Lemma 3.2 and $N - |H| \geq N - s$ imply

$$\left\| \mathsf{Q}_x^\dagger \widehat{C}_j \mathsf{Q}_x - \widehat{C}_j \right\| \leq \frac{c_j}{\sqrt{N-s}} \sqrt{C_{j-1}(H)}.$$

The full controlled query cpO preserves the orthogonal decomposition indexed by (x, H), we therefore obtain

$$\begin{aligned} |m_j(s+1) - m_j(s)| &\leq \frac{c_j}{\sqrt{N-s}} \sum_{x,H} w_{x,H} \sqrt{C_{j-1}(H)} \\ &\leq \frac{c_j}{\sqrt{N-s}} \left(\sum_{x,H} w_{x,H} C_{j-1}(H) \right)^{1/2} \\ &\leq c_j \sqrt{\frac{m_{j-1}(s)}{N-s}}. \end{aligned}$$

The second inequality is Cauchy–Schwarz. The last inequality follows from (3.12), applied at level $j-1$, and the positive semidefiniteness of $B_{j-1,H}$. It follows that

$$\sum_{x,H} w_{x,H} C_{j-1}(H) \leq \langle v, \widehat{C}_{j-1} v \rangle = m_{j-1}(s).$$

This proves (3.10).

For the single-target bound, let $P = P_{j,1}$. If $C_j(H) \geq 1$, every database in the subspace belongs to the range of P, so $(I-P)v_{x,H} = 0$. If $C_j(H) = 0$, the restriction of P to the subspace is $B_{j,H}$. The boundary-crossing estimate from Lemma 3.2 therefore gives

$$\|P \mathsf{Q}_x (I-P) v_{x,H}\|^2 \leq \frac{c_j^2 C_{j-1}(H)}{N-s} \|(I-P)v_{x,H}\|^2.$$

The resulting vectors lie in mutually orthogonal subspaces. Summing their squared norms yields

$$\begin{aligned} \|P \mathsf{cpO} (I-P) v\|^2 &\leq \frac{c_j^2}{N-s} \sum_{x,H} C_{j-1}(H) \|(I-P)v_{x,H}\|^2 \\ &\leq \frac{c_j^2}{N-s} \sum_{x,H} C_{j-1}(H) w_{x,H} \\ &\leq \frac{c_j^2 m_{j-1}(s)}{N-s}. \end{aligned} \tag{3.13}$$

Since cpO is unitary and P is an orthogonal projection,

$$\|P \mathsf{cpO} P v\| \leq \|Pv\| = a_{j,1}(s).$$

The triangle inequality now gives

$$\begin{aligned} a_{j,1}(s+1) &= \|P \mathsf{cpO} v\| \\ &\leq \|P \mathsf{cpO} P v\| + \|P \mathsf{cpO} (I-P) v\| \end{aligned}$$

$$\leq a_{j,1}(s) + c_j \sqrt{\frac{m_{j-1}(s)}{N-s}},$$

proving (3.11). □

Combining the preceding lemma with the first-moment recurrence yields an upper bound on $a_{r,1}(T)$. Formally, we obtain the following theorem.

Theorem 3.5 (Single-target bounds of every order without truncation)**.** *For $1 \leq j \leq k$, define*

$$A_j = \prod_{i=2}^{j} (8(k-i+1))^{2^{i-j-1}} \leq (8k)^{1-2^{1-j}},$$

where an empty product equals 1. Then $A_1 = 1$ and $A_j = c_j \sqrt{A_{j-1}}$ for $j \geq 2$. For every integer $1 \leq T < N$ and every $1 \leq j \leq k$, with λ_j as in (3.2),

$$\sup_{0 \leq s \leq T} m_j(s) \leq A_j \lambda_j, \qquad a_{j,1}(T) \leq \min\{1, A_j \lambda_j\}. \tag{3.14}$$

In particular, for fixed k, r and $T \leq N/2$,

$$a_{r,1}(T) \leq \frac{(16k)^{1-2^{1-r}} T^{2-2^{1-r}}}{N^{1-2^{1-r}}} = O_{k,r}\left(\frac{T^{2-2^{1-r}}}{N^{1-2^{1-r}}} \right).$$

Proof. Put $d = N - T$. Recall that the scales in (3.2) satisfy

$$\lambda_1 = T, \qquad \lambda_j = \frac{T}{\sqrt{d}} \sqrt{\lambda_{j-1}}.$$

The first-moment bound for $j = 1$ follows from $m_1(s) \leq s \leq T = A_1 \lambda_1$. The corresponding single-target bound follows from $a_{1,1}(T) \leq 1$ and $T \geq 1$.

Suppose that $m_{j-1}(s) \leq A_{j-1} \lambda_{j-1}$ for every $s \leq T$. The database is initially empty, so $m_j(0) = 0$. Summing (3.10) and using $N - s \geq d$ gives, for every $0 \leq u \leq T$,

$$m_j(u) \leq \sum_{s=0}^{u-1} c_j \sqrt{\frac{m_{j-1}(s)}{N-s}} \leq \frac{T c_j}{\sqrt{d}} \sqrt{A_{j-1} \lambda_{j-1}} = A_j \lambda_j.$$

This proves the first-moment bound by induction on j.

Similarly, $a_{j,1}(0) = 0$, and summing (3.11) gives

$$a_{j,1}(T) \leq \frac{T c_j}{\sqrt{d}} \sqrt{A_{j-1} \lambda_{j-1}} = A_j \lambda_j.$$

The additional upper bound 1 follows from normalization. Finally, $T \leq N/2$ implies $d \geq N/2$, and the stated asymptotic bound follows from (3.2). □

Remark 3.6 (First moments and multi-collision tails). For every database D and every $\ell \geq 1$,

$$\ell \, \mathbf{1}_{\{C_r(D) \geq \ell\}} \leq C_r(D).$$

Since $P_{r,\ell}$ and $\widehat{C}_r$ are diagonal in the database basis, this pointwise inequality gives

$$\ell P_{r,\ell} \preceq \widehat{C}_r.$$

As $P_{r,\ell}$ is an orthogonal projection, we obtain

$$\begin{aligned} a_{r,\ell}(T)^2 = \|P_{r,\ell}\zeta_T\|^2 &= \langle \zeta_T, P_{r,\ell}\zeta_T\rangle \\ &\le \frac{1}{\ell}\langle \zeta_T, \widehat{C}_r\zeta_T\rangle = \frac{m_r(T)}{\ell} \le \frac{A_r\lambda_r}{\ell}, \end{aligned}$$

where the last inequality follows from (3.14). Thus, for any constant $p_0 \in (0,1)$, achieving database success probability at least p_0 requires $\Omega_{k,r,p_0}(\tau_{r,\ell})$ queries. Combining this estimate with the success-to-database reduction already proves the second assertion of Theorem 3.1, for $1 \le \ell \le N^{2^{-r}}$.

It remains to establish exponential decay in ℓ, as required by the first assertion of Theorem 3.1. At $T = c\tau_{r,\ell}$, for a sufficiently small constant $c > 0$, the first-moment estimate gives only $a_{r,\ell}(T)^2 = O_{k,r}(c^{\alpha_r})$. This bound does not decay with ℓ and therefore does not establish $a_{r,\ell}(T) \le \theta^\ell$.

For $\ell > 1$, extending the single-target argument requires controlling a joint quantity involving two counting levels. Writing

$$|\zeta_s\rangle = \sum_D |\zeta_{s,D}\rangle \otimes |D\rangle,$$

where $|\zeta_{s,D}\rangle$ are unnormalized vectors in the algorithm registers, the relevant boundary-crossing quantity is

$$\left(\sum_{\substack{D: \\ C_r(D)=\ell-1}} \left(C_{r-1}(D) - \ell + 1\right) \|\zeta_{s,D}\|^2 \right)^{1/2}.$$

Bounding this quantity by $\sqrt{m_{r-1}(s)}\, a_{r,\ell-1}(s)$ would require an additional correlation estimate, which does not follow from the first-moment bound alone. The next subsection uses sequential truncation to control this joint quantity and obtain exponential tail bounds.

3.3 Sequential truncation and multi-collision tail bounds

The first-moment estimate already yields the query threshold for constant database success probability. We now strengthen this conclusion to exponential decay in ℓ. Throughout this subsection, we retain the notation $a_{r,\ell}(s)$, c_r, and $d = N - T$ introduced above. The estimates apply to arbitrary joint algorithm–database states arising from the allowed compressed algorithms. They also apply to unequal bucket sizes, provided that every bucket has size at most k. As before, ℓ counts distinct buckets.

We first apply Theorem 3.3 to a contractive truncated trajectory, and then choose the truncation thresholds recursively across collision orders. Formally, we have the following lemma.

Lemma 3.7 (A contractive truncated trajectory)**.** *Let $2 \le r \le k$, let $1 \le T < N$ be an integer, and let $L \ge 0$ be an integer. For every integer $\ell \ge 1$,*

$$a_{r,\ell}(T) \le F_{r,\ell}(T, L) + \sum_{s=1}^{T} a_{r-1,L+1}(s), \tag{3.15}$$

where

$$F_{r,\ell}(T,L) = \begin{cases} \binom{T}{\ell}\left(\frac{c_r}{\sqrt{d}}\right)^{\ell}\sqrt{\frac{L!}{(L-\ell)!}}, & 1 \le \ell \le L, \\ 0, & \ell > L. \end{cases} \tag{3.16}$$

We use the convention $\binom{T}{\ell} = 0$ *for* $\ell > T$*. In particular, for* $L \ge 1$*,*

$$F_{r,\ell}(T,L) \le \left(\frac{ec_rT}{\ell}\sqrt{\frac{L}{d}}\right)^{\ell}. \tag{3.17}$$

Proof. Recall that

$$P^{\perp}_{r-1,L+1} = I - P_{r-1,L+1} = \mathbf{1}_{\{C_{r-1}\le L\}}.$$

Define the auxiliary trajectory by

$$\xi_0 = \zeta_0, \qquad \xi_{s+1} = P^{\perp}_{r-1,L+1}U_{s+1}\mathsf{cpO}\xi_s \qquad (0 \le s < T).$$

Each operator $P^{\perp}_{r-1,L+1}U_{s+1}\mathsf{cpO}$ is contractive, so $\|\xi_s\| \le 1$. The initial database is empty and satisfies the cutoff. Thus $P^{\perp}_{r-1,L+1}\xi_s = \xi_s$ for $0 \le s \le T$. Moreover, ξ_s is supported on $|D| \le s$: each query increases database size by at most one, and the cutoff projection only removes database components.

Since $\zeta_{s+1} = U_{s+1}\mathsf{cpO}\zeta_s$,

$$\zeta_{s+1} - \xi_{s+1} = P^{\perp}_{r-1,L+1}U_{s+1}\mathsf{cpO}(\zeta_s - \xi_s) + P_{r-1,L+1}\zeta_{s+1}.$$

Contractivity and $\xi_0 = \zeta_0$ therefore give

$$\|\zeta_T - \xi_T\| \le \sum_{s=1}^{T}\|P_{r-1,L+1}\zeta_s\| = \sum_{s=1}^{T} a_{r-1,L+1}(s). \tag{3.18}$$

For every integer $\ell \ge 0$, write

$$b_{r,\ell}(s) = \|P_{r,\ell}\xi_s\|.$$

Since $C_r \le C_{r-1}$ and ξ_s satisfies the cutoff, we have $b_{r,\ell}(s) = 0$ whenever $\ell > L$. For these values of ℓ, the claimed bound follows from (3.18). This also covers all $\ell \ge 1$ when $L = 0$. It remains to consider $L \ge 1$ and $1 \le \ell \le L$.

The cutoff projection commutes with $P_{r,\ell}$, and $P_{r-1,L+1}\xi_s = 0$. Applying (3.7) to the possibly unnormalized vector ξ_s therefore gives

$$\begin{aligned} b_{r,\ell}(s+1) &= \left\|P_{r,\ell}P^{\perp}_{r-1,L+1}U_{s+1}\mathsf{cpO}\xi_s\right\| \\ &\le \|P_{r,\ell}U_{s+1}\mathsf{cpO}\xi_s\| \\ &\le b_{r,\ell}(s) + c_r\sqrt{\frac{L-\ell+1}{N-s}}\,b_{r,\ell-1}(s). \end{aligned}$$

Using $N - s \ge d$, we obtain

$$b_{r,\ell}(s+1) \le b_{r,\ell}(s) + \frac{c_r}{\sqrt{d}}\sqrt{L-\ell+1}\,b_{r,\ell-1}(s). \tag{3.19}$$

To solve this recurrence, define

$$w_{r,\ell} = \left(\frac{c_r}{\sqrt{d}}\right)^{\ell} \sqrt{\frac{L!}{(L-\ell)!}}, \qquad u_{r,\ell}(s) = \frac{b_{r,\ell}(s)}{w_{r,\ell}} \qquad (0 \le \ell \le L).$$

These weights are positive, with $w_{r,0} = 1$, and satisfy

$$w_{r,\ell} = \frac{c_r}{\sqrt{d}} \sqrt{L-\ell+1}\, w_{r,\ell-1} \qquad (1 \le \ell \le L).$$

Dividing (3.19) by $w_{r,\ell}$ gives

$$u_{r,\ell}(s+1) \le u_{r,\ell}(s) + u_{r,\ell-1}(s) \qquad (1 \le \ell \le L).$$

Contractivity gives $u_{r,0}(s) = b_{r,0}(s) = \|\xi_s\| \le 1$. Since the initial database is empty, $u_{r,\ell}(0) = 0$ for $1 \le \ell \le L$.

