Quantum Time-Space Tradeoffs for Short Merkle-Damgård Collisions

Shashwat Agrawal* Ashrujit Ghoshal† Venkata Koppula‡

Abstract

We study time-space tradeoffs for finding short collisions in Merkle-Damgård (MD) hashing in the quantum auxiliary-input quantum random-oracle model (QAI-QROM). An adversary may preprocess the compression function into S qubits of arbitrary oracle-dependent quantum advice and, after receiving a fresh salt, make T superposition queries to find a collision consisting of at most B blocks.

Previous quantum analyses gave an $\widetilde{O}(ST^3/N)$ security bound for MD collisions with classical advice. Unlike the classical theory of short MD collisions, however, this bound is insensitive to the collision length B: existing quantum techniques did not exploit the combinatorial structure forced by a short collision.

We develop a structure-sensitive compressed-oracle analysis that overcomes this obstacle. For two-block collisions, we prove an upper bound

$$\widetilde{O}\left(\min\left\{1, \frac{ST^2+T^3}{N}\right\}\right),$$

even against arbitrary quantum advice. For $M = N$, together with the known attack, this gives a tight characterization up to logarithmic factors, closes the previous factor-T gap in the preprocessing term, and resolves an open problem of Akshima, Guo, and Liu (CRYPTO '22, JoC '24).

For general $B \le T$, we show in the regime $(2S+1)(T+2B)^3 \le N$ that the success probability is at most

$$\widetilde{O}\left(\min\left\{1, \frac{ST^2B+T^3}{N}\right\}\right).$$

Thus, when $B \ll T$, the preprocessing term improves over the previous ST^3/N bound by a factor of roughly T/B. Motivated by the tight two-block result, we propose the ST^2B conjecture, asserting that $ST^2B = \widetilde{\Theta}(N)$ is the correct constant-success tradeoff for short MD collisions in the regime $T^3 = o(N)$.

1 Introduction

The study of time-space tradeoffs in cryptography dates back to the seminal work of Hellman [Hel80], who showed that an adversary can use preprocessing and memory to reduce the online cost of inverting a one-way function. This basic phenomenon of trading storage acquired in an offline phase for a faster online attack has since become a central theme in the concrete analysis of cryptographic primitives. Understanding such tradeoffs is important both for assessing the security of practical constructions and for characterizing the power of non-uniform adversaries. These questions take on renewed urgency in the quantum setting, as steady progress in quantum hardware brings practical quantum attackers closer to reality [Goo25].

*IIT Delhi. Email: csz248012@cse.iitd.ac.in

†IIT Madras. Email: ashrujit@cse.iitm.ac.in

‡IIT Delhi. Email: kvenkata@cse.iitd.ac.in

Time-space tradeoffs for MD collisions. In this work, we study time-space tradeoffs for collision finding in *Merkle-Damgård (MD) hashing*, one of the most widely used paradigms for building a variable-input-length hash function from a fixed-input-length compression function. Variants of the MD paradigm underlie the SHA-2 family, which remains one of the most widely deployed families of cryptographic hash functions. Fix a compression function $h : [N] \times [M] \to [N]$. Starting from an initial *salt* $a_0 \in [N]$, the MD hash of a B-block message $m = (m_1, \ldots, m_B)$ is obtained by the iteration $a_i = h(a_{i-1}, m_i)$ for $i = 1, \ldots, B$, with the final value a_B taken as the output. A *collision* is a pair of distinct messages that map to the same output under the same salt. We work in the *auxiliary-input random-oracle model* [Unr07, DGK17] (AI-ROM), in which h is modeled as a uniformly random function: an unbounded preprocessing stage examines the entire compression function and produces an S-bit advice string, after which a fresh salt a_0 is sampled and the online algorithm, given the advice and the salt, is allowed T oracle queries to h.[1] We refer to such adversaries as (S, T)-*adversaries* in the AI-ROM.

MD collisions in AI-ROM were first studied by Coretti, Dodis, Guo, and Steinberger (CDGS) [CDGS18], who proved that the success probability of any (S, T)-adversary is at most $\widetilde{O}\left(ST^2/N\right)$. They also showed that this bound is tight up to logarithmic factors. There is, however, an important caveat to the CDGS attack: the collisions it produces are *long*, with messages consisting of roughly T blocks. This significantly limits the relevance of the attack to realistic collision-finding scenarios.

Short collisions and the STB conjecture. Motivated by this observation, Akshima, Cash, Drucker, and Wee [ACDW20] initiated the systematic study of *short MD collisions*. For collisions consisting of at most $B \geq 2$ blocks,[2] they gave an (S, T)-adversary with success probability $\widetilde{\Omega}\left(\left(STB + T^2\right)/N\right)$ and put forward the *STB conjecture*, asserting that this attack is optimal, up to logarithmic factors, for every $2 \leq B \leq T$. Their main technical result established the conjecture for $B = 2$, giving the tight bound $\widetilde{\Theta}\left(\left(ST + T^2\right)/N\right)$, but extending the security proof even to $B = 3$ appeared to require qualitatively new ideas.

Substantial subsequent work has been devoted to the STB conjecture. Ghoshal and Komargodski [GK22] developed structural characterizations of bounded-length MD collisions and proved the conjecture for every constant B, as well as when $S^4B^2 \in \widetilde{O}(T)$. Akshima, Guo, and Liu [AGL22, AGL24] subsequently introduced a different framework, inspired by multi-instance techniques [IK10, ACDW20, CGLQ20], and obtained the general security bound $\widetilde{O}\left(\frac{STB}{N} \cdot \max\left\{1, \frac{ST^2}{N}\right\} + \frac{T^2}{N}\right)$. Consequently, the STB conjecture holds throughout the regime $ST^2 \leq N$, for arbitrary $2 < B < T$; outside this regime their bound is within a factor S of the conjectured optimum. Thus, although the complete classical picture remains elusive in some parameter regimes, the short-collision problem is by now well understood over a broad range of parameters.

A recurring feature of these classical analyses is that they exploit the combinatorial structure imposed by a bounded-length collision. Carrying such structural reasoning to the quantum setting is substantially harder: the online algorithm no longer reveals a classical set of queried edges, and existing quantum preprocessing techniques do not directly retain enough information about the shape of the recorded MD computation.

Quantum time-space tradeoffs. The situation becomes considerably more delicate when the online algorithm is quantum. There are two distinct sources of difficulty: the online algorithm may query the random oracle in superposition, and the preprocessing stage may itself produce quantum

[1] Looking ahead, when we work with quantum adversaries, we will assume the online adversary is a quantum algorithm that receives an S-qubit advice and is allowed T superposition queries to h.

[2] When $B = 1$, the Merkle–Damgård construction is simply a salted random oracle, and its collision resistance was analyzed in [DGK17].

advice. Following prior work, we refer to the resulting model as the *quantum auxiliary-information quantum random-oracle model* (QAI-QROM). These difficulties already arise in the fundamental problem of *function inversion* (finding the preimage of $h(x)$ for a random $x \in [M]$ and random $h : [M] \to [N]$). Nayebi, Aaronson, Belovs, and Trevisan [NABT15] initiated the study of quantum time–space tradeoffs for inversion, proving an $ST^2 = \widetilde{\Omega}(N)$ lower bound for random permutations with classical advice; subsequent works by Chung, Guo, Liu and Qian [CGLQ20] and Liu [Liu23] extended this line to random functions and to quantum advice, showing that an (S, T)-adversary in the QAI-QROM can invert a random function with probability at most $O((ST + T^2)/N)$. Further, Dong, Liu, and Wu [DLW24] utilized this framework to yield a $\widetilde{O}((S/K) + (T^2/N))$ bound for salted preimage-finding where $h : [K] \times [M] \to [N]$ and $[K]$ is the salt space.

Quantum time-space tradeoffs for collision finding. Without preprocessing, quantum collision search in a random 2-to-1 function succeeds with probability $\Theta(T^3/N)$ [BHT98], which is also optimal [AS04]. Chung, Guo, Liu, and Qian [CGLQ20] also gave the first quantum time-space lower bounds for salted collision finding in the preprocessing setting. Dong, Liu, and Wu [DLW24] obtain $\widetilde{O}(S/K + T^3/N)$ for salted collision resistance when the salt space has size K; equivalently, this gives the corresponding bound for one-block Merkle-Damgård hashing.

For general Merkle–Damgård hashing, the main prior non-uniform quantum security bound is due to Guo, Li, Liu, and Zhang (GLLZ) [GLLZ21]. Using quantum presampling together with the compressed-oracle framework, they showed that an adversary with S bits of *classical advice* and T quantum queries finds an MD collision with probability at most $\widetilde{O}\left(\frac{ST^3}{N}\right)$.

This was an important first step, but the bound does not exploit the length of the collision: the same ST^3/N preprocessing term is obtained for two-block collisions as for unrestricted MD collisions. In particular, the classical strategy of gaining security by analyzing the structure of short collision paths had no known quantum analogue.

On the attack side, GLLZ [GLLZ21] observed that the classical two-block attack of ACDW [ACDW20] admits a quantum speedup, giving success probability $\Omega((ST^2 + T^3)/N)$. This leaves a factor of T between the known preprocessing upper and lower bounds even for the shortest nontrivial case $B = 2$. The resulting question, explicitly left open in [AGL22, AGL24], is therefore already fundamental:

What is the optimal quantum time–space tradeoff for two-block Merkle–Damgård collisions?

We resolve this question completely in the natural setting $M = N$: even allowing arbitrary oracle-dependent S-qubit advice, the known attack is optimal up to logarithmic factors.

Theorem 1.1 (Two-block collisions with quantum advice, informal)**.** *Let $N \geq 2$, $M \geq 1$, $S \geq 0$, and $T \geq 1$. Then,*

$$\mathrm{Adv}^{\mathsf{MD\text{-}2col\text{-}q}}_{N,M}(S, T) = \widetilde{O}\Big(\min\{1, (ST^2 + T^3)/N\}\Big), \tag{1}$$

where the tilde hides factors polynomial in $\log N$.

Together with the quantum attack of [GLLZ21], Theorem 1.1 gives a tight characterization of the two-block problem, up to logarithmic factors. In particular, it closes the factor-T gap between the previously known ST^3/N security bound and the ST^2/N preprocessing attack. Moreover, whereas the previous ST^3/N upper bound applied only to classical advice, *our result allows arbitrary oracle-dependent quantum advice.*

The main technical novelty is that our compressed-oracle analysis is sensitive to the *structure of the collision.* We view the recorded oracle database as a directed graph and identify the configurations that can be completed to a two-block collision from the subsequently sampled salt. We then bound

Table 1: Quantum bounds for short Merkle–Damgård collisions. All bounds suppress logarithmic factors.

Length	Previous upper bound	Our upper bound	Best known attack	Tight?
$B = 2$	$\frac{ST^3}{N}$	$\frac{ST^2+T^3}{N}$	$\frac{ST^2+T^3}{N}$	✓
$2 < B \le T$	$\frac{ST^3}{N}$	$\frac{BST^2+T^3}{N}$ †	$\frac{ST^2+T^3}{N}$	✗

The previous upper bound [GLLZ21] holds for classical advice, whereas ours allows arbitrary oracle-dependent quantum advice. The ✓ denotes tightness up to logarithmic factors for $M \ge N$. †The general-B bound holds when $(2S+1)(T+2B)^3 \le N$.

how quickly quantum queries can build amplitude on such configurations. This is precisely the type of structural reasoning that underlies the classical short-collision results, but that was missing from previous quantum analyses.

General bounded-length collisions. The structural difficulty becomes substantially richer once $B > 2$. For two blocks, the useful structure already present in the oracle database is essentially local: once the online computation reaches a previously recorded state, only one further recorded transition is relevant. For larger B, reaching a single state may instead make an entire previously recorded path useful. A successful analysis must therefore control not merely individual recorded collisions, but the collection of short paths already present in the compressed-oracle database and the ways in which the online computation can connect to them.

We develop such an analysis and obtain the first quantum security bound for bounded-length MD collisions whose preprocessing term improves as the collision length B decreases. In the sparse regime specified below, the preprocessing contribution becomes ST^2B/N, rather than the previously known ST^3/N term [GLLZ21]. Thus, when $B \ll T$, our bound improves the preprocessing term by a factor of roughly T/B.

Theorem 1.2 (General-B collisions with quantum advice, informal)**.** *Let $N \ge 2$, $M \ge 1$, $S \ge 0$, $T \ge 1$, and $B > 2$ with $B \le T$. If $(2S+1)(T+2B)^3 \le N$, then*

$$\mathrm{Adv}^{\mathsf{MD}\text{-}B\mathsf{col}\text{-}\mathsf{q}}_{N,M}(S,T) = \widetilde{O}\left(\min\left\{1, \frac{ST^2B+T^3}{N}\right\}\right). \tag{2}$$

Tightness and the ST^2B conjecture. The bound of Theorem 1.2 is not known to be tight for $B > 2$. Motivated by the sharp two-block result, we formulate the following conjecture.

Conjecture 1.3 (ST^2B conjecture). For every $2 \le B \le T$, in the regime $T^3 = o(N)$, the constant-success time–space tradeoff for B-block Merkle–Damgård collisions (when $M = \Omega(N)$) is $ST^2B = \widetilde{\Theta}(N)$.

The restriction $T^3 = o(N)$ isolates the regime in which ordinary quantum collision finding, without preprocessing, has subconstant success. For $B = 2$, the upper bound of Theorem 1.1, together with the attack of [GLLZ21], establishes the ST^2B conjecture up to logarithmic factors. For $B > 2$, the optimal dependence on B, S, and T remains open. In the parameter regime of Theorem 1.2, our upper bound establishes the security direction of the conjecture: when $T^3 = o(N)$, constant success requires $ST^2B = \widetilde{\Omega}(N)$. We do not, however, obtain this upper bound in full generality, nor do we give a matching general-B attack.

Finally, since $B \le T$, the condition $(2S+1)(T+2B)^3 \le N$ is, up to constant factors, essentially $(S+1)T^3 \lesssim N$. This is precisely the sparse regime in which the previous ST^3/N preprocessing bound is informative. The condition is needed only for the clean statement of Theorem 1.2: in Section 5 we prove a more general bound that remains valid outside this regime.

1.1 Related Works

Nayebi, Aaronson, Belovs, and Trevisan [NABT15] showed an $ST^2 = \widetilde{\Omega}(N)$ lower bound for inverting random permutations with classical advice. Later, Chung, Guo, Liu, and Qian [CGLQ20] developed a general multi-instance framework together with the compressed-oracle technique of Zhandry [Zha19] and showed that an (S,T)-quantum-adversary with quantum advice can invert a random function with probability at most $\widetilde{O}\Big(\left(ST+T^2\right)/N\big)^{1/3}\Big)$. For classical advice they obtained the optimal success bound $\widetilde{\Theta}\left(\left(ST+T^2\right)/N\right)$. More recently, Liu [Liu23], using the alternating-measurement framework of Marriott-Watrous [MW05], removed the cube-root loss in the success probability for quantum advice as well, proving that any adversary with S-qubit advice and T quantum queries inverts with probability at most $\widetilde{\Theta}\left(\left(ST+T^2\right)/N\right)$.

Time-space tradeoffs for quantum collision finding have also been studied in the bounded-workspace model: Hamoudi and Magniez [HM23], for example, proved that finding K disjoint collision pairs with S qubits of working memory and T queries requires $T^3S = \Omega(K^3N)$. This model is different from the auxiliary-input model considered here as it bounds the algorithm's live workspace rather than allowing an unbounded preprocessing phase that produces S bits or qubits of advice.

Paper organization. Section 2 gives a detailed overview of our proof techniques. Section 3 introduces the collision game, the advice-removal reduction, and the compressed-oracle framework. Section 4 proves the tight two-block upper bound, while Section 5 gives the general-B analysis.

2 Technical Overview

Databases and compressed oracles. For classical queries to a uniformly random function $h : [N] \times [M] \to [N]$, it is useful to generate the random function only as needed. On a previously unseen input (s,m), sample a uniform output $y \in [N]$ and record it; on a repeated input, return the recorded output. This is the usual lazy-sampling implementation of a random oracle. Its record is a *database* $D : [N] \times [M] \longrightarrow [N] \cup \{\perp\}$, where $D(s,m) = \perp$ means that no value is recorded at (s,m). We call (s,m) a *coordinate*, write $|D|$ for the number of coordinates with a recorded value, and call the coordinates with first input s the *row* s.

A quantum query can involve a superposition of inputs. Zhandry's compressed-oracle framework [Zha19], in the formulation of Chung et al. [CFHL21], instead gives an equivalent representation of random-oracle access with an additional quantum database register D. For a computation initialized independently of h, this register starts in the empty-database state. The adversary has an input register X, response register Y, workspace register W, and the register D belongs to the oracle and is inaccessible to the adversary. We assume without loss of generality that the oracle is a phase oracle. Each oracle query is implemented by a unitary O acting jointly on the registers X, Y, and D. For each basis query input (s,m), the query changes only the database entry at (s,m), possibly inserting, modifying, or erasing it. Consequently, after q queries starting from the empty database, every database in the support has at most q recorded entries.

After purifying the computation, write its normalized joint state as $|\Psi\rangle = \sum_D |D\rangle_{\mathsf{D}}|\psi_D\rangle$, where $|\psi_D\rangle$ contains all remaining registers. We use the following properties.

1. *Database properties and weighted norms.* Let $\mathcal{C}$ denote the set of databases that satisfy some property, and let $\Pi_{\mathcal{C}}$ project onto the corresponding database basis states. Then

$$\|\Pi_{\mathcal{C}}|\Psi\rangle\|^2 = \sum_{D\in\mathcal{C}} \||\psi_D\rangle\|^2.$$

   This is the probability that measuring the database yields an element of $\mathcal{C}$.

2. *Counting answers bounds one-query progress.* Fix the salt, and let $\mathcal{C}$ be the set of databases containing a recorded collision. For a database $D \notin \mathcal{C}$ and a query coordinate $x = (s, m)$, write $D[x \leftarrow y]$ for the database obtained by replacing the entry at x by y. Define

$$Y_{D,x} := \{y \in [N] : D[x \leftarrow y] \in \mathcal{C}\}.$$

   Thus $Y_{D,x}$ contains exactly the values that would create a recorded collision if written at the queried coordinate. Suppose that $|Y_{D,x}| \le z$ for every collision-free database D and coordinate x.

   Let O denote the compressed-oracle query unitary. Leaving an entry unchanged cannot create a collision, and erasing an entry cannot create one either. Hence every transition from a collision-free database to a collision-containing database writes a finite value belonging to $Y_{D,x}$. We will often use the following inequality which can be derived from the definition of the compressed oracle unitary:

$$\begin{aligned}\|\Pi_{\mathcal{C}} \cdot \mathsf{O}|\Psi\rangle\| &\le \|\Pi_{\mathcal{C}} \cdot \mathsf{O} \cdot \Pi_{\mathcal{C}}|\Psi\rangle\| + \|\Pi_{\mathcal{C}} \cdot \mathsf{O} \cdot (\mathsf{I} - \Pi_{\mathcal{C}})|\Psi\rangle\| \\ &\le \|\Pi_{\mathcal{C}}|\Psi\rangle\| + 4\sqrt{\frac{z}{N}}.\end{aligned}$$

   Since the adversary's operations between queries leave the database register unchanged, we can iterate this inequality. Starting with no recorded collision, if the corresponding counting bound at query j is z_j, then after t queries,

$$\|\Pi_{\mathcal{C}}|\Psi_t\rangle\|^2 \le \min\left\{1,\ \frac{16}{N}\left(\sum_{j=1}^{t}\sqrt{z_j}\right)^2\right\}.$$

   This is how our combinatorial bounds on possible oracle values become bounds on the weight of collision-containing databases.

3. *Conditioning on a preprocessing event.* Suppose an event $\mathcal{W}$, read from the adversary's workspace, has probability $p > 0$. If the unconditioned state has norm at most ε on databases in $\mathcal{C}$, the normalized state conditioned on $\mathcal{W}$ has norm at most $\varepsilon/\sqrt{p}$ there. Indeed, the event projector commutes with $\Pi_{\mathcal{C}}$, and conditioning renormalizes the retained state by $1/\sqrt{p}$.

A HIGH-LEVEL OUTLINE OF OUR APPROACH. We study short collisions in salted Merkle-Damgård hashing against non-uniform quantum adversaries. The compression function is modeled as random. Before the salt is chosen, the adversary receives S-qubit advice that may depend arbitrarily on the compression function; it then learns a fresh uniformly random salt and makes its T online queries. Its goal is to output two distinct nonempty messages of at most B blocks that collide when hashing from that salt. The main challenge is that such advice may encode the outcome of unbounded preprocessing. Consequently, it cannot in general be modeled as the transcript of, or replaced by, S ordinary oracle queries made before the salt is revealed. Our proof proceeds in three steps.

1. We first apply the advice-removal framework of Liu [Liu23] and Dong–Liu–Wu [DLW24] to replace the arbitrary quantum advice by a bounded-query preprocessing procedure, at the cost of conditioning on a preprocessing event of sufficiently large probability.
2. We next analyze an arbitrary adversary in a simpler model called the *offline-online query model*. In this model, the adversary starts independently of h and first makes a bounded number of *offline* queries before the salt is known; a fresh uniform salt is then sampled; and the adversary makes its remaining *online* queries after seeing the salt (and the residual state left by the offline adversary).
3. Finally, we show that this offline-online model analysis remains valid after conditioning on the preprocessing event introduced by the advice-removal step. This lifts the bound for bounded-query preprocessing back to the original adversary with arbitrary quantum advice.

The main technical work is in the second step. Its key simplification is to separate the Merkle-Damgård-specific combinatorics from the quantum analysis. Fix a compressed-oracle database containing no recorded collision and a query input. We first ask a purely combinatorial question: among all possible answers to this query, how many would cause the resulting database to contain a short Merkle-Damgård collision? The core combinatorial argument bounds this number in terms of database statistics that we control during the computation. Standard compressed-oracle estimates [Zha19, CFHL21] then translate this cardinality bound into a bound on how much amplitude a single quantum query can place on transitions into the collision event. Applying this bound to every database in the adversary's superposition and accumulating the resulting growth over its successive queries gives the desired quantum bound. Thus, essentially all of the problem-specific work is carried out on a fixed classical database; the quantum argument lifts these pointwise bounds to the adversary's full state.

From quantum advice to a conditioned preprocessing game. As in prior work, our first step is to replace oracle-dependent quantum advice by a simpler two-stage experiment that is more amenable to analysis. A natural starting point is the work of Liu [Liu23]. Liu considers a broad class of games in the QAI-QROM, specified by two deterministic oracle algorithms Samp^H and Verify^H. Let H be a uniformly random function mapping $[N] \times [M] \to [N]$ where $M \geq N$. Given uniformly random coins r, the sampling algorithm generates a challenge ch ; the online adversary receives ch together with the non-uniform (quantum) advice corresponding to H, makes at most T (quantum) queries to H, and outputs an answer ans; finally, $\mathsf{Verify}^H(r, \mathsf{ans})$ decides whether the adversary wins.

For this general class of games, Liu [Liu23] showed, using the alternating-measurement framework, how to reduce security against S qubits of oracle-dependent advice to security in the bit-fixing quantum random oracle model (P-BF-QROM). In the P-BF-QROM, a preprocessing algorithm first makes at most P queries to H, postselects on an event $\mathcal{W}$, and leaves a residual quantum state ρ; the ordinary online game is then executed relative to the resulting conditioned oracle distribution, with the online adversary receiving the state ρ.

Analysing the 2-block salted Merkle-Damgård collision game in the P-BF-QROM. The 2-block salted Merkle-Damgård collision game fits Liu's framework. Consequently, to obtain our desired bound of $\widetilde{O}\left(\left(ST^2 + T^3\right)/N\right)$ for 2-block salted MD in QAI-QROM, it suffices to show that any offline algorithm that makes at most $P = 2S(T+4)$ queries, any positive-probability preprocessing event $\mathcal{W}$, and any T-query adversary $\mathcal{A}$, the probability that $\mathcal{A}$ succeeds in finding a 2-block collision for a fresh salt, conditioned on the event $\mathcal{W}$, is at most $\tilde{O}\left(\left(PT + T^3\right)/N\right)$.

Unfortunately, the above is false because the P-BF-QROM places no restriction on how unlikely the conditioning event $\mathcal{W}$ may be. [3] Fix $L = P/2$ distinct strings $s_1, \ldots, s_L$ and distinct pairs $u_i \neq v_i$ for all $i \leq L$. During preprocessing, query $H(s_i, u_i), H(s_i, v_i)$, and condition on the event that $H(s_i, u_i) = H(s_i, v_i)$ for all $i \in [L]$. This event happens with probability $N^{-\Theta(P)}$. Conditioned on this event, however, a T-query adversary can find a 2-block MD-collision with probability $\Omega\left(\min\left\{1, (PT^2/N)\right\}\right)$ as follows. After receiving a fresh salt a, the online adversary uses Grover search to find m such that $H(a, m) \in \{s_1, \ldots, s_L\}$.

Our modified model. The problem is precisely that conditioning on the extremely unlikely event W allows preprocessing to plant a lot of useful structure in the random oracle. This motivates a thresholded version of the bit-fixing model. The experiment is identical to the P-BF-QROM, except that we only require a conditional security guarantee for preprocessing events satisfying $\Pr[\mathcal{W}] \geq N^{-2S}$. We show that Liu's alternating-measurement argument can be suitably adapted to this thresholded setting. For completeness, we include a proof of this in Section 3.1.

Why we first analyze the ordinary offline-online game. To control the effect of postselection, we first prove structural bounds for the unconditioned offline-online computation. Let G be the set of databases satisfying the structural conditions needed for the online analysis. As explained before, conditioning on $\mathcal{W}$, with $p := \Pr[\mathcal{W}]$, increases an unconditioned norm bound ε outside G to at most $\varepsilon/\sqrt{p}$. We therefore establish sufficiently strong tail bounds at each relevant query time to absorb this loss.

We also need the fresh salt a to have an empty database row at the end of preprocessing. Every basis database produced by Q offline queries has at most Q nonempty rows. Since a is sampled uniformly and independently after conditioning, the weight on databases with a nonempty row at a is at most Q/N. Removing this component costs at most $\sqrt{Q/N}$ in norm, without a $1/\sqrt{p}$ loss. Thus, up to these errors, the online analysis starts from a state supported on databases in G with an empty salt row.

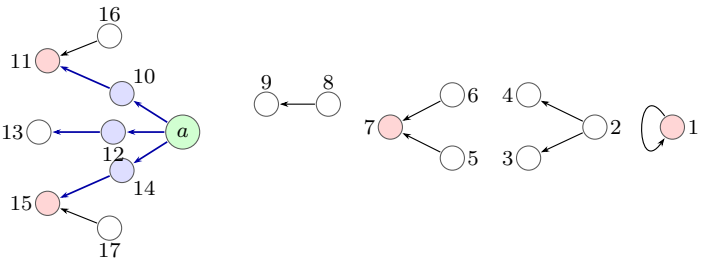
Finally, we evaluate the two output messages using at most $2B$ additional queries. A standard recording bound relates successful verification to the weight on databases containing a recorded collision, with a small additive error in norm. These reductions are common to both proofs; the two-block and general-B cases differ in their choice of G and their bounds on the growth of recorded collisions.

2.1 Two-block collisions

We view a database D as a directed labeled multigraph on $[N]$, where each edge has a label from $[M]$. Whenever the database contains $D(s, m) = y$, we draw a directed edge $s \xrightarrow{m} y$. Thus, evaluating the Merkle-Damgård hash on message $\widetilde{m} = (m_1, \ldots, m_k)$ $(k \leq B)$ with salt a traces a directed walk starting at a (and the edges on this path are labeled using $\widetilde{m}$). An edge from a vertex to itself is a *self-loop*, and two recorded evaluations have the same output exactly when their corresponding edges have the same endpoint. For two-block messages, we can describe very explicitly what a recorded collision must look like. This allows a sharper analysis than the one we use for general B.

The classical picture. Fix a database D and a salt a. A message of at most two blocks can only use compression-function evaluations whose first input is either a itself or a value obtained by one compression-function evaluation step from a. Thus, once a is fixed, only these first inputs can

[3] A classical analogue of this barrier was shown by [AGL22, AGL24].



$$R_a(D) = \{a, 10, 12, 14\}, \qquad J(D) = 1, \qquad K_a(D) = 8, \qquad \Gamma(D) = 20.$$

Figure 1: A sample compressed-oracle database D. Blue vertices are exactly the rows in $R_a(D)$, and blue edges are the relevant edges, whose sources lie in $R_a(D)$. The self-loop at 1 gives $J(D) = 1$. For $K_a(D)$, the outputs $10, 12, 13, 14$ contribute one each, while 11 and 15 contribute two each. Finally, $7, 11, 15$ are exactly the outputs of multiplicity two, giving $\Gamma(D) = 8 \cdot 1^2 + 3 \cdot 2^2 = 20$.

participate in a two-block computation. A recorded two-block collision can then arise in one of two ways:

(i) among the recorded evaluations starting from one of these values there is a self-loop; or

(ii) two distinct recorded evaluations that can occur in such computations have the same output.

The subtlety comes from the offline phase. Suppose an online query (a, m) returns y. The value y can now occur after the first block of a message. Consequently, offline queries of the form (y, m'), which were made before y was known to matter, may now complete a collision. This can happen in two ways. The recorded evaluations starting from y may already contain a self-loop or two evaluations with the same output. Alternatively, one of their outputs may equal the output of another recorded evaluation that can already occur in a two-block computation starting from a. We keep track of exactly the information needed to control these possibilities using three quantities, denoted by J, K_a, and Γ.

- J counts values x for which the offline database already contains a self-loop starting from x, or two distinct evaluations starting from x with the same output; formally

$$J(D) = \left|\{s : \exists m,\ D(s, m) = s,\ \text{or } \exists m, m', m \neq m',\ D(s, m) = D(s, m') \neq \perp\}\right|$$

- K_a counts the number of recorded pairs of edges with the same end-point, where the first edge is a relevant edge w.r.t. a (the two edges may coincide). An edge is relevant when its source is either a or a node at distance 1 from a. Let R_a denote the set of all nodes at distance at most 1 from a. Formally

$$\begin{aligned} K_a(D) &= \sum_{z \in [N]} n_z(D) r_z(D; a) \ \text{ where} \\ n_z(D) &:= \#\text{ of edges in } D \text{ with target } z, \\ r_z(D; a) &:= \#\text{ of edges with target } z \text{ that are relevant w.r.t. } a \end{aligned}$$

- Γ is the sum of the squared numbers of recorded evaluations producing each possible output. Formally

$$\Gamma(D) = \sum_{z \in [N]} n_z(D)^2$$

The role of Γ is easiest to see when an online query (a, m) returns a new value y. All evaluations recorded during the offline phase with first input y can now participate in a two-block computation from a. Consider one such evaluation with output z. It can create a new pair counted by K_a with every other relevant recorded evaluation having the same output z. Thus the increase in K_a is controlled by how many recorded evaluations share each output. If n_z many evaluations in the database have output z, then there are at most n_z choices for each side of such a pair, and hence at most n_z^2 pairs associated with output z. Summing over all possible outputs gives the bound $\sum_z n_z^2 = \Gamma$. A toy example in Figure 1 illustrates the definitions of J, K_a, Γ.

Controlling the offline database. We need strong bounds on two properties of the database produced during the offline phase. First, J should be small: there should be few values s for which the recorded evaluations starting from s already contain a self-loop or two evaluations with the same output. Second, we need to control the quantity Γ defined above.

For both quantities, a bound on the expectation is not enough. Recall that the advice-removal step later conditions on an event $\mathcal{W}$ whose probability may be as small as N^{-2S}. Conditioning can increase the norm of any part of the state by a factor $1/\sqrt{\Pr[\mathcal{W}]}$. We therefore need the norm on databases with large J or large Γ to decrease much faster than the polynomial tail obtained from an expectation bound and Markov's inequality. This is why we prove exponential tail bounds.

We first consider J. Suppose there are already d recorded evaluations with first input s. A new query with the same first input has at most $d+1$ answers that can create one of the patterns counted by J: the answer can equal s, creating a self-loop, or it can equal one of the d outputs already recorded for that first input. A generic compressed-oracle argument would upper bound d by the total number of previous queries, which is too coarse for the two-block theorem.

Instead, we charge each first input gradually as evaluations are recorded there. Fix a threshold L. Before a self-loop or repeated output appears, a first input with $d < L$ recorded evaluations receives weight d/L. Once it contains one of these patterns, or once d reaches L, its weight is one. Let $F(D)$ be the sum of these weights over all first inputs. Since every value counted by J contributes weight 1 to F, we have $J(D) \le F(D)$.

This choice gives a sharp one-query bound. Recording an additional answer that does not create one of the above patterns increases F by only $1/L$. A larger jump can occur only when the answer is one of at most L values, and the compressed-oracle estimate contributes order $\sqrt{L/N}$ for these answers. Balancing $\frac{1}{L}$ and $\sqrt{\frac{L}{N}}$ gives L of order $N^{1/3}$.

To turn this one-query estimate into the strong tail that we need, we track an exponential moment of the total weight F, rather than only its expectation. More concretely, we track $\sum_D e^{F(D)} \||\psi_D\rangle\|^2$ for a state $|\psi\rangle = \sum_D |D\rangle |\psi_D\rangle$ as $|\psi\rangle$ evolves with queries. The one-query bound shows that this exponential moment grows slowly with each offline query. Iterating the bound over all offline queries and using $J \le F$ then gives a bound on the amplitude of databases with $J \ge k$ that is exponentially decreasing in k.

The bound on Γ uses a different potential. Define

$$Z(D) := \sum_z n_z(n_z - 1) = \Gamma(D) - |D|.$$

If a new query receives output z, then Z increases by $2n_z$. These increments can be large when some output already appears many times, so we do not track Z directly. After q queries, we instead consider

$$g_q(D) := \sqrt{q + Z(D)}.$$

Since a q-query database has at most q recorded entries, $\Gamma(D) \le g_q(D)^2$.

The reason for taking the square root is that the possible changes in g_q are small in aggregate. Fix the database H outside the queried coordinate and write $b = q + Z(H)$. If the answer to the query is z, the corresponding change is $\delta_z = \sqrt{b + 2n_z} - \sqrt{b}$. Although an individual δ_z can be larger when n_z is large, the changes over all possible answers satisfy $\sum_z \delta_z^2 \le 1$. The key point is that, although some individual changes δ_z may be large, their squared sum over all possible answers is at most one. The standard compressed-oracle one-query estimate therefore gives a uniform bound on the growth of an exponential moment of g_q. Iterating this bound over the offline queries yields a tail bound for $\Gamma \ge \gamma(R, \theta)$ that is exponentially decreasing in θ; here $R \ge q$ is an appropriately chosen threshold parameter.

Thus, before conditioning on $\mathcal{W}$, the offline computation has only very small norm on databases for which either J or Γ is too large. The exponential tails are strong enough to survive the later $1/\sqrt{\Pr[\mathcal{W}]}$ loss from conditioning. After this step, we may restrict the online analysis to databases with both J and Γ below suitable thresholds.

Tracking equal-output matches during the online phase. The quantity K_a need not be small on every individual database appearing in the adversary's quantum state. What we need is that it is small on average with respect to the amplitudes of those databases.

Accordingly, after j online queries we track $\sum_D K_a(D) \left\| |v_{j,D}\rangle \right\|^2$, where $|v_j\rangle := \sum_D |D\rangle |v_{j,D}\rangle$ is the part of the online state retained after projecting away the events already of large Γ or large J. This quantity is the amplitude-weighted average of $K_a(D)$.

To bound its growth in one query, fix the database and query input and vary only the oracle answer. If the first input of the query is not a, changing the answer affects only a small amount of the information counted by K_a. If the query is (a, m) and returns y, all recorded evaluations beginning at y can become usable. Nevertheless, when we sum this possible increase over all values of y, the result is controlled by Γ.

The compressed-oracle one-query bound converts this deterministic statement into a recurrence for $\sum_D K_a(D) \left\| |v_{j,D}\rangle \right\|^2$. Iterating the recurrence shows that the amplitude-weighted amount of equal-output information that can be useful from the salt remains controlled throughout the online phase.

Creating the first collision. Finally, consider a database in which no two-block collision has yet been recorded, immediately before an online query. We count the possible answers that could create the first collision.

There is at most one answer corresponding to the relevant self-loop case. There are also answers that make the computation enter one of the first-input values counted by J. The remaining possibility is an equal-output collision with information already usable from the salt, and the number of such possibilities is controlled by K_a.

Thus the number of collision-causing answers is bounded by a term depending on J together with a term depending on K_a. The compressed-oracle one-query estimate converts this count into a bound on the amplitude that can enter the collision event. Summing over the online queries and using the weighted bound on K_a gives the desired two-block collision bound.

We then apply the conditioning argument above, perform the final message evaluations, and invoke the advice-removal theorem. This gives the result for the original adversary with S qubits of advice and T online queries.

Up to logarithmic factors, the resulting bound has two sources. One is the advantage obtained by storing useful collision information during preprocessing and searching for it after the salt is known. The other is the usual quantum collision-finding advantage available even without advice. The first can already be achieved using classical advice together with Grover search, while the second is achieved by ordinary quantum collision finding [BHT98]. Thus, for two-block messages,

quantum advice gives no asymptotic improvement over classical advice.

Our two-block theorem is the quantum analogue of the classical preprocessing bound of Akshima et al. [ACDW20]. The proof, however, is different. The classical analysis handles the S-bit advice through multi-instance and compression arguments. In our setting, advice removal instead leads to a conditioned bounded-query quantum computation. We then need strong compressed-oracle bounds that survive this conditioning, together with amplitude-weighted control of K_a during the online computation.

2.2 B-block collisions

For messages of up to B blocks, a new difficulty appears. Suppose an online query returns y. For $B = 2$, only previously recorded evaluations starting directly from y can become useful. For larger B, those evaluations may themselves lead to further preprocessing queries, producing a whole recorded path starting from y. A single online answer can therefore make several old compression-function evaluations usable at once.

The general proof must control these short recorded paths.

What we need from the preprocessing database. We first control two features of the recorded graph of a database D. Let $c(D)$ be the maximum number of disjoint pairs of recorded edges with the same output, and let $z_B(D)$ be the maximum number of pairwise edge-disjoint directed cycles of length at most B. Standard compressed-oracle estimates show that, except for small amplitude, both quantities are at most a suitable threshold k. Let us henceforth assume that D has this property.

Simplifying the recorded graph. Fix such a database D and a salt a, and suppose moreover that there are no two distinct recorded paths of length at most B starting from a and ending at the same vertex, where for this part of the analysis we also allow the path of length zero. We can delete at most $3k$ recorded edges so that the remaining graph has two properties:

(i) every vertex has at most one incoming edge; and

(ii) there is no directed cycle of length at most B.

Moreover, the deleted edges can be chosen so that no path starting from a of length at most B is removed. A toy example of this graph simplification is illustrated in Figure 2.

The deletions are chosen so that we do not remove any short path starting from the salt.

To obtain the first property, whenever several edges have the same output we keep at most one of them, giving priority to an edge that lies on a short path already reachable from the salt[4]. Since there are only few repeated outputs, only few edges are removed. We then delete one edge from each remaining short cycle. Because the database is collision-free, these deletions can be chosen without destroying the short paths already reachable from the salt. Similarly, for the second property, we can delete one edge from a cycle, and we can delete an edge that is not part of a short path from a.

Counting by following edges backwards. The advantage of this simpler graph is that a vertex has at most one incoming edge. Hence, if we start from a target vertex and trace edges backwards, there are very few possible short histories; in particular, we can encounter at most $B + 1$ vertices (including the target vertex) while tracing back B steps. The absence of short cycles prevents the same short history from repeatedly returning to vertices already seen.

We use this observation in three ways (formally done in Lemma 5.6).

[4] at most one such edge can exist assuming no short paths starting from a collide

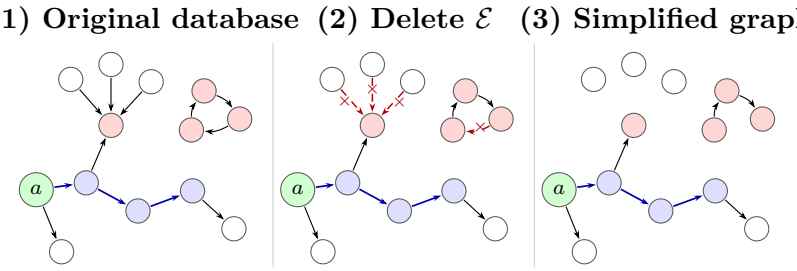


Figure 2: Graph simplification in the general-B proof (the example is for $B = 3$). Blue edges indicate short paths already reachable from the salt a. We delete a small set $\mathcal{E}$ of edges while preserving every such path. The vertex set remains unchanged; the resulting graph has indegree at most one and no short directed cycle.

(i) It bounds the number of vertices from which one can reach, in at most B steps, a vertex already reachable from the salt. In particular, these vertices can only be the reachable vertices themselves plus at most B vertices in the backward traced path from a.

(ii) It bounds the number of vertices from which one can reach the starting point of one of the edges deleted above. This can be simply bounded by $(B + 1)$ times the size of set of sources with deleted edges.

(iii) It bounds the total length of short (at most B length) recorded paths, when summed over their possible starting vertices. This can be bounded by B times the size of the remaining graph.

These elementary graph bounds provide the main combinatorial input to the general-B proof.

Answers whose continuation uses a deleted edge. Consider a single online query for a fixed salt a, query coordinate $x = (s, m)$. Temporarily remove the queried coordinate from the database, call the remaining (fixed) database H and simplify the remaining graph as above. Now consider a possible answer y.

The simplified graph correctly describes any recorded path starting from y unless that path eventually reaches the starting point of an edge that we deleted. Let Y_H be the set of answers for which this happens. Because only few edges were deleted and only paths of bounded length matter, observation (ii) above gives that Y_H is small. The compressed-oracle one-query estimate therefore bounds the amplitude of queries whose answer lies in Y_H.

Now consider an answer $y \notin Y_H$. Every old edge that can be used after reaching y is still present in the simplified graph. If this answer creates a new short collision, then a recorded path starting from y must meet a vertex that was already reachable from the salt. The backward-counting observation in point (i) above therefore controls the number of such answers in terms of the current number of salt-reachable vertices, together with a small additive term. An important point is that this count does not introduce an additional factor of B in front of the number of salt-reachable vertices. This is exactly why we insisted that simplifying the graph preserve every short path that was already reachable from the salt.

Tracking how much becomes reachable. As in the two-block proof, we do not need a worst-case bound for every database in the quantum state. We instead track the amplitude-weighted number of vertices that can currently be reached from the salt within at most B blocks.

Let $V_a(D)$ denote the number of vertices excluding a that are reachable from a within B blocks. After j online queries we consider $\sum_D V_a(D) \, \| |\widehat{v}_{j,D}\rangle \|^2$ where $|\widehat{v}_j\rangle := \sum_D |D\rangle \, |\widehat{v}_{j,D}\rangle$ is the restricted

evolution of the state wherein the databases don't contain B-block collisions, represent simplified graphs, and satisfy $y \notin Y_H$ for the value of the query. The one-query bound again comes from a deterministic calculation on a fixed database. For each possible answer y, let $w(y)$ be the number of previously recorded vertices that would become reachable if the new edge pointed to y. In the simplified graph, the backward-counting observations bound $\sum_y w(y)$. The compressed-oracle one-query estimate then turns this classical bound into a recurrence for the weighted quantity above.

We combine this recurrence with the bounds for answers in Y_H and for answers that create the first collision. Summing over all online queries gives the collision bound for the bounded-query preprocessing model. The conditioning argument and advice-removal theorem then give the result for arbitrary bounded-qubit quantum advice.

In the main parameter regime, the bound has two familiar contributions. One depends linearly on B and captures the possibility that preprocessing stores a useful continuation that is found after the salt is known. The other is the ordinary quantum collision-finding term. For the full range of parameters, our general theorem has an additional loss coming from the bounds on repeated outputs and short cycles.

Why the two-block proof is sharper. The two proofs differ in how much information they retain about the preprocessing database.

For general B, we use only two global bounds: how many repeated outputs and how many short cycles the database contains. This is enough to simplify the graph and control arbitrary short paths, but it loses some information.

For $B = 2$, we can keep track of the database much more precisely. We measure separately how much information has accumulated for each possible first input, and we control the multiplicity of outputs only to the extent needed for K_a. These more precise quantities give a stronger preprocessing bound.

This is why the two-block theorem is not obtained simply by setting $B = 2$ in the general theorem. Its proof uses additional structure that is no longer easy to exploit once a newly revealed value can lead into a path of several preprocessed evaluations.

Classical and quantum parts of the proof. Most of the problem-specific reasoning is classical. For a fixed database, we determine which recorded evaluations can participate in a collision and how many oracle answers can make new recorded information useful. In the two-block proof this leads to the quantities J, K_a, and Γ. In the general-B proof it leads to the graph simplification and the backward-counting bounds.

The quantum part then turns these fixed-database statements into bounds for a quantum adversary. The compressed-oracle one-query estimate converts a bound on the number of relevant oracle answers into an amplitude bound. Weighted norms allow us to average quantities such as K_a or the number of salt-reachable vertices over the superposition of databases. Finally, the conditioning argument and the advice-removal theorem connect this bounded-query analysis to the original model with arbitrary oracle-dependent quantum advice.

This separation also indicates where the gap between the two-block and general-B results comes from. Improving the general theorem would require a more precise classical description of the preprocessing information that can become useful after the salt is revealed, analogous to the finer analysis available when $B = 2$.

3 Preliminaries

Let $\mathbb{N} := \{0, 1, \ldots\}$ be the set of natural numbers and let $\mathbb{N}^+ := \{1, 2, \ldots\}$. For any $N \in \mathbb{N}$, let $[N] := \{0, 1, \ldots, N-1\}$. For $k \in \mathbb{N}$ such that $k \leq N$, let $\binom{[N]}{k}$ denote the set of k-sized subsets of

$[N]$. For a set X, let $|X|$ denote its size and let X^+ denote the set of all nonempty lists of elements of X. Let $x \leftarrow X$ denote choosing $x \in X$ uniformly at random. For an event E, let $\mathbf{1}[E]$ be the indicator function which is 1 when E is true, and 0 otherwise. We let $*$ denote a wildcard element. For example, $(*, z) \in L$ means that L contains an ordered pair whose second element is z; the type of the wildcard will be clear from context. For a bit string s, we use $|s|$ to denote the number of bits in s. When referring to directed graphs in this paper, we mean directed multigraphs with labelled edges. In particular, parallel edges are allowed, but parallel edges have distinct labels. All logarithms in this paper are base 2 unless otherwise stated.

Merkle-Damgård Hashing. For $N, M \in \mathbb{N}^+$, let $h : [N] \times [M] \to [N]$ be a compression function. When discussing tightness up to logarithmic factors, we assume that $M = N^{O(1)}$, so that $\log M = O(\log N)$. This restriction is needed only for the matching attacks, all of our upper bounds hold for arbitrary M. For $a \in [N]$ and a nonempty message $\alpha = (\alpha_1, \ldots, \alpha_\ell) \in [M]^+$, define $h_0 = a$ and $h_i = h(h_{i-1}, \alpha_i)$ for $i = 1, \ldots, \ell$, and set $\mathrm{MD}_h(a, \alpha) := h_\ell$. The elements of $[M]$ are referred to as *blocks*, and a, the first input to MD_h, is referred to as the *salt*. This is the same abstraction of plain Merkle–Damgård hashing used in prior work on time-space tradeoffs for MD collisions [ACDW20, GK22, AGL22, AGL24].

Adversaries with quantum advice. For $S, T \in \mathbb{N}^+$, an (S, T) adversary first receives an arbitrary S-qubit state ρ_h depending on the entire compression function h. There is no bound on the computation or oracle queries used to prepare ρ_h. A uniformly random salt $a \in [N]$, independent of h and the advice, is then revealed. The online adversary makes at most T quantum queries to h and outputs distinct nonempty messages α, α', each of length at most B. It succeeds if $\mathrm{MD}_h(a, \alpha) = \mathrm{MD}_h(a, \alpha')$. For fixed N, M, let $\mathrm{Adv}_{N,M}^{\mathsf{MD}\text{-}B\mathsf{col}\text{-}\mathsf{q}}(S, T)$ denote the supremum of this success probability, averaged over h, a, and the adversary's measurements. We use the following query-bounded offline-online experiment as a means to upper bound $\mathrm{Adv}_{N,M}^{\mathsf{MD}\text{-}B\mathsf{col}\text{-}\mathsf{q}}(S, T)$.

Offline-online adversaries. Fix $B, Q, T \in \mathbb{N}^+$. We consider quantum adversaries starting independently of h and making quantum oracle queries to a uniformly random compression function $h : [N] \times [M] \to [N]$ in two phases. The adversary first makes at most Q queries before learning the salt; we call these the *offline queries*. A uniformly random salt $a \in [N]$ is then revealed to the adversary, after which it makes at most T further queries; we call these the *online queries*. At the end of the online phase, the adversary outputs two distinct messages $\alpha, \alpha' \in [M]^+$, each containing at most B blocks. It succeeds if $\mathrm{MD}_h(a, \alpha) = \mathrm{MD}_h(a, \alpha')$.

3.1 Removing quantum advice

The advice state in our (S, T) model may depend on the entire random function h, so it cannot simply be replaced by a bounded number of ordinary oracle queries. The reduction below uses a different route. It starts from an oracle-independent state, performs a bounded-query preprocessing computation, and postselects on an event $\mathcal{W}$. Conditioned on this event, the residual state can play the role of the original S-qubit advice for the subsequent online computation.

This is why the collision bounds proved later are stated for an arbitrary offline event $\mathcal{W}$: the event is not a "bad event" of the collision analysis, but the successful postselection event introduced by the advice-removal reduction. The reduction guarantees that, whenever the original advice adversary has nontrivial success, the relevant event satisfies $\Pr[\mathcal{W}] \geq N^{-2S}$. It therefore suffices to prove an online bound that holds uniformly after conditioning on every such event.

Theorem 3.1 (Advice removal; following Liu [Liu23] and Dong–Liu–Wu [DLW24])**.** *Let $N, M, B, S, T \in \mathbb{N}^+$ and define $\tau := T + 2B$, $Q := 2S\tau$. Let $h : [N] \times [M] \to [N]$ be a uniformly random function. Suppose that the following holds for every Q-query offline algorithm $\mathcal{P}^h$*

initialized independently of h, every event $\mathcal{W}$ determined during the execution of $\mathcal{P}^h$ satisfying $\Pr[\mathcal{W}] \geq N^{-2S}$, and every τ-query online algorithm $\mathcal{B}^h$ (that devotes its last $2B$ queries for the verification procedure): conditioned on $\mathcal{W}$, if $\mathcal{B}^h$ is given the residual quantum state produced by $\mathcal{P}^h$ together with an independently sampled uniform salt $a \leftarrow [N]$, then

$$\Pr\left[\begin{matrix}\mathcal{B}^h \text{ outputs a valid collision between two distinct messages} \\ \text{of length at most } B \text{ for salt } a\end{matrix} \middle| \mathcal{W}\right] \leq \delta.$$

Then every adversary that receives an arbitrary S-qubit advice state ρ_h depending on h, subsequently receives an independently sampled uniform salt $a \leftarrow [N]$, and makes at most T quantum queries to h, outputs a valid collision between two distinct messages of length at most B for salt a with probability at most

$$\sqrt{2}\max\left\{\delta, \frac{1}{N}\right\}.$$

The proof of this theorem closely follows [Liu23, DLW24], and is included here for completeness.

Proof. Let $\mathcal{A}^h$ be any (S,T) adversary that uses S qubits of advice, makes at most T queries to h, and attempts to produce an MD collision between two distinct messages of length at most B. Without loss of generality, we assume the following about $\mathcal{A}$:

- $\mathcal{A}^h$ operates on a register A containing the salt, Adv which contains the advice, and work registers $\mathsf{Work} = (\mathsf{C}_1, \mathsf{C}_2, \mathsf{W})$ which are initialised to $|0\rangle_{\mathsf{Work}}$.
- Each operation of $\mathcal{A}^h$ is either a query to h, or a unitary applied on the registers $(\mathsf{A}, \mathsf{Adv}, \mathsf{Work})$. Moreover, all operations leave the register A unchanged (that is, every non-oracle unitary is of the form $\sum_a |a\rangle\langle a|_{\mathsf{A}} \otimes (U_a)_{\mathsf{Adv},\mathsf{Work}}$).
- Finally, the registers $\mathsf{C}_1, \mathsf{C}_2$ contain the candidate B-block collision.