We prove by induction on s that

$$u_{r,\ell}(s) \le \binom{s}{\ell} \qquad (0 \le \ell \le L).$$

The claim holds at $s = 0$, and the case $\ell = 0$ follows from contractivity at every step. For $1 \le \ell \le L$, the induction hypothesis and Pascal's identity give

$$u_{r,\ell}(s+1) \le \binom{s}{\ell} + \binom{s}{\ell-1} = \binom{s+1}{\ell}.$$

Multiplying by $w_{r,\ell}$ yields

$$b_{r,\ell}(s) \le \binom{s}{\ell} \left(\frac{c_r}{\sqrt{d}}\right)^{\ell} \sqrt{\frac{L!}{(L-\ell)!}}.$$

In particular, $b_{r,\ell}(T) \le F_{r,\ell}(T, L)$. Combining this with (3.18) gives

$$\begin{aligned} a_{r,\ell}(T) &\le \|P_{r,\ell}\xi_T\| + \|P_{r,\ell}(\zeta_T - \xi_T)\| \\ &\le F_{r,\ell}(T, L) + \sum_{s=1}^{T} a_{r-1,L+1}(s), \end{aligned}$$

proving (3.15).

Finally, for $1 \le \ell \le L$, use

$$\binom{T}{\ell} \le \frac{T^{\ell}}{\ell!}, \qquad \frac{L!}{(L-\ell)!} \le L^{\ell}, \qquad \ell! \ge (\ell/e)^{\ell}$$

to obtain (3.17). For $\ell > L$, the same bound holds because $F_{r,\ell}(T, L) = 0$. □

The cumulative contribution of the bad terms is bounded by $\sum_{s=1}^{T} a_{r-1,L+1}(s)$. If each summand is at most ε, the total error is at most $T\varepsilon$. We next choose the threshold L to balance the good term $F_{r,\ell}(T, L)$ against the bad term $\sum_{s=1}^{T} a_{r-1,L+1}(s)$, thereby obtaining a sharper upper bound on the database amplitude.

Recursive truncation thresholds. First we consider the case that $r = 2$. The two-collision layer provides a base case with no bad term. For $1 \le T < N$, the database size bound gives $a_{1,T+1}(s) = 0$ for all $0 \le s \le T$. Taking $L = T$ in Lemma 3.7, and using (3.17), yields

$$a_{2,\ell}(T) \le \min\left\{1, \left(\frac{ec_2T^{3/2}}{\ell\sqrt{N-T}}\right)^{\ell}\right\}. \tag{3.20}$$

We next extend this estimate to every fixed collision order, allowing ℓ to grow with N.

Theorem 3.8 (An exponential tail at the database query threshold)**.** *Fix $2 \le r \le k$ and $\theta \in (0,1)$, and let $N = kn$. There is a constant $c_{k,r,\theta} > 0$ such that, for sufficiently large N, we have*

$$a_{r,\ell}(T) \le \theta^{\ell}$$

for every integer $1 \le \ell \le n$ and every nonnegative integer T satisfying

$$T \le c_{k,r,\theta}\, \ell^{1/\alpha_r} N^{\beta_r/\alpha_r}. \tag{3.21}$$

Proof. The claim is immediate for $T = 0$, so assume $T \ge 1$. For convenience, we write $c = c_{k,r,\theta}$, to be chosen sufficiently small. We require $c \le 1/2$. Since $\ell \le n \le N$ and $\alpha_r = 1 + \beta_r$, the assumed query bound gives $T \le N/2$, and hence $d = N - T \ge N/2$.

For $r = 2$, (3.20) gives

$$a_{2,\ell}(T) \le \left(4e\sqrt{k-1}\, c^{3/2}\right)^{\ell}.$$

Choosing c sufficiently small proves the claim. Assume henceforth that $r \ge 3$. Fix the truncation thresholds by

$$L_1 = T, \qquad L_j = \left\lceil e^2 c_j T\sqrt{\frac{L_{j-1}}{d}} + (r+2)\log(N+1) + \ell\log\frac{2}{\theta}\right\rceil, \quad 2 \le j \le r-1. \tag{3.22}$$

We show that, with these thresholds, the good term and the accumulated bad terms in Lemma 3.7 are each at most $(\theta/2)^{\ell}$.

The bad terms. For $1 \le j \le r-1$, write

$$\varepsilon_j = \max_{0 \le s \le T} a_{j,L_j+1}(s).$$

The database size bound gives $\varepsilon_1 = 0$. For $2 \le j \le r-1$ and $1 \le s \le T$, apply Lemma 3.7 with horizon s, target count $L_j + 1$, and cutoff L_{j-1}. Using (3.17), $s \le T$, and $N - s \ge d$, we obtain

$$\begin{aligned} F_{j,L_j+1}(s, L_{j-1}) &\le \left(\frac{ec_jT\sqrt{L_{j-1}/d}}{L_j+1}\right)^{L_j+1} \\ &\le e^{-(L_j+1)} \le \frac{(\theta/2)^{\ell}}{(N+1)^{r+2}}. \end{aligned}$$

Both inequalities in the second line follow from (3.22). Since the tails at $s = 0$ are zero, it follows that

$$\varepsilon_j \le \frac{(\theta/2)^{\ell}}{(N+1)^{r+2}} + T\varepsilon_{j-1}.$$

Starting from $\varepsilon_1 = 0$ and iterating over j therefore gives

$$\sum_{s=1}^{T} a_{r-1,L_{r-1}+1}(s) \le T\varepsilon_{r-1} \le \frac{(\theta/2)^\ell}{(N+1)^{r+2}} \sum_{q=1}^{r-2} T^q \le \frac{(r-2)T^{r-2}}{(N+1)^{r+2}} \left(\frac{\theta}{2}\right)^\ell \le \left(\frac{\theta}{2}\right)^\ell. \tag{3.23}$$

The last inequality uses $T \le N$ and $N \ge k \ge r$.

The good term. Choose a sufficiently large constant $C_{k,r,\theta} \ge 1$, depending only on k, r, θ. We first prove by induction on j that

$$L_j \le C_{k,r,\theta}(\lambda_j + \ell + \log(N+1)), \qquad 1 \le j \le r-1. \tag{3.24}$$

The base case follows from $L_1 = \lambda_1 = T$. For the induction step, recall that

$$\lambda_j = T\sqrt{\frac{\lambda_{j-1}}{d}}, \qquad \frac{d\lambda_j}{T^2} = \left(\frac{d}{T}\right)^{2-\alpha_j} \ge 1.$$

Using $\sqrt{u+v} \le \sqrt{u} + \sqrt{v}$ for $u, v \ge 0$, the induction hypothesis gives

$$T\sqrt{\frac{L_{j-1}}{d}} \le \sqrt{C_{k,r,\theta}} \left(\lambda_j + T\sqrt{\frac{\ell + \log(N+1)}{d}}\right) \le \sqrt{C_{k,r,\theta}} \left(\frac{3}{2}\lambda_j + \frac{1}{2}(\ell + \log(N+1))\right).$$

Here we used

$$2T\sqrt{\frac{\ell + \log(N+1)}{d}} \le \frac{T^2}{d} + \ell + \log(N+1) \le \lambda_j + \ell + \log(N+1).$$

Substituting $c_j = \sqrt{8(k-j+1)}$ into (3.22), and using $j \ge 2$ and $1 \le \ell + \log(N+1)$, gives

$$L_j \le \left(\frac{3}{2}e^2\sqrt{8(k-1)C_{k,r,\theta}} + r + 3 + \log\frac{2}{\theta}\right)(\lambda_j + \ell + \log(N+1)).$$

For fixed k, r, θ, we can choose the sufficiently large constant $C_{k,r,\theta}$ such that

$$\frac{3}{2}e^2\sqrt{8(k-1)C_{k,r,\theta}} + r + 3 + \log\frac{2}{\theta} \le C_{k,r,\theta}.$$

This choice applies to every $2 \le j \le r-1$ and completes the induction.

Applying (3.17) and (3.24), and using $\lambda_r = T\sqrt{\lambda_{r-1}/d}$, we obtain

$$F_{r,\ell}(T, L_{r-1}) \le \left(\frac{ec_rT}{\ell}\sqrt{\frac{L_{r-1}}{d}}\right)^\ell \le \left[\frac{ec_r\sqrt{C_{k,r,\theta}}}{\ell}\left(T\sqrt{\frac{\lambda_{r-1}}{d}} + T\sqrt{\frac{\ell}{d}} + T\sqrt{\frac{\log(N+1)}{d}}\right)\right]^\ell$$

$$\leq \left[ec_r \sqrt{C_{k,r,\theta}} \left(\frac{\lambda_r}{\ell} + \frac{T}{\sqrt{d\ell}} + \frac{T}{\ell} \sqrt{\frac{\log(N+1)}{d}} \right) \right]^{\ell}. \tag{3.25}$$

The assumed query bound and $d \geq N/2$ imply

$$\frac{\lambda_r}{\ell} \leq 2^{\beta_r} c^{\alpha_r}, \qquad \frac{T}{\sqrt{d\ell}} \leq \sqrt{2}c \left(\frac{\ell}{N} \right)^{1/\alpha_r - 1/2} \leq \sqrt{2}c.$$

Moreover,

$$\begin{aligned} \frac{T}{\ell} \sqrt{\frac{\log(N+1)}{d}} &\leq \sqrt{2}c\, \ell^{1/\alpha_r - 1} N^{1/2 - 1/\alpha_r} \sqrt{\log(N+1)} \\ &\leq \sqrt{2}c\, N^{1/2 - 1/\alpha_r} \sqrt{\log(N+1)} = o(c), \end{aligned}$$

uniformly over $1 \leq \ell \leq n$, since $1 < \alpha_r < 2$. Thus the bracketed factor in (3.25) is at most

$$e\sqrt{8(k-r+1)C_{k,r,\theta}} \cdot \left(2^{\beta_r} c^{\alpha_r} + \sqrt{2}c + o(c) \right).$$

Choose $c > 0$ sufficiently small. For sufficiently large N, this factor is at most $\theta/2$, and hence $F_{r,\ell}(T, L_{r-1}) \leq (\theta/2)^{\ell}$.

Finally, Lemma 3.7 and (3.23) give

$$a_{r,\ell}(T) \leq F_{r,\ell}(T, L_{r-1}) + \sum_{s=1}^{T} a_{r-1, L_{r-1}+1}(s) \leq 2 \left(\frac{\theta}{2} \right)^{\ell} \leq \theta^{\ell}.$$

This completes the proof. □

This proves the database tail assertion in Theorem 3.1. The corresponding database event probability is $a_{r,\ell}(T)^2$, and is therefore at most $\theta^{2\ell}$. Furthermore, viewing $a_{r,\ell}(T)$ as a function of T, we obtain the following corollary by an argument similar to the proof above.

Corollary 3.9 (A query-dependent probability bound)**.** *Fix $2 \leq r \leq k$ and positive constants c_0, c_1. For sufficiently large N, uniformly over integers satisfying*

$$1 \leq \ell \leq N^{2^{-r}}, \qquad c_0 N^{\beta_r/\alpha_r} \leq T \leq c_1 \sqrt{N},$$

the database success probability satisfies

$$a_{r,\ell}(T)^2 \leq \min \left\{ 1, \left(\frac{C_{k,r} T^{\alpha_r}}{\ell N^{\beta_r}} \right)^{2\ell} \right\}. \tag{3.26}$$

Here $C_{k,r}$ depends only on k, r, while the threshold for sufficiently large N may also depend on c_0, c_1.

Proof. For sufficiently large N, we have $d = N - T \geq N/2$. The case $r = 2$ follows directly from (3.20). Assume henceforth that $r \geq 3$. It suffices to prove

$$a_{r,\ell}(T) \leq \min \left\{ 1, \left(\frac{C_{k,r} \lambda_r}{\ell} \right)^{\ell} \right\},$$

since $\lambda_r \leq 2^{\beta_r} T^{\alpha_r} / N^{\beta_r}$. The case $\lambda_r \geq \ell$ follows from normalization, so assume $\lambda_r < \ell$.

Fix the truncation thresholds by

$$L_1 = T, \qquad L_j = \left\lceil e^2 c_j T \sqrt{\frac{L_{j-1}}{d}} + (r+2)\log(N+1) + \ell \log \frac{2\ell}{\lambda_r} \right\rceil, \quad 2 \le j \le r-1.$$

The bad-term estimate in the preceding proof, with $\theta = \lambda_r/\ell$, gives

$$\sum_{s=1}^{T} a_{r-1,L_{r-1}+1}(s) \le \left(\frac{\lambda_r}{2\ell}\right)^{\ell}.$$

We next bound the good term. The lower bound on T and $d \le N$ imply

$$\lambda_r \ge c_0^{\alpha_r}, \qquad \lambda_{r-1} \ge c_0^{\alpha_{r-1}} N^{1/(2^r-1)}.$$

Consequently, uniformly over the stated parameter range,

$$(r+2)\log(N+1) + \ell \log \frac{2\ell}{\lambda_r} = O_{r,c_0}\Big(N^{2^{-r}} \log N\Big) = o(\lambda_{r-1}),$$

where the last equality uses $2^{-r} < 1/(2^r-1)$.