Next, we define unitaries $\mathsf{CheckCol}$, 'verification unitary' V^h and projective measurements Π^h, Γ. The unitary $\mathsf{CheckCol}^h$ acts on $(\mathsf{A}, \mathsf{C}_1, \mathsf{C}_2, \mathsf{B}, \mathsf{Z})$ as follows:

$$\mathsf{CheckCol}^h : |a\rangle_{\mathsf{A}} |m\rangle_{\mathsf{C}_1} |m'\rangle_{\mathsf{C}_2} |b\rangle_{\mathsf{B}} |z\rangle_{\mathsf{Z}} \longmapsto |a\rangle_{\mathsf{A}} |m\rangle_{\mathsf{C}_1} |m'\rangle_{\mathsf{C}_2} |b \oplus c\rangle_{\mathsf{B}} \left|z \oplus g^h(a,m,m')\right\rangle_{\mathsf{Z}},$$

where $c = 1$ exactly when (m, m') is a valid collision between two distinct messages of length at most B for a, and $g^h(a,m,m')$ is the intermediate computation needed to compute $\mathrm{MD}_h(a,m)$ and $\mathrm{MD}_h(a,m')$ (it makes at most $2B$ oracle calls to h).

The 'verification unitary' V^h acts on registers $(\mathsf{A}, \mathsf{Adv}, \mathsf{Work}, \mathsf{B}, \mathsf{Z})$ as follows: it applies $\mathcal{A}^h$ on the registers $(\mathsf{A}, \mathsf{Adv}, \mathsf{Work})$, and then applies the unitary $\mathsf{CheckCol}^h$ on the registers $(\mathsf{A}, \mathsf{C}_1, \mathsf{C}_2, \mathsf{B}, \mathsf{Z})$. Note that since $\mathcal{A}^h$ makes at most T queries to h, V^h makes at most $T + 2B$ queries to h.

The two-outcome projective measurement $\Pi^h = (\Pi^h_{\mathsf{acc}}, \mathsf{I} - \Pi^h_{\mathsf{acc}})$ on registers $(\mathsf{A}, \mathsf{Adv}, \mathsf{Work}, \mathsf{B}, \mathsf{Z})$ works as follows:

1. apply V^h on registers $(\mathsf{A}, \mathsf{Adv}, \mathsf{Work}, \mathsf{B}, \mathsf{Z})$
2. measure the register B in the computational basis.
3. apply $\left(V^h\right)^\dagger$ on registers $(\mathsf{A}, \mathsf{Adv}, \mathsf{Work}, \mathsf{B}, \mathsf{Z})$.

Finally, the two-outcome projective measurement $\Gamma = (\Gamma_{\mathsf{acc}}, \mathsf{I} - \Gamma_{\mathsf{acc}})$ on registers $(\mathsf{A}, \mathsf{Work}, \mathsf{B}, \mathsf{Z})$ is defined as follows:

$$|\mathrm{unif}\rangle_{\mathsf{A}} = \frac{1}{\sqrt{N}} \sum_{a \in [N]} |a\rangle \qquad \Gamma_{\mathsf{acc}} = |\mathrm{unif}\rangle\langle\mathrm{unif}|_{\mathsf{A}} \otimes |0\rangle\langle 0|_{\mathsf{Work}} \otimes |0\rangle\langle 0|_{\mathsf{B}} \otimes |0\rangle\langle 0|_{\mathsf{Z}}.$$

We also define the following positive semidefinite matrix which will be useful for our analysis:

$$M^h := \Big(\langle \mathrm{unif}|_{\mathsf{A}} \otimes \mathsf{I}_{\mathsf{Adv}} \otimes \langle 0|_{\mathsf{Work,B,Z}} \Big) \Pi^h_{\mathsf{acc}} \Big(|\mathrm{unif}\rangle_{\mathsf{A}} \otimes \mathsf{I}_{\mathsf{Adv}} \otimes |0\rangle_{\mathsf{Work,B,Z}} \Big). \tag{3}$$

Offline adversary $\mathcal{P}$ and the postselection event $\mathcal{W}$. The event $\mathcal{W}$ below is the postselection event anticipated in Theorem 3.1. Its purpose is to manufacture, from an oracle-independent initial state using only boundedly many queries, a residual state that can replace the original oracle-dependent advice. It should not be confused with any "bad event" used later in the collision analysis.

The offline algorithm $\mathcal{P}^h$ and event $\mathcal{W}$ are defined as follows.

P1. Prepare $|\mathrm{unif}\rangle\langle\mathrm{unif}|_{\mathsf{A}} \otimes \left(\mathsf{I}/2^S\right)_{\mathsf{Adv}} \otimes |0\rangle\langle 0|_{\mathsf{Work}} \otimes |0\rangle\langle 0|_{\mathsf{B}} \otimes |0\rangle\langle 0|_{\mathsf{Z}}$, this initial state is completely independent of h.

P2. For $j = 1, \ldots, S$, perform the following two measurements in order:

(a) perform projective measurement $\Pi^h = \left(\Pi^h_{\mathsf{acc}}, \mathsf{I} - \Pi^h_{\mathsf{acc}}\right)$

(b) perform projective measurement $\Gamma = (\Gamma_{\mathsf{acc}}, \mathsf{I} - \Gamma_{\mathsf{acc}})$

The event $\mathcal{W}$ is defined as follows

$$\mathcal{W} := \{\text{all } 2S \text{ measurements in Step } \textbf{P2.} \text{ return "accept"}\}.$$

Conditioned on the event $\mathcal{W}$, the state is $|\mathrm{unif}\rangle\langle\mathrm{unif}|_{\mathsf{A}} \otimes (\rho_{h,\mathcal{W}})_{\mathsf{Adv}} \otimes |0\rangle\langle 0|_{\mathsf{Work}} \otimes |0\rangle\langle 0|_{\mathsf{B}} \otimes |0\rangle\langle 0|_{\mathsf{Z}}$. The offline adversary passes the state $(\rho_{h,\mathcal{W}})_{\mathsf{Adv}}$ to the online adversary. Each Π^h measurement results in at most $2(T + 2B)$ calls to h, hence the offline adversary makes at most $2S(T + 2B)$ calls to h.

Online adversary $\mathcal{B}^h$. The online adversary $\mathcal{B}^h$ receives a uniformly random salt a from the challenger and the state $(\rho_{h,\mathcal{W}})_{\mathsf{Adv}}$ from the offline-phase adversary. It initializes the register A as $|a\rangle\langle a|$, registers $\mathsf{Work}, \mathsf{B}, \mathsf{Z}$ as $|0\rangle\langle 0|_{\mathsf{Work,B,Z}}$. It simply runs the *forward* verified computation:

B1. Apply V^h on registers $(\mathsf{A}, \mathsf{Adv}, \mathsf{Work}, \mathsf{B}, \mathsf{Z})$.

B2. measure the B register; if the output is 1, measure the registers C_1 and C_2.

The query complexity of $\mathcal{B}$ is at most $T + 2B$.

Analysis of $(\mathcal{P}, \mathcal{B})$. Throughout this part we fix an arbitrary (S, T) adversary $\mathcal{A}$ normalised as at the beginning of the proof. Let ϵ denote the winning probability. We will show that if $\epsilon > \sqrt{2}/N$,

- $\Pr[\mathcal{W}] \geq N^{-2S}$
- $\Pr\left[\mathcal{B} \text{ outputs a valid collision between two distinct messages of length at most } B \mid \mathcal{W}\right] \geq \epsilon/\sqrt{2}$.

For any hash function h, let ϵ_h denote the probability that $\mathcal{A}^h$, with advice ρ_h, outputs a valid collision between two distinct messages of length at most B (for random salt). We are given that $\epsilon := \mathbb{E}_h[\epsilon_h]$.

Claim 3.2. *For any hash function h,*

$$\epsilon_h = \mathrm{Tr}\left[M^h \rho_h\right].$$

As a result, the largest eigenvalue of M^h is at least ϵ_h.

Proof. Let

$$J : |\phi\rangle_{\mathsf{Adv}} \longmapsto |\mathrm{unif}\rangle_{\mathsf{A}} |\phi\rangle_{\mathsf{Adv}} |0\rangle_{\mathsf{Work,B,Z}} .$$

By definition, $M^h = J^\dagger \Pi^h_{\mathsf{acc}} J$. The success probability of $\mathcal{A}^h$ with advice ρ_h and a uniform salt is therefore

$$\mathrm{Tr}\Big(\Pi^h_{\mathsf{acc}} J \rho_h J^\dagger\Big) = \mathrm{Tr}(M^h \rho_h).$$

Since $M^h \succeq 0$ and ρ_h is a density matrix, $\mathrm{Tr}(M^h\rho_h) \le \lambda_{\max}(M^h)$. □

Claim 3.3. *Fix any h, and let σ be any density matrix on the S-qubit advice register Adv. Apply the projective measurement Π^h, followed by Γ, to the state*

$$\tau = |\mathrm{unif}\rangle\langle\mathrm{unif}|_{\mathsf{A}} \otimes \sigma_{\mathsf{Adv}} \otimes |0\rangle\langle 0|_{\mathsf{Work,B,Z}} .$$

The probability that both measurements accept is

$$\theta = \mathrm{Tr}\left(M^h \sigma M^h\right) = \mathrm{Tr}\left[\left(M^h\right)^2 \sigma\right].$$

If $\theta > 0$, the state conditioned on both measurements accepting is

$$|\mathrm{unif}\rangle\langle\mathrm{unif}|_{\mathsf{A}} \otimes \left(\frac{M^h \sigma M^h}{\theta}\right)_{\mathsf{Adv}} \otimes |0\rangle\langle 0|_{\mathsf{Work,B,Z}} .$$

Proof. Using the isometry J from Claim 3.2, the initial state is $J\sigma J^\dagger$ and $\Gamma_{\mathsf{acc}} = JJ^\dagger$. Hence the probability that both projective measurements accept is

$$\begin{aligned}\theta &= \mathrm{Tr}\Big(\Gamma_{\mathsf{acc}} \Pi^h_{\mathsf{acc}} J\sigma J^\dagger \Pi^h_{\mathsf{acc}} \Gamma_{\mathsf{acc}}\Big) \\ &= \mathrm{Tr}\Big(J^\dagger \Pi^h_{\mathsf{acc}} J\, \sigma\, J^\dagger \Pi^h_{\mathsf{acc}} J\Big) = \mathrm{Tr}(M^h \sigma M^h).\end{aligned}$$

On this branch the normalized state is

$$J \frac{M^h \sigma M^h}{\theta} J^\dagger,$$

which is the state stated in the claim. □

We can now analyse the offline algorithm $\mathcal{P}$ and the event $\mathcal{W}$. From the above claim, we can conclude the following.

Claim 3.4. *For any h,*

$$\Pr[\mathcal{W} \mid h] = \frac{\mathrm{Tr}\left[\left(M^h\right)^{2S}\right]}{2^S}$$

If $\Pr[\mathcal{W} \mid h] > 0$,

$$\rho_{h,\mathcal{W}} = \frac{\left(M^h\right)^{2S}}{\mathrm{Tr}\left[\left(M^h\right)^{2S}\right]}$$

Recall $\rho_{h,\mathcal{W}}$ is the state received by $\mathcal{B}$, conditioned on h and $\mathcal{W}$.

Proof. Let $d := 2^S$. Initially the advice state is I/d. By Claim 3.3, one accepted Π^h–Γ pair maps an unnormalized advice state σ to $M^h \sigma M^h$. After j accepted pairs, the unnormalized advice state is therefore $(M^h)^{2j}/d$. For $j = S$, its trace is the probability of $\mathcal{W}$, giving

$$\Pr[\mathcal{W} \mid h] = \frac{\mathrm{Tr}[(M^h)^{2S}]}{2^S}.$$

Normalizing gives the displayed expression for $\rho_{h,\mathcal{W}}$. □

Claim 3.5. *Fix any h such that $\Pr[\mathcal{W} \mid h] > 0$, and let E_h be the event that $\mathcal{B}$, on the fresh salt a, outputs a valid collision between two distinct messages of length at most B. Then*

$$\Pr[E_h \mid \mathcal{W}, h] = \mathrm{Tr}[M^h \rho_{h,\mathcal{W}}] = \frac{\mathrm{Tr}[(M^h)^{2S+1}]}{\mathrm{Tr}[(M^h)^{2S}]}.$$

The probability is over the fresh uniformly random salt and the randomness and measurement outcomes of $\mathcal{B}$.

Proof. Conditioned on $\mathcal{W}$ and h, the online algorithm starts with advice state $\rho_{h,\mathcal{W}}$, a fresh uniformly random salt, and zeroed work and verification registers, and then runs V^h. By the definition of M^h in (3), the probability of the event E_h is therefore

$$\Pr[E_h \mid \mathcal{W}, h] = \mathrm{Tr}[M^h \rho_{h,\mathcal{W}}].$$

Substituting Claim 3.4 gives

$$\mathrm{Tr}[M^h \rho_{h,\mathcal{W}}] = \frac{\mathrm{Tr}[(M^h)^{2S+1}]}{\mathrm{Tr}[(M^h)^{2S}]}.$$

□

Claim 3.6. *Suppose $\epsilon > \sqrt{2}/N$. Let* Win *denote the event that $\mathcal{B}$ outputs a valid collision between two distinct messages of length at most B. Then*

$$\Pr[\mathcal{W}] > N^{-2S} \qquad \textit{and} \qquad \Pr[\mathsf{Win} \mid \mathcal{W}] \geq \frac{\epsilon}{\sqrt{2}}.$$

Proof. Let $d := 2^S$, and define

$$t_k := \mathbb{E}_h\Big[\mathrm{Tr}[(M^h)^k]\Big] \qquad \text{for } k \geq 1.$$

By Claim 3.2, the largest eigenvalue of M^h is at least ϵ_h. Since M^h is positive semidefinite,

$$\mathrm{Tr}[(M^h)^{2S}] \geq \epsilon_h^{2S}.$$

Taking expectations and applying Jensen's inequality to the convex function $x \mapsto x^{2S}$ gives

$$t_{2S} \geq \mathbb{E}_h[\epsilon_h^{2S}] \geq \big(\mathbb{E}_h[\epsilon_h]\big)^{2S} = \epsilon^{2S}.$$

Consequently,

$$\begin{aligned}\Pr[\mathcal{W}] = \frac{t_{2S}}{d} &\geq \frac{\epsilon^{2S}}{2^S} \\ &> \frac{(\sqrt{2}/N)^{2S}}{2^S} = N^{-2S}.\end{aligned}$$

For every h with $\Pr[\mathcal{W} \mid h] > 0$, Claims 3.4 and 3.5 give

$$\Pr[\mathcal{W} \wedge \mathsf{Win} \mid h] = \frac{\mathrm{Tr}[(M^h)^{2S+1}]}{d}.$$

The same identity holds when $\Pr[\mathcal{W} \mid h] = 0$. In that case, $\mathrm{Tr}[(M^h)^{2S}] = 0$, and positivity of M^h implies $M^h = 0$, so both sides vanish. Averaging over h therefore yields

$$\Pr[\mathcal{W} \wedge \mathsf{Win}] = \frac{t_{2S+1}}{d}.$$

Thus

$$p := \Pr[\mathsf{Win} \mid \mathcal{W}] = \frac{t_{2S+1}}{t_{2S}}. \tag{4}$$

For each h, let $\lambda_{h,1}, \ldots, \lambda_{h,d}$ be the nonnegative eigenvalues of M^h. Sample h from its original distribution and sample i independently and uniformly from $\{1, \ldots, d\}$.

Applying Jensen's inequality to the random variable $\lambda_{h,i}^{2S}$ and the convex function $x \mapsto x^{(2S+1)/(2S)}$ gives

$$\begin{aligned} \frac{t_{2S+1}}{d} &= \mathbb{E}_{h,i}\left[(\lambda_{h,i}^{2S})^{(2S+1)/(2S)}\right] \\ &\geq \left(\mathbb{E}_{h,i}\left[\lambda_{h,i}^{2S}\right]\right)^{(2S+1)/(2S)} \\ &= \left(\frac{t_{2S}}{d}\right)^{(2S+1)/(2S)}. \end{aligned} \tag{5}$$

Dividing by $t_{2S}/d > 0$ and using (4) and (5), we obtain

$$p \geq \left(\frac{t_{2S}}{d}\right)^{1/(2S)} \geq \left(\frac{\epsilon^{2S}}{2^S}\right)^{1/(2S)} = \frac{\epsilon}{\sqrt{2}}.$$

□

This completes the proof of Theorem 3.1.

□

3.2 Compressed Oracles

We analyze quantum queries to the random compression function using the compressed-oracle framework. In this framework, the information recorded about the oracle is represented by a *database*. A database specifies the oracle values that have been recorded so far; coordinates whose values have not been recorded are marked by $\perp$. We first introduce notation for such databases and then describe how a quantum oracle query acts on them. A database is a function $D : [N] \times [M] \to [N] \cup \{\perp\}$. We refer to an ordered pair in $[N] \times [M]$ as a *coordinate*. For a coordinate $x = (s, m)$, the value $D(x) = \perp$ means that no oracle value is recorded at x. If $D(s, m) = y \in [N]$, then the database records the oracle value $h(s, m) = y$ at that coordinate. Define $\mathrm{Dom}(D) := \{x \in [N] \times [M] : D(x) \neq \perp\}$ and $|D| := |\mathrm{Dom}(D)|$. For $s \in [N]$, the *row* s is the set of coordinates $\{(s, m) : m \in [M]\}$. We also identify a database with a directed edge-labelled multigraph on vertex set $[N]$: every recorded entry $D(s, m) = y$ gives the edge $s \xrightarrow{m} y$. Each recorded coordinate $(s, m) \in \mathrm{Dom}(D)$ therefore corresponds to one edge $s \xrightarrow{m} D(s, m)$. Thus distinct recorded coordinates are regarded as distinct edges, even if they have the same output

vertex. The indegree of a vertex is the number of recorded coordinates whose output is that vertex. For a set E of coordinates, $D \setminus E$ is obtained by replacing $D(x)$ by $\perp$ for every $x \in E$. For a single coordinate x, write $D^{-x} := D \setminus \{x\}$. A *recorded path of length* d from v_0 to v_d is a sequence $v_0, m_1, v_1, \ldots, m_d, v_d$ such that $D(v_{i-1}, m_i) = v_i$ for every $i = 1, \ldots, d$. We allow $d = 0$, in which case the path consists only of its starting vertex. Define

$$\mathrm{Reach}_D^{\le d}(v) := \{w : \text{there is a recorded path in } D \text{ from } v \text{ to } w \text{ of length at most } d\}.$$

For a message $\alpha = (\alpha_1, \ldots, \alpha_\ell) \in [M]^+$, we say that its evaluation from a is *recorded in* D if there exist $v_0, \ldots, v_\ell \in [N]$ with $v_0 = a$ such that $D(v_{i-1}, \alpha_i) = v_i$ for every $i = 1, \ldots, \ell$. When this holds, we write $\mathrm{MD}_a^D(\alpha) := v_\ell$.

Definition 3.7 (Recorded collision)**.** For $a \in [N]$ and $B \in \mathbb{N}^+$, let $\mathcal{C}_{a,B}$ be the set of databases D for which there exist distinct messages $\alpha, \alpha' \in [M]^+$, each containing at most B blocks, such that $\mathrm{MD}_a^D(\alpha)$ and $\mathrm{MD}_a^D(\alpha')$ are both defined and $\mathrm{MD}_a^D(\alpha) = \mathrm{MD}_a^D(\alpha')$.

For the analysis, we represent the uniform salt coherently as $N^{-1/2} \sum_{a \in [N]} |a\rangle_A$. During the offline phase the adversary has no access to A. During the online phase, its operations may depend on the value of the salt but do not modify A; equivalently, every online unitary is block diagonal in the computational basis of A. Since all projectors used in the analysis are also diagonal in this basis, the resulting squared norms are exactly the corresponding probabilities averaged over a uniformly sampled classical salt.

We next extend the database-level definitions above to quantum states. We use A for the salt register and D for the compressed-oracle database register. Any state can therefore be written uniquely as $|\xi\rangle = \sum_{a,D} |a\rangle_A |D\rangle_{\mathsf{D}} |\xi_{a,D}\rangle$, where $|\xi_{a,D}\rangle$ contains all registers other than the salt and database registers. Throughout, by a *projector* we mean an orthogonal projector, i.e., an operator P satisfying $P^2 = P = P^\dagger$. For any set $\mathcal{S}$ of pairs (a, D), define the projector $\Pi_{\mathcal{S}}$ by $\Pi_{\mathcal{S}} |\xi\rangle := \sum_{(a,D) \in \mathcal{S}} |a\rangle_A |D\rangle_{\mathsf{D}} |\xi_{a,D}\rangle$. Thus $\Pi_{\mathcal{S}}$ keeps exactly those basis components for which $(a, D) \in \mathcal{S}$. In particular, define the projector onto states containing a recorded collision of length at most B by

$$\mathsf{Col}^{\le B} := \Pi_{\{(a,D): D \in \mathcal{C}_{a,B}\}} = \sum_a |a\rangle\langle a|_A \otimes \sum_{D \in \mathcal{C}_{a,B}} |D\rangle\langle D|_{\mathsf{D}}.$$

When $B = 2$, we denote this projector by $\mathsf{Col}^{\le 2}$. For any two sets $\mathcal{S}$ and $\mathcal{T}$ of pairs (a, D), $\Pi_{\mathcal{S}}\Pi_{\mathcal{T}} = \Pi_{\mathcal{S} \cap \mathcal{T}} = \Pi_{\mathcal{T}}\Pi_{\mathcal{S}}$. Hence any two such projectors commute. Further, for $\mathcal{S} \subseteq \mathcal{T}$, we have $\Pi_{\mathcal{T}} - \Pi_{\mathcal{S}} = \Pi_{\mathcal{T} \setminus \mathcal{S}}$, which is also a projector.

For a nonnegative function $g(a, D)$ (sometimes written $g_a(D)$), we use the shorthand $\| \sqrt{g}\, |\xi\rangle \|^2 := \sum_{a,D} g(a, D) \| |\xi_{a,D}\rangle \|^2$. Thus this notation measures the squared norm of $|\xi\rangle$ with each (a, D) component weighted by $g(a, D)$. The proofs in the paper track how the weight on various sets of databases changes under oracle queries. For this, we need the action of a single compressed-oracle query.

The action of a compressed-oracle query. We use the standard compressed-oracle representation of Zhandry [Zha19], in the formulation of Chung et al. [CFHL21]. We only need the action of a query on the value stored at the queried coordinate. We consider the *compressed phase oracle*. The query has an input register X with basis $\{|x\rangle_X : x \in [N] \times [M]\}$ and a register Y with basis $\{|u\rangle_Y : u \in \mathbb{Z}_N\}$. Thus $x = (s, m)$ specifies the coordinate being queried. Let $\omega := e^{2\pi i/N}$. Identifying each $y \in [N]$ with the corresponding element of $\mathbb{Z}_N$, a phase query to h acts as

$$|x\rangle_X |u\rangle_Y \longmapsto \omega^{uh(x)} |x\rangle_X |u\rangle_Y.$$

We now describe the corresponding action on the database. Fix a coordinate $x = (s, m)$. Once all database entries other than $D(x)$ are fixed, the possible values of $D(x)$ are the elements of $[N] \cup \{\perp\}$. We write $|\perp\rangle_x$ when $D(x) = \perp$ and $|y\rangle_x$ when $D(x) = y \in [N]$. When x is clear from context, we omit the subscript. Let $|+\rangle := N^{-1/2} \sum_{y\in[N]} |y\rangle$. Define S to swap $|\perp\rangle$ and $|+\rangle$ and to act as the identity on the subspace orthogonal to both. Define Φ_u by $\Phi_u |\perp\rangle = |\perp\rangle$ and $\Phi_u |y\rangle = \omega^{uy} |y\rangle$. The local action of the compressed phase oracle on the value stored at coordinate x is $R_u := S\Phi_u S$. For $u = 0$, we have $R_0 = I$. For $u \neq 0$,

$$R_u |\perp\rangle = \frac{1}{\sqrt{N}} \sum_{y\in[N]} \omega^{uy} |y\rangle , \tag{6}$$

$$R_u |r\rangle = \frac{\omega^{ur}}{\sqrt{N}} |\perp\rangle + \frac{1 + (N-2)\omega^{ur}}{N} |r\rangle + \sum_{y\neq r} \frac{1 - \omega^{uy} - \omega^{ur}}{N} |y\rangle . \tag{7}$$

Thus, if the value of $D(x)$ before the query is $r \in [N] \cup \{\perp\}$ and its value after the query is $t \in [N] \cup \{\perp\}$, then $\langle t| R_u |r\rangle$ is the amplitude of this transition. We will use only the following two consequences:

$$|\langle y| R_u |\perp\rangle| = \frac{1}{\sqrt{N}} \qquad (y \in [N],\ u \neq 0), \tag{8}$$

$$|\langle y| R_u |r\rangle| \leq \frac{3}{N} \qquad (r, y \in [N],\ y \neq r,\ u \neq 0). \tag{9}$$

The first follows from (6), while the second follows from (7) and $|1-\omega^{uy}-\omega^{ur}| \leq 3$. For $t \in [N]\cup\{\perp\}$, let $D[x \leftarrow t]$ denote the database that agrees with D at every coordinate other than x and has value t at x. Let W denote all registers other than A, X, Y, and the database register. We denote the compressed-oracle query by $\mathcal{O}$. For every basis state of A, X, Y, and the database register, and every state $|\eta\rangle_W$,

$$\mathcal{O} |a\rangle_A |x\rangle_X |u\rangle_Y |D\rangle_{\mathsf{D}} |\eta\rangle_W = \sum_{t\in[N]\cup\{\perp\}} (\langle t| R_u |D(x)\rangle) |a\rangle_A |x\rangle_X |u\rangle_Y |D[x \leftarrow t]\rangle_{\mathsf{D}} |\eta\rangle_W .$$

In particular, a query changes at most the database entry at the queried coordinate x. Starting from the empty database, every basis database appearing after q oracle queries therefore contains at most q recorded coordinates. We next isolate the quantitative consequence of (8)–(9) that will be used throughout the proofs. For an operator C, let $\|C\|$ denote its operator norm. We use the elementary inequality $\|C\| \leq (\sum_{i,j} |C_{i,j}|^2)^{1/2}$. The next lemma formalizes the following observation: if, for every basis database and queried coordinate, at most z possible new oracle values are designated as bad, then the operator consisting of transitions to these values has norm $O(\sqrt{z/N})$.

Lemma 3.8. *Let $\mathcal{P}$ be a set of pairs (a,D) for databases D and $a \in [N]$. For every $a \in [N]$, queried coordinate $x \in [N] \times [M]$, and database D, let $L_{a,x,D} \subseteq [N]$ satisfy $|L_{a,x,D}| \leq z$. Define*

$$\widehat{B} := \sum_{\substack{x\in[N]\times[M],\\ u\in\mathbb{Z}_N,(a,D)\in\mathcal{P}}} \sum_{\substack{y\in L_{a,x,D}\\ y\neq D(x)}} (\langle y| R_u |D(x)\rangle) |a\rangle\langle a|_A \otimes |x\rangle\langle x|_X \otimes |u\rangle\langle u|_Y \otimes |D[x \leftarrow y]\rangle\langle D|_{\mathsf{D}} .$$

Then $\left\|\widehat{B}\right\| \leq 4\sqrt{\frac{z}{N}}$. The same statement holds with an identity operator tensored on any registers that are not modified by the oracle query.