Since $T \le d$, the scales satisfy $\lambda_1 \ge \cdots \ge \lambda_{r-1}$. Thus, for sufficiently large N, the additive terms in each threshold are at most λ_j. The same induction as in the preceding proof now gives

$$L_j \le C_{k,r}\lambda_j, \qquad 1 \le j \le r-1.$$

Indeed, the induction step uses

$$T\sqrt{\frac{L_{j-1}}{d}} \le \sqrt{C_{k,r}}\lambda_j,$$

and the ceiling is absorbed by increasing $C_{k,r}$. It follows from (3.17) that

$$F_{r,\ell}(T, L_{r-1}) \le \left(\frac{C_{k,r}T\sqrt{\lambda_{r-1}/d}}{\ell}\right)^{\ell} = \left(\frac{C_{k,r}\lambda_r}{\ell}\right)^{\ell}.$$

Combining the good and bad terms in Lemma 3.7, and increasing $C_{k,r}$ to absorb their sum, proves the desired norm bound. Squaring and using $d \ge N/2$ completes the proof. □

3.4 Multi-Collision Query Lower Bounds

Let $N = kn$, where $k \ge r \ge 2$ are fixed, and recall

$$\alpha_r = 2 - 2^{1-r}, \qquad \beta_r = 1 - 2^{1-r}.$$

Combining Theorem 3.8 with (2.9) gives the following lower bound for the original function-query problem.

Corollary 3.10 (Multi-collision query lower bound)**.** *Fix $p_0 \in (0,1)$ and any alphabet size $q \ge n$. For every integer ℓ satisfying*

$$1 \le \ell \le N^{2^{-r}}, \tag{3.27}$$

finding ℓ r-tuples in distinct equality classes with success probability at least p_0 under the exact

(k^n)-type distribution requires

$$t = \Omega_{k,r,p_0}\Big(\ell^{1/\alpha_r} N^{\beta_r/\alpha_r}\Big) \tag{3.28}$$

queries to O_f, with the implicit constant independent of q.

Proof. Since $\beta_r + 2^{-r} = \alpha_r/2$, the range (3.27) gives

$$\begin{aligned} &\ell^{1/\alpha_r} N^{\beta_r/\alpha_r} \le \sqrt{N}, \\ &\frac{\ell}{\ell^{1/\alpha_r} N^{\beta_r/\alpha_r}} = \left(\frac{\ell}{N}\right)^{\beta_r/\alpha_r} \le N^{-(1-2^{-r})\beta_r/\alpha_r} = o(1). \end{aligned} \tag{3.29}$$

All asymptotic estimates below are uniform over this range.

Suppose that a t-query algorithm succeeds with probability $p \ge p_0$ and that $t \le c\,\ell^{1/\alpha_r} N^{\beta_r/\alpha_r}$, where $c > 0$ depends only on k, r, p_0 and will be chosen sufficiently small. By (3.29), the additional $r\ell$ queries are $o(\ell^{1/\alpha_r} N^{\beta_r/\alpha_r})$. Thus, for sufficiently small c and sufficiently large N, Theorem 3.8 yields

$$a_{r,\ell}(2t + r\ell) \le \left(\frac{\sqrt{p_0}}{4}\right)^{\ell} \le \frac{\sqrt{p_0}}{4}.$$

Moreover, (3.29) implies $(2t+2r\ell)/\sqrt{N} \le 2c+o(1)$, so the query-budget condition $3(2t+2r\ell)-1 \le N$ holds for sufficiently large N. The two error terms in (2.9) satisfy

$$\begin{aligned} r\ell\sqrt{\frac{2}{N - 2t - r\ell}} &= O_r\Big(N^{2^{-r}-1/2}\Big) = o(1), \\ \delta_N(2t + 2r\ell) &\le 120c + 2c^2 + o(1), \end{aligned}$$

where the second estimate follows from (2.6). Choose c smaller if necessary so that $120c+2c^2 < \sqrt{p_0}/4$. Substituting into (2.9) gives

$$\sqrt{p} \le \frac{\sqrt{p_0}}{4} + 120c + 2c^2 + o(1) < \sqrt{p_0}$$

for sufficiently large N. This contradicts $p \ge p_0$ and proves (3.28). □

Finally, we prove Theorem 1.1 by combining the preceding lower bound with a matching upper bound obtained from the Liu–Zhandry construction.

Proof of Theorem 1.1. Taking $p_0 = 2/3$ in Corollary 3.10 gives the lower bound for a uniformly random function of exact type (k^n). Any algorithm that succeeds with probability at least $2/3$ on every function of this type also succeeds with that probability under the uniform distribution. The corollary therefore implies the stated worst-case lower bound.

For the upper bound, use the construction of Liu and Zhandry [26] with list sizes

$$L_j = \Big\lceil \ell^{(2^r-2^{r-j})/(2^r-1)} N^{(2^{r-j}-1)/(2^r-1)} \Big\rceil, \qquad 1 \le j \le r.$$

Here $L_r = \ell$, and the assumed range of ℓ gives $L_j = o(L_{j-1})$ for $2 \le j \le r$, uniformly over that range. Querying kL_1 distinct inputs provides representatives of at least L_1 distinct equality classes, from which we form the initial list.

At stage j, extend L_j of the $(j-1)$-collisions in the preceding list, removing each source entry after its extension. After $u < L_j$ extensions, there remain $L_{j-1} - u$ active classes, each with exactly

$k-j+1$ preimages outside its stored tuple. The number of marked inputs is therefore known exactly, and exact amplitude amplification [11] performs each extension using $O_{k,r}(\sqrt{N/L_{j-1}})$ function queries. Removing the source entries ensures that the resulting collisions belong to distinct equality classes.

The total worst-case query count is

$$O_{k,r}\left(L_1+\sum_{j=2}^{r}L_j\sqrt{\frac{N}{L_{j-1}}}\right)=O_{k,r}\left(\ell^{2^{r-1}/(2^r-1)}N^{(2^{r-1}-1)/(2^r-1)}\right).$$

This construction works for every function of the prescribed type, with a query bound independent of q. Together with the lower bound, this proves the theorem. □

4 Multi-collision bounds for arbitrary exact types

We extend the preceding analysis to arbitrary exact types by counting intermediate collisions only in buckets that can contain an r-collision. After deriving local transition and first-moment bounds, we truncate all intermediate layers simultaneously and sum their leakage over time. This yields exponential database tails and function-query lower bounds in the range covered by the success-to-database interface. We conclude with a matching upper bound for the auxiliary model with public buckets and forward permutation access.

4.1 Eligible counters and local transition bounds

Similarly we first introduce some notation. Fix constants $k \geq r \geq 2$ and an exact type

$$\lambda=(1^{n_1},\dots,k^{n_k}),\qquad N=\sum_{i=1}^{k}in_i,\qquad n_*=\sum_{i=r}^{k}n_i. \tag{4.1}$$

Some n_i may be zero; no lower layer is required to contain $\Theta(N)$ equality classes. Let $Q_1,\dots,Q_h$ be the public bucket partition used in the permutation representation, and write

$$E=\{b:|Q_b|\geq r\},\qquad C_0^*(D)=n_*,\qquad C_j^*(D)=\sum_{b\in E}\mathbf{1}_{\{o_b(D)\geq j\}}\quad(1\leq j\leq r). \tag{4.2}$$

Here $o_b(D)=|\operatorname{im}(D)\cap Q_b|$, as before. Every intermediate counter uses the same eligible set E. Thus buckets of size less than r are excluded even when counting j-collisions for $j<r$. Pointwise on databases,

$$n_*=C_0^*(D)\geq C_1^*(D)\geq\cdots\geq C_r^*(D),$$

and

$$C_r^*(D)=C_r(D). \tag{4.3}$$

The target event is therefore unchanged. As with C_j, we suppress the argument D in event subscripts when the meaning is clear.

Define the good-event projections and their amplitudes by

$$P_{j,m}^*=\mathbf{1}_{\{C_j^*\geq m\}},\qquad a_{j,m}^*(s)=\left\|P_{j,m}^*\zeta_s\right\|,\qquad P_{j,0}^*=I,$$

for $0\leq j\leq r$ and integers $m\geq 0$. In particular, $P_{0,m}^*$ is I or 0 according as $m\leq n_*$ or $m>n_*$. We

write

$$P^{*,\perp}_{j,m} = (P^*_{j,m})^\perp = I - P^*_{j,m}.$$

The state ζ_s is the normalized joint state of an arbitrary forward compressed-query algorithm, with the database initially empty. If $n_* = 0$ or $\ell > n_*$, the target event is empty. Henceforth, assume $n_* \geq 1$ and $1 \leq \ell \leq n_*$. Each of the ℓ requested r-collisions must belong to a different equality class.

We first count the eligible extensions that cross a collision threshold and derive the corresponding one-query recurrence.

Lemma 4.1 (Eligible boundary counts and query recurrence)**.** *For each subspace $\mathcal{F}_{x,H}$ from Section 3.1, the number of available outputs whose addition increases C^*_j is*

$$b_j(H) = \sum_{\substack{b \in E \\ o_b(H) = j-1}} (|Q_b| - j + 1) \leq (k - j + 1)\big(C^*_{j-1}(H) - C^*_j(H)\big), \qquad 1 \leq j \leq r. \tag{4.4}$$

Use $c_j = \sqrt{8(k-j+1)}$, extending the earlier notation to $j = 1$. Let $0 \leq s < N$, $m \geq 1$, and $L \geq m-1$ be integers. For every vector v supported on databases of size at most s,

$$\left\| P^*_{j,m} U_{s+1} \mathsf{cpO} v \right\| \leq \left\| P^*_{j,m} v \right\| + c_j \sqrt{\frac{L-m+1}{N-s}} \left\| P^*_{j,m-1} v \right\| + \left\| P^*_{j-1,L+1} v \right\|. \tag{4.5}$$

In particular,

$$a^*_{j,m}(s+1) \leq a^*_{j,m}(s) + c_j \sqrt{\frac{L-m+1}{N-s}}\, a^*_{j,m-1}(s) + a^*_{j-1,L+1}(s). \tag{4.6}$$

For $j = 1$ and $L = n_*$, the last term is zero.

Proof. An eligible bucket increases the layer-j counter exactly when its occupancy changes from $j-1$ to j. It then has $|Q_b| - j + 1$ available outputs. This proves (4.4), including $j = 1$, where $C^*_0(H) = n_*$.

The algorithmic unitary U_{s+1} commutes with all database projections. A compressed query changes C^*_j by at most one. Consequently, among databases below the threshold, only those with $C^*_j = m-1$ can contribute to $P^*_{j,m}$ after a query. Split this boundary into

$$G = \mathbf{1}_{\{C^*_j = m-1,\ C^*_{j-1} \leq L\}}, \qquad R = \mathbf{1}_{\{C^*_j = m-1,\ C^*_{j-1} \geq L+1\}}.$$

These are orthogonal projections satisfying

$$G \preceq P^*_{j,m-1}, \qquad R \preceq P^*_{j-1,L+1}.$$

If $\mathcal{F}_{x,H}$ contains databases on both sides of the threshold, then $C^*_j(H) = m-1$. If it also contains a source database D in G with $|D| \leq s$, then $H \subseteq D$ gives $C^*_{j-1}(H) \leq L$ and $N - |H| \geq N - s$. Lemma 3.2 and (4.4) therefore give

$$\left\| P^*_{j,m} \mathsf{cpO} G \mathbf{1}_{\{|D| \leq s\}} \right\| \leq c_j \sqrt{\frac{L-m+1}{N-s}}. \tag{4.7}$$

Here we take the orthogonal direct sum over H and the query input x. The remaining source components are bounded by unitarity and contractivity of projections, exactly as in Theorem 3.3. The projection comparisons above give (4.5). Substituting $v = \zeta_s$ proves (4.6). □

4.2 First moments starting from the eligible bucket count

For the moment estimates only, distinguish the diagonal counting operators from the database statistics:

$$\widehat{C}_j^* = \sum_D C_j^*(D)|D\rangle\langle D|, \qquad \mu_j(s) = \langle \zeta_s, \widehat{C}_j^* \zeta_s \rangle.$$

Thus $\widehat{C}_0^* = n_* I$ and $\mu_0(s) = n_*$.

We next bound the first moment at each layer in terms of the preceding eligible counter.

Lemma 4.2 (Eligible first moments)**.** *For $1 \le j \le r$ and $0 \le s < N$,*

$$|\mu_j(s+1) - \mu_j(s)| \le c_j \sqrt{\frac{\mu_{j-1}(s)}{N-s}}, \tag{4.8}$$

$$a_{j,1}^*(s+1) - a_{j,1}^*(s) \le c_j \sqrt{\frac{\mu_{j-1}(s)}{N-s}}. \tag{4.9}$$

Proof. On $\mathcal{F}_{x,H}$, the counting operator has the form $C_j^*(H) I + B_{j,H}$, where $B_{j,H}$ projects onto the extensions that increase C_j^*. The constant part cancels under conjugation by the query. For any unitary V and orthogonal projection B, $\left\| V^\dagger B V - B \right\| = \|[B, V]\|$; the latter norm is the maximum of the two off-diagonal block norms of V relative to B. Applying Lemma 3.2 and (4.4) bounds this change on a subspace meeting $|D| \le s$ by $c_j \sqrt{C_{j-1}^*(H)/(N-s)}$.