Proof. Fix a, x, u, and the values of all database entries at coordinates other than x. On the subspace spanned by the states $|D[x \leftarrow r]\rangle_{\mathsf{D}}$, the operator $\widehat{B}$ acts on the value stored at x as

$$B = \sum_{r \in [N] \cup \{\perp\}} \sum_{\substack{y \in L_{a,x,D[x \leftarrow r]} \\ y \neq r}} \left(\langle y| R_u |r\rangle\right) |y\rangle \langle r| .$$

If $u = 0$, then $R_u = I$, so $B = 0$. Assume $u \neq 0$. Corresponding to $r = \perp$, there are at most z nonzero entries, each of magnitude $1/\sqrt{N}$ by (8). Hence the Euclidean norm of this column is at most $\sqrt{z/N}$. For each $r \in [N]$, there are at most z entries of the form $\langle y| R_u |r\rangle$ with $y \neq r$. Over all $r \in [N]$, there are therefore at most zN such entries. By (9), each has magnitude at most $3/N$. Hence the square root of the sum of their squared magnitudes is at most $\sqrt{zN \cdot \frac{9}{N^2}} = 3\sqrt{\frac{z}{N}}$. Using the operator-norm bound stated above and then the triangle inequality, we obtain $\|B\| \leq 4\sqrt{z/N}$[5]. Finally, different fixed choices of a, x, u, and the database entries other than $D(x)$ determine mutually orthogonal subspaces, and $\widehat{B}$ does not mix these subspaces. Thus $\widehat{B}$ is block diagonal, with every block having norm at most $4\sqrt{z/N}$. Therefore $\left\|\widehat{B}\right\| \leq 4\sqrt{z/N}$. □

We will repeatedly use Lemma 3.8 in the following form. Let $\mathcal{P} \subseteq \mathcal{P}'$ be sets of pairs (a, D). Suppose that for every $(a, D) \in \mathcal{P}$ and every queried coordinate x, there are at most z values $y \in [N] \setminus \{D(x)\}$ such that $(a, D[x \leftarrow y]) \notin \mathcal{P}'$. Suppose also that $(a, D^{-x}) \in \mathcal{P}'$ for every $(a, D) \in \mathcal{P}$ and every x. Then

$$\|(I - \Pi_{\mathcal{P}'})\mathcal{O}\Pi_{\mathcal{P}}\| \leq 4\sqrt{\frac{z}{N}}. \tag{10}$$

Indeed, for fixed $(a, D) \in \mathcal{P}$ and queried coordinate x, let $L_{a,x,D}$ be the set of values $y \in [N] \setminus \{D(x)\}$ for which $(a, D[x \leftarrow y]) \notin \mathcal{P}'$. By assumption, $|L_{a,x,D}| \leq z$. Since $\mathcal{P} \subseteq \mathcal{P}'$, leaving $D(x)$ unchanged cannot move the pair outside $\mathcal{P}'$. By the second assumption, changing $D(x)$ to $\perp$, which produces D^{-x}, cannot move it outside $\mathcal{P}'$ either. Hence every transition from $\mathcal{P}$ to outside $\mathcal{P}'$ must change $D(x)$ to some value in $L_{a,x,D}$, and Lemma 3.8 gives (10).

3.3 Fresh salts, output verification, and conditioning

The remaining reductions in this section are common to both collision proofs. The two-block and general-B analyses use different structural projectors on the compressed-oracle database, and those projectors are therefore introduced in their respective sections. What they share are three facts. First, the salt is sampled only after the offline phase, so its row is almost always untouched by the offline database. Second, after the adversary outputs two messages, we must evaluate them in order to turn acceptance into a recorded collision in the compressed database. Third, after the advice-removal reduction we are working with a state conditioned on the postselection event $\mathcal{W}$, so we need to understand how a small structural tail behaves under that conditioning.

The salt row is empty after the offline phase. The salt is independent of the offline phase and is hidden from the adversary throughout that phase. After Q offline queries, every basis database contains at most Q recorded coordinates and hence has nonempty entries in at most Q rows. Consequently, when the fresh uniform salt a is sampled, its row is empty except on a component of small norm. We record this fact on its own; it will later be combined with the structural projector appropriate to each collision proof.

For $a \in [N]$, let $\mathcal{E}_a$ be the set of databases satisfying $D(a, m) = \perp$ for every $m \in [M]$. Thus $\mathcal{E}_a$ is the set of databases whose row a is empty. Define $\Pi_{\text{empty}} := \Pi_{\{(a,D):D \in \mathcal{E}_a\}}$. Write the state after Q queries as $|\phi_Q\rangle = \sum_D |D\rangle_{\mathsf{D}} |\eta_D\rangle$, where $|\eta_D\rangle$ contains all registers other than the database register.

[5] In fact, we can obtain $\|B\| \leq \sqrt{10z/N}$ since the cases $r = \perp$ and $r \in [N]$ pertain to orthogonal states.

Lemma 3.9. *At the end of offline phase, before the adversary is given access to the salt register,*

$$\left\| (I - \Pi_{\text{empty}}) \left(\frac{1}{\sqrt{N}} \sum_{a \in [N]} |a\rangle_A |\phi_Q\rangle \right) \right\| \leq \sqrt{\frac{Q}{N}}.$$

Proof. Every basis database D appearing in the support of $|\phi_Q\rangle$ contains at most Q recorded coordinates. Consequently, at most Q values $a \in [N]$ have a nonempty row in D. Therefore

$$\begin{aligned} &\left\| (I - \Pi_{\text{empty}}) \frac{1}{\sqrt{N}} \sum_{a,D} |a\rangle_A |D\rangle_{\mathsf{D}} |\eta_D\rangle \right\|^2 \\ &\quad = \sum_D \frac{|\{a \in [N] : \text{row } a \text{ of } D \text{ is nonempty}\}|}{N} \||\eta_D\rangle\|^2 \leq \frac{Q}{N} \sum_D \||\eta_D\rangle\|^2 = \frac{Q}{N}. \end{aligned}$$

Taking square roots proves the claim. □

Recording the output collision. The online bounds control the norm of states whose compressed database already contains a recorded collision. The adversary, however, ends by outputting two classical candidate messages. We therefore append the reversible evaluations of those messages. The next lemma quantifies the small recording error incurred when translating acceptance of this verification into the recorded-collision projector. The proof of this lemma follows via a similar argument as in Lemma 5 from [Zha19].

Lemma 3.10 (Output recording)**.** *Suppose a computation has produced two distinct nonempty messages α, α', each of length at most B. Verify the collision by evaluating both messages from the sampled salt, writing each oracle answer into a fresh, initially zero register and retaining these registers. Let $L \leq 2B$ be an upper bound on the number of verification queries on every branch, and let $|\psi_{\text{ver}}\rangle$ denote the resulting normalized compressed-oracle state. Then*

$$\sqrt{\Pr[\text{verification accepts}]} \leq \left\| \mathsf{Col}^{\leq B} |\psi_{\text{ver}}\rangle \right\| + \sqrt{\frac{L}{N}}. \tag{11}$$

For $B = 2$, the same statement holds with $\mathsf{Col}^{\leq 2}$ in place of $\mathsf{Col}^{\leq B}$.

Proof. For the analysis, represent the random function h by one register for each input x, containing $h(x)$. Let $|\phi_{\text{ver}}\rangle$ be the state after verification in this representation. Applying S to every oracle register gives the compressed-oracle state:

$$|\psi_{\text{ver}}\rangle = \widehat{S} |\phi_{\text{ver}}\rangle, \qquad \widehat{S} := \bigotimes_x S_x.$$

Let τ consist of the salt, the two messages, and all answers obtained during verification. These values are kept in the work registers, so we can write

$$|\phi_{\text{ver}}\rangle = \sum_\tau |\tau\rangle |\phi_\tau\rangle.$$

Write $\mathcal{A}$ for the set of records τ in which the two message evaluations have the same final output. Then

$$p := \Pr[\text{verification accepts}] = \sum_{\tau \in \mathcal{A}} \||\phi_\tau\rangle\|^2.$$

Fix a record τ with nonzero amplitude. Let $x_1, \ldots, x_d$ be the distinct inputs queried during verification, with answers $y_1, \ldots, y_d$, respectively. Here $d \le L$. Since these are the actual oracle answers, the oracle register at x_i contains y_i for every i. Consequently,

$$|\phi_\tau\rangle = \left(\bigotimes_{i=1}^{d} |y_i\rangle_{x_i}\right) \otimes |\eta_\tau\rangle ,$$

where $|\eta_\tau\rangle$ contains all remaining registers and $\||\eta_\tau\rangle\| = \||\phi_\tau\rangle\|$.

Let Π_τ project onto databases satisfying $D(x_i) = y_i$ for every i. Recall that

$$S|y\rangle = |y\rangle - \frac{1}{N}\sum_{z\in[N]} |z\rangle + \frac{1}{\sqrt{N}}|\perp\rangle , \qquad \langle y| S |y\rangle = 1 - \frac{1}{N}.$$

Thus, applying S and then projecting onto the specified answer contributes a factor $1 - 1/N$ at each of the d queried inputs. On the other oracle registers, S acts unitarily. Therefore,

$$\left\|\Pi_\tau \widehat{S} |\phi_\tau\rangle\right\|^2 = \left(1 - \frac{1}{N}\right)^{2d} \||\phi_\tau\rangle\|^2 .$$

If $\tau \in \mathcal{A}$, every database satisfying these constraints contains the complete evaluations of two distinct messages of length at most B with the same final output. Such a database is therefore in $\mathsf{Col}^{\le B}$. Since different records τ occupy orthogonal work-register states, their squared norms add, giving

$$\begin{aligned}
\left\|\mathsf{Col}^{\le B} |\psi_{\mathrm{ver}}\rangle\right\|^2 &\ge \sum_{\tau\in\mathcal{A}} \left\|\Pi_\tau \widehat{S} |\phi_\tau\rangle\right\|^2 \\
&\ge \left(1 - \frac{1}{N}\right)^{2L} \sum_{\tau\in\mathcal{A}} \||\phi_\tau\rangle\|^2 \\
&= \left(1 - \frac{1}{N}\right)^{2L} p.
\end{aligned}$$

Taking square roots and using $p \le 1$, we obtain

$$\begin{aligned}
\sqrt{p} - \left\|\mathsf{Col}^{\le B} |\psi_{\mathrm{ver}}\rangle\right\| &\le \left[1 - \left(1 - \frac{1}{N}\right)^{L}\right] \sqrt{p} \\
&\le \min\left\{1, \frac{L}{N}\right\} \\
&\le \sqrt{\frac{L}{N}}.
\end{aligned}$$

For $B = 2$, the recorded evaluations also witness membership in $\mathsf{Col}^{\le 2}$, so the same argument applies. □

Conditioning on the offline event. We finally package the effect of conditioning on the event $\mathcal{W}$ from the advice-removal theorem. Suppose an unconditioned preprocessing/online execution has only ε norm outside some structural projector G. After conditioning on an event of probability p, this structural component may be amplified by a factor $1/\sqrt{p}$. In contrast, the fresh-salt-row error does *not* acquire this factor: the salt is sampled only after $\mathcal{W}$ has already been fixed, so it remains uniform and independent of the conditioned offline state. The next lemma isolates exactly these two losses and is the common bridge from the offline phase to the projected online analyses in both collision sections.

Lemma 3.11 (Conditional projection comparison)**.** *Consider an offline computation initialized independently of h that makes at most Q oracle queries. Let $\mathcal{W}$ be an event determined during the offline phase with probability $p > 0$. Condition on $\mathcal{W}$ and only then sample a fresh uniform salt $a \in [N]$. Let G be a projector that depends only on the compressed-oracle database. Suppose that, in the corresponding unconditioned computation in which the event $\mathcal{W}$ is retained in a flag register, the norm outside G is at most ε at the start of the online phase and after each of at most t online oracle queries.*

Then there is a projected evolution $|v_0\rangle, \ldots, |v_t\rangle$ such that $\||v_0\rangle\| \le 1$, every database in the support of $|v_0\rangle$ contains at most Q recorded coordinates, $G|v_j\rangle = |v_j\rangle$ for every j, and the salt row is empty on the support of $|v_0\rangle$. Moreover, for every projector C depending only on the salt and database registers,

$$\left\| C \left| \psi_t^{\mathcal{W}} \right\rangle \right\| \le (t+1)\frac{\varepsilon}{\sqrt{p}} + \sqrt{\frac{Q}{N}} + \|C|v_t\rangle\|, \tag{12}$$

where $\left|\psi_t^{\mathcal{W}}\right\rangle$ is the normalized online state conditioned on $\mathcal{W}$.

Proof. We separate the argument into two steps. First we control the effect of conditioning on $\mathcal{W}$ on the component outside G. We then remove the component on which the freshly sampled salt already has a recorded database entry.

Conditioning on $\mathcal{W}$. Record the offline event $\mathcal{W}$ coherently in a flag register, and let $P_{\mathcal{W}}$ denote the projector onto the branch on which $\mathcal{W}$ occurs. This flag is fixed at the end of the offline phase and is not modified during the online phase.

Let $|\psi_j\rangle$ denote the unconditioned state at the corresponding point after j online queries. Since $\Pr[\mathcal{W}] = p$ and the flag is unchanged online, $\|P_{\mathcal{W}}|\psi_j\rangle\|^2 = p$. Hence the normalized state conditioned on $\mathcal{W}$, $\left|\psi_j^{\mathcal{W}}\right\rangle$ satisfies $\left\|\left|\psi_j^{\mathcal{W}}\right\rangle\right\| = \frac{\|P_{\mathcal{W}}|\psi_j\rangle\|}{\sqrt{p}}$. Thus the factor $1/\sqrt{p}$ is simply the normalization factor arising from conditioning on an event of probability p.

The projector $P_{\mathcal{W}}$ acts on the flag register, whereas G acts on the database register, so they commute. Therefore, using the assumed unconditioned bound $\|(I-G)|\psi_j\rangle\| \le \varepsilon$, we obtain

$$\begin{aligned}
\left\|(I-G)\left|\psi_j^{\mathcal{W}}\right\rangle\right\| &= \frac{1}{\sqrt{p}}\|(I-G)P_{\mathcal{W}}|\psi_j\rangle\| && (13)\\
&= \frac{1}{\sqrt{p}}\|P_{\mathcal{W}}(I-G)|\psi_j\rangle\| && (14)\\
&\le \frac{1}{\sqrt{p}}\|(I-G)|\psi_j\rangle\| && (15)\\
&\le \frac{\varepsilon}{\sqrt{p}} && (16)
\end{aligned}$$

In other words, conditioning on an event of probability p can amplify the norm of a pre-existing component by at most a factor $1/\sqrt{p}$.

We now compare the conditioned execution with the evolution obtained by projecting onto G. Let U_0 denote the online unitary before the first oracle query and, for $j = 1, \ldots, t$, let U_j denote the unitary following the j-th query and preceding the next one. This means $|\psi_j\rangle = \mathcal{O}U_{j-1}|\psi_{j-1}\rangle$, $\left|\psi_j^{\mathcal{W}}\right\rangle = \mathcal{O}U_{j-1}\left|\psi_{j-1}^{\mathcal{W}}\right\rangle$ for $j \ge 1$. Also, these unitaries do not modify the database register and hence commute with G.

Starting from the conditioned execution, project onto G before the first query and again after

every oracle query. Let $|\widetilde{v}_j\rangle$ denote the resulting projected state after j queries, i.e.

$$|\widetilde{v}_0\rangle = G\left|\psi_0^{\mathcal{W}}\right\rangle$$
$$|\widetilde{v}_j\rangle = G\mathcal{O}U_{j-1}\,|\widetilde{v}_{j-1}\rangle \quad \text{for } j \geq 1$$

Set $d_j := \left\|\left|\psi_j^{\mathcal{W}}\right\rangle - |\widetilde{v}_j\rangle\right\|$. Before the first query, $d_0 = \left\|(I-G)\left|\psi_j^{\mathcal{W}}\right\rangle\right\| \leq \frac{\varepsilon}{\sqrt{p}}$. Let us bound d_j for $j \geq 1$ by a recurrence argument.

$$\begin{aligned}
d_j &= \left\|\left|\psi_j^{\mathcal{W}}\right\rangle - |\widetilde{v}_j\rangle\right\| \\
&\leq \left\|\left|\psi_j^{\mathcal{W}}\right\rangle - G\left|\psi_j^{\mathcal{W}}\right\rangle\right\| + \left\|G\left|\psi_j^{\mathcal{W}}\right\rangle - |\widetilde{v}_j\rangle\right\| \quad \text{(by triangle inequality)} \\
&= \left\|\left|\psi_j^{\mathcal{W}}\right\rangle - G\left|\psi_j^{\mathcal{W}}\right\rangle\right\| + \left\|G\mathcal{O}U_{j-1}\left|\psi_{j-1}^{\mathcal{W}}\right\rangle - G\mathcal{O}U_{j-1}\,|\widetilde{v}_{j-1}\rangle\right\| \quad \text{(by recurrences of } \left|\psi_j^{\mathcal{W}}\right\rangle, |\widetilde{v}_j\rangle) \\
&\leq \left\|(I-G)\left|\psi_j^{\mathcal{W}}\right\rangle\right\| + \|G\mathcal{O}U_{j-1}\| \left\|\left|\psi_{j-1}^{\mathcal{W}}\right\rangle - |\widetilde{v}_{j-1}\rangle\right\| \\
&\leq \frac{\varepsilon}{\sqrt{p}} + d_{j-1} \quad \text{(using } \|G\|, \|U_{j-1}\|, \|\mathcal{O}\| \leq 1, \text{ equation (16), definition of } d_{j-1})
\end{aligned}$$

By induction, $d_t \leq (t+1)\frac{\varepsilon}{\sqrt{p}}$.

The fresh salt row. It remains to ensure that the salt row is empty at the beginning of the online phase. The important point is that the salt a is sampled *after* conditioning on $\mathcal{W}$. Thus conditioning may change the offline state, but it cannot bias the distribution of a.

Write the conditioned offline state, including all registers other than the salt, in the form $|\phi\rangle = \sum_D |D\rangle\,|\xi_D\rangle$, where D ranges over basis databases and $\sum_D \||\xi_D\rangle\|^2 = 1$. For a database D, let

$$\mathrm{Rows}(D) := \{a \in [N] : \text{the } a\text{-row of } D \text{ is nonempty}\}.$$

Since the offline phase makes at most Q oracle queries, every basis database contains at most Q recorded coordinates. Consequently, $|\mathrm{Rows}(D)| \leq Q$ for every D.

Let E_a denote the projector onto databases whose a-row is nonempty. For a fresh uniformly random salt a,

$$\begin{aligned}
\mathbb{E}_{a\leftarrow[N]}\left[\|E_a\,|\phi\rangle\|^2\right] &= \frac{1}{N}\sum_{a\in[N]} \|E_a\,|\phi\rangle\|^2 \\
&= \frac{1}{N}\sum_{a\in[N]}\sum_D \||\xi_D\rangle\|^2\,\mathbf{1}\left[a^{\text{th}} \text{ row in } D \text{ is non-empty}\right] \\
&= \frac{1}{N}\sum_D \||\xi_D\rangle\|^2 \sum_{a\in[N]} \mathbf{1}\left[a^{\text{th}} \text{ row in } D \text{ is non-empty}\right] \\
&= \sum_D \||\xi_D\rangle\|^2 \frac{|\mathrm{Rows}(D)|}{N} \leq \frac{Q}{N}.
\end{aligned}$$

Equivalently, after adjoining the fresh uniform salt register, the joint component on which the selected salt row is already nonempty has squared norm at most Q/N, and therefore

$$\left\|\frac{1}{\sqrt{N}}\sum_{a\in[N]} |a\rangle \otimes E_a\,|\phi\rangle\right\| \leq \sqrt{\frac{Q}{N}}$$

This is the same bound as in Lemma 3.9. Notice that there is no factor $1/\sqrt{p}$ in this term: the salt is sampled only after the conditioning event has been fixed.

Remove this nonempty-salt-row component from the initial G-projected state. Since both the empty-row projector and G are diagonal in the database basis, they commute, so the resulting initial state still lies in the image of G. Call this state $|v_0\rangle$, and evolve it using the same G-projected online evolution to obtain $|v_0\rangle, \ldots, |v_t\rangle$. In other words,

$$\begin{aligned} |v_0\rangle &= \Pi_{\text{empty}} |\widetilde{v}_0\rangle, \\ |v_j\rangle &= G\mathcal{O}U_{j-1} |v_{j-1}\rangle \end{aligned}$$

The removed component satisfies $\||v_0\rangle - |\widetilde{v}_0\rangle\| \le \sqrt{Q/N}$. By induction hypothesis, assume $\||v_{j-1}\rangle - |\widetilde{v}_{j-1}\rangle\| \le \sqrt{Q/N}$ for some $j \ge 1$. Then

$$\begin{aligned} \||v_j\rangle - |\widetilde{v}_j\rangle\| &= \|G\mathcal{O}U_{j-1} |v_{j-1}\rangle - G\mathcal{O}U_{j-1} |\widetilde{v}_{j-1}\rangle\| \\ &\le \|G\mathcal{O}U_{j-1}\| \, \||v_{j-1}\rangle - |\widetilde{v}_{j-1}\rangle\| \\ &\le \sqrt{Q/N} \end{aligned}$$

Hence

$$\begin{aligned} \left\| \left|\psi_t^{\mathcal{W}}\right\rangle - |v_t\rangle \right\| &\le \left\| \left|\psi_t^{\mathcal{W}}\right\rangle - |\widetilde{v}_t\rangle \right\| + \||\widetilde{v}_t\rangle - |v_t\rangle\| \quad \text{(by triangle inequality)} \\ &\le (t+1)\frac{\varepsilon}{\sqrt{p}} + \sqrt{\frac{Q}{N}} \end{aligned}$$

Finally, applying the projector C cannot increase norm. Therefore, by the triangle inequality,

$$\left\| C \left|\psi_t^{\mathcal{W}}\right\rangle \right\| \le \|C |v_t\rangle\| + (t+1)\frac{\varepsilon}{\sqrt{p}} + \sqrt{\frac{Q}{N}},$$

which is exactly (12). □

A DIMENSION BOUND FOR QUANTUM ADVICE. We will use the following elementary fact when passing from an arbitrary S-qubit advice state to the maximally mixed state. Fix the oracle and all parts of an experiment except for the advice state. The probability of any fixed success event can then be written as $\mathrm{Tr}(M\rho)$, where ρ is the advice state and $M \succeq 0$ is the corresponding acceptance operator on the S-qubit advice register. The next lemma (observed in [Aar05, Liu23]) shows that replacing ρ by the maximally mixed state costs at most the dimension 2^S of this register. Thus, whenever it is convenient to analyze the maximally mixed advice state, the resulting bound can be converted back to arbitrary S-qubit advice with a factor 2^S.

Lemma 3.12 (Replacing an arbitrary advice state by the maximally mixed state, [Aar05])**.** *Let M be a positive semidefinite operator on an S-qubit register and let ρ be any density matrix on that register. Then*

$$\mathrm{Tr}(M\rho) \le 2^S \, \mathrm{Tr}\left(M \frac{I}{2^S}\right).$$

Proof. Since ρ is a density matrix, all its eigenvalues lie in $[0,1]$. Hence $\rho \preceq I$. Because $M \succeq 0$, this implies $\mathrm{Tr}(M\rho) \le \mathrm{Tr}(MI) = \mathrm{Tr}(M)$ Finally, $\mathrm{Tr}(M) = 2^S \, \mathrm{Tr}\left(M \frac{I}{2^S}\right)$, which proves the claim. □

4 The two-block case

We begin with the case $B = 2$. A message of length at most two can only use an edge leaving the salt a, followed by at most one edge leaving the vertex reached by that first edge. Thus, once the recorded first-block outputs from a are fixed, we know exactly which rows can supply the final edge of a two-block evaluation. This simple structure allows us to characterize precisely which parts of the database at the end of the offline phase can help the online adversary create a collision.

Fix a salt a. A one-block message uses an edge leaving a. A two-block message first follows an edge $a \xrightarrow{m} s$ leaving a and then an edge leaving s. Hence every possible last edge of a recorded message of length at most two has source either a or a vertex reached from a by one recorded edge. We first turn this observation into an exact characterization of recorded two-block collisions.

The proof is then organized around a single question: starting from a database that does not yet contain such a collision, which answers to one new oracle query can create one? Answering this question will tell us what must be controlled in the database at the end of the offline phase.

Theorem 4.1 (Two-block collisions with quantum advice)**.** *Let $N \geq 2$, $M \geq 1$, $S \geq 0$, and $T \geq 1$, and put $\tau := T + 4$. For $S = 0$, interpret the advice as an oracle-independent initial state. Then*

$$\mathrm{Adv}^{\mathsf{MD\text{-}2col\text{-}q}}_{N,M}(S,T) = O\left(\min\left\{1, \frac{2^S\tau^3}{N}, \frac{S\tau^2\log_2 N + \tau^3}{N}\right\}\right). \tag{17}$$

In particular,

$$\mathrm{Adv}^{\mathsf{MD\text{-}2col\text{-}q}}_{N,M}(S,T) = \widetilde{O}\Big(\min\{1, (ST^2 + T^3)/N\}\Big),$$

where the tilde hides factors polynomial in $\log N$. For $M = N$, classical advice and quantum queries achieve the same rate up to logarithmic factors.

We prove the upper bound first, including its conditioned-on-an-offline-event form, and give the classical-advice attack at the end of the section. The upper bound holds for every block alphabet $[M]$; the matching-attack statement uses $M = N$.

We have compiled a table of notations which are used in this section in Table 2 of Appendix A, which may be helpful for the reader while going through the proof.

Characterizing a two-block collision. For a salt a, define

$$\mathcal{R}_a(D) := \mathrm{Reach}^{\leq 1}_D(a) = \{a\} \cup \{D(a,m) : D(a,m) \neq \perp\}. \tag{18}$$

Definition 4.2. A recorded edge $s \xrightarrow{m} y$ is *relevant* when $s \in \mathcal{R}_a(D)$.

These are exactly the edges that can be the last edge of a recorded nonempty message of length at most two. We continue to use $\mathsf{Col}^{\leq 2}$ for the recorded-collision projector defined in the preliminaries.

Lemma 4.3. *A database D belongs to $\mathcal{C}_{a,2}$ if and only if at least one of the following holds:*

(i) there is a recorded self-loop $D(s,m) = s$ with $s \in \mathcal{R}_a(D)$;

(ii) two distinct relevant coordinates have the same recorded output.

Proof. Suppose first that there is such a self-loop. If $s = a$, the messages (m) and (m,m) both end at a. If $s \neq a$, let u be such that $D(a,u) = s$ (since $s \in \mathcal{R}_a(D)$, such u exists). The messages (u) and (u,m) both end at s. In either case they are distinct, nonempty, and have length at most two.

Now suppose two distinct relevant coordinates have the same output. Let the relevant coordinates be $(s,m),(s',m')$. If $s = a$, then let $\alpha = (m)$. If $s \neq a$, let u be such that $D(a,u) = s$ (since



(a) Self-loop (b) Parallel edges

Figure 3: Types of within-row collisions

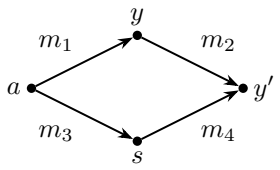


Figure 4: Two paths from a to y'

$s \in \mathcal{R}_a(D)$, such u exists) and set $\alpha = (u, m)$. Similarly, if $s' = a$, then let $\alpha' = (m')$. If $s' \neq a$, let u' be such that $D(a, u') = s'$ (since $s' \in \mathcal{R}_a(D)$, such u' exists) and set $\alpha' = (u', m')$. By construction, the last edge in the evaluation of α is exactly (s, m) and the last edge in the evaluation of α' is exactly (s', m'). Hence the two coordinates having the same output implies that α and α' have the same endpoint.

It remains only to check that these messages are distinct. Suppose otherwise that $\alpha' = \alpha$. If the common message has length one, then $s = s' = a$ and $m = m'$, so the two coordinates are equal. If it has length two, then $u = u'$ and $m = m'$. Since $D(a, u) = s$ and $D(a, u') = s'$, we also have $s = s'$, and again the two coordinates are equal. Both cases contradict the assumption that the coordinates are distinct. Thus, α, α' form a recorded two-block collision with respect to salt a.

Conversely, take two distinct recorded colliding messages. Their last coordinates have relevant sources and the same output. If these coordinates are distinct, (ii) holds. Suppose they are the same. If one message has one block and the other two, the common last coordinate has source a. The first edge of the two-block message must therefore return to a, giving (i). If both messages have two blocks, their second blocks and intermediate states agree. Their first blocks must differ, so the two first coordinates in row a have equal outputs, giving (ii). □

The characterization above identifies the condition for a two-block collision. We now work backwards from that condition and determine what an individual oracle query can do.

Fix a database D with no recorded two-block collision, and suppose a query changes one coordinate $x = (s, m)$. If $s \neq a$, then changing the value of $D(x)$ does not change which rows are reachable from the salt in one step. If $s \notin \mathcal{R}_a(D)$ is not reachable, the new edge also cannot be relevant. If $s \in \mathcal{R}_a(D)$, the new edge can create a collision only by becoming a self-loop or by taking an output that is already used by another relevant edge.

The more interesting case is $s = a$. If the new answer is y, then row y becomes newly reachable and every old edge leaving that row becomes relevant. There are two different ways in which this can create a collision.

(i) Row y may already contain enough information to produce a collision by itself: it may contain a self-loop, or two distinct edges in the row may already have the same output. Figure 3 depicts the two cases.

(ii) An edge in row y may have the same output as an edge whose source was already reachable before the query. Figure 4 depicts this.

These two mechanisms require different bookkeeping. The first asks how many rows already contain a self-loop or a within-row collision. The second asks how many rows contain an edge that shares its output with a relevant edge. We now define exactly the quantities that measure these two phenomena.

Rows that contain a self-loop or a within-row collision. Let

$$\mathcal{L}(D) := \{s : \exists m,\ D(s,m) = s,\ \text{or } \exists m, m', m \neq m',\ D(s,m) = D(s,m') \neq \perp\},$$
$$J(D) := |\mathcal{L}(D)|$$

Thus $J(D)$ is exactly the number of rows of type (i): once such a row is reached, the old database already contains a collision-producing pattern.

Rows contain an edge that shares its output with a relevant edge. For each $z \in [N]$, let $n_z(D) := |\{(s,m) \in \mathrm{Dom}(D) : D(s,m) = z\}|$, and let $r_z(D;a) := |\{(s,m) \in \mathrm{Dom}(D) : s \in \mathcal{R}_a(D),\ D(s,m) = z\}|$. The first quantity counts all recorded edges with output z, while the second counts only the relevant ones. Define

$$K_a(D) := \sum_{z \in [N]} n_z(D) \cdot r_z(D;a). \tag{19}$$

$K_a(D)$ is the number of ordered pairs (e, e') of recorded edges with the same output for which the first edge e is relevant. Thus it counts precisely the equal-output structure that can become useful once another source row s becomes reachable (in one step) by a.

To see the connection with number of rows of type (ii), define

$$O_a(D) := \{z : r_z(D;a) > 0\}$$
$$P_a(D) := \{s : \text{row } s \text{ contains a recorded edge whose output lies in } O_a(D)\}.$$

Then

$$|O_a(D)| \leq K_a(D), \qquad |P_a(D)| \leq \sum_{z \in O_a(D)} n_z(D) \leq K_a(D). \tag{20}$$

Indeed, every $z \in O_a(D)$ has $r_z(D;a) \geq 1$ and $n_z(D) \geq 1$. The second inequality follows because every row in $\sum_{z \in O_a(D)} n_z(D)$ is exactly the number of recorded edges whose output belongs to $O_a(D)$. By definition, every row in $P_a(D)$ contains at least one such edge. Choosing one such edge from each row therefore gives

$$|P_a(D)| \leq \sum_{z \in O_a(D)} n_z(D).$$

Finally, since $r_z(D;a) \geq 1$ for every $z \in O_a(D)$,

$$\sum_{z \in O_a(D)} n_z(D) \leq \sum_{z \in O_a(D)} n_z(D) \cdot r_z(D;a) \leq K_a(D).$$

Hence, a query in row a can cause a two block collision if its answer selects one of the $J(D)$ locally bad rows, or if it selects one of the rows counted by $P_a(D)$ (which we upper bound using $K_a(D)$).

For a classical database it would be enough to bound $K_a(D)$. A quantum execution, however, is a superposition of databases, and there is no reason for every database in the support to have a small value of K_a, nor do we need such a pointwise statement. The collision-entry estimate will be linear in $K_a(D)$, so the quantity we actually need is its average with respect to squared amplitudes. For a state $|\xi\rangle = \sum_{a,D} |a\rangle\,|D\rangle\,|\xi_{a,D}\rangle$, this average is exactly $\left\|\sqrt{K}\,|\xi\rangle\right\|^2 = \sum_{a,D} K_a(D)\,\||\xi_{a,D}\rangle\|^2$.

We need to understand how much K_a can grow under one query. Consider again a query in row a. For this local calculation, fix the queried coordinate (a, m), and let D denote the database with this coordinate set to $\perp$. For each possible answer y, write $D_y := D[(a, m) \to y]$. For each z, let $c_z(D; y)$ be the number of recorded edges in D in row y with output z, i.e.

$$c_z(D; y) := |\{(y, m) : D(y, m) = z\}|$$

If $y \notin \mathcal{R}_a(D)$, the old edges in row y become relevant when the new edge is inserted. The contribution from pairing these newly relevant old edges with previously recorded edges is

$$\sum_z n_z(D) \cdot c_z(D; y). \tag{21}$$

Summing this expression over all possible rows y gives

$$\sum_{y\in[N]} \sum_{z\in[N]} n_z(D) \cdot c_z(D; y) = \sum_{z\in[N]} n_z(D)^2, \tag{22}$$

because, for each fixed z, the edges counted by $c_z(D; y)$ over all rows y are exactly the recorded edges with output z. We let $\Gamma(D) := \sum_{z\in[N]} n_z(D)^2$.

Thus $\Gamma(D)$ bounds, after summing over all possible answers, the contribution from newly relevant old edges paired with previously recorded edges.

This is only part of the increase in K_a: we must also count pairs involving the newly inserted edge. Expanding the definition of K_a gives the full increment

$$\begin{aligned} K_a(D_y) - K_a(D) = 1 + n_y(D) + r_y(D; a) \\ + \mathbf{1}[y \notin \mathcal{R}_a(D)] \left(c_y(D; y) + \sum_z n_z(D) c_z(D; y) \right). \end{aligned}$$

Here $n_y(D)$ counts pairs in which the new edge is first and an old edge is second. The terms $r_y(D; a)$ and $\mathbf{1}[y \notin R_a(D)]c_y(D; y)$ count pairs in which an old relevant edge is first and the new edge is second; the latter term accounts for old self-loops in the newly relevant row. Finally, the constant 1 counts the new edge paired with itself.

Each of $n_y(D)$, $r_y(D; a)$, and $\mathbf{1}[y \notin \mathcal{R}_a(D)]c_y(D; y)$ sums to at most $|D|$ over all y. Combining this with (22), and using $|D| \le \Gamma(D)$, gives

$$\sum_{y\in[N]} (K_a(D_y) - K_a(D) - 1) \le \Gamma(D) + 3\,|D| \le 4\Gamma(D) \tag{23}$$

We treat the constant contribution 1 separately rather than including it in this sum. In the local quantum calculation, this contribution is the indicator that the queried coordinate is non-$\perp$. Its expectation lies between zero and the squared norm of the local state, so its increase is at most that squared norm. Summing over the orthogonal local query blocks therefore contributes at most 1 per query to the squared-amplitude-weighted expectation of K_a. The remaining contributions are controlled by the bound above on their sum over possible answers.

We now prove the bounds needed to control a single online query. Fix the database outside the queried coordinate. Our goal is to bound the number of possible oracle answers y for which inserting y at that coordinate creates a recorded two-block collision.

As discussed above, there are two sources of such answers. A query in row a may reach a row that already contains a self-loop or a within-row collision; the number of these rows is measured

by $J(D)$. Alternatively, the newly reached row may contain an edge whose output agrees with the output of an already relevant edge; the number of such possibilities is controlled by $K_a(D)$.

We therefore first bound the part of the compressed-oracle state on which $J(D)$ is large. We then introduce the additional bound on $\Gamma(D) = \sum_z n_z(D)^2$, which is needed to control how much $K_a(D)$ can increase during an online query. After bounding the growth of $K_a(D)$, we return to the single-query question above and bound the number of answers that can create a collision.

UPPER BOUNDING $J(D)$ AND $\Gamma(D)$ We first bound the number of rows that already contain enough information to produce a collision as soon as that row becomes reachable.

Recall that $\mathcal{L}(D)$ is the set of rows containing either a self-loop or a within-row collision, and $J(D) := |\mathcal{L}(D)|$. Our goal is to show that a q-query algorithm has very little amplitude on databases for which $J(D)$ is large.

A direct application of the generic tail lemma from the preliminaries is too weak. Indeed, if a queried row currently contains d recorded entries, then there are at most $d+1$ answers that can make this row enter $\mathcal{L}$: the answer may equal the source of the row, creating a self-loop, or it may equal one of the d outputs already recorded in that row, creating a within-row collision. Bounding $d+1$ by the total number of previous queries loses the row-by-row structure and leads to a non-tight bound.

Instead, we charge a row gradually as entries are recorded in it. Fix $L := \lceil N^{1/3} \rceil$. For each row s, let $d_s(D) := |\{m \in [M] : D(s,m) \neq \perp\}|$ and define

$$f_s(D) := \begin{cases} 1, & s \in \mathcal{L}(D), \\ \min\{d_s(D)/L, 1\}, & s \notin \mathcal{L}(D). \end{cases} \qquad F(D) := \sum_{s \in [N]} f_s(D).$$

Thus every row in $\mathcal{L}(D)$ contributes one to $F(D)$, and hence $J(D) \leq F(D)$. A collision-free row with $d < L$ recorded entries contributes only d/L. The purpose of this fractional charge is that filling such a row by one ordinary entry increases F by only $1/L$.

We will prove the following tail bound.

Lemma 4.4 (Upper bounding $J(D)$)**.** *Let $|\Psi_q\rangle$ be the state after q quantum queries to the random compression function, starting from a state independent of the oracle. Then, for every $k \geq 0$,*

$$\left\| \Pi_{\{J \geq k\}} |\Psi_q\rangle \right\| \leq \exp\left(\frac{8q}{N^{1/3}} - \frac{k}{2} \right).$$

Proof. We first explain how the potential F will be used. For a state written according to the database register as $|\Psi\rangle = \sum_D |D\rangle |\psi_D\rangle$, let $e^{F/2}$ denote the diagonal operator $e^{F/2} |D\rangle = e^{F(D)/2} |D\rangle$. Then $\left\| e^{F/2} |\Psi\rangle \right\|^2 = \sum_D e^{F(D)} \||\psi_D\rangle\|^2$. Thus controlling $\left\| e^{F/2} |\Psi\rangle \right\|$ controls the tail of F. Indeed,

$$\begin{aligned} \left\| \Pi_{\{F \geq k\}} |\Psi\rangle \right\|^2 &= \sum_{D : F(D) \geq k} \||\psi_D\rangle\|^2 \\ &\leq e^{-k} \sum_{D : F(D) \geq k} e^{F(D)} \||\psi_D\rangle\|^2 \\ &\leq e^{-k} \left\| e^{F/2} |\Psi\rangle \right\|^2. \end{aligned}$$

Taking square roots gives

$$\left\| \Pi_{\{F \geq k\}} |\Psi\rangle \right\| \leq e^{-k/2} \left\| e^{F/2} |\Psi\rangle \right\|. \tag{24}$$

It therefore remains to bound how much one oracle query can increase the weighted norm $\left\| e^{F/2} |\Psi\rangle \right\|$.

Fix one oracle query. We analyze it after fixing the queried coordinate $x = (s, m)$, the value $u \in \mathbb{Z}_N$ in the response register, and all database entries except the one at x. Let $H := D^{-x}$. With these quantities fixed, the only database value that varies is $D(x) \in [N] \cup \{\perp\}$, and the compressed-oracle query acts on this value by the local matrix R_u defined in the preliminaries.

Only the contribution f_s of row s can vary. If row s already contains a self-loop or a within-row collision in H, then it remains in $\mathcal{L}$ for every possible value of $D(x)$, so $f_s = 1$ throughout. Likewise, if row s already contains at least L recorded entries in H, then $f_s = 1$ throughout. In either case F is constant on this part of the query, and there is nothing to prove.

We may therefore assume that row s of H contains neither a self-loop nor a within-row collision and has $d := d_s(H) < L$ recorded entries.

Which values assigned to $D(x)$ make row s enter $\mathcal{L}$? They are contained in

$$Y := \{s\} \cup \{H(s, m') : H(s, m') \neq \perp\}.$$

The answer s creates a self-loop, while an answer equal to an output already present in row s creates a within-row collision. Hence $|Y| \leq d + 1 \leq L$.

Hence, we have[6]

$$f_s(D(x)) = \begin{cases} d/L, & D(x) = \perp, \\ (d+1)/L, & D(x) = y \notin Y, \\ 1, & D(x) = y \in Y. \end{cases} \tag{25}$$

Thus there are two effects to control. Moving from $\perp$ to an ordinary non-$\perp$ value changes f_s only by $1/L$. A transition into Y may change f_s by much more, but there are at most L such answers.

We make this separation explicit. Set $a := e^{1/(2L)}$ and $c := \exp\left(\frac{1-(d+1)/L}{2}\right)$. Since $d + 1 \leq L$, we have $1 \leq c \leq e^{1/2}$. Let P_Y denote the projector onto the local basis vectors $\{|y\rangle : y \in Y\}$. Define two diagonal operators acting on the xth register of D by

$$A\,|\perp\rangle = |\perp\rangle\,, \qquad A\,|y\rangle = a\,|y\rangle \quad (y \in [N]),$$

and

$$B := I + (c - 1)P_Y.$$

Equation (25) implies that, on the xth register of D, $e^{f_s/2} = e^{d/(2L)}AB$. Indeed, after removing the common factor $e^{d/(2L)}$, the remaining multipliers are 1 on $|\perp\rangle$, $a = e^{1/(2L)}$ outside Y, and, for $y \in Y$, $ac = e^{1/(2L)}e^{(1-(d+1)/L)/2} = e^{(1-d/L)/2}$, which is exactly the remaining factor required to reach $e^{1/2}$.

Hence

$$e^{f_s/2}R_u e^{-f_s/2} = ABR_uB^{-1}A^{-1}.$$

We first bound the effect of B. Since

$$B = cP_Y + (I - P_Y), \qquad B^{-1} = c^{-1}P_Y + (I - P_Y),$$

we have

$$\begin{aligned} BR_uB^{-1} = R_u &+ (c-1)P_YR_u(I - P_Y) \\ &+ (c^{-1} - 1)(I - P_Y)R_uP_Y. \end{aligned}$$

[6]Here we slightly abuse notation by defining f_s over $[N] \cup \{\perp\}$ rather than the space of databases, when x, D^{-x} are fixed.

The local-query estimate from Lemma 3.8 of the preliminaries gives

$$\|P_Y R_u (I - P_Y)\| \le 4\sqrt{\frac{|Y|}{N}} \le 4\sqrt{\frac{L}{N}}$$

Also,

$$\|(I - P_Y) R_u P_Y\| = \left\| P_Y^\dagger R_u^\dagger (I - P_Y)^\dagger \right\| = \|P_Y R_{-u} (I - P_Y)\| \le 4\sqrt{\frac{L}{N}}$$

Hence we obtain

$$\left\| B R_u B^{-1} \right\| \le \|R_u\| + (c-1)\,\|P_Y R_u (I - P_Y)\| + (1 - c^{-1})\,\|(I - P_Y) R_u P_Y\| \tag{26}$$

$$\le 1 + 5\sqrt{\frac{L}{N}} \quad (\text{using } c - c^{-1} \le e^{1/2} - e^{-1/2} < 5/4) \tag{27}$$

Since A acts by 1 on $|\perp\rangle$ and by $a = e^{1/(2L)} > 1$ on every non-$\perp$ basis state,

$$A^{-1}\,|\perp\rangle = |\perp\rangle\,, \qquad A^{-1}\,|y\rangle = a^{-1}\,|y\rangle \quad (y \in [N]).$$

Hence

$$\|A\| = a, \qquad \left\| A^{-1} \right\| = \max\{1, a^{-1}\} = 1.$$

This gives us

$$\left\| e^{f_s/2} R_u e^{-f_s/2} \right\| \le e^{1/(2L)} \left(1 + 5\sqrt{\frac{L}{N}} \right).$$

The two losses above have orders $\frac{1}{L}$ and $\sqrt{\frac{L}{N}}$. Balancing them gives $L \asymp N^{1/3}$. We choose $L = \lceil N^{1/3} \rceil$, which entails $N^{1/3} \le L \le 2N^{1/3}$, and therefore

$$e^{1/(2L)} \left(1 + 5\sqrt{\frac{L}{N}} \right) \le \exp(8N^{-1/3}).$$

All other rows contribute the same value to F throughout the local calculation and hence cancel in the conjugation. Since the full compressed-oracle query is the orthogonal direct sum of these local actions, we have

$$\left\| e^{F/2} \mathcal{O} e^{-F/2} \right\| \le \exp(8N^{-1/3}).$$

The unitaries between oracle queries do not modify the database register, so they commute with the diagonal operator $e^{F/2}$. The initial database is empty and hence has $F = 0$. Iterating for q oracle queries gives

$$\left\| e^{F/2}\,|\Psi_q\rangle \right\| = \left\| e^{F/2} \mathcal{O} e^{-F/2} \cdot e^{F/2}\,|\Psi_{q-1}\rangle \right\| \le \left\| e^{F/2} \mathcal{O} e^{-F/2} \right\| \left\| e^{F/2}\,|\Psi_{q-1}\rangle \right\| \tag{28}$$

$$\dots \le \left\| e^{F/2} \mathcal{O} e^{-F/2} \right\|^q \le \exp\left(\frac{8q}{N^{1/3}} \right) \tag{29}$$

Finally, since $\{J \ge k\} \subseteq \{F \ge k\}$. Applying (24) and (29),

$$\left\| \Pi_{\{J \ge k\}}\,|\Psi_q\rangle \right\| \le \exp\left(\frac{8q}{N^{1/3}} - \frac{k}{2} \right),$$

as claimed. □

We next bound the part of the state on which $\Gamma(D) = \sum_z n_z(D)^2$ is large. This quantity is needed because, as discussed above, it controls the aggregate contribution to the growth of $K_a(D)$ when a first-block answer makes a new row reachable. More precisely, the full local calculation in equation (23) shows that, after separating the constant contribution of the newly inserted edge, the remaining increments summed over all possible answers are at most $4\Gamma(D)$.

The difficulty is that $\Gamma(D)$ does not increase by a fixed amount when one database entry changes. If a new coordinate is assigned an output z that already occurs $n_z(D)$ times, then the increase in $\Gamma(D)$ depends on $n_z(D)$. We therefore introduce a related quantity whose change under one query is easier to control. Define $Z(D) := \sum_z n_z(D)(n_z(D) - 1) = \Gamma(D) - |D|$. If a new recorded coordinate is assigned output z, then $Z(D)$ increases by exactly $2n_z(D)$. We will control these changes through the square-root potential $g_q(D) := \sqrt{q + Z(D)}$. Every database appearing after q oracle queries contains at most q recorded coordinates, and therefore $\Gamma(D) = Z(D) + |D| \le Z(D) + q = g_q(D)^2$. Consequently, it is enough to prove a tail bound for $g_q(D)$.

Lemma 4.5 (Upper bounding $\Gamma(D)$)**.** *Let $|\Psi_q\rangle$ be as in Lemma 4.4. Fix $R \ge q$ and $\theta \ge 0$. Then $\left\| \Pi_{\{\Gamma > (\sqrt{R} + 8R/\sqrt{N} + \theta)^2\}} |\Psi_q\rangle \right\| \le e^{-\theta/2}$.*

Proof. We first prove

$$\left\| e^{g_q/2} |\Psi_q\rangle \right\| \le \exp\left(\frac{\sqrt{q}}{2} + \frac{4q}{\sqrt{N}} \right).$$

Let us bound the effect of one oracle query on the exponential moment associated with g_q.

Fix the queried coordinate x, the value u in the response register, and all database entries outside x. Let $H := D^{-x}$ and

$$b := q + Z(H).$$

For each $y \in [N]$, let $n_y := n_y(H)$. If the queried coordinate is assigned the value y, then, as observed above, $Z(H[x \leftarrow y]) = Z(H) + 2n_y$. Hence $g_q(H[x \leftarrow y]) = \sqrt{b + 2n_y}$. If instead the queried coordinate is $\perp$, then $g_q(H[x \leftarrow \perp]) = \sqrt{b}$. It is therefore convenient to define

$$\delta_y := \sqrt{b + 2n_y} - \sqrt{b} \qquad (y \in [N]), \qquad \delta_\perp := 0.$$

Thus δ_y is exactly the increase in g_q caused by assigning output y to the queried coordinate. If $b = 0$, then necessarily $q = 0$ and $Z(H) = 0$. Since $|H| \le q$, the database H is empty, so $n_y = 0$ and hence $\delta_y = 0$ for every y. The two bounds below are therefore immediate. We may thus assume $b > 0$.

We first record two bounds on these increments. Rationalizing the difference of square roots gives

$$\delta_y = \frac{2n_y}{\sqrt{b + 2n_y} + \sqrt{b}} \le \frac{n_y}{\sqrt{b}}.$$

Moreover,

$$\Gamma(H) = Z(H) + |H| \le Z(H) + q = b.$$

Since $n_y^2 \le \sum_z n_z^2 = \Gamma(H) \le b$, we obtain $0 \le \delta_y \le 1$. Also,

$$\sum_y \delta_y^2 \le \frac{\sum_y n_y^2}{b} = \frac{\Gamma(H)}{b} \le 1. \tag{30}$$

We now compare the query after weighting by $e^{g_q/2}$ with the ordinary query.

On the local basis state corresponding to the value $r \in [N] \cup \{\perp\}$ at the queried coordinate, the factor $e^{g_q/2}$ is, up to the common scalar $e^{\sqrt{b}/2}$, $e^{\delta_r/2}$. The common scalar cancels in the conjugation. Therefore, if $(R_u)_{y,r}$ is the matrix entry of the ordinary compressed-oracle query from value r to value y, then the corresponding entry of $e^{g_q/2} R_u e^{-g_q/2}$ is $e^{(\delta_y-\delta_r)/2}(R_u)_{y,r}$. Let $\Delta := e^{g_q/2} R_u e^{-g_q/2} - R_u$. We will show that $\|\Delta\| \le 4/\sqrt{N}$.

First consider non-$\perp$ input and output values $r, y \in [N]$. When $y = r$, the exponential factor is one, so the diagonal entry of Δ is zero. For $y \neq r$, $\Delta_{y,r} = \left(e^{(\delta_y-\delta_r)/2} - 1\right)(R_u)_{y,r}$. Since $0 \le \delta_y, \delta_r \le 1$, we have $|\delta_y - \delta_r| \le 1$, and therefore $\left|e^{(\delta_y-\delta_r)/2} - 1\right| \le \frac{e^{1/2}}{2}|\delta_y - \delta_r|$. The preliminaries ((6),(7)) give $|(R_u)_{y,r}| \le 3/N$ for $y \neq r$. Hence the squared Frobenius norm of the non-$\perp$ to non-$\perp$ part of Δ is at most $\frac{9e}{4N^2} \sum_{y,r\in[N]} (\delta_y - \delta_r)^2$. Now

$$\begin{aligned}
\sum_{y,r} (\delta_y - \delta_r)^2 &= \sum_{y,r} \left(\delta_y^2 + \delta_r^2 - 2\delta_y\delta_r\right) \\
&= 2N \sum_y \delta_y^2 - 2\left(\sum_y \delta_y\right)^2 \\
&\le 2N,
\end{aligned}$$

where the final inequality uses (30). Thus the non-$\perp$-to-non-$\perp$ part contributes at most $\frac{9e}{2N}$ to the squared Frobenius norm.

It remains to consider transitions involving $\perp$. For a transition from $\perp$ to $y \in [N]$, the extra factor is $e^{\delta_y/2} - 1$, while for a transition from r to $\perp$ it is $e^{-\delta_r/2} - 1$. Using the same elementary bound and the compressed-oracle estimate $|(R_u)_{y,\perp}|,\ |(R_u)_{\perp,r}| \le \frac{1}{\sqrt{N}}$, the two parts together contribute at most $\frac{e}{2N} \sum_y \delta_y^2 \le \frac{e}{2N}$ to the squared Frobenius norm.