Decompose the pre-query state into orthogonal components $v_{x,H}$ and put $w_{x,H} = \|v_{x,H}\|^2$. Since $\sum_{x,H} w_{x,H} = 1$ and $C_{j-1}^*(H) I \preceq \widehat{C}_{j-1}^*$ on the subspace,

$$\begin{aligned} |\mu_j(s+1) - \mu_j(s)| &\le \frac{c_j}{\sqrt{N-s}} \sum_{x,H} w_{x,H} \sqrt{C_{j-1}^*(H)} \\ &\le \frac{c_j}{\sqrt{N-s}} \sqrt{\sum_{x,H} w_{x,H} C_{j-1}^*(H)} \le c_j \sqrt{\frac{\mu_{j-1}(s)}{N-s}}. \end{aligned}$$

For the second assertion, the same orthogonal decomposition gives

$$\left\| P_{j,1}^* \mathsf{cpO} P_{j,1}^{*,\perp} \zeta_s \right\|^2 \le \frac{c_j^2}{N-s} \sum_{x,H} C_{j-1}^*(H) \, \|v_{x,H}\|^2 \le \frac{c_j^2 \mu_{j-1}(s)}{N-s}.$$

The already-good component contributes at most $a_{j,1}^*(s)$. □

For a fixed integer horizon $1 \le T < N$, retain $d = N - T$ and the earlier exponents $\alpha_j = 2 - 2^{1-j}$. The scales for the eligible counters are

$$\Lambda_0 = n_*, \qquad \Lambda_j = \frac{T}{\sqrt{d}} \sqrt{\Lambda_{j-1}} = \frac{T^{2-2^{1-j}} n_*^{2^{-j}}}{d^{1-2^{-j}}} \quad (1 \le j \le r). \tag{4.10}$$

In particular,

$$\Lambda_r = \frac{T^{2-2^{1-r}} n_*^{2^{-r}}}{(N-T)^{1-2^{-r}}}. \tag{4.11}$$

Unlike the earlier scales λ_j, these start at layer zero with the eligible bucket count n_*.

Iterating these inequalities from the eligible bucket count gives a bound for a single target collision.

Proposition 4.3 (Single-target bound)**.** *Set* $A_0^* = 1$ *and*

$$A_j^* = \prod_{i=1}^{j} \bigl(8(k-i+1)\bigr)^{2^{i-j-1}} \le (8k)^{1-2^{-j}} \qquad (1 \le j \le r).$$

These constants satisfy $A_j^* = c_j\sqrt{A_{j-1}^*}$*. The superscript distinguishes them from the earlier constants* A_j*, whose recurrence starts at* $A_1 = 1$*. For every* $1 \le j \le r$*,*

$$\sup_{0\le s\le T} \mu_j(s) \le A_j^*\Lambda_j, \qquad a_{j,1}^*(T) \le \min\{1, A_j^*\Lambda_j\}.$$

Proof. Starting from $\mu_0 = n_*$, induct on j and sum (4.8), using $\mu_j(0) = 0$ and $N - s \ge d$:

$$\mu_j(u) \le \frac{Tc_j}{\sqrt{d}}\sqrt{A_{j-1}^*\Lambda_{j-1}} = A_j^*\Lambda_j \qquad (0 \le u \le T).$$

Summing (4.9) gives the norm bound. □

As before, $\ell P_{r,\ell}^* \preceq \widehat{C}_r^*$ gives $(a_{r,\ell}^*(T))^2 \le A_r^*\Lambda_r/\ell$. The following argument strengthens this first-moment bound to exponential decay in the number of target classes.

4.3 Simultaneous truncation and multi-collision tails

Using the falling-factorial notation from the preliminaries, define

$$F_{j,m}(s,L) = \begin{cases} 1, & m = 0, \\ \dbinom{s}{m}\left(\dfrac{c_j}{\sqrt{d}}\right)^m\sqrt{(L)_m}, & 1 \le m \le \min\{s,L\}, \\ 0, & m > \min\{s,L\}. \end{cases} \tag{4.12}$$

The denominator $d = N - T$ is fixed throughout the interval $0 \le s \le T$. For integers $L \ge 0$ and $m \ge 1$,

$$F_{j,m}(T,L) \le \left(\frac{ec_jT\sqrt{L}}{m\sqrt{d}}\right)^m. \tag{4.13}$$

This follows from $\binom{T}{m} \le T^m/m!$, $(L)_m \le L^m$, and $m! \ge (m/e)^m$ whenever the left-hand side is nonzero.

To control multiple target classes, we truncate all intermediate layers simultaneously and sum the resulting leakage over time.

Theorem 4.4 (Finite bound by simultaneous truncation)**.** *Let* $L_0 = n_*$ *and choose nonnegative integer cutoffs* $L_1, \dots, L_{r-1}$*. For every forward compressed-query algorithm and* $1 \le T < N$*,*

$$a_{r,\ell}^*(T) \le F_{r,\ell}(T, L_{r-1}) + \sum_{j=1}^{r-1} F_{j,L_j+1}(T, L_{j-1}). \tag{4.14}$$

No monotonicity assumption on the cutoffs is required.

Proof. Define the contractive auxiliary trajectory

$$\xi_0 = \zeta_0, \qquad \xi_{s+1} = \left(\prod_{j=1}^{r-1} P^{*,\perp}_{j,L_j+1} \right) U_{s+1} \mathsf{cpO} \xi_s.$$

All cutoff projections commute. Thus $\|\xi_s\| \le 1$, its database support has size at most s, and $P^*_{j,L_j+1}\xi_s = 0$ for every $j < r$. These projections are used only in the proof.

For $1 \le j \le r$ and $m \ge 0$, denote $b_{j,m}(s) = \left\| P^*_{j,m}\xi_s \right\|$. We first prove

$$b_{j,m}(s) \le F_{j,m}(s, L_{j-1}). \tag{4.15}$$

If $m > L_{j-1}$, the left-hand side is zero by the cutoff at layer $j-1$ and counter monotonicity; for $j = 1$, use $C^*_1 \le n_*$. For $m = 0$, the assertion is contractivity. For $1 \le m \le L_{j-1}$, commutation of the cutoff projections with $P^*_{j,m}$ and the vector recurrence (4.5) give

$$\begin{aligned} b_{j,m}(s+1) &\le \left\| P^*_{j,m} U_{s+1} \mathsf{cpO} \xi_s \right\| \\ &\le b_{j,m}(s) + c_j \sqrt{\frac{L_{j-1} - m + 1}{d}}\, b_{j,m-1}(s). \end{aligned}$$

The bad-source term vanishes because the source ξ_s satisfies $C^*_{j-1} \le L_{j-1}$. The query itself need not preserve that constraint.

All positive-order tails are initially zero. Assuming (4.15) at time s, the last recurrence gives

$$\begin{aligned} b_{j,m}(s+1) &\le F_{j,m}(s, L_{j-1}) + c_j \sqrt{\frac{L_{j-1} - m + 1}{d}} F_{j,m-1}(s, L_{j-1}) \\ &= \left[\binom{s}{m} + \binom{s}{m-1} \right] \left(\frac{c_j}{\sqrt{d}} \right)^m \sqrt{(L_{j-1})_m} \\ &= F_{j,m}(s+1, L_{j-1}). \end{aligned}$$

Here $(L_{j-1} - m + 1)(L_{j-1})_{m-1} = (L_{j-1})_m$, and the last line is Pascal's identity. We use $\binom{s}{m} = 0$ for $m > s$. This proves (4.15) by induction on time.

The difference between the actual and truncated trajectories satisfies

$$\begin{aligned} \zeta_{s+1} - \xi_{s+1} = {} & U_{s+1} \mathsf{cpO} (\zeta_s - \xi_s) \\ & + \left(I - \prod_{j=1}^{r-1} P^{*,\perp}_{j,L_j+1} \right) U_{s+1} \mathsf{cpO} \xi_s. \end{aligned} \tag{4.16}$$

Expand the complementary product using the commuting projections:

$$I - \prod_{j=1}^{r-1} P^{*,\perp}_{j,L_j+1} = \sum_{j=1}^{r-1} \left(\prod_{i=1}^{j-1} P^{*,\perp}_{i,L_i+1} \right) P^*_{j,L_j+1}.$$

Unitarity, contractivity, and $\xi_0 = \zeta_0$ then imply

$$\|\zeta_T - \xi_T\| \le \sum_{j=1}^{r-1} \sum_{s=0}^{T-1} \left\| P^*_{j,L_j+1} U_{s+1} \mathsf{cpO} \xi_s \right\|. \tag{4.17}$$

In particular, every leakage term is evaluated on the truncated state.

If $L_j < L_{j-1}$, only the boundary $C^*_j = L_j$ can leak. Applying (4.7) to this boundary and then (4.15) gives

$$\left\| P^*_{j,L_j+1} U_{s+1} \mathsf{cpO} \xi_s \right\| \le \frac{c_j \sqrt{L_{j-1} - L_j}}{\sqrt{d}} F_{j,L_j}(s, L_{j-1}). \tag{4.18}$$

If $L_j \ge L_{j-1}$, the leakage is zero: a crossing subspace would require both $C^*_j(H) = L_j$ and an eligible bucket of occupancy exactly $j-1$, forcing $C^*_{j-1}(H) \ge L_j + 1 > L_{j-1}$. This contradicts the source constraint, also for $j = 1$.

For $L_j < L_{j-1}$, the summand on the right of (4.18) is

$$\binom{s}{L_j} \left(\frac{c_j}{\sqrt{d}}\right)^{L_j+1} \sqrt{(L_{j-1})_{L_j+1}}.$$

Its time dependence is retained exactly:

$$\sum_{s=0}^{T-1} \binom{s}{L_j} = \sum_{s=0}^{T-1} \left[\binom{s+1}{L_j+1} - \binom{s}{L_j+1} \right] = \binom{T}{L_j+1}.$$

Consequently,

$$\sum_{s=0}^{T-1} \left\| P^*_{j,L_j+1} U_{s+1} \mathsf{cpO} \xi_s \right\| \le F_{j,L_j+1}(T, L_{j-1}). \tag{4.19}$$

For $L_j \ge L_{j-1}$, both sides are zero. Finally,

$$a^*_{r,\ell}(T) \le b_{r,\ell}(T) + \|\zeta_T - \xi_T\|,$$

so (4.15), (4.17), and (4.19) prove the claim. □

The bad sources have not been discarded. Their contributions are the last $r-1$ terms in (4.14). Summing the binomial factors gives the next factorial tail, rather than multiplying a time-uniform error by T.

Fixed cutoffs and the database threshold. Define

$$\tau_{r,\ell}(N, n_*) = \frac{\sqrt{N}\, \ell^{1/(2-2^{1-r})}}{n_*^{1/(2^{r+1}-2)}}, \tag{4.20}$$

we now choose the cutoffs so that the target term and all leakage terms decay exponentially in ℓ.

Theorem 4.5 (Exponential database tail for all target counts)**.** *Fix $\theta \in (0,1)$. There is a constant $c_{k,r,\theta} > 0$ such that, for every type with $n_* \ge 1$ and every integer $1 \le \ell \le n_*$,*

$$a^*_{r,\ell}(T) \le \theta^\ell \quad \textit{whenever} \quad 0 \le T \le c_{k,r,\theta} \tau_{r,\ell}(N, n_*). \tag{4.21}$$

The query count T is an integer.

Proof. The case $T = 0$ is immediate. Write $c = c_{k,r,\theta}$ and require $c \le 1/4$. Since $\ell \le n_* \le N$, the threshold is at most $\sqrt{Nn_*} \le N$. Hence $T \le N/4$, $d \ge N/2$, and $T^2/d \le 2c^2 n_* \le n_*$. Choose, once and for all,

$$A = \max\left\{1, \log \frac{2(r-1)}{\theta}\right\},$$

and fix the cutoffs

$$L_0 = n_*, \qquad L_j = \left\lceil e^2\sqrt{8(k-j+1)}\frac{T\sqrt{L_{j-1}}}{\sqrt{d}} + A\ell \right\rceil \quad (1 \le j \le r-1). \tag{4.22}$$

Leakage terms. From (4.22), we have $L_j + 1 \ge e^2 c_j T\sqrt{L_{j-1}/d}$ and $L_j + 1 \ge A\ell$. Thus (4.13) implies

$$F_{j,L_j+1}(T, L_{j-1}) \le e^{-(L_j+1)} \le e^{-A\ell}.$$

This remains true when the tail is identically zero. By the choice of A,

$$\sum_{j=1}^{r-1} F_{j,L_j+1}(T, L_{j-1}) \le (r-1)e^{-A\ell} \le (\theta/2)^\ell.$$

The target term. Use the single constant $C_k = 32e^4 k$. We claim

$$L_j \le C_k(\Lambda_j + A\ell) \qquad (0 \le j \le r-1). \tag{4.23}$$

The case $j = 0$ is immediate. The scale recurrence and $n_* \ge T^2/d$ give $\Lambda_j \ge T^2/d$ at every layer. If the claim holds at $j-1$, then

$$\begin{aligned} L_j &\le e^2\sqrt{8kC_k}\left(\Lambda_j + \frac{T\sqrt{A\ell}}{\sqrt{d}}\right) + 2A\ell \\ &\le e^2\sqrt{8kC_k}\left(\frac{3}{2}\Lambda_j + \frac{1}{2}A\ell\right) + 2A\ell \\ &= \frac{3}{4}C_k\Lambda_j + \left(\frac{1}{4}C_k + 2\right)A\ell \le C_k(\Lambda_j + A\ell). \end{aligned}$$

The first line absorbs the ceiling using $A\ell \ge 1$. The second uses $2T\sqrt{A\ell/d} \le T^2/d + A\ell \le \Lambda_j + A\ell$. This proves (4.23) with a constant independent of the layer.