Combining the two contributions, $\|\Delta\|^2 \le \|\Delta\|_{\mathrm{F}}^2 \le \frac{5e}{N}$. Since $\sqrt{5e} < 4$, $\|\Delta\| < \frac{4}{\sqrt{N}}$. Because R_u is unitary,

$$\left\| e^{g_q/2} R_u e^{-g_q/2} \right\| \le 1 + \frac{4}{\sqrt{N}}. \tag{31}$$

This controls the query while keeping the index q fixed. After the query, however, we want to use g_{q+1} rather than g_q. For every database D,

$$\begin{aligned}
g_{q+1}(D) - g_q(D) &= \sqrt{q + 1 + Z(D)} - \sqrt{q + Z(D)} \\
&\le \sqrt{q+1} - \sqrt{q},
\end{aligned}$$

because the difference $\sqrt{x+1} - \sqrt{x}$ decreases with x. Consequently,

$$\left\| e^{g_{q+1}/2} e^{-g_q/2} \right\| \le \exp\left(\frac{\sqrt{q+1} - \sqrt{q}}{2}\right). \tag{32}$$

Combining (31) and (32), one query satisfies

$$\left\| e^{g_{q+1}/2} |\Psi_{q+1}\rangle \right\| \le \exp\left(\frac{\sqrt{q+1} - \sqrt{q}}{2}\right)\left(1 + \frac{4}{\sqrt{N}}\right)\left\| e^{g_q/2} |\Psi_q\rangle \right\|.$$

The operations between oracle queries leave the database unchanged and therefore commute with g_q. Iterating the last inequality, using $1 + x \le e^x$, gives

$$\left\| e^{g_q/2} |\Psi_q\rangle \right\| \le \exp\left(\frac{\sqrt{q}}{2} + \frac{4q}{\sqrt{N}}\right). \tag{33}$$

We finally convert this moment bound into the desired bound on $\Gamma(D)$. Every database appearing after q queries satisfies $|D| \le q$, and therefore

$$g_q(D)^2 = q + Z(D) = q + \Gamma(D) - |D| \ge \Gamma(D).$$

Hence, on the event

$$\Gamma(D) > \left(\sqrt{R} + \frac{8R}{\sqrt{N}} + \theta\right)^2,$$

we have

$$g_q(D) > \sqrt{R} + \frac{8R}{\sqrt{N}} + \theta.$$

Exactly as in the preceding exponential-moment argument (in proof of Lemma 4.4, see (24)),

$$\begin{aligned}&\left\|\Pi_{\left\{\Gamma > (\sqrt{R}+8R/\sqrt{N}+\theta)^2\right\}} |\Psi_q\rangle\right\| \\ &\quad\le \exp\!\left(-\frac{1}{2}\left(\sqrt{R} + \frac{8R}{\sqrt{N}} + \theta\right)\right) \left\|e^{g_q/2} |\Psi_q\rangle\right\|.\end{aligned}$$

Using $q \le R$,

$$-\frac{\sqrt{R}}{2} - \frac{4R}{\sqrt{N}} - \frac{\theta}{2} + \frac{\sqrt{q}}{2} + \frac{4q}{\sqrt{N}} \le -\frac{\theta}{2}.$$

Therefore (33) gives us

$$\left\|\Pi_{\left\{\Gamma > (\sqrt{R}+8R/\sqrt{N}+\theta)^2\right\}} |\Psi_q\rangle\right\| \le e^{-\theta/2},$$

as claimed.

□

Fix a total query bound $R \ge 1$ and a parameter $\theta \ge 0$, and define

$$k := \left\lceil \frac{16R}{N^{1/3}} + \theta \right\rceil, \qquad \gamma := \left(\sqrt{R} + \frac{8R}{\sqrt{N}} + \theta\right)^2, \qquad G_2 := \Pi_{\{J \le k,\ \Gamma \le \gamma\}}. \tag{34}$$

By Lemmas 4.4 and 4.5, at every time $q \le R$ in an execution initialized independently of h,

$$\|(I - G_2)\,|\Psi_q\rangle\| \le 2e^{-\theta/2}. \tag{35}$$

Moreover,

$$k = O\!\left(\frac{R}{N^{1/3}} + \theta + 1\right), \qquad \gamma = O\!\left(R + \frac{R^2}{N} + \theta^2\right). \tag{36}$$

We now analyze the online computation after inserting the projection G_2 after every oracle query. The comparison with the actual computation is made later, after we condition on the offline event.

Controlling the growth of $K_a(D)$ during the online phase. We now return to the quantity $K_a(D)$ introduced earlier. Recall that $K_a(D)$ measures the equal-output structure that is already useful from the current set of relevant rows: in particular, it upper bounds $|P_a(D)|$, the number of rows containing an edge whose output is also attained by a relevant edge.

The two bounds proved above have different roles in the rest of the argument. The restriction $J(D) \le k$ controls the number of rows that already contain a self-loop or a within-row collision. The restriction $\Gamma(D) \le \gamma$ will be used here to control how quickly $K_a(D)$ can increase when a new row becomes relevant.

Indeed, consider a query in row a whose answer is y. As discussed above, besides the old edges in row y becoming relevant, the newly inserted edge itself creates additional contributions to K_a. The exact local calculation in equation (23) shows that, after separating the constant contribution 1 of the new edge paired with itself, the remaining increment can be written as a nonnegative weight $w(y)$ satisfying

$$\sum_y w(y) \le 4\Gamma(D).$$

Thus, after restricting to databases with $\Gamma(D) \le \gamma$, the aggregate new contribution over all possible oracle answers is at most 4γ.

Since the quantum state is a superposition of databases, we do not try to bound $K_a(D)$ separately on every database in its support. What we need later is the average of $K_a(D)$ with respect to the squared amplitudes. Let $\left|v_0^{(2)}\right\rangle$ have norm at most one, lie in the image of G_2, and have an empty salt row on its support. Define

$$\left|v_j^{(2)}\right\rangle := G_2 U_j \mathcal{O} G_2 \left|v_{j-1}^{(2)}\right\rangle \qquad (j \ge 1), \tag{37}$$

and set

$$E_j := \left\| \sqrt{K} \left|v_j^{(2)}\right\rangle \right\|^2 = \sum_{a,D} K_a(D) \left\| \left|(v_j^{(2)})_{a,D}\right\rangle \right\|^2. \tag{38}$$

Thus E_j is precisely the average value of $K_a(D)$ in the projected state after j online queries. Since the salt row is empty on the support of $\left|v_0^{(2)}\right\rangle$, no recorded edge is relevant initially, and hence $E_0 = 0$.

To compare E_j with E_{j-1}, we need one elementary consequence of the local compressed-oracle formulas. Suppose that, after fixing every database entry except the queried one, assigning the value y to that coordinate creates an additional nonnegative contribution $w(y)$ to the quantity being tracked. The following inequality bounds the total effect of these possible contributions under one quantum query.

For any function $w : [N] \to \mathbb{R}_{\ge 0}$, extended by $w(\perp) = 0$, and every local vector $|\eta\rangle$,

$$\|\sqrt{w}\, R_u |\eta\rangle\| \le \|\sqrt{w}\Phi_u |\eta\rangle\| + \|\sqrt{w}(R_u - \Phi_u)|\eta\rangle\| \tag{39}$$

$$\le \|\sqrt{w}\,|\eta\rangle\| + 4\sqrt{\frac{\sum_y w(y)}{N}}\, \||\eta\rangle\| \tag{40}$$

Indeed, Φ_u (recall from Section 3.2 that $\Phi_u|\perp\rangle = |\perp\rangle$ and $\Phi_u|y\rangle = \omega^{uy}|y\rangle$) changes only phases and therefore preserves the w-weighted norm. The $\perp$ input column of $R_u - \Phi_u$ has squared w-weighted norm $\sum_y w(y)/N$, while its non-$\perp$-input part has squared w-weighted Frobenius norm at most $9\sum_y w(y)/N$. The triangle inequality gives (40).

Lemma 4.6 (Growth of K_a)**.** *For every $j \ge 1$,*

$$E_j \le E_{j-1} + 1 + \frac{64\gamma}{N} + 16\sqrt{\frac{\gamma}{N}}\sqrt{E_{j-1}}. \tag{41}$$

Consequently, for every $j \ge 0$ and $t \ge 1$,

$$E_j \le 256\left(j + \frac{4\gamma j^2}{N}\right), \qquad \sum_{j=0}^{t-1} E_j \le 128t^2 + \frac{1024\gamma t^3}{3N}. \tag{42}$$

Proof. The projection G_2 can only decrease the quantity in (38), and U_j leaves the database unchanged. It is therefore enough to analyze one oracle query. Fix a salt a, one local block at coordinate $x = (s, m)$ and let $H := D^{-x}$. Write $n_y := n_y(H)$, $r_y := r_y(H; a)$, $\mathcal{R} := \mathcal{R}_a(H)$, and $K_0 := K_a(H)$. Since deleting a coordinate cannot increase Γ, the input restriction implies $\Gamma(H) \le \gamma$.

Queries outside row a. Suppose first that $s \neq a$. Changing the value of $D(x)$ does not change the set of relevant source rows. Let $c = 1$ if $s \in \mathcal{R}$ and $c = 0$ otherwise. If the new value is y, then n_y increases by one and r_y increases by c. Expanding the definition of K_a gives

$$K_a(H[x \leftarrow y]) = K_0 + c + w(y), \qquad w(y) = c\, n_y + r_y.$$

Moreover,

$$\sum_y w(y) = c \sum_y n_y + \sum_y r_y \leq 2|H| \leq 2\Gamma(H) \leq 2\gamma.$$

Queries in row a. Now suppose $s = a$. If the new answer is $y \notin \mathcal{R}$, the old edges in row y become relevant. For each z, let

$$b_z(y) := \mathbf{1}[y \notin \mathcal{R}]\, |\{m' : H(y, m') = z\}| \;\; = \mathbf{1}[y \notin \mathcal{R}] c_z(H; y)$$

Thus $b_z(y)$ is the number of previously recorded edges in row y with output z that become relevant because of the new first-block answer. After inserting the new edge, n_z becomes $n_z + \mathbf{1}[z = y]$, while r_z becomes $r_z + b_z(y) + \mathbf{1}[z = y]$. Expanding the product gives

$$\begin{aligned} K_a(H[x \leftarrow y]) &= K_0 + 1 + w(y), \\ w(y) &= n_y + r_y + b_y(y) + \sum_z n_z b_z(y). \end{aligned}$$

The first three terms in $w(y)$ have total at most $3|H|$ when summed over y. For the last term, fix z. The edges counted by $b_z(y)$ for different rows y are distinct, so $\sum_y b_z(y) \leq n_z$. Hence

$$\sum_y w(y) \leq 3|H| + \sum_z n_z^2 \leq 4\Gamma(H) \leq 4\gamma.$$

This is the point at which the bound on Γ is used.

In both cases, the local value of K_a is K_0 on $\perp$ and $K_0 + c + w(y)$ on a non-$\perp$ value y, where $0 \leq c \leq 1$ and $\sum_y w(y) \leq 4\gamma$. Applying (40) gives

$$\left\| \sqrt{w}\, R_u \left|\eta\right\rangle \right\| \leq \left\| \sqrt{w} \left|\eta\right\rangle \right\| + 8\sqrt{\frac{\gamma}{N}} \left\| \left|\eta\right\rangle \right\|$$

Combining this with unitarity for the K_0 contribution and the bound $c \leq 1$ yields

$$\begin{aligned} \left\| \sqrt{K} R_u \left|\eta\right\rangle \right\|^2 &\leq (K_0 + 1) \left\| R_u \left|\eta\right\rangle \right\|^2 + \left\| \sqrt{w} R_u \left|\eta\right\rangle \right\|^2 \\ &\leq (K_0 + 1) \left\| \left|\eta\right\rangle \right\|^2 + \left(\left\| \sqrt{w} \left|\eta\right\rangle \right\| + 8\sqrt{\frac{\gamma}{N}} \left\| \left|\eta\right\rangle \right\| \right)^2 \\ &\leq \left\| \sqrt{K} \left|\eta\right\rangle \right\|^2 + \left(1 + \frac{64\gamma}{N}\right) \left\| \left|\eta\right\rangle \right\|^2 + 16\sqrt{\frac{\gamma}{N}} \left\| \left|\eta\right\rangle \right\| \left\| \sqrt{w} \left|\eta\right\rangle \right\|. \end{aligned}$$

Summing over the mutually orthogonal local blocks, the ordinary squared input norms sum to at most one. The w-weighted squared input norms sum to at most E_{j-1}. Cauchy-Schwarz therefore gives (41).

To solve the recurrence, put $\alpha^2 := 4\gamma/N$ and $\overline{E}_j := 256(j + \alpha^2 j^2)$. If $E_j \leq \overline{E}_j$, then

$$8\alpha\sqrt{E_j} \leq 128\alpha\sqrt{j} + 128\alpha^2 j \leq 64 + 192\alpha^2 j,$$

while $\overline{E}_{j+1} - \overline{E}_j = 256 + 256\alpha^2 + 512\alpha^2 j$. Thus induction gives $E_j \leq \overline{E}_j$ for all j. Summing this bound over $0 \leq j < t$ proves the second inequality in (42). □

WHICH ANSWERS TO ONE QUERY CAN CREATE A COLLISION? We now return to the question that motivated J and K_a. Fix the salt a, the queried coordinate $x = (s, m)$, $H := D^{-x}$. Assume that H does not already contain a recorded two-block collision.

Recall that $O_a(H)$ is the set of outputs of relevant edges, and $P_a(H)$ is the set of rows containing an edge whose output lies in $O_a(H)$. By (20),

$$|O_a(H)| \le K_a(H), \qquad |P_a(H)| \le K_a(H).$$

For the queried coordinate $x = (s, m)$, define

$$\mathcal{Y}_{a,x,H} := \begin{cases} \varnothing, & s \neq a,\ s \notin \mathcal{R}_a(H), \\ \{s\} \cup O_a(H), & s \neq a,\ s \in \mathcal{R}_a(H), \\ \{a\} \cup O_a(H) \cup \mathcal{L}(H) \cup P_a(H), & s = a. \end{cases} \tag{43}$$

We claim that every non-$\perp$ answer that creates a recorded two-block collision belongs to this set.

If $s \neq a$ and $s \notin \mathcal{R}_a(H)$, the new edge cannot be the last edge of a one- or two-block message, and the set of relevant rows does not change. Hence no collision can be created.

If $s \neq a$ and $s \in \mathcal{R}_a(H)$, the new edge is relevant. By Lemma 4.3, it can create a collision only by becoming a self-loop, which requires answer s, or by taking an output already used by a relevant edge, which requires an answer in $O_a(H)$.

Finally suppose $s = a$. The new edge is relevant, and its answer y may also make row y relevant. The new edge itself can be a self-loop only when $y = a$, and it can match an old relevant output only when $y \in O_a(H)$. Among the old edges of the newly reached row, a self-loop or a within-row collision is possible only when $y \in \mathcal{L}(H)$. If an old edge in row y shares its output with an edge that was already relevant, then $y \in P_a(H)$. These are exactly the possibilities in the collision characterization of Lemma 4.3.

Since $J(H) \le k$ on the part of the state kept by G_2, (20) gives

$$|\mathcal{Y}_{a,x,H}| \le k + 1 + 2K_a(H). \tag{44}$$

Lemma 4.7 (One query creating a recorded collision)**.** *Let $G_2 |\zeta\rangle = |\zeta\rangle$ and suppose $\mathsf{Col}^{\le 2} |\zeta\rangle = 0$. Then*

$$\left\| \mathsf{Col}^{\le 2} G_2 \mathcal{O} |\zeta\rangle \right\|^2 \le \frac{16}{N} \left((k+1) \||\zeta\rangle\|^2 + 2 \left\| \sqrt{K} |\zeta\rangle \right\|^2 \right). \tag{45}$$

Proof. Decompose $|\zeta\rangle$ into the orthogonal local query blocks. In one block, the salt a, the query $x = (s, m)$ and $H := D^{-x}$ is fixed. Deleting the queried coordinate cannot create a collision or increase J, so $H \notin \mathcal{C}_{a,2}$ and $J(H) \le k$.

Leaving the value at the queried coordinate unchanged cannot create a new collision, and changing it to $\perp$ only deletes an edge. Every remaining answer that can create a collision lies in $\mathcal{Y}_{a,x,H}$, whose size is bounded by (44). Applying Lemma 3.8 in this block gives squared output norm at most $\frac{16}{N}(k + 1 + 2K_a(H))$ times the squared input norm. The output projection G_2 can only reduce that norm. Summing over the local blocks and using $K_a(H) \le K_a(D)$ for every input database D in the block proves the claim. □

Lemma 4.8 (Accumulating the online queries)**.** *For the projected evolution (37) and every $t \ge 1$,*

$$\left\| \mathsf{Col}^{\le 2} \left| v_t^{(2)} \right\rangle \right\|^2 \le \frac{16}{N} \left((k+1)t^2 + 256t^3 + \frac{2048\gamma t^4}{3N} \right). \tag{46}$$

Proof. Set $b_j := \left\| \mathsf{Col}^{\le 2} \left| v_j^{(2)} \right\rangle \right\|$. Since the salt row is empty on the support of $v_0^{(2)}$, we have $b_0 = 0$.

Before the j-th query, split the state into the component that already contains a recorded collision and its orthogonal complement. Because the projected query is a contraction, the first component contributes at most b_{j-1}. Apply Lemma 4.7 to the second component. The projection onto the collision-free component cannot increase either the ordinary norm or the quantity in (38). Let us formally show the above steps:

$$\begin{aligned} b_j &= \left\| \mathsf{Col}^{\le 2} \left| v_j^{(2)} \right\rangle \right\| = \left\| \mathsf{Col}^{\le 2} G_2 U_j \mathcal{O} \left| v_{j-1}^{(2)} \right\rangle \right\| = \left\| \mathsf{Col}^{\le 2} G_2 \mathcal{O} \left| v_{j-1}^{(2)} \right\rangle \right\| \\ &\le \left\| \mathsf{Col}^{\le 2} G_2 \mathcal{O} \mathsf{Col}^{\le 2} \left| v_{j-1}^{(2)} \right\rangle \right\| + \left\| \mathsf{Col}^{\le 2} G_2 \mathcal{O} \left(I - \mathsf{Col}^{\le 2} \right) \left| v_{j-1}^{(2)} \right\rangle \right\| \\ &\le \left\| \mathsf{Col}^{\le 2} \left| v_{j-1}^{(2)} \right\rangle \right\| + \frac{4}{\sqrt{N}} \sqrt{(k+1) \left\| \left(I - \mathsf{Col}^{\le 2} \right) \left| v_{j-1}^{(2)} \right\rangle \right\|^2 + 2 \left\| \sqrt{K} \left(I - \mathsf{Col}^{\le 2} \right) \left| v_{j-1}^{(2)} \right\rangle \right\|^2} \\ &\le b_{j-1} + \frac{4}{\sqrt{N}} \sqrt{k + 1 + 2E_{j-1}} \end{aligned}$$

Summing this inequality and applying Cauchy-Schwarz gives

$$b_t^2 \le \frac{16t}{N} \sum_{j=0}^{t-1} (k + 1 + 2E_j).$$

Substituting (42) proves (46). □

Conditioning on offline event. The previous lemma analyzes a state already restricted to $J(D) \le k$ and $\Gamma(D) \le \gamma$. We now compare this projected computation with the state produced by an arbitrary offline procedure after conditioning on an arbitrary offline event.

The advice-removal theorem will apply this bound to a computation that already includes verification of the two output messages. We therefore count those verification queries as part of the online query budget below.

Theorem 4.9 (Two-block bound after conditioning)**.** *Consider an offline algorithm initialized independently of h that makes at most Q quantum queries and an event $\mathcal{W}$ determined during the offline phase with probability $p > 0$. Condition on $\mathcal{W}$, retain the resulting state, and sample a fresh independent uniform salt. Suppose the subsequent online computation, including the evaluation of the two candidate messages used to verify a collision, makes at most t oracle queries in total. Put*

$$R := Q + t, \qquad \theta := \ln(1/p) + 40 \ln(2NR).$$

Then

$$\begin{aligned} \Pr[\text{win} \mid \mathcal{W}] = O\Big(\frac{Q+1}{N} + \frac{t^2}{N} \left(\frac{R}{N^{1/3}} + \theta + 1 \right) + \frac{t^3}{N} \\ + \frac{t^4}{N^2} \left(R + \frac{R^2}{N} + \theta^2 \right) \Big). \end{aligned} \tag{47}$$

The constant in the $O(\cdot)$ notation is universal.

Proof. Use the values of k and γ from (34) with the R and θ above. By (35), at the start of the online phase and after every subsequent query, the corresponding unconditioned execution has norm at most $2e^{-\theta/2}$ outside G_2. After conditioning on $\mathcal{W}$, this becomes

$$\eta := \frac{2e^{-\theta/2}}{\sqrt{p}} = 2(2NR)^{-20}.$$

Apply the conditional projection comparison Lemma 3.11 with $G = G_2$ and online budget t. Its initial projected vector has norm at most one, has an empty salt row, and contains at most Q recorded coordinates. It therefore satisfies the hypotheses of Lemma 4.8. Since the last part of the online computation verifies the two candidate messages, the common output-recording lemma, Lemma 3.10, gives

$$\sqrt{\Pr[\text{win} \mid \mathcal{W}]} \le (t+1)\eta + \sqrt{\frac{Q}{N}} + \left\| \mathsf{Col}^{\le 2} \left| v_t^{(2)} \right\rangle \right\| + \frac{2}{\sqrt{N}}. \tag{48}$$

The term $\sqrt{Q/N}$ comes from the possibility that the fresh salt row was already touched during the offline phase; it is not multiplied by $1/\sqrt{p}$. The final $2/\sqrt{N}$ is the statewise output-recording error for two messages of length at most two.

Square (48), apply (46), and then use (36). The term $(t+1)^2\eta^2$ is smaller than $O(1/N)$ and is absorbed by the displayed bound. This proves (47). □

Applying the quantum-advice reduction. We now prove the polynomial-advice term in Theorem 4.1. Assume first that $S \ge 1$. The desired expression contains two terms, so write

$$x := \frac{S\tau^2 \log_2 N}{N}, \qquad y := \frac{\tau^3}{N}.$$

If $x + y \ge 1$, the trivial upper bound one already proves the claim. We may therefore assume $x + y < 1$.

The advice-removal theorem produces a conditional online computation using at most $\tau = T + 4$ queries in total: the original T online queries followed by the four queries used to evaluate the two candidate messages. We therefore apply Theorem 4.9 with

$$t = \tau, \qquad Q = 2S\tau, \qquad R = Q + t = (2S+1)\tau.$$

Since $y < 1$, we have $\tau < N^{1/3}$. Since $x < 1$ and $S \ge 1$,

$$R \le 3S\tau < \frac{3N}{\tau \log_2 N} \le N.$$

Every conditioning event required by Theorem 3.1 has $p \ge N^{-2S}$. Hence

$$\theta = \ln(1/p) + 40 \ln(2NR) = O(S \log_2 N).$$

Substituting these estimates into (47) gives, uniformly over all the conditioning events used by the advice-removal theorem,

$$\begin{aligned} \Pr[\text{win} \mid \mathcal{W}] &= O\left(x + y + \frac{S\tau^5}{N^2} + \frac{S^2\tau^4 \log_2^2 N}{N^2} \right) \\ &= O(x + y + xy + x^2) = O(x + y). \end{aligned}$$

Here $S\tau^5/N^2 = xy/\log_2 N \le xy$, while the final term equals x^2 up to a universal constant. Since $x + y < 1$, both xy and x^2 are absorbed by $x + y$.

The alternative $1/N$ in Theorem 3.1 is also absorbed, because $y = \tau^3/N \ge 1/N$. We conclude that

$$\mathrm{Adv}^{\mathsf{MD\text{-}2col\text{-}q}}_{N,M}(S,T) = O\left(\min\left\{ 1, \frac{S\tau^2 \log_2 N + \tau^3}{N} \right\} \right). \tag{49}$$

Oracle-independent input. For $S = 0$, a simpler argument gives the required $O(\tau^3/N)$ bound. Before the $(j+1)^{\text{th}}$ query, assume the database contains no equal-output pair and no self-loop. At most j output values have already been recorded, and one additional answer: the source of the queried coordinate, would create a self-loop. Hence at most $j+1$ non-$\perp$ answers can create the first equal-output pair or self-loop. By Lemma 3.8, the norm entering this event in query $j+1$ is at most $4\sqrt{(j+1)/N}$. Summing over $j < \tau$ gives recorded norm $O(\tau^{3/2}/\sqrt{N})$. By Lemma 4.3, every recorded two-block collision is contained in this event. The common output-recording lemma, Lemma 3.10, therefore gives success $O(\tau^3/N)$.

A complementary bound for small advice. The theorem also states the bound $O(2^S\tau^3/N)$. This follows by comparing arbitrary advice with the maximally mixed state.

Lemma 4.10 (Replacing the advice by an oracle-independent state)**.** *For every $S \geq 0$,*

$$\mathrm{Adv}_{N,M}^{\mathsf{MD\text{-}2col\text{-}q}}(S,T) = O\left(\min\left\{1, \frac{2^S(T+4)^3}{N}\right\}\right). \tag{50}$$

Proof. Fix the compression function h, and let M^h be the positive semidefinite acceptance operator on the S-qubit advice register from (3). By Lemma 3.12,

$$\mathbb{E}_h \operatorname{Tr}(M^h \rho_h) \leq 2^S \mathbb{E}_h \operatorname{Tr}\left(M^h \frac{I}{2^S}\right).$$

The state $I/2^S$ is independent of h. The oracle-independent bound proved above applies to the experiment on the right and gives $O(\tau^3/N)$. Multiplying by 2^S and capping the result at one proves the lemma. □

Taking the smaller of (49), (50), and the trivial bound one proves (17).

4.1 Classical advice and matching attacks

We now show that the two-block upper bound is tight, up to logarithmic factors, already against classical advice. The matching attack applies for every $M \geq N$: restrict all block choices in both preprocessing and the online attack to $[N]$.

Using preprocessing. Choose arbitrary distinct public states $v_1, \ldots, v_r \in [N]$, where $r \leq \lfloor N/2 \rfloor$. During preprocessing, for each v_i find two distinct blocks $b_i, b_i' \in [N]$ such that

$$h(v_i, b_i) = h(v_i, b_i').$$

Since the restriction of $h(v_i, \cdot)$ to $[N]$ is a uniformly random function from $[N]$ to $[N]$, it is injective with probability $\frac{N!}{N^N} \leq e^{-(N-1)/2}$.

Hence, by a union bound over the $r \leq N/2$ selected rows, except with negligible probability every selected row contains the required pair.

Store (b_i, b_i') for each i. This uses $2r\lceil \log_2 N \rceil$ bits of classical advice; the states v_i are fixed publicly and need not be stored.