Substituting into (4.13) gives

$$F_{r,\ell}(T, L_{r-1}) \le \left[e\sqrt{8kC_k}\left(\frac{\Lambda_r}{\ell} + \frac{T\sqrt{A}}{\sqrt{d\ell}}\right)\right]^\ell.$$

The assumed query bound yields

$$\frac{\Lambda_r}{\ell} \le 2^{1-2^{-r}} c^{2-2^{1-r}}, \qquad \frac{T}{\sqrt{d\ell}} \le \sqrt{2}c\left(\frac{\ell}{n_*}\right)^{1/(2^{r+1}-2)} \le \sqrt{2}c. \tag{4.24}$$

Choose $c > 0$ sufficiently small that

$$e\sqrt{8kC_k}\left(2^{1-2^{-r}} c^{2-2^{1-r}} + \sqrt{2A}\, c\right) \le \theta/2.$$

The target term is then at most $(\theta/2)^\ell$. Theorem 4.4 gives $a^*_{r,\ell}(T) \le 2(\theta/2)^\ell \le \theta^\ell$. □

A uniform bound as a function of the query count. Keeping the truncation thresholds fixed while varying the query count gives an explicit exponential tail for all target counts.

Corollary 4.6 (A uniform query-dependent tail)**.** *There is a constant $C_{k,r} > 0$ such that, for every integer $1 \le \ell \le n_*$ and every integer $0 \le T < N$,*

$$a^*_{r,\ell}(T) \le \min\left\{1, \left[\frac{C_{k,r}\, T^{\,2-2^{1-r}} n_*^{2^{-r}}}{\ell N^{1-2^{-r}}}\right]^{\ell/(2-2^{1-r})}\right\}. \tag{4.25}$$

The constant is uniform over the input type and the number n_* of eligible buckets.

Proof. Fix $\theta = 1/2$ in the proof of Theorem 4.5, and let $c \le 1/4$ and $A \ge 1$ be the constants chosen there. Set

$$U = c\,\tau_{r,\ell}(N, n_*).$$

Since $\ell \le n_* \le N$, we have $U \le N/4$. Choose the cutoffs in (4.22) with T replaced by U and d replaced by $N - U$. These cutoffs are fixed throughout the following argument, and $L_j + 1 \ge A\ell \ge \ell$ for every $j < r$.

The coarse estimates in the proof of Theorem 4.5 give

$$\left(\frac{ec_r U\sqrt{L_{r-1}}}{\ell\sqrt{N-U}}\right)^\ell + \sum_{j=1}^{r-1}\left(\frac{ec_j U\sqrt{L_{j-1}}}{(L_j+1)\sqrt{N-U}}\right)^{L_j+1} \le 2^{-\ell}.$$

These estimates remain valid for real U, since only the cutoffs are rounded to integers.

For an integer $1 \le T \le U$, we have $N - T \ge N - U$. Hence (4.13) implies, for every $m \ge 1$,

$$F_{j,m}(T, L) \le \left(\frac{T}{U}\right)^m \left(\frac{ec_j U\sqrt{L}}{m\sqrt{N-U}}\right)^m.$$

In (4.14), the target term has order $m = \ell$, and every leakage term has order $m = L_j + 1 \ge \ell$. Since $T/U \le 1$, it follows that

$$a^*_{r,\ell}(T) \le 2^{-\ell}\left(\frac{T}{U}\right)^\ell \le \left(\frac{T}{c\,\tau_{r,\ell}(N, n_*)}\right)^\ell.$$

For $T > U$, the last expression exceeds 1, so the norm bound $a^*_{r,\ell}(T) \le 1$ applies. The case $T = 0$ is immediate. Substituting (4.20) and adjusting the constant proves (4.25). □

The exponent $\ell/(2-2^{1-r})$ is $\Theta_r(\ell)$. Thus the estimate gives exponential decay below a sufficiently small constant multiple of $\tau_{r,\ell}(N, n_*)$ throughout the full range $1 \le \ell \le n_*$ and $0 \le T < N$.

4.4 Function-query lower bounds for arbitrary exact types

Let f be uniform among functions of exact type $\lambda = (1^{n_1}, \dots, k^{n_k})$, with alphabet size $q \ge h = \sum_{i=1}^k n_i$. The injective-label representation and the simulation from Proposition 2.6 impose no further condition on q. By (4.3), Theorem 2.7 gives

$$\sqrt{p} \le a^*_{r,\ell}(2t + r\ell) + r\ell\sqrt{\frac{2}{N - 2t - r\ell}} + \delta_N(2t + 2r\ell), \tag{4.26}$$

provided $3(2t + 2r\ell) - 1 \le N$. Here δ_N is defined in (2.6). The $2t$ queries simulate the original function queries. The additional $r\ell$ reads expose the permutation images of the reported inputs, and the output-witness lemma contributes another $r\ell$ to the soundness term. The database estimates hold uniformly over auxiliary registers and the independent injective label table. Thus, one may condition on that table and average the resulting success-probability bound.

Combining Theorem 4.5 with (4.26) gives the following lower bound.

Theorem 4.7 (Function-query lower bound for arbitrary types)**.** *Fix constants $k \ge r \ge 2$ and $p_0 \in (0, 1)$. For sufficiently large N, uniformly over all exact types $\lambda = (1^{n_1}, \dots, k^{n_k})$ with $n_* = \sum_{i=r}^{k} n_i \ge 1$, all $q \ge h$, and all integers*

$$1 \le \ell \le n_*^{2^{-r}}, \tag{4.27}$$

any quantum algorithm that, for every function f of exact type λ, finds r-collisions in ℓ distinct equality classes with probability at least p_0 requires

$$\Omega_{k,r,p_0}\left(\frac{\sqrt{N}\,\ell^{1/(2-2^{1-r})}}{n_*^{1/(2^{r+1}-2)}}\right) \tag{4.28}$$

queries to O_f in the worst case.

Proof. Write

$$\tau = \frac{\sqrt{N}\,\ell^{1/(2-2^{1-r})}}{n_*^{1/(2^{r+1}-2)}}.$$

The permitted target counts imply

$$\tau \le \sqrt{N}, \qquad \frac{\ell}{\tau} \le \frac{n_*^{2^{-r}}}{\sqrt{N}} \le N^{2^{-r}-1/2} = o(1). \tag{4.29}$$

These estimates are uniform over the stated range.

Suppose such an algorithm has worst-case query count $t \le c\tau$, where constant $c > 0$ will be chosen sufficiently small. Its success probability p on a uniformly random function of type λ satisfies $p \ge p_0$. By (4.29),

$$2t + r\ell \le (2c + o(1))\tau.$$

For sufficiently small c and sufficiently large N, Theorem 4.5, applied with $\theta = \sqrt{p_0}/4$, gives

$$a^*_{r,\ell}(2t + r\ell) \le \left(\frac{\sqrt{p_0}}{4}\right)^{\ell} \le \frac{\sqrt{p_0}}{4}.$$

Moreover, $2t + 2r\ell \le (2c + o(1))\sqrt{N}$, so the query-budget condition in (4.26) holds for sufficiently large N. The remaining terms satisfy

$$r\ell\sqrt{\frac{2}{N - 2t - r\ell}} = O_r\Big(N^{2^{-r}-1/2}\Big) = o(1), \qquad \delta_N(2t + 2r\ell) \le 120c + 2c^2 + o(1),$$

where the second estimate follows from (2.6). Decrease c if necessary so that $120c + 2c^2 < \sqrt{p_0}/4$. Substituting into (4.26) gives

$$\sqrt{p} \le \frac{\sqrt{p_0}}{4} + 120c + 2c^2 + o(1) < \sqrt{p_0}$$

for sufficiently large N, contradicting $p \geq p_0$. □

For the pure type (k^n), we have $n_* = n = N/k$, so (4.28) recovers the scale of Corollary 3.10. The same asymptotic scale holds whenever $n_* = \Theta(N)$.

4.5 A matching upper bound with forward permutation access

We now give an algorithm attaining the preceding query scale when the bucket partition is public and the algorithm has forward access to P_π. The algorithm combines the search-domain preprocessing used for non-uniform collision finding [29] with the layered collision lists of [26]. In the present setting, the public bucket sizes determine the number of solutions to every search exactly. This yields an upper bound for every permutation, including when all eligible buckets must contribute a collision.

Theorem 4.8 (Upper bound for arbitrary public bucket types)**.** *Fix $k \geq r \geq 2$ and a public bucket partition $Q_1, \ldots, Q_h$ of $[N]$ of type $(1^{n_1}, \ldots, k^{n_k})$. Let $n_* = \sum_{i=r}^{k} n_i \geq 1$. For every integer $1 \leq \ell \leq n_*$, there is a quantum algorithm that, for every permutation $\pi \in S_N$, finds r distinct preimages in each of ℓ distinct buckets using*

$$O_{k,r}\left(\frac{\sqrt{N}\,\ell^{1/(2-2^{1-r})}}{n_*^{1/(2^{r+1}-2)}}\right) \tag{4.30}$$

forward swap queries to P_π. The algorithm succeeds with certainty and does not require queries to π^{-1}.

Proof. Call a bucket eligible if its size is at least r. Choose integers $n_* \geq t_1 \geq \cdots \geq t_r = \ell$, to be specified below. At stage 1, the algorithm finds one preimage in each of t_1 distinct eligible buckets. At stage $j \geq 2$, it extends records from t_j distinct buckets in the preceding list, obtaining t_j j-collisions. All lists record bucket identities and the corresponding inputs and permutation images.

Extending a list. Suppose the current list contains m eligible buckets, each with $j-1$ recorded preimages, and we wish to extend $t \leq m$ of them. For $j = 1$, the current list consists of all n_* eligible buckets, initially with no recorded preimages. Maintain a list of pending buckets, initially containing all m buckets. Search for an input x such that $\pi(x)$ belongs to a pending bucket and x is not one of its recorded preimages. After a successful search, append x to the record of its bucket, move the resulting j-collision to the new list, and remove that bucket from the pending list.

After u extensions, the number of marked inputs is exactly

$$\sum_{b \text{ pending}} \left(|Q_b| - j + 1\right) \geq m - u.$$

This number is known from the public bucket sizes and the pending list. The search predicate is evaluated by computing $\pi(x)$, testing the bucket and recorded-input conditions, and uncomputing $\pi(x)$. Both oracle calls are forward swap queries because $P_\pi^2 = I$. Exact amplitude amplification [11] therefore finds the next marked input with certainty using $O(\sqrt{N/(m-u)})$ forward queries. One additional query records its permutation image, which is absorbed in the same bound.

The total cost of the t extensions is thus at most

$$O\left(\sqrt{N}\sum_{u=0}^{t-1}\frac{1}{\sqrt{m-u}}\right) = O\left(t\sqrt{\frac{N}{m}}\right).$$

Indeed,

$$\sum_{u=0}^{t-1} \frac{1}{\sqrt{m-u}} \le \int_{m-t}^{m} \frac{dv}{\sqrt{v}} = \frac{2t}{\sqrt{m}+\sqrt{m-t}} \le \frac{2t}{\sqrt{m}}.$$

This bound also holds for $t = m$. Every search succeeds with certainty, and the query bound holds for every sequence of measured outcomes.

Choosing the list sizes. Applying the preceding procedure at each stage gives the total query bound

$$O_r\left(t_1\sqrt{\frac{N}{n_*}} + \sum_{j=2}^{r} t_j\sqrt{\frac{N}{t_{j-1}}}\right). \tag{4.31}$$

Using the exponents $\alpha_j = 2 - 2^{1-j}$ defined above, choose

$$t_j = \left\lceil n_*\left(\frac{\ell}{n_*}\right)^{\alpha_j/\alpha_r}\right\rceil, \qquad 1 \le j \le r. \tag{4.32}$$

These integers satisfy $n_* \ge t_1 \ge \cdots \ge t_r = \ell$. Every quantity inside a ceiling lies between ℓ and n_*, so rounding increases it by at most a factor of two. Since $\alpha_1 = 1$ and $\alpha_j - \alpha_{j-1}/2 = 1$, we obtain

$$t_1\sqrt{\frac{N}{n_*}} \le 2\sqrt{Nn_*}\left(\frac{\ell}{n_*}\right)^{1/\alpha_r},$$

and, for $2 \le j \le r$,

$$t_j\sqrt{\frac{N}{t_{j-1}}} \le 2\sqrt{Nn_*}\left(\frac{\ell}{n_*}\right)^{(\alpha_j-\alpha_{j-1}/2)/\alpha_r} = 2\sqrt{Nn_*}\left(\frac{\ell}{n_*}\right)^{1/\alpha_r}.$$

Summing the fixed number of stages and using

$$\sqrt{Nn_*}\left(\frac{\ell}{n_*}\right)^{1/\alpha_r} = \frac{\sqrt{N}\,\ell^{1/(2-2^{1-r})}}{n_*^{1/(2^{r+1}-2)}}$$

proves (4.30). Each final record contains r distinct preimages of one eligible bucket, and the ℓ records use distinct buckets, as required. □

For every injective label table a, the output also consists of ℓ r-collisions of $f_{\pi,a} = a \circ b \circ \pi$ in distinct equality classes. The algorithm does not need to know a. Theorem 4.8 matches the scale in Theorem 4.5 for all $1 \le \ell \le n_*$. The following corollary states the resulting tight real-query bound in the range covered by our simulation and witness estimates.

Corollary 4.9 (Tight complexity with forward permutation access)**.** *Fix $k \ge r \ge 2$. For sufficiently large N, uniformly over all public bucket partitions of the specified type and all integers $1 \le \ell \le n_*^{2^{-r}}$, the quantum query complexity of finding r preimages in each of ℓ distinct buckets, with success probability at least $2/3$ for a uniformly random permutation, is*

$$\Theta_{k,r}\left(\frac{\sqrt{N}\,\ell^{1/(2-2^{1-r})}}{n_*^{1/(2^{r+1}-2)}}\right). \tag{4.33}$$

Only forward swap queries to P_π are allowed.

Proof. The upper bound follows from Theorem 4.8. For the lower bound, apply the argument of

Theorem 4.7 directly to a T-query permutation algorithm. Since no function-query simulation is needed, the success-to-database bound is (4.26) with $2t$ replaced by T. The same threshold and soundness estimates then yield the matching lower bound. □

The matching upper bound in the preceding corollary applies to the enlarged permutation-query model. It does not imply that our lower bound is tight for every exact type in the original function-query model. For example, consider type $(1^n, 2)$ on $N = n + 2$ inputs, with $q \geq N$. Our lower bound is $\Omega(\sqrt{N})$, and the matching algorithm uses $O(\sqrt{N})$ queries to P_π. However, finding the unique collision using only O_f is the unique-collision search version of element distinctness, whose bounded-error quantum query complexity is $\Theta(N^{2/3})$ [31, 4].

This gap reflects the additional information supplied by the permutation oracle. If Q_* is the unique public bucket of size two, access to P_π allows the algorithm to test whether $\pi(x) \in Q_*$, and hence whether x belongs to the colliding pair. A query to O_f reveals only $f(x)$ and does not directly reveal the size of its equality class. The original algorithm has access only to O_f, whereas the matching upper bound allows arbitrary forward queries to P_π. Consequently, that algorithm need not be implementable with the same query complexity in the original function-query model. In the next section, we use a sampling reduction to relate our collision lower bounds to k-distinctness on sparse exact types, and recover the resulting lower-bound exponents within Belovs's framework.

5 A sampling reduction to promised k-distinctness

This section connects our pure-type lower bound to a sparse promise for k-distinctness. We prove the random-restriction reduction in subsection 5.1 and then derive the same exponent directly from Belovs's framework in subsection 5.2.

5.1 A sampling reduction to sparse-type distinctness

Random restriction connects the pure-type collision problem to a distinctness problem with a sparse multiplicity profile. We first determine the distribution of that profile, including the probability that the restriction contains exactly one complete bucket.

Fix an integer $k \geq 2$ and a constant $c > 0$. Let $N = kn$ and

$$m = \left\lfloor cN^{(k-1)/k} \right\rfloor.$$

Choose $f : [N] \to \mathbb{Z}_q$ uniformly from the exact (k^n)-type distribution, where $q \geq n$, and independently choose a uniformly random injection $\iota : [m] \hookrightarrow [N]$. Write $g = f \circ \iota$. For $0 \leq j \leq k$, let H_j be the number of equality classes of f containing exactly j sampled inputs. Thus the exact type of g is $(1^{H_1}, 2^{H_2}, \ldots, k^{H_k})$ and $\sum_{j=1}^{k} jH_j = m$. First, we present an estimation for the expectation of H_j for all $1 \leq j \leq k$.

Lemma 5.1 (The multiplicity profile of a random restriction)**.** *For each $1 \le j \le k$,*

$$\mu_j := \mathbb{E}H_j = \frac{N}{k}\binom{k}{j}\frac{(m)_j(N-m)_{k-j}}{(N)_k} \sim \frac{1}{k}\binom{k}{j}c^j N^{1-j/k}. \tag{5.1}$$

For every $1 \le j < k$, we have $H_j/\mu_j \to 1$ in probability. Moreover, H_k converges in distribution to a Poisson random variable with mean c^k/k. In particular,

$$\Pr[H_k = 1] \longrightarrow \frac{c^k}{k}\exp(-c^k/k) > 0. \tag{5.2}$$

The simultaneous concentration statements for $j < k$ also hold conditional on $H_k = 1$.

Proof. The sampled set is a uniform m-subset of $[N]$. A fixed k-element bucket has exactly j sampled elements with probability

$$\binom{k}{j}\frac{(m)_j(N-m)_{k-j}}{(N)_k}.$$

Summing over the N/k buckets gives the first equality in (5.1). Since k is fixed, $m \sim cN^{(k-1)/k}$, and $m/N \to 0$, the stated asymptotic formula follows.

For two distinct buckets, the corresponding joint probability gives

$$\mathbb{E}(H_j)_2 = \binom{N}{k}_2\binom{k}{j}^2\frac{(m)_{2j}(N-m)_{2(k-j)}}{(N)_{2k}}.$$

Dividing by μ_j^2 and expanding the fixed-length falling factorials shows that

$$\frac{\mathbb{E}(H_j)_2}{\mu_j^2} = 1 + O_k\!\left(\frac{1}{m} + \frac{1}{N-m} + \frac{1}{N}\right) = 1 + O_k(1/m).$$

Consequently,

$$\frac{\mathrm{Var}(H_j)}{\mu_j^2} = O_k\!\left(\frac{1}{\mu_j} + \frac{1}{m}\right). \tag{5.3}$$

For $j < k$, the mean μ_j tends to infinity, so Chebyshev's inequality proves concentration. There are only $k-1$ such indices, and hence the concentration is simultaneous.

For every fixed integer $a \ge 1$, requiring a specified buckets to be fully sampled gives

$$\mathbb{E}(H_k)_a = \binom{N}{k}_a\frac{(m)_{ka}}{(N)_{ka}} \longrightarrow \left(\frac{c^k}{k}\right)^a.$$

The factorial-moment criterion for Poisson convergence proves (5.2). Finally, conditioning an event of probability $1 - o(1)$ on $H_k = 1$ preserves probability $1 - o(1)$, because $\Pr[H_k = 1]$ is bounded below by a positive constant. □

We next record the symmetry needed to apply a solver whose guarantee is stated for a uniform exact-type distribution.

Lemma 5.2 (Uniformity conditional on the sampled type)**.** *For every vector $\mathbf{h} = (h_1, \dots, h_k)$ with $\Pr[(H_1, \dots, H_k) = \mathbf{h}] > 0$, the conditional distribution of g is uniform among all functions $[m] \to \mathbb{Z}_q$*

of exact type $(1^{h_1}, \dots, k^{h_k})$.

Proof. The distribution of g is invariant under every permutation of its m inputs and every permutation of its q output labels. These actions preserve its multiplicity profile and act transitively on the functions of any fixed exact type. The conditional distribution on each nonempty such orbit is therefore uniform. □

For a precise promise family, set $\varepsilon_N = N^{-1/(4k)}$ and let $\mathcal{H}_N$ consist of the vectors $\mathbf{h} = (h_1, \dots, h_k) \in \mathbb{Z}_{\geq 0}^k$ satisfying

$$h_k = 1, \qquad \sum_{j=1}^{k} j h_j = m, \qquad |h_j - \mu_j| \leq \varepsilon_N \mu_j \quad (1 \leq j < k). \tag{5.4}$$

Let E_N be the event $(H_1, \dots, H_k) \in \mathcal{H}_N$, and let $\mathcal{D}_N$ be the distribution of g conditional on E_N. Equation (5.3) and a union bound show that the inequalities $|H_j - \mu_j| \leq \varepsilon_N \mu_j$ hold simultaneously for all $1 \leq j < k$ with probability $1 - o(1)$. Since membership in $\mathcal{H}_N$ additionally requires $H_k = 1$, equation (5.2) gives

$$\Pr[E_N] = \Pr[H_k = 1] + o(1) = \frac{c^k}{k} \exp(-c^k/k) + o(1). \tag{5.5}$$

Indeed, $\min_{j<k} \mu_j = \Theta_{k,c}(N^{1/k})$, so the probability of violating any of the concentration inequalities in (5.4) is $O_{k,c}(N^{-1/(2k)})$. Every type in this family satisfies

$$h_j = \Theta_{k,c}\Big(N^{1-j/k}\Big) = \Theta_{k,c}\Big(m^{(k-j)/(k-1)}\Big) \quad (1 \leq j < k), \qquad h_k = 1. \tag{5.6}$$

In particular, $h_1 = (1 - o(1))m$. By Lemma 5.2, $\mathcal{D}_N$ is a mixture of uniform exact-type distributions with profiles in $\mathcal{H}_N$.

The following reduction transfers the pure-type lower bound to this mixture, and consequently to any algorithm required to solve the whole typical-profile promise family.

Proposition 5.3 (A lower bound for the sampled promise family)**.** *Fix $p_0 \in (0, 1)$. Suppose a quantum algorithm, given oracle access to $g : [m] \to \mathbb{Z}_q$, finds a k-collision with probability at least p_0 when g is distributed according to $\mathcal{D}_N$. The algorithm may depend on the public parameters N, k, c, q, but is not additionally given the realized vector $(H_1, \dots, H_k)$. Its query complexity satisfies*

$$T = \Omega_{k,c,p_0}\Big(N^{(2^{k-1}-1)/(2^k-1)}\Big) = \Omega_{k,c,p_0}\left(m^{\frac{k(2^{k-1}-1)}{(k-1)(2^k-1)}}\right), \tag{5.7}$$

uniformly over $q \geq N/k$. In particular, the same bound holds for a single algorithm that succeeds with probability at least p_0 under the uniform exact-type distribution for every profile in $\mathcal{H}_N$, without receiving the realized profile as additional input.

Proof. Given oracle access to a uniformly random exact $(k^{N/k})$-type function $f : [N] \to \mathbb{Z}_q$, choose ι independently as above and run the proposed algorithm on $g = f \circ \iota$. A query to g can be implemented with one query to f: reversibly compute $\iota(x)$ in an input register, query f into the algorithm's response register, and uncompute $\iota(x)$. Computing the sampled injection requires no oracle queries. If the algorithm outputs a k-collision $(x_1, \dots, x_k)$ of g, then $(\iota(x_1), \dots, \iota(x_k))$ is a k-collision of f. The algorithm can be truncated after its stated query budget, so its behavior outside the promise does not increase that budget.

The reduction need not recognize E_N or determine the sampled profile. Its unconditional success probability is at least

$$p_0 \Pr[E_N] = p_0 \frac{c^k}{k} \exp(-c^k/k) + o(1),$$

which is bounded below by a positive constant depending only on k, c, p_0. Apply Corollary 3.10 with $r = k$ and $\ell = 1$ to obtain the first bound in (5.7). The second follows from $m \sim cN^{(k-1)/k}$. If the algorithm instead has the stated guarantee for every uniform exact type in $\mathcal{H}_N$, averaging these guarantees over the mixture $\mathcal{D}_N$ gives the hypothesis already proved. □

5.2 Comparison with the profiled partition framework

The sampled profiles also admit a direct analysis in Belovs's partition framework. We retain the number of blocks at each level in his query-gain argument, rather than imposing a linear number of blocks at every level. This gives the same exponent as the sampling reduction for each fixed profile, even when its block counts are given to the algorithm, under the alphabet-size assumption $q = \Omega(m^2)$.

Proposition 5.4 (Lower bound for a fixed sampled-scale profile)**.** *Fix an integer $k \geq 2$ and positive constants c_j, C_j for $1 \leq j < k$. Let N tend to infinity and let*

$$m = k + \sum_{j=1}^{k-1} j b_j, \qquad c_j N^{1-j/k} \leq b_j \leq C_j N^{1-j/k} \quad (1 \leq j < k).$$

For an alphabet of size $q \geq c_q m^2$, where $c_q > 0$ is fixed, finding the unique k-collision in a uniformly random function of exact type $(1^{b_1}, 2^{b_2}, \ldots, (k-1)^{b_{k-1}}, k^1)$ with constant success probability requires

$$\Omega\left(N^{\frac{2^{k-1}-1}{2^k-1}}\right) = \Omega\left(m^{\frac{k(2^{k-1}-1)}{(k-1)(2^k-1)}}\right) \quad \text{queries.}$$

The implicit constant can be chosen uniformly over all profiles satisfying the displayed inequalities. The algorithm may be given the exact numbers $b_1, \ldots, b_{k-1}$. Consequently, the same lower bound holds for any mixture of these uniform exact-type distributions on a common domain and alphabet. In particular, it applies to the sampled distribution $\mathcal{D}_N$ over profiles in $\mathcal{H}_N$ whenever $q \geq c_q m^2$.