After the fresh salt a is revealed, use quantum search to find a block $m \in [M]$ such that

$$h(a, m) \in \{v_1, \ldots, v_r\}.$$

Condition on $a \notin \{v_1, \ldots, v_r\}$, which occurs with probability at least $1/2$. Then the restriction of $h(a, \cdot)$ to $[N]$ is independent of the preprocessing information, and each block satisfies this

condition with probability r/N. Thus, using $O(T)$ queries, quantum search succeeds with probability $\Omega\left(\min\left\{1, \frac{rT^2}{N}\right\}\right)$ in the nontrivial regime $T^2 \leq N$.

If $h(a, m) = v_i$, output

$$(m, b_i) \qquad \text{and} \qquad (m, b_i').$$

The two messages are distinct and satisfy

$$\mathrm{MD}_h(a, (m, b_i)) = h(v_i, b_i) = h(v_i, b_i') = \mathrm{MD}_h(a, (m, b_i')).$$

Taking

$$r = \min\left\{\left\lfloor \frac{S}{2\lceil \log_2 N\rceil}\right\rfloor, \left\lfloor \frac{N}{2}\right\rfloor\right\}$$

gives success

$$\widetilde{\Omega}\left(\min\left\{1, \frac{ST^2}{N}\right\}\right),$$

where the tilde hides logarithmic factors in N.

Without preprocessing. Independently, after receiving the salt a, the adversary may apply the standard quantum collision-finding algorithm [BHT98] to the random function

$$m \longmapsto h(a, m).$$

For $M \geq N$, this gives success probability

$$\Omega\left(\min\left\{1, \frac{T^3}{N}\right\}\right)$$

using $O(T)$ queries and no advice.

Combining the two attacks, and observing that the maximum of the two success probabilities is within a constant factor of their sum, gives

$$\mathrm{Adv}_{N,M}^{\mathsf{MD\text{-}2col\text{-}q}}(S, T) = \widetilde{\Theta}\left(\min\left\{1, \frac{ST^2 + T^3}{N}\right\}\right), \qquad M \geq N, \tag{51}$$

where the upper bound holds for arbitrary S-qubit advice, while the matching lower bound uses only classical advice.

Thus, for $M \geq N$, arbitrary oracle-dependent quantum advice gives no asymptotic advantage over classical advice for finding two-block Merkle-Damgård collisions.

5 The general B-block case

We now consider messages of length at most B, where $B > 2$. The main new feature is that the answer to one query can make an entire previously recorded path useful. More precisely, if a query changes the value at a coordinate (s, m) to y, then a message can traverse this new edge and continue from y along recorded edges for several more blocks. Thus the one-step description used in the two-block proof is no longer sufficient.

To control these longer continuations, we first restrict attention to databases with only a bounded amount of two kinds of structure that make backward path counting difficult: repeated outputs and short directed cycles. We denote this set of databases by $\mathcal{G}_{k,B}$ and prove below that the component outside it has small norm after a bounded number of queries. From a database in $\mathcal{G}_{k,B}$ we can then

delete only a few recorded edges so that the remaining graph has indegree at most one and no short directed cycle, while preserving every path of length at most B already reachable from the salt. After this deletion, paths can be traced backwards uniquely. This will let us count, for one query, how many answers can make a new path interact with the part of the graph already reachable from the salt.

Theorem 5.1 (General-B collisions with quantum advice)**.** *Let $N \geq 2$, $M \geq 1$, $S \geq 0$, $T \geq 1$, and $B > 2$ with $B \leq T$. Let*

$$\tau := T + 2B, \qquad R := (2S+1)\tau, \qquad \Lambda := \lceil \log_2(2NR) \rceil.$$

Define

$$\mathcal{F}_B := \frac{(B+1)\tau^2}{N}\left(\frac{R^{3/2}}{\sqrt{N}} + \Lambda + S\log_2 N\right) + \frac{\tau^3}{N}. \tag{52}$$

Then

$$\mathrm{Adv}^{\mathsf{MD}\text{-}B\mathsf{col}\text{-}\mathsf{q}}_{N,M}(S,T) = O\left(\min\left\{1,\ \frac{2^S\tau^3}{N},\ \mathcal{F}_B\right\}\right). \tag{53}$$

If $(2S+1)\tau^3 \leq N$, then

$$\mathrm{Adv}^{\mathsf{MD}\text{-}B\mathsf{col}\text{-}\mathsf{q}}_{N,M}(S,T) = O\left(\min\left\{1,\ \frac{2^S\tau^3}{N},\ \frac{(B+1)S\tau^2\log_2 N + \tau^3}{N}\right\}\right). \tag{54}$$

Since $B \leq T$, the last term is $\widetilde{O}((BST^2+T^3)/N)$.

Unlike the two-block theorem, this bound is not known to be tight for $B > 2$.

We have compiled a table of notations which are used in this section in Table 3 of Appendix A, which may be helpful for the reader while going through the proof.

Restricting the database structure. Before analyzing paths from the salt, we isolate the part of the compressed oracle state whose recorded graph has controlled global structure. The two statistics below are tailored to the general-B argument: one measures how many repeated outputs must be removed to obtain indegree at most one, and the other measures how many short cycles must be broken. Neither statistic is needed by the sharper two-block proof, which is why this material is placed here rather than in the common preliminaries.

Fix a total query bound $R \geq 1$. The statements in this subsection apply to any compressed-oracle execution initialized independently of h and making at most R queries, later we instantiate R with the total query budget of the relevant offline/online experiment. We write $\varnothing$ for the empty database, namely the database satisfying $D(x) = \perp$ for every coordinate x.

Bounding the structural bad events. We begin by defining the two database statistics. For $y \in [N]$, like in Section 4, let $n_y(D) := |\{x \in \mathrm{Dom}(D) : D(x) = y\}|$, and define

$$c(D) := \sum_{y \in [N]} \left\lfloor \frac{n_y(D)}{2} \right\rfloor.$$

An *equal-output pair* is a pair of distinct recorded coordinates x, x' satisfying $D(x) = D(x')$. A collection of such pairs is *disjoint* if no recorded coordinate occurs in two pairs. For each y, the quantity $\lfloor n_y(D)/2 \rfloor$ is the maximum number of disjoint equal-output pairs having output y. Hence $c(D)$ is the maximum number of disjoint equal-output pairs in the entire database. A *directed cycle* is a nonempty recorded path whose first and last vertices are equal and which has no other repeated

vertex. The length of a cycle is the number of edges in it. In particular, a self-loop is a directed cycle of length one. Two directed cycles are *edge-disjoint* if they share no recorded coordinate. Let $z_B(D)$ be the maximum number of pairwise edge-disjoint directed cycles in D whose lengths are at most B. Fix an integer $k \geq 0$ and define

$$\mathcal{G}_{k,B} := \{D : c(D) \leq k \text{ and } z_B(D) \leq k\}.$$

Define $\Pi_{\mathcal{G}_{k,B}} := \Pi_{\{(a,D):D\in\mathcal{G}_{k,B}\}}$ (understood as tensored with the identity on all other registers). Deleting recorded coordinates cannot increase either $c(D)$ or $z_B(D)$. Indeed, deleting a coordinate decreases one of the multiplicities $n_y(D)$ by one, so it cannot increase $c(D)$. Likewise, every directed cycle present after deleting coordinates was already present before the deletion. Hence any family of pairwise edge-disjoint cycles in the smaller database is also such a family in the original database, and therefore $z_B(D)$ cannot increase. Both statistics will be bounded using the same generic argument. The next lemma isolates that argument.

Lemma 5.2. *Let f be a nonnegative integer-valued function of the database satisfying $f(\varnothing) = 0$. Suppose that the following hold.*

(a) For every database D, coordinate x, and $y \in [N] \setminus \{D(x)\}$, changing $D(x)$ to y can increase f by at most one.

(b) Deleting a recorded coordinate cannot increase f.

(c) For every database D and coordinate x, let $r := D(x)$. If $|D^{-x}| \leq q$, then there are at most $z(q)$ values $y \in [N] \setminus \{r\}$ such that $f(D[x \leftarrow y]) > f(D)$.

Define $\Pi^f_{\geq h} := \Pi_{\{(a,D):f(D)\geq h\}}$. Let $|\Phi_q\rangle$ be a normalized state obtained after q compressed-oracle queries starting from the empty database. Between oracle queries, arbitrary unitaries may be applied, provided they do not modify the database register. Define $\Delta_{q,h} := \left\|\Pi^f_{\geq h}|\Phi_q\rangle\right\|$. Then

$$\Delta_{q+1,h+1} \leq \Delta_{q,h+1} + 4\sqrt{\frac{z(q)}{N}}\,\Delta_{q,h}. \tag{55}$$

Moreover, if $z(q) \leq z_*$ for every $q < r$, then for every $0 \leq h \leq q \leq r$,

$$\Delta_{q,h} \leq \binom{q}{h}\left(4\sqrt{\frac{z_*}{N}}\right)^h. \tag{56}$$

For $h > q$, we have $\Delta_{q,h} = 0$.

Proof. For $h \geq 0$, let $E_h := \Pi^f_{\geq h} - \Pi^f_{\geq h+1}$. Since $\{(a,D) : f(D) \geq h\}$ contains $\{(a,D) : f(D) \geq h+1\}$, the operator E_h is a projector onto basis components whose database satisfies $f(D) = h$. Let U_q denote the unitary applied between the q-th and $(q+1)$-st oracle queries, and set $|\Psi_q\rangle := U_q|\Phi_q\rangle$. Since U_q does not modify the database register, it commutes with every $\Pi^f_{\geq h}$. Hence $\left\|\Pi^f_{\geq h}|\Psi_q\rangle\right\| = \Delta_{q,h}$ for every h. The state immediately after the next oracle query is $|\Phi_{q+1}\rangle = \mathcal{O}|\Psi_q\rangle$. Using $I = \Pi^f_{\geq h+1} + E_h + (I - \Pi^f_{\geq h})$, we have

$$\Pi^f_{\geq h+1}|\Phi_{q+1}\rangle = \Pi^f_{\geq h+1}\mathcal{O}\Pi^f_{\geq h+1}|\Psi_q\rangle + \Pi^f_{\geq h+1}\mathcal{O}E_h|\Psi_q\rangle + \Pi^f_{\geq h+1}\mathcal{O}(I - \Pi^f_{\geq h})|\Psi_q\rangle. \tag{57}$$

We first show that the last term in (57) is zero. Consider a basis state in the image of $I - \Pi^f_{\geq h}$, with database D. Thus $f(D) \leq h-1$. A compressed-oracle query at coordinate x can only replace

$D(x)$ by some $t \in [N] \cup \{\perp\}$. If $t = D(x)$, the database is unchanged. If $t = \perp$, the resulting database is D^{-x}, and assumption (b) gives $f(D^{-x}) \le f(D) \le h-1$. Finally, if $t \in [N]$ and $t \neq D(x)$, assumption (a) gives $f(D[x \leftarrow t]) \le f(D) + 1 \le h$. Thus no basis state in the image of $I - \Pi^f_{\ge h}$ can be mapped in one query to a basis state in the image of $\Pi^f_{\ge h+1}$. Therefore $\Pi^f_{\ge h+1}\mathcal{O}(I - \Pi^f_{\ge h}) = 0$.

We next bound the second term in the RHS of (57). Every basis state in the image of E_h has a database D satisfying $f(D) = h$. Since the state is obtained after q oracle queries, every such D contains at most q recorded coordinates, and hence $|D^{-x}| \le q$ for every queried coordinate x. By assumption (c), there are therefore at most $z(q)$ values $y \in [N] \setminus \{D(x)\}$ for which $f(D[x \leftarrow y]) > h$. By assumption (b), replacing $D(x)$ by $\perp$ cannot increase f. Applying (10) with $\mathcal{P} := \{(a, D) : f(D) = h\}$ and $\mathcal{P}' := \{(a, D) : f(D) \le h\}$, and observing that $\Pi_{\mathcal{P}} = E_h$ and $\Pi_{\mathcal{P}'} = I - \Pi^f_{\ge h+1}$, gives

$$\left\|\Pi^f_{\ge h+1}\mathcal{O}E_h\Pi_{|D|\le q}\right\| \le 4\sqrt{\frac{z(q)}{N}}.$$

Since $E_h^2 = E_h$ and $|\Psi_q\rangle$ has databases of size at most q in its support,

$$\begin{aligned}\left\|\Pi^f_{\ge h+1}\mathcal{O}E_h\,|\Psi_q\rangle\right\| &= \left\|(\Pi^f_{\ge h+1}\mathcal{O}E_h)E_h\,|\Psi_q\rangle\right\| \\ &\le 4\sqrt{\frac{z(q)}{N}}\,\left\|E_h\,|\Psi_q\rangle\right\|.\end{aligned}$$

Since $\Pi^f_{\ge h+1}$ and $\mathcal{O}$ both have operator norm one, using (57)

$$\begin{aligned}\Delta_{q+1,h+1} &= \left\|\Pi^f_{\ge h+1}\,|\Phi_{q+1}\rangle\right\| \\ &\le \left\|\Pi^f_{\ge h+1}\,|\Psi_q\rangle\right\| + 4\sqrt{\frac{z(q)}{N}}\,\left\|E_h\,|\Psi_q\rangle\right\| \\ &\le \Delta_{q,h+1} + 4\sqrt{\frac{z(q)}{N}}\,\Delta_{q,h},\end{aligned}$$

where the last inequality uses $E_h \preccurlyeq \Pi^f_{\ge h}$. This proves (55). For the second claim, let $\alpha := 4\sqrt{z_*/N}$. Since the initial database is empty and $f(\varnothing) = 0$, we have $\Delta_{0,0} = 1$ and $\Delta_{0,h} = 0$ for every $h \ge 1$. We prove by induction on q that

$$\Delta_{q,h} \le \binom{q}{h}\alpha^h \qquad \text{for every } 0 \le h \le q.$$

The claim holds for $q = 0$. Suppose it holds for some $q < r$. For $0 \le h \le q$, (55) and $z(q) \le z_*$ give

$$\begin{aligned}\Delta_{q+1,h+1} &\le \Delta_{q,h+1} + \alpha\Delta_{q,h} \\ &\le \binom{q}{h+1}\alpha^{h+1} + \binom{q}{h}\alpha^{h+1} \\ &= \binom{q+1}{h+1}\alpha^{h+1},\end{aligned}$$

where the term $\binom{q}{h+1}$ is interpreted as 0 when $h = q$. The case $h = 0$ follows from $\Delta_{q+1,0} \le \||\Phi_{q+1}\rangle\| = 1$. This completes the induction and proves (56). □

We now apply the lemma to $c(D)$ and $z_B(D)$.

Lemma 5.3. *Let $\kappa := k+1$ and define*

$$\varepsilon_{R,k} := \begin{cases} \min\left\{1,\, 2\binom{R}{\kappa}\left(4\sqrt{\dfrac{R}{N}}\right)^{\kappa}\right\}, & \kappa \le R, \\ 0, & \kappa > R. \end{cases} \tag{58}$$

At every point in an execution in which at most R oracle queries have been made, the norm of the state component whose database does not lie in $\mathcal{G}_{k,B}$ is at most $\varepsilon_{R,k}$.

Proof. We apply Lemma 5.2 separately to $c(D)$ and $z_B(D)$. First, consider $c(D)$. Replacing the value stored at one coordinate removes at most one occurrence of one output value and adds at most one occurrence of another, so $c(D)$ can increase by at most one. Deleting a coordinate cannot increase $c(D)$. Finally, fix a database D and queried coordinate x. Changing $D(x)$ to $y \in [N]$ can increase $c(D)$ only if y already occurs as the output of some coordinate in D^{-x}. If $|D^{-x}| \le q$, there are at most q such values y. Thus Lemma 5.2 applies to $c(D)$ with $z(q) \le q$.

Now consider $z_B(D)$. We first show that changing the value stored at one coordinate can increase $z_B(D)$ by at most one. Fix a coordinate $x = (s, m)$ and a new value $y \in [N]$, and let $D' := D[x \leftarrow y]$. Let $\mathcal{C}$ be any family of pairwise edge-disjoint directed cycles of length at most B in D'. Since the cycles in $\mathcal{C}$ are edge-disjoint, at most one of them can use the edge corresponding to the coordinate x. After removing this cycle, if it exists, every remaining cycle uses only coordinates different from x, and hence is already present in D^{-x}. Since D^{-x} is obtained from D by deleting a coordinate, every such cycle is also present in D. Therefore $|\mathcal{C}| - 1 \le z_B(D)$, and hence $|\mathcal{C}| \le z_B(D) + 1$. Since this holds for every such family $\mathcal{C}$ in D', we obtain $z_B(D') \le z_B(D) + 1$. Deleting a coordinate cannot increase $z_B(D)$ either. Indeed, every directed cycle present in D^{-x} is also present in D, so every family of pairwise edge-disjoint directed cycles in D^{-x} is also such a family in D. Hence $z_B(D^{-x}) \le z_B(D)$. It remains to bound the number of values y that can increase $z_B(D)$. Fix $x = (s, m)$ and suppose $z_B(D[x \leftarrow y]) > z_B(D)$. Then some cycle in a maximum edge-disjoint family in $D[x \leftarrow y]$ must use the edge $s \xrightarrow{m} y$, otherwise the entire family would already be present in D^{-x}, and hence in D. Removing this edge from that cycle leaves a recorded path in D^{-x} from y to s of length at most $B - 1$. If this path has length zero, then $y = s$. Otherwise, y is the source vertex of its first edge. If D^{-x} contains at most q recorded coordinates, there are at most q source vertices of recorded edges, and hence at most $q + 1$ possible values of y that can increase $z_B(D)$. Thus Lemma 5.2 applies to $z_B(D)$ with $z(q) \le q + 1$. Before any of the first R oracle queries, we have $q + 1 \le R$. Hence, for both statistics, we may use $z_* = R$ in (56). Taking $h = k + 1 = \kappa$ shows that the norm of the component satisfying $c(D) \ge \kappa$ is at most $\binom{R}{\kappa}(4\sqrt{R/N})^{\kappa}$, and the same bound holds for the component satisfying $z_B(D) \ge \kappa$. Since a database lies outside $\mathcal{G}_{k,B}$ only if at least one of these two events occurs, the triangle inequality gives (58). If $\kappa > R$, every basis database obtained after at most R queries has at most R recorded coordinates and therefore cannot contain κ disjoint equal-output pairs or κ edge-disjoint cycles. In this case the component outside $\mathcal{G}_{k,B}$ is zero. □

Deleting a few edges while preserving all short paths. For the analysis below, we will use a slightly larger collision event. Define the evaluation of the empty message by $\mathrm{MD}_a^D(\emptyset) := a$. Let $\mathcal{C}_{a,B}^+$ be the set of databases D for which there exist distinct messages $\alpha, \alpha' \in \{\emptyset\} \cup [M]^+$, $|\alpha|, |\alpha'| \le B$, such that both evaluations from the salt a are recorded in D and $\mathrm{MD}_a^D(\alpha) = \mathrm{MD}_a^D(\alpha')$. Since $\mathcal{C}_{a,B}$ only allows nonempty messages, $\mathcal{C}_{a,B} \subseteq \mathcal{C}_{a,B}^+$.

We introduce the larger event only because it gives the following useful property: if $D \notin \mathcal{C}^{+}_{a,B}$, then no two distinct recorded paths of length at most B starting from a have the same endpoint. In particular, no nonempty recorded path of length at most B can return to a, since it would have the same endpoint as the empty path.

Lemma 5.4 (Uniqueness of short paths)**.** *Let C be a recorded graph with indegree at most one and no directed cycle of length at most B. If two paths in C have the same starting vertex, the same endpoint, and length at most B, then the two paths are identical.*

Proof. Trace the two paths backwards from their common endpoint. Since every vertex has at most one incoming edge, the paths agree for as long as both still have an edge left. Hence the shorter path is a suffix of the longer. If the lengths are equal, the paths are identical. Otherwise the extra prefix of the longer path starts and ends at the common starting vertex, so it contains a directed cycle of length at most B, a contradiction. □

The next lemma extracts the structural consequence of membership in $\mathcal{G}_{k,B}$ that is needed in the online analysis. Starting from a database with few equal-output pairs and few short cycles, we delete a small set of recorded coordinates so that the remaining graph has indegree at most one and no directed cycle of length at most B. Crucially, the deleted coordinates can be chosen outside every recorded path of length at most B starting from the salt.

Lemma 5.5 (Deleting edges while preserving the short paths from a)**.** *Let $a \in [N]$. Suppose $H \in \mathcal{G}_{k,B}$ and $H \notin \mathcal{C}^{+}_{a,B}$. There is a set $E \subseteq \mathrm{Dom}(H)$ with $|E| \le 3k$ such that, for $C := H \setminus E$,*

(i) every vertex of C has indegree at most one,

(ii) C contains no directed cycle of length at most B, and

(iii) no edge whose source lies in $\mathrm{Reach}_{H}^{\le B-1}(a)$ is deleted.

Consequently,

$$\mathrm{Reach}_{C}^{\le B}(a) = \mathrm{Reach}_{H}^{\le B}(a).$$

Proof. Consider all recorded edges whose sources lie in $\mathrm{Reach}_{H}^{\le B-1}(a)$. Every such edge occurs on a recorded path of length at most B from a, so these are precisely the edges that we must not delete.

Because $H \notin \mathcal{C}^{+}_{a,B}$, every vertex in $\mathrm{Reach}_{H}^{\le B-1}(a)$ has a unique short path from a. It follows that two of the edges just described cannot enter the same vertex: otherwise appending the two edges to their respective paths from a would give two different short paths to the same endpoint. These edges also contain no directed cycle, since following such a cycle would give two different short paths to one of its vertices.

Now consider each vertex of indegree one or more in H. Keep one incoming edge, choosing the edge that lies on a short path from a whenever such an edge exists, and delete all the other incoming edges. If a vertex has indegree $n \ge 2$, this deletes $n - 1 \le 2\lfloor n/2 \rfloor$ edges. Summing over all vertices, at most $2c(H) \le 2k$ edges are deleted. None of the short paths from a is affected.

The remaining graph has indegree at most one. Distinct directed cycles in such a graph are vertex-disjoint: if two cycles shared a vertex, tracing the unique incoming edges backwards would force the two cycles to be the same. Hence the cycles of length at most B form an edge-disjoint family already present in H, and there are at most $z_B(H) \le k$ of them. None of these cycles can consist entirely of edges on short paths from a, because those edges contain no directed cycle. Delete one other edge from each short cycle. This uses at most another k deletions and proves all three claims. □

We will need three elementary counting consequences of indegree at most one.

Lemma 5.6 (Counting short paths backwards)**.** *Let $a \in [N]$, C be a database having indegree of all vertices at most one, and set $\mathcal{R} := \mathrm{Reach}_C^{\le B}(a)$.*

(i) The number of vertices y from which a path of length at most B in C reaches some vertex of $\mathcal{R}$ is at most $|\mathcal{R}| + B$. Equivalently,

$$\left|\{y \in [N] : \mathrm{Reach}_C^{\le B}(y) \cap \mathcal{R} \neq \varnothing\}\right| \le |\mathcal{R}| + B.$$

(ii) For every set of vertices $Z \subseteq [N]$,

$$\left|\{y \in [N] : \mathrm{Reach}_C^{\le B}(y) \cap Z \neq \varnothing\}\right| \le (B+1)|Z|.$$

(iii) Moreover,

$$\sum_{y \in [N]} (|\,\mathrm{Reach}_C^{\le B}(y)| - 1) \le B|C|.$$

Proof. (i) For the first claim, fix y such that $\mathrm{Reach}_C^{\le B}(y) \cap \mathcal{R} \neq \varnothing$, and let $z \in \mathrm{Reach}_C^{\le B}(y) \cap \mathcal{R}$. Since $z \in \mathcal{R} = \mathrm{Reach}_C^{\le B}(a)$, there is also a path from a to z of length at most B.

Because C has indegree at most one, tracing these two paths backwards from z determines the preceding edge uniquely at every step. Hence the two paths agree until the shorter one terminates.

If the path from y terminates first, then y lies on the path from a to z, and therefore $y \in \mathcal{R}$. Otherwise, the path from y contains the entire path from a to z as a suffix. Its remaining prefix is therefore a nonempty path from y to a of length at most B.

For each length $d \in \{1, \dots, B\}$, indegree at most one implies that there is at most one vertex having a path of length exactly d to a. Thus there are at most B such vertices outside $\mathcal{R}$, and hence

$$\left|\{y : \mathrm{Reach}_C^{\le B}(y) \cap \mathcal{R} \neq \varnothing\}\right| \le |\mathcal{R}| + B.$$

(ii) For the second claim, fix $z \in Z$ and a distance $d \in \{0, \dots, B\}$. Tracing incoming edges backwards determines at most one starting vertex at that distance. Hence each z contributes at most $B+1$ possible starting vertices.

(iii) For the third claim, count ordered pairs (y, z) with $y \neq z$ and $z \in \mathrm{Reach}_C^{\le B}(y)$. The endpoint z must be the output of a recorded edge, so there are at most $|C|$ choices for it. For each endpoint there is at most one starting vertex at each positive distance $1, \dots, B$.

□

What one query can change. For a salt a and database D, define

$$V_a(D) := |\,\mathrm{Reach}_D^{\le B}(a)| - 1.$$

Thus $V_a(D)$ is the number of vertices other than a that are already reachable by a recorded message of length at most B. This is the quantity whose average size we will control during the online phase. Deleting coordinates cannot increase it, and $V_a(D) \le |D|$.

Fix integers $Q \ge 0$ and $t \ge 1$, and let $R := Q + t$. Let $|v_0\rangle$ have norm at most one and be supported on databases in $\mathcal{G}_{k,B}$, of size at most Q, with empty row at the salt. Define the projected online evolution by $|v_{j+1}\rangle := \Pi_{\mathcal{G}_{k,B}} U_{j+1} \mathcal{O} \Pi_{\mathcal{G}_{k,B}} |v_j\rangle$ for $0 \le j < t$.

For the analysis we also keep only components that do not yet contain the larger collision event $\mathcal{C}^+_{a,B}$. For $0 \le j \le t$, let

$$\mathcal{P}_j := \{(a, D) : D \in \mathcal{G}_{k,B},\ |D| \le Q + j,\ D \notin \mathcal{C}^+_{a,B}\}, \qquad P_j := \Pi_{\mathcal{P}_j}.$$

The sets $\mathcal{P}_j$ are increasing with j. Moreover, if $(a, D) \in \mathcal{P}_j$, then $(a, D^{-x}) \in \mathcal{P}_j$ for every coordinate x: deleting a recorded coordinate cannot increase $c(D)$ or $z_B(D)$, cannot create a collision, and cannot increase the size of the database.