Proof. We first work in the relaxed hidden-partition model of [6, Definition 4.3]: the partition is uniform within its orbit, and its blocks receive independent uniform labels in $\mathbb{Z}_q$. The required output is the distinguished k-block of the underlying partition. Distinct blocks need not have distinct labels in this intermediate model.

For $1 \leq j \leq k$, let $\mathcal{M}_j$ be the orbit obtained by replacing the distinguished k-block by one highlighted j-block and $k - j$ singletons. For $j < k$, let $\mathcal{M}_j^\circ$ be the orbit obtained by forgetting the highlight. These are the auxiliary orbits used in [6, Section 6.1], with their block counts left explicit. Their sizes satisfy

$$|\mathcal{M}_j| = \frac{m!}{j!\,(b_1 + k - j)!\,\prod_{i=2}^{k-1} (i!)^{b_i} b_i!}, \tag{5.8}$$

$$\frac{|\mathcal{M}_{j+1}|}{|\mathcal{M}_j|} = \frac{b_1 + k - j}{j+1} = \Theta(m), \tag{5.9}$$

$$\frac{|\mathcal{M}_j|}{|\mathcal{M}_j^\circ|} = h_j, \qquad h_1 = b_1 + k, \qquad h_j = b_j + 1 \quad (2 \leq j < k). \tag{5.10}$$

In particular, splitting a highlighted block still contributes a factor of order $m^{-1/2}$; only the cost of forgetting the highlight depends on the number of blocks at that level.

We use the transfer and knowledge operators of [6, Sections 3 and 6]. Write ψ_t for the uniform algorithm state after t queries, and ψ'_t for the state immediately before its next query. The operator Υ_j^+ retains the Fourier components whose support contains the highlighted j-block. Put

$$A_j(t) = \|\Upsilon_j^+ \psi_t\| = \|\Upsilon_j^+ \psi'_t\|.$$

The equality holds because intervening unitaries act only on the algorithm's registers. Every unhighlighted transfer operator sends these uniform states to the algorithm states for a normalized input distribution; therefore

$$\|\Upsilon_j^\circ \psi_t\| = \|\Upsilon_j^\circ \psi'_t\| = 1. \tag{5.11}$$

This is a statement about these algorithm states, not a bound on the operator norms of the transfer or knowledge operators.

We next retain the block counts in the two estimates that control query gain. Lemma 6.6 of [6] and (5.9) give, for every vector φ,

$$\|\Psi_{j+1}^\partial \varphi\| \le \sqrt{\frac{|\mathcal{M}_j|}{|\mathcal{M}_{j+1}|}} \|\Upsilon_j^+ \varphi\| = O(m^{-1/2}) \|\Upsilon_j^+ \varphi\|. \tag{5.12}$$

The highlighted-singleton estimate [6, Lemma 6.8], together with (5.11), gives

$$A_1(t)^2 \le \frac{t}{b_1 + k} = O(t/m). \tag{5.13}$$

For $2 \le j < k$, let Ξ_j project onto the components for which the query index lies in the highlighted j-block. The key adaptation to sparse profiles is to retain the orbit ratios in the highlighted-query estimate. The local-alteration proof of [6, Lemma 6.14], before its final substitution of these ratios, yields

$$\|\Xi_j \Upsilon_j^+ \psi'_t\|^2 \le C_k \sum_{i=1}^{j} m^{j-i} \frac{|\mathcal{M}_i^\circ|}{|\mathcal{M}_j|} \|\Upsilon_i^\circ \psi'_t\|^2 \le C \sum_{i=1}^{j} \frac{1}{h_i} = O(b_j^{-1}). \tag{5.14}$$

For completeness, the counting behind the first inequality does not require linear block counts. Splitting a highlighted j-block along a subset of size i gives a partition in $\mathcal{M}_i^\circ$. Inclusion–exclusion expresses the coefficient with the whole highlighted block in the Fourier support as a signed sum of these split coefficients. For a fixed query index in the original highlighted block, a given split partition has at most $O_k(m^{j-i})$ preimages: reconstruct the highlighted block by merging an i-block with $j - i$ singletons, with one of the merged blocks containing that fixed index. The empty-subset term is included with $i = 1$ and has at most $O_k(m^{j-1})$ preimages. Cauchy–Schwarz and summation give the first inequality in (5.14). The second uses (5.9)–(5.11). The last uses $b_i = \Omega(b_j)$ for $i \le j$, uniformly over the profiles under consideration.

We now apply the refined query-gain inequality while retaining its quadratic term. Set $B_{j-1}(t) = A_{j-1}(t) + A_{j-1}(t+1)$. Lemma 6.13 of [6], (5.12), and (5.14) imply

$$A_j(t+1)^2 - A_j(t)^2 \le C \left(\frac{B_{j-1}(t)}{\sqrt{m b_j}} + \frac{B_{j-1}(t)^2}{m} \right), \qquad 2 \le j < k. \tag{5.15}$$

We next solve (5.15) by induction. Let $\theta_k = (2^{k-1}-1)/(2^k-1)$, and define

$$a_j = 1 - 2^{-j}, \qquad d_j = 1 - \frac{j}{2k} - 2^{-j} \qquad (1 \le j < k).$$

For every integer $0 \le t \le \lfloor N^{\theta_k} \rfloor$, we prove by induction on j that

$$A_j(t) \le C'_j \frac{t^{a_j}}{N^{d_j}} \qquad (1 \le j < k). \tag{5.16}$$

The case $j = 1$ follows from (5.13), since $m = \Theta(N^{1-1/k})$.

For the induction step, first observe that

$$\varepsilon_j := d_j - a_j\theta_k = (1-2^{-j})(1-\theta_k) - \frac{j}{2k} > 0 \qquad (1 \le j < k). \tag{5.17}$$

Indeed, $(1-2^{-j})/j$ is decreasing in j, so

$$\frac{\varepsilon_j}{j} \ge \frac{\theta_k}{k-1} - \frac{1}{2k} > 0,$$

where the last inequality is equivalent to $2^k - 1 > k$. Thus, for $2 \le j < k$ and $0 \le t < \lfloor N^{\theta_k} \rfloor$, the induction hypothesis gives

$$B_{j-1}(t) = O(N^{-\varepsilon_{j-1}}) = O(1).$$

Since $b_j \le m$, it follows that

$$\frac{B_{j-1}(t)^2}{m} = O\left(\frac{B_{j-1}(t)}{\sqrt{mb_j}}\right).$$

Hence the quadratic term in (5.15) is controlled by the linear term using the bound already established at the preceding level. As $A_j(0) = 0$, summing the recurrence gives

$$A_j(t)^2 = O\left(\frac{t^{a_{j-1}+1}}{N^{d_{j-1}+(2k-j-1)/(2k)}}\right) = O\left(\frac{t^{2a_j}}{N^{2d_j}}\right),$$

which proves (5.16).

At the last level, we apply the ordinary amplitude query-gain inequality [6, Corollary 3.24]. Together with (5.12) and (5.16), it gives

$$\begin{aligned} A_k(T) &\le \frac{C}{\sqrt{m}} \sum_{t=0}^{T-1} \left(A_{k-1}(t) + A_{k-1}(t+1)\right) \\ &\le C\frac{T^{2-2^{1-k}}}{N^{1-2^{1-k}}} = C\left(\frac{T}{N^{\theta_k}}\right)^{2-2^{1-k}} \end{aligned} \tag{5.18}$$

for every integer $0 \le T \le \lfloor N^{\theta_k} \rfloor$. This also covers $k = 2$, for which there are no intermediate applications of the refined recurrence.

The target orbit has $b_1 = \Theta(m)$ singletons and bounded block size, so the anti-concentration theorem [6, Theorem 5.4] applies for $T = o(m)$. Its low-knowledge subspace is $O(m^{-1/2})$-anti-concentrated. Since $\|\Upsilon_k \psi_T\| = 1$, the low-knowledge component has norm at most $1 + A_k(T)$.

Consequently, the success probability $p_{\mathrm{rel}}(T)$ in the relaxed model satisfies

$$\sqrt{p_{\mathrm{rel}}(T)} \le A_k(T) + O(m^{-1/2})(1 + A_k(T)),$$
$$p_{\mathrm{rel}}(T) \le C\left(m^{-1} + \frac{T^{4-2^{2-k}}}{N^{2-2^{2-k}}}\right).$$

Here $N^{\theta_k} = o(m)$, so the required query range is satisfied.

Finally, let $B = 1 + \sum_{j<k} b_j \le m$ be the number of blocks in the target partition. Under independent uniform block labels, the probability that all labels are distinct is

$$\frac{(q)_B}{q^B} = \Omega_{c_q}(1) \qquad \text{when } q \ge c_q m^2.$$

Conditioned on this event, the resulting function is uniform within the stated exact type, and its unique k-collision is precisely the distinguished block. Thus the same success-probability bound, with a constant depending additionally on c_q, holds in the exact-type model. Taking T to be a sufficiently small constant multiple of N^{θ_k} rules out any prescribed constant success probability and proves the claimed lower bound.

All constants in these estimates are uniform over the admissible profiles. Averaging the success-probability bound therefore gives the same lower bound for any mixture of their uniform exact-type distributions on a common domain and alphabet, including $\mathcal{D}_N$ under the stated alphabet-size assumption. □

Combining the sampling reduction above with Proposition 5.4, we establish a concrete connection between our collision bounds and Belovs's transfer framework on sparse exact types. In the next section, we explore whether the two approaches can be combined to obtain tight bounds for types with missing intermediate layers.

6 Toward $(k+1)$-distinctness of type $(k^n, k+1)$

Belovs [6] establishes a tight quantum query lower bound for $(k+1)$-distinctness under the assumption that the input contains a unique $(k+1)$-block and $\Omega(N)$ blocks of each size $1, \dots, k$, where N is the input size. His analysis does not establish the same bound for arbitrary exact types. In particular, he asks whether the bound remains valid for type $(k^n, k+1)$, where all layers below size k are absent. We use this open problem to examine the complementary strengths and current limitations of the two approaches, and to identify a possible direction for combining them.

Our database recurrence has a useful feature for this problem: its intermediate levels describe partial occupancy of buckets, rather than the presence of actual blocks of the corresponding sizes in the input. Thus, a type consisting almost entirely of k-blocks still supports a hierarchy of database events at levels $1, \dots, k$. Theorem 3.3 controls the accumulation of multiple collisions through this hierarchy. However, the arbitrary-type analysis also permits independent access to the permutation oracle P_π. As discussed after Corollary 4.9, this additional access allows the unique exceptional bucket to be found in $O(\sqrt{N})$ queries. A stronger lower bound for the original problem must therefore exploit the restriction to function queries.

Belovs's transfer analysis retains this restriction and relates knowledge across different partition types. Its existing application, however, does not directly provide an analogue of the multi-target recurrence in Theorem 3.3. As explained in Remark 2.10, our good-database events provide an alternative notion of high knowledge, defined by orthogonal projections after transfer and

embedding. Belovs's decomposition instead splits source Fourier supports before transfer. The resulting progress measures cannot be identified directly, so combining the two analyses requires an additional comparison. A possible combination would use our recurrence to control the number of available intermediate collisions, and a transfer argument to control how this knowledge can reveal the unique exceptional block. The following construction makes this proposed connection precise.

Fix $k \geq 2$, let $N = k(n+1)+1$, and take $q \geq n+2$. Consider the two types

$$\lambda_- = (1, k^{n+1}), \qquad \lambda_+ = (k^n, k+1). \tag{6.1}$$

They admit a simple coupling. Sample f_- uniformly from the functions of type λ_-, let z be its unique singleton input, and choose a uniformly random k-class B. Obtain f_+ by changing only $f_-(z)$ to the common output of B. The resulting f_+ is uniform among functions of type λ_+: each such function has exactly $(k+1)(q-n-1)$ inverse constructions, corresponding to the choice of z in its exceptional block and an unused former label for z. The marked class is independent of the negative execution, but it determines the positive oracle.

Run the same function-query algorithm on the two distributions, with the same initial state and inter-query operations. Let ζ_t^- and ζ_t^+ be the corresponding normalized compressed states, starting from an empty database. Here t counts complete function-query blocks; the algorithm has no independent access to P_π or to intermediate simulation registers. Let $\mathcal{B}_-$ index the $n+1$ negative buckets of size k, and let b_* denote the unique positive bucket of size $k+1$. Writing o_b for database occupancy, define

$$\widehat{C}_k^- = \sum_{b \in \mathcal{B}_-} \mathbf{1}_{\{o_b = k\}}, \quad \mu_k^-(t) = \langle \zeta_t^-, \widehat{C}_k^- \zeta_t^- \rangle, \quad P_{k+1}^+ = \mathbf{1}_{\{o_{b_*} = k+1\}}, \quad A_{k+1}^+(t) = \|P_{k+1}^+ \zeta_t^+\|. \tag{6.2}$$

The following lemma shows that negative first moment has an exact interpretation when a bucket is marked independently of the execution.

Lemma 6.1 (Independent marking)**.** *Let*

$$\bar{\zeta}_t^- = \frac{1}{\sqrt{n+1}} \sum_{b \in \mathcal{B}_-} |b\rangle \otimes \zeta_t^-,$$

and let P_k^{mark} project onto databases in which the marked bucket has occupancy k. Then

$$\|P_k^{\mathrm{mark}} \bar{\zeta}_t^-\|^2 = \frac{\mu_k^-(t)}{n+1} = \frac{1}{n+1} \sum_{\ell=1}^{n+1} \|\mathbf{1}_{\{\widehat{C}_k^- \geq \ell\}} \zeta_t^-\|^2. \tag{6.3}$$

Proof. Orthogonality of the mark register gives $(n+1)^{-1} \sum_b \|\mathbf{1}_{\{o_b = k\}} \zeta_t^-\|^2$. The two identities follow from

$$\sum_{b \in \mathcal{B}_-} \mathbf{1}_{\{o_b = k\}} = \widehat{C}_k^- = \sum_{\ell=1}^{n+1} \mathbf{1}_{\{\widehat{C}_k^- \geq \ell\}}.$$

Taking the expectation in ζ_t^- and dividing by $n+1$ establishes the claimed identities. □

Thus, our collision counters quantify knowledge of an independently marked negative bucket. They do not yet control the positive execution, whose oracle depends on the mark. A fixed hidden permutation does not remove this distinction: later queries may become correlated with that permutation through earlier oracle responses.

The splitting construction identifies another obstacle. Let $\mathcal{M}_{k+1}$ be the orbit of partitions of type λ_+ with the exceptional block highlighted. Splitting one element off that block gives the orbit $\mathcal{M}_k$ of partitions of type λ_- with one k-block highlighted. If $\mathcal{M}_k^\circ$ denotes the same orbit without a highlight, then

$$\frac{|\mathcal{M}_k|}{|\mathcal{M}_{k+1}|} = k+1, \qquad |\mathcal{M}_k| = (n+1)|\mathcal{M}_k^\circ|. \tag{6.4}$$

Indeed, each positive partition has $k+1$ splittings, and each resulting highlighted partition has a unique inverse obtained by merging its singleton with its highlighted block. The second identity counts the possible highlights.

Consequently, the normalization factor in Belovs's splitting lemma [6, Lemma 6.6] is constant for these two orbits. The $O(N^{-1/2})$ factor in [6, Corollary 6.7] cannot simply be carried over. Independent marking supplies one factor of order $N^{-1/2}$ through (6.3); an additional gain would require a new argument that uses the function-query restriction. A concrete target for such an argument is the following estimate.

Conjecture 6.2 (Cross-type moment transfer)**.** *For every fixed $k \geq 2$, there are constants $C_k, c_k > 0$ such that, for all sufficiently large n, every $q \geq n+2$, and every function-query algorithm described above,*

$$A_{k+1}^+(T) \leq \frac{C_k}{N} \sum_{t=1}^{T} \sqrt{\mu_k^-(t)} \qquad (1 \leq T \leq c_k N). \tag{6.5}$$

The constants are independent of n, q, and the algorithm.

The first-moment estimate already proved in Proposition 4.3 gives

$$\mu_k^-(t) = O_k\left(\frac{t^{\,2-2^{1-k}}}{N^{\,1-2^{1-k}}}\right).$$

The constant c_k may be decreased to keep the compressed-query overhead within the range of that estimate. Substituting into Conjecture 6.2 would yield

$$A_{k+1}^+(T) = O_k\left(\frac{T^{\,2-2^{-k}}}{N^{\,3/2-2^{-k}}}\right). \tag{6.6}$$

In particular, constant target amplitude would require

$$T = \Omega_k\left(N^{\,\frac{3}{4}-\frac{1}{4(2^{k+1}-1)}}\right). \tag{6.7}$$

For $k=2$, this is the scale $N^{5/7}$ for type $(3, 2^n)$.

The conjecture also addresses the alphabet-size restriction. Belovs's lower bound assumes $q = \Omega(N^2)$ [6], whereas Conjecture 6.2 has constants independent of q for every $q \geq n+2$. If proved, it would yield the conditional database bound above without the quadratic alphabet-size assumption. Converting this database conclusion into a function-query lower bound would additionally require a soundness estimate effective at the scale in (6.7), uniformly over $q \geq n+2$. The current CPO soundness bound only reaches the square-root scale.

The proposed combination therefore has two specific tasks: relating the transfer boundary to our collision counters without exposing the hidden partition, and connecting the resulting progress measure to success without a supply of singletons. Our recurrence addresses the accumulation of intermediate collisions; the transfer framework offers a way to compare the hidden target with those

intermediate structures. Establishing the missing comparison would extend the connection proved in the preceding section to the singleton-free problem posed by Belovs.

Another direction is to construct a compressed oracle directly for uniformly random k-to-one functions. The goal is to retain an orthogonal database and the original function-query interface without first passing through a permutation oracle. The completion-state construction and normalization method of Carolan and Majenz [14] provide a possible starting point, but establishing useful query-gain and soundness estimates already presents a substantial challenge for pure types. General path-recording constructions can perfectly simulate random elements of closed subgroups of the unitary group [17]. The question here is whether a representation for exact-type functions supports the local collision-count recurrences and success comparison needed for our lower bounds.

To see an additional difficulty in the representation-theoretic analysis, let $\Omega_{k,n}$ be the family of k-to-one functions from $[kn]$ to $[n]$. Young's rule gives

$$\mathbb{C}[\Omega_{k,n}] \cong \bigoplus_{\nu \vdash kn} V_\nu \otimes \mathbb{C}^{K_{\nu,(k^n)}},$$

where V_ν is the irreducible S_{kn}-module indexed by ν and $K_{\nu,(k^n)}$ is a Kostka number. Unlike the permutation case, these multiplicities need not equal $\dim V_\nu$. The hook-length formula controls the irreducible dimensions, but does not determine the additional multiplicities or the operators acting on their spaces. Moreover, conditioning on recorded input-output pairs changes the remaining multiplicities, even when the original type is pure.

Extending the representation-theoretic analysis of Carolan and Majenz to k-to-one functions may therefore require uniform estimates for Kostka multiplicities, together with spectral and branching estimates for the associated Gram operators. Moreover, a successful construction would also need local query bounds that support recursive control analogous to Theorem 3.3; an isometric representation alone does not provide such bounds.

Statement on AI Use

The author proposed the central ideas of this work: representing exact-type functions using CPO, deriving lower bounds through database-amplitude recurrences, and the sampling reduction to sparse-type k-distinctness. GPT-6 Astra assisted throughout the development and preparation of the manuscript. For example, the author supplied the recurrence in Theorem 3.3 and an initial proof inspired by Liu and Zhandry [26]; subsequent discussions with Astra refined the truncation thresholds and database-tail estimates without changing the asymptotic query lower bounds. Astra also assisted with the sampling analysis, conjecture formulation, English exposition, and bibliography preparation. The introduction and technical overview combine author-written text with model-generated drafts edited by the author. The author takes responsibility for the final manuscript, including its mathematical arguments and references.

References

[1] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004.

[2] Andris Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, STOC '00, pages 636–643.

Association for Computing Machinery, 2000. Full version: https://arxiv.org/abs/quant-ph/0002066.

[3] Andris Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing*, 1(3):37–46, 2005.

[4] Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007. https://arxiv.org/abs/quant-ph/0311001.

[5] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001.

[6] Aleksandrs Belovs. Tight quantum lower bound for k-distinctness. arXiv:2604.05133v1, 2026. Version 1, 6 April 2026.

[7] Aleksandrs Belovs and Ansis Rosmanis. Adversary lower bounds for the collision and the set equality problems. *Quantum Information and Computation*, 18(3–4):198–222, 2018.

[8] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In *Theory of Cryptography – TCC 2019, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 61–90. Springer, 2019.

[9] Xavier Bonnetain, André Chailloux, André Schrottenloher, and Yixin Shen. Finding many collisions via reusable quantum walks: Application to lattice sieving. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 221–251. Springer, 2023. doi:10.1007/978-3-031-30589-4_8.

[10] Xavier Bonnetain, Johanna Loyer, André Schrottenloher, and Yixin Shen. A tight quantum algorithm for multiple collision search. Cryptology ePrint Archive, Paper 2025/1690, 2025. https://eprint.iacr.org/2025/1690.

[11] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In Samuel J. Lomonaco, Jr. and Howard E. Brandt, editors, *Quantum Computation and Information*, volume 305 of *Contemporary Mathematics*, pages 53–74. American Mathematical Society, 2002. Full version: https://arxiv.org/abs/quant-ph/0005055.

[12] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In Cláudio L. Lucchesi and Arnaldo V. Moura, editors, *LATIN'98: Theoretical Informatics*, volume 1380 of *Lecture Notes in Computer Science*, pages 163–169. Springer, 1998.

[13] Joseph Carolan. Compressed permutation oracles. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, STOC '26, pages 150–161. Association for Computing Machinery, 2026. Full version: https://arxiv.org/abs/2509.18586v1.

[14] Joseph Carolan and Christian Majenz. Compressed permutation oracles revisited. arXiv:2609.28469v1, 2026. https://arxiv.org/abs/2609.28469v1.

[15] Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In *Advances in Cryptology – EUROCRYPT 2021, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 598–629. Springer, 2021.

[16] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. In *Advances in Cryptology – EUROCRYPT 2022, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 677–706. Springer, 2022.

[17] Ben Foxman, Alex Lombardi, Fermi Ma, Barak Nehoran, and John Wright. Quantum lazy sampling and path recording for any group. Cryptology ePrint Archive, Paper 2026/1510, 2026. `https://eprint.iacr.org/2026/1510`.

[18] Jiangxia Ge, Tianshu Shan, and Rui Xue. Tighter QCCA-secure key encapsulation mechanism with explicit rejection in the quantum random oracle model. In *Advances in Cryptology – CRYPTO 2023, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 292–324. Springer, 2023.

[19] Amin Shiraz Gilani, Daochen Wang, Pei Wu, and Xingyu Zhou. Quantum algorithms on edge lists: Hiding, shuffling, and cycle finding. In *53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026)*, volume 374 of *Leibniz International Proceedings in Informatics*, pages 97:1–97:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026.

[20] Yassine Hamoudi and Frédéric Magniez. Quantum time–space tradeoff for finding multiple collision pairs. *ACM Transactions on Computation Theory*, 15(1–2):3:1–3:22, 2023.

[21] Akinori Hosoyamada, Yu Sasaki, Seiichiro Tani, and Keita Xagawa. Improved quantum multicollision-finding algorithm. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography – PQCrypto 2019*, volume 11505 of *Lecture Notes in Computer Science*, pages 350–367. Springer, 2019. doi:10.1007/978-3-030-25510-7_19.

[22] Akinori Hosoyamada, Yu Sasaki, and Keita Xagawa. Quantum multicollision-finding algorithm. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 179–210. Springer, 2017. doi:10.1007/978-3-319-70697-9_7.

[23] Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 526–535. Association for Computing Machinery, 2007. Full version: `https://arxiv.org/abs/quant-ph/0611054`.

[24] Sunyeop Kim. Improved soundness for compressed permutation oracles and tight quantum preimage and collision bounds for the sponge. Cryptology ePrint Archive, Paper 2026/2135, 2026. `https://eprint.iacr.org/2026/2135`.

[25] Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 344–353. IEEE Computer Society, 2011. Full version: `https://arxiv.org/abs/1011.3020`.

[26] Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 189–218. Springer, 2019. Full version: `https://arxiv.org/abs/1811.05385v2`.

[27] Frédéric Magniez and Sebastian Zur. Tight time-space lower bounds for collision finding and element distinctness under label symmetry. arXiv:2609.10808v2, 2026. Version 2, 2 October 2026.

[28] Christian Majenz, Giulio Malavolta, and Michael Walter. Permutation superposition oracles for quantum query lower bounds. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 1508–1519. Association for Computing Machinery, 2025.

[29] Tianci Peng, Shujiao Cao, and Rui Xue. On quantum query complexities of collision-finding in non-uniform random functions. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology – ASIACRYPT 2025, Part VIII*, volume 16252 of *Lecture Notes in Computer Science*, pages 193–223. Springer, 2026.

[30] Ben W. Reichardt. Reflections for quantum query algorithms. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 560–569. Society for Industrial and Applied Mathematics, 2011.

[31] Ansis Rosmanis. Quantum adversary lower bound for element distinctness with small range. arXiv:1401.3826, 2014. `https://arxiv.org/abs/1401.3826`.

[32] Ansis Rosmanis. Tight bounds for inverting permutations via compressed oracle arguments. arXiv:2103.08975, 2021. `https://arxiv.org/abs/2103.08975`.

[33] Dominique Unruh. Towards compressed permutation oracles. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part IV*, volume 14441 of *Lecture Notes in Computer Science*, pages 369–400. Springer, 2023.

[34] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 239–268. Springer, 2019. Full version: `https://eprint.iacr.org/2018/276`.