Every database in the support of $|v_0\rangle$ has an empty row at a. Hence it contains no recorded nonempty path starting from a, and in particular does not belong to $\mathcal{C}^+_{a,B}$. Since these databases also lie in $\mathcal{G}_{k,B}$ and have size at most Q, we have $P_0 |v_0\rangle = |v_0\rangle$.

We now examine one oracle query. Fix the salt a, the queried coordinate $x = (s, m)$, the value $u \in \mathbb{Z}_N$ in the response register, and every database entry except the one at x. Let $H := D^{-x}$, so the query can only change the value stored at x. With everything else fixed, the query acts on this value by the matrix R_u from the preliminaries.

If $s \notin \mathrm{Reach}^{\le B-1}_H(a)$, changing the value at x cannot affect any recorded path of length at most B from a: such a path would have to reach s before using x. We therefore only have to consider the case $s \in \mathrm{Reach}^{\le B-1}_H(a)$.

Delete from H a set E supplied by Lemma 5.5, and write $C := H \setminus E$ and $\mathcal{R} := \mathrm{Reach}^{\le B}_H(a) = \mathrm{Reach}^{\le B}_C(a)$. The choice of E depends only on a and H, not on the old value stored at x.

There is one issue with reasoning only inside C: after assigning a new value y to x, a path starting at y might reach the source of one of the deleted edges in E and then use that edge in the original database. We separate precisely those values. For a set E of coordinates, write $\mathrm{src}(E) := \{s : \exists m,\ (s, m) \in E\}$ for the set of their source vertices. Let

$$Y_H := \{y \in [N] : \mathrm{Reach}^{\le B}_C(y) \cap \mathrm{src}(E) \ne \varnothing\}.$$

Since $|E| \le 3k$, part (ii) of Lemma 5.6 gives $|Y_H| \le 3(B + 1)k$.

By Lemma 3.8, the part of the oracle query that changes the value at x to some $y \in Y_H$ has operator norm at most $\beta := 4\sqrt{3(B + 1)k/N}$. Let A_j denote this part of the query, with its input restricted to the image of P_j, and define

$$\widehat{\mathcal{O}}_j := \Pi_{\mathcal{G}_{k,B}}(\mathcal{O} P_j - A_j).$$

Thus $\widehat{\mathcal{O}}_j$ agrees with the projected oracle query except that the transitions to values in Y_H have been omitted. We have $\|A_j\| \le \beta$, and hence $\left\|\widehat{\mathcal{O}}_j\right\| \le 1 + \beta$ by the triangle inequality.

The operator $\widehat{\mathcal{O}}_j$ is introduced only for the analysis, it need not itself be a valid oracle operation. In particular, deleting selected matrix entries from a unitary operator need not produce an operator of norm at most one. Set $\chi := 1 + \beta$. Since $\left\|\widehat{\mathcal{O}}_j\right\| \le \chi$, the rescaled operator $\chi^{-1}\widehat{\mathcal{O}}_j$ has norm at most one. We will use this rescaled operator below so that each step of the comparison evolution is a contraction, i.e., an operator M satisfying $\|M\| \le 1$.

$$\left\|\widehat{\mathcal{O}}_j\right\| \le \chi, \qquad \left\|\chi^{-1}\widehat{\mathcal{O}}_j\right\| \le 1. \tag{59}$$

The next lemma explains what remains after the values in Y_H have been removed.

Lemma 5.7. *Let $H = D^{-x}$, $H \in \mathcal{G}_{k,B} \setminus \mathcal{C}^+_{a,B}$, let $x = (s, m)$ with $s \in \mathrm{Reach}^{\le B-1}_H(a)$, and let E be the set given by Lemma 5.5. Set*

$$C := H \setminus E, \qquad \mathcal{R} := \mathrm{Reach}^{\le B}_C(a),$$

and

$$Y_H := \{y \in [N] : \mathrm{Reach}_{\widehat{C}}^{\leq B}(y) \cap \mathrm{src}(E) \neq \varnothing\}.$$

Fix $y \in [N] \setminus Y_H$.

(i) Suppose $H[x \leftarrow y] \notin \mathcal{C}_{a,B}^{+}$. *Let* P *be a recorded path in* $H[x \leftarrow y]$ *of length at most* B *starting at* a *and containing the edge* $s \xrightarrow{m} y$. *Then every edge of* P *appearing after* $s \xrightarrow{m} y$ *belongs to* C.

(ii) If $H[x \leftarrow y] \in \mathcal{C}_{a,B}^{+}$, *then*

$$\mathrm{Reach}_{\widehat{C}}^{\leq B}(y) \cap \mathcal{R} \neq \varnothing.$$

Consequently,

$$\left|\left\{y \in [N] \setminus Y_H : H[x \leftarrow y] \in \mathcal{C}_{a,B}^{+}\right\}\right| \leq |\mathcal{R}| + B.$$

Proof. For (i), suppose that some edge of P occurring after $s \xrightarrow{m} y$ does not belong to C, and consider the first such edge. Since $C = H \setminus E$ and H has no entry at x, this edge is either in E or is again the edge $s \xrightarrow{m} y$.

In the first case, the segment of P from y to the source of this edge lies entirely in C. Hence $\mathrm{Reach}_{\widehat{C}}^{\leq B}(y) \cap \mathrm{src}(E) \neq \varnothing$, contradicting $y \notin Y_H$.

In the second case, the segment of P between the two occurrences of $s \xrightarrow{m} y$ gives a path in C from y to s. Since $s \in \mathrm{Reach}_{H}^{\leq B-1}(a)$ and the pruning preserves all paths of length at most B from a, we have $s \in \mathcal{R}$. Thus $\mathrm{Reach}_{\widehat{C}}^{\leq B}(y) \cap \mathcal{R} \neq \varnothing$. Moreover, the prefix of P ending at this second occurrence of s and the path from a to s already present in H are distinct recorded paths of length at most B with the same endpoint. This contradicts $H[x \leftarrow y] \notin \mathcal{C}_{a,B}^{+}$. This proves (i).

For (ii), suppose, towards a contradiction, that $\mathrm{Reach}_{\widehat{C}}^{\leq B}(y) \cap \mathcal{R} = \varnothing$. We show that $H[x \leftarrow y] \notin \mathcal{C}_{a,B}^{+}$.

Any recorded path in $H[x \leftarrow y]$ of length at most B from a that does not use the edge $s \xrightarrow{m} y$ is already present in H. By the choice of E, every such path is also present in C, and hence its endpoint belongs to $\mathcal{R}$.

Now consider a recorded path of length at most B from a that uses $s \xrightarrow{m} y$. We claim that every edge following $s \xrightarrow{m} y$ belongs to C. Otherwise, consider the first subsequent edge outside C. If this edge belongs to E, then the preceding segment gives a path in C from y to $\mathrm{src}(E)$, contradicting $y \notin Y_H$. If it is again $s \xrightarrow{m} y$, then the preceding segment gives a path in C from y to s. Since $s \in \mathcal{R}$, this contradicts $\mathrm{Reach}_{\widehat{C}}^{\leq B}(y) \cap \mathcal{R} = \varnothing$. Thus the endpoint of every path using $s \xrightarrow{m} y$ belongs to $\mathrm{Reach}_{\widehat{C}}^{\leq B}(y)$.

It follows that a path using $s \xrightarrow{m} y$ cannot have the same endpoint as a path avoiding this edge, since their endpoints lie in the disjoint sets $\mathrm{Reach}_{\widehat{C}}^{\leq B}(y)$ and $\mathcal{R}$, respectively.

It remains to compare two paths that both use $s \xrightarrow{m} y$. Their prefixes ending at s must be identical, since otherwise H would contain two distinct recorded paths of length at most B from a to s, contrary to $H \notin \mathcal{C}_{a,B}^{+}$. After the edge $s \xrightarrow{m} y$, both paths lie in C. If they have the same endpoint, their remaining parts are identical by Lemma 5.4. Hence the two paths themselves are identical.

Therefore $H[x \leftarrow y]$ contains no two distinct recorded paths of length at most B from a with the same endpoint. This contradicts the assumption that $H[x \leftarrow y] \in \mathcal{C}_{a,B}^{+}$, proving (ii).

Finally, (ii) implies

$$\left\{y \in [N] \setminus Y_H : H[x \leftarrow y] \in \mathcal{C}_{a,B}^{+}\right\} \subseteq \left\{y \in [N] : \mathrm{Reach}_{\widehat{C}}^{\leq B}(y) \cap \mathcal{R} \neq \varnothing\right\}.$$

The latter set has size at most $|\mathcal{R}| + B$ by Lemma 5.6 part (i), which proves the final claim. □

We can now bound the part of one query that creates a recorded collision without using a value in Y_H. Let $F_j := (I - P_{j+1})\widehat{\mathcal{O}}_j$. Every output of $\widehat{\mathcal{O}}_j$ already lies in $\mathcal{G}_{k,B}$ and contains at most $Q + j + 1$ recorded coordinates, so leaving P_{j+1} is the same as entering $\mathcal{C}^+_{a,B}$. Let $\mathsf{Col}^{\le B,+}$ denote $\Pi_{(a,D):D\in\mathcal{C}^+_{a,B}}$. Hence $F_j = \left(\mathsf{Col}^{\le B,+}\right)\widehat{\mathcal{O}}_j$.

For a fixed H, the preceding lemma says that at most $|\mathcal{R}| + B = V_a(H) + B + 1$ remaining non-$\perp$ values can create a collision. Lemma 3.8 therefore gives the following bound.

Lemma 5.8 (One-query collision bound)**.** *If $P_j\,|\zeta\rangle = |\zeta\rangle$, then*

$$\|F_j\,|\zeta\rangle\|^2 \le \frac{16}{N}\left((B+1)\,\||\zeta\rangle\|^2 + \left\|\sqrt{V}\,|\zeta\rangle\right\|^2\right).$$

Proof. For fixed a, x, u, H, (the proof of) Lemma 3.8 gives squared norm at most $16(V_a(H)+B+1)/N$ times the squared input norm. Summing over the mutually orthogonal choices of the fixed data gives the displayed bound, using that deleting x cannot increase V_a. □

The important feature of this estimate is that the term involving V is not multiplied by B. This is exactly where preserving all previously reachable short paths in Lemma 5.5 is used.

Analysis of online phase. We next control the average value of $V_a(D)$ in an auxiliary evolution that omits the values in Y_H and remains outside the enlarged collision event. Set $|\widehat{v}_0\rangle := |v_0\rangle$ and define

$$|\widehat{v}_{j+1}\rangle := \chi^{-1}U_{j+1}P_{j+1}\widehat{\mathcal{O}}_j\,|\widehat{v}_j\rangle\,.$$

The factor χ^{-1} is present only to make every step a contraction, indeed, (59) implies $\||\widehat{v}_j\rangle\| \le 1$ and $P_j\,|\widehat{v}_j\rangle = |\widehat{v}_j\rangle$.

Define $e_j := \left\|\sqrt{V}\,|\widehat{v}_j\rangle\right\|^2$. Explicitly, e_j is the average value of $V_a(D)$ with respect to the squared amplitudes of the database components of $|\widehat{v}_j\rangle$. Since the salt row is empty initially, $e_0 = 0$. Let $\alpha^2 := (B+1)R/N$.

Lemma 5.9 (Growth of the reachable set)**.** *For $0 \le j < t$,*

$$e_{j+1} \le e_j + 1 + 16\alpha^2 + 8\alpha\sqrt{e_j}. \tag{60}$$

Consequently,

$$e_j \le 256(j + \alpha^2 j^2), \qquad \sum_{j=0}^{t-1} e_j \le 128t^2 + \frac{256\alpha^2 t^3}{3}. \tag{61}$$

Proof. Fix a, the queried coordinate $x = (s, m)$, the response-register value u, and all database entries other than x, and again let $H := D^{-x}$. If $s \notin \mathrm{Reach}_H^{\le B-1}(a)$, changing the value at x does not change V_a, so this case cannot increase e_j. Assume from now on that $s \in \mathrm{Reach}_H^{\le B-1}(a)$.

Let $V_0 := V_a(H)$, and use the graph $C = H \setminus E$ as in Lemma 5.5. Consider a non-$\perp$ value y that is retained by the modified query $\widehat{\mathcal{O}}_j$ and for which the resulting database still lies in $\mathcal{P}_{j+1}$. The new edge makes y reachable from a in $\le B$ steps. Let $w(y)$ be the number of additional vertices, beyond y itself, that become reachable. Thus

$$V_a(H[x \leftarrow y]) \le V_0 + 1 + w(y).$$

For all other values set $w(y) = 0$.

By part (i) of Lemma 5.7, after the new edge $s \xrightarrow{m} y$ for y satisfying above properties, every newly reached old vertex lies on a path of length $\le B$ in C starting at y. Hence, for each retained off-diagonal transition to y, $w(y) \le |\mathrm{Reach}_C^{\le B}(y)| - 1$. Part (iii) of Lemma 5.6, together with $|C| \le R$, therefore gives $\sum_y w(y) \le BR$ for each fixed old value at the queried coordinate.

We now use the matrix bounds from the preliminaries. Let M denote the part of $P_{j+1}\widehat{\mathcal{O}}_j$ acting on the possible values stored at x, with all other data fixed, and let Φ_u be the diagonal matrix from the preliminaries.

Consider $\mathcal{A}_H := \{r \in [N] \cup \{\perp\} : (a, H[x \leftarrow r]) \in \mathcal{P}_j\}$ and let $\Pi_{\mathcal{A}_H}$ be its corresponding projection operator. We bound the operator norm of $\sqrt{w}(M - \Phi_u \Pi_{\mathcal{A}_H})$ by its Frobenius norm. Let r for the input value stored at x and y for the output value. For $r \neq \perp$ and $y \neq r$, the matrix bounds from equation (9) of the preliminaries (together with the observation that $(\Phi_u)_{y,r} = 0$) gives

$$|(M - \Phi_u \Pi_{\mathcal{A}_H})_{y,r}| \le \frac{3}{N}.$$

For each fixed $r \neq \perp$, we showed above that $\sum_y w(y) \le BR$. Hence the off-diagonal entries in the non-$\perp$ columns contribute at most

$$\sum_{r \neq \perp} \sum_{y \neq r} w(y) |(M - \Phi_u \Pi_{\mathcal{A}_H})_{y,r}|^2 \le N \cdot \frac{9}{N^2} \cdot BR = \frac{9BR}{N}.$$

For the diagonal entries, equation (7) of the preliminaries (together with the observation that $(\Phi_u)_{r,r} = \omega^{ur}$) gives $|(M - \Phi_u \Pi_{\mathcal{A}_H})_{r,r}| \le 3/N$. Since $w(r) \le R$, their total contribution is at most

$$\sum_{r \neq \perp} w(r) |(M - \Phi_u \Pi_{\mathcal{A}_H})_{r,r}|^2 \le N \cdot R \cdot \frac{9}{N^2} = \frac{9R}{N}.$$

Finally, for the column corresponding to the empty value, the matrix bound from (8) of the preliminaries gives $|(M - \Phi_u \Pi_{\mathcal{A}_H})_{y,\perp}| \le 1/\sqrt{N}$. Its contribution is therefore at most

$$\sum_y w(y) |(M - \Phi_u \Pi_{\mathcal{A}_H})_{y,\perp}|^2 \le \frac{1}{N} \sum_y w(y) \le \frac{BR}{N}.$$

Consequently,

$$\|\sqrt{w}(M - \Phi_u \Pi_{\mathcal{A}_H})\|_{\mathrm{op}}^2 \le \|\sqrt{w}(M - \Phi_u \Pi_{\mathcal{A}_H})\|_F^2 \le \frac{(10B+9)R}{N} \le \frac{16(B+1)R}{N}.$$

Thus for a vector $|\eta\rangle$ satisfying $|\eta\rangle = \Pi_{\mathcal{A}_H} |\eta\rangle$,

$$\left\| \sqrt{w} \left(M - \Phi_u \Pi_{\mathcal{A}_H} \right) |\eta\rangle \right\| \le 4 \sqrt{\frac{(B+1)R}{N}} \, \| |\eta\rangle \| = 4\alpha \, \| |\eta\rangle \| .$$

Since Φ_u changes only phases and is diagonal, it commutes with $\sqrt{w}$ and

$$\left\| \sqrt{w}\, \Phi_u \Pi_{\mathcal{A}_H} |\eta\rangle \right\| = \left\| \sqrt{w}\, |\eta\rangle \right\| .$$

Hence

$$\left\| \sqrt{w}\, M |\eta\rangle \right\| \le \left\| \sqrt{w}\, |\eta\rangle \right\| + 4\alpha \, \| |\eta\rangle \| . \tag{62}$$

For every output value y, we have

$$V_a(H[x \leftarrow y]) \le V_0 + 1 + w(y).$$

Therefore

$$\left\|\sqrt{V}M\left|\eta\right\rangle\right\|^2 \le (V_0+1)\left\|M\left|\eta\right\rangle\right\|^2 + \left\|\sqrt{w}\,M\left|\eta\right\rangle\right\|^2.$$

Using $\|M\| \le \chi$, $\chi \ge 1$, and (62), we obtain

$$\begin{aligned}\chi^{-2}\left\|\sqrt{V}M\left|\eta\right\rangle\right\|^2 &\le (V_0+1)\left\||\eta\rangle\right\|^2 + \left(\left\|\sqrt{w}\left|\eta\right\rangle\right\| + 4\alpha\left\||\eta\rangle\right\|\right)^2 \\ &= V_0\left\||\eta\rangle\right\|^2 + \left\|\sqrt{w}\left|\eta\right\rangle\right\|^2 + (1+16\alpha^2)\left\||\eta\rangle\right\|^2 \\ &\quad + 8\alpha\left\||\eta\rangle\right\|\left\|\sqrt{w}\left|\eta\right\rangle\right\|.\end{aligned}$$

On the input side, adding the value stored at x cannot destroy any vertex already reachable in H, and the vertices counted by $w(r)$ are additional reachable vertices. Thus, pointwise in the input value r,

$$V_a(H[x \leftarrow r]) \ge V_0 + w(r).$$

Consequently,

$$V_0\left\||\eta\rangle\right\|^2 + \left\|\sqrt{w}\left|\eta\right\rangle\right\|^2 \le \left\|\sqrt{V}\left|\eta\right\rangle\right\|^2.$$

Substituting this above gives

$$\chi^{-2}\left\|\sqrt{V}M\left|\eta\right\rangle\right\|^2 \le \left\|\sqrt{V}\left|\eta\right\rangle\right\|^2 + (1+16\alpha^2)\left\||\eta\rangle\right\|^2 + 8\alpha\left\||\eta\rangle\right\|\left\|\sqrt{w}\left|\eta\right\rangle\right\|.$$

Sum this inequality over the mutually orthogonal choices of a, x, u, H. The ordinary squared input norms sum to at most one, while the w-weighted squared input norms sum to at most e_j. Cauchy-Schwarz therefore gives (60).

The recurrence is the same numerical recurrence that appears in the proof of Lemma 4.6. For completeness, if $\overline{e}_j := 256(j+\alpha^2 j^2)$ and $e_j \le \overline{e}_j$, then $8\alpha\sqrt{e_j} \le 64 + 192\alpha^2 j$, whereas $\overline{e}_{j+1} - \overline{e}_j = 256 + 256\alpha^2 + 512\alpha^2 j$. Induction gives $e_j \le \overline{e}_j$ for all j, and summing over $j < t$ gives the second bound. □

Accumulating the online queries. We now compare the real projected evolution $|v_j\rangle$ with the auxiliary evolution $|\widehat{v}_j\rangle$. In one query there are three possible sources of discrepancy. First, we removed the transitions to values in Y_H, which costs at most β. Second, we keep the auxiliary state inside P_{j+1}, the part that would leave this subspace is exactly $F_j\left|\widehat{v}_j\right\rangle$. Third, the factor χ^{-1} used to make the auxiliary step a contraction costs at most another β.

We formalize this calculation below:

$$\begin{aligned}\left\||v_{j+1}\rangle - |\widehat{v}_{j+1}\rangle\right\| &= \left\|\Pi_{\mathcal{G}_{k,B}}U_{j+1}\mathcal{O}\left|v_j\right\rangle - \chi^{-1}U_{j+1}P_{j+1}\widehat{\mathcal{O}}_j\left|\widehat{v}_j\right\rangle\right\| \\ &= \left\|\Pi_{\mathcal{G}_{k,B}}\mathcal{O}\left|v_j\right\rangle - \chi^{-1}P_{j+1}\widehat{\mathcal{O}}_j\left|\widehat{v}_j\right\rangle\right\| \quad (U_{j+1} \text{ commutes with } \Pi_{\mathcal{G}_{k,B}} \text{ and preserves norm}) \\ &\le \left\|\Pi_{\mathcal{G}_{k,B}}\mathcal{O}\left|v_j\right\rangle - \chi^{-1}\widehat{\mathcal{O}}_j\left|\widehat{v}_j\right\rangle\right\| + \chi^{-1}\left\|F_j\left|\widehat{v}_j\right\rangle\right\| \quad (\text{by triangle inequality, definition of } F_j) \\ &= \left\|\Pi_{\mathcal{G}_{k,B}}\mathcal{O}\left|v_j\right\rangle - \chi^{-1}\Pi_{\mathcal{G}_{k,B}}\left(\mathcal{O}P_j - A_j\right)\left|\widehat{v}_j\right\rangle\right\| + \chi^{-1}\left\|F_j\left|\widehat{v}_j\right\rangle\right\| \quad (\text{by definition of } \widehat{\mathcal{O}}_j) \\ &\le \left\|\mathcal{O}\left|v_j\right\rangle - \chi^{-1}\mathcal{O}\left|\widehat{v}_j\right\rangle\right\| + \chi^{-1}\left\|A_j\left|\widehat{v}_j\right\rangle\right\| + \chi^{-1}\left\|F_j\left|\widehat{v}_j\right\rangle\right\| \\ &\le \left\|\mathcal{O}\left|v_j\right\rangle - \mathcal{O}\left|\widehat{v}_j\right\rangle\right\| + (1-\chi^{-1})\left\|\mathcal{O}\left|\widehat{v}_j\right\rangle\right\| + \chi^{-1}\left\|A_j\left|\widehat{v}_j\right\rangle\right\| + \chi^{-1}\left\|F_j\left|\widehat{v}_j\right\rangle\right\| \\ &\le \left\||v_j\rangle - |\widehat{v}_j\rangle\right\| + (1-\chi^{-1}) + \chi^{-1}\beta + \chi^{-1}\left\|F_j\left|\widehat{v}_j\right\rangle\right\| \quad (\text{using } \|\mathcal{O}\| \le 1, \|A_j\| \le \beta) \\ &\le \left\||v_j\rangle - |\widehat{v}_j\rangle\right\| + 2\beta + \left\|F_j\left|\widehat{v}_j\right\rangle\right\| \quad (\text{using } \chi^{-1} = 1/(1+\beta), \beta/(1+\beta) \le \beta, 1/(1+\beta) \le 1)\end{aligned}$$

The two initial vectors are the same, and $\mathsf{Col}^{\le B,+}\,|\widehat{v}_t\rangle = 0$ because the auxiliary state always lies in P_t. We sum the preceding inequality, apply Lemma 5.8, and then use Cauchy-Schwarz inequality to get a bound on $\left\|\mathsf{Col}^{\le B,+}\,|v_t\rangle\right\|$. We show the calculation below:

$$\begin{aligned}
&\sum_{j<t} \| |v_{j+1}\rangle - |\widehat{v}_{j+1}\rangle \| \le \sum_{j<t} (\| |v_j\rangle - |\widehat{v}_j\rangle \| + 2\beta + \|F_j\,|\widehat{v}_j\rangle\|) \\
&\qquad \Rightarrow \| |v_t\rangle - |\widehat{v}_t\rangle \| \le 2t\beta + \sum_{j<t} \|F_j\,|\widehat{v}_j\rangle\| \quad \text{(telescoping the summations, using } |v_0\rangle = |\widehat{v}_0\rangle) \\
\Rightarrow \left\|\mathsf{Col}^{\le B,+}\,(|v_t\rangle - |\widehat{v}_t\rangle)\right\| &\le 8t\sqrt{\frac{3(B+1)k}{N}} + \frac{4}{\sqrt{N}} \sum_{j<t} \sqrt{(B+1)\,\| |\widehat{v}_j\rangle\|^2 + \left\|\sqrt{V}\,|\widehat{v}_j\rangle\right\|^2} \\
\Rightarrow \left\|\mathsf{Col}^{\le B,+}\,|v_t\rangle\right\| &\le 8t\sqrt{\frac{3(B+1)k}{N}} + 4\sqrt{\frac{t}{N}\sum_{j<t}(B+1+e_j)}
\end{aligned}$$

Substituting (61) yields the following bound.

Lemma 5.10 (Online bound for messages of length at most B)**.** *The projected evolution satisfies*

$$\begin{aligned}
\left\|\mathsf{Col}^{\le B,+}\,|v_t\rangle\right\| \le 8t\sqrt{\frac{3(B+1)k}{N}} \\
+ 4\sqrt{\frac{(B+1)t^2 + 128t^3}{N} + \frac{256(B+1)Rt^4}{3N^2}}.
\end{aligned}$$

In particular,

$$\left\|\mathsf{Col}^{\le B,+}\,|v_t\rangle\right\|^2 = O\left(\frac{(B+1)(k+1)t^2 + t^3}{N} + \frac{(B+1)Rt^4}{N^2}\right). \tag{63}$$

When $t \ge B$, the term $(B+1)t^2/N$ is absorbed by $O(t^3/N)$.

Conditioned on offline event. The previous lemma analyzes a state that has already been restricted to $\mathcal{G}_{k,B}$ and whose salt row is empty. We now explain why these assumptions can be imposed after an arbitrary offline phase, even after conditioning on the offline event $\mathcal{W}$ produced by the advice-removal theorem. The structural tail Lemma 5.3, proved above for the general-B restriction, supplies the required bound outside $\mathcal{G}_{k,B}$, Lemma 3.11 then combines this with the fresh-salt-row reduction.

Theorem 5.11 (General-B bound after conditioning)**.** *Consider an offline algorithm initialized independently of h that makes at most Q quantum queries, and let $\mathcal{W}$ be an event determined during the offline phase with probability $p > 0$. Condition on $\mathcal{W}$, retain the resulting state, and sample a fresh uniform salt. Suppose the subsequent online computation makes at most T queries before outputting two messages of length at most B. Let $\tau := T + 2B$, $R := Q + \tau$, and $\Lambda := \lceil \log_2(2NR) \rceil$. Let* win *be the event that the two messages output form a* MD *collision with respect to the salt. Then*

$$\begin{aligned}
\Pr[\text{win} \mid \mathcal{W}] = O\Big(\frac{Q+B}{N} + \frac{(B+1)\tau^2}{N}\left(\frac{R^{3/2}}{\sqrt{N}} + \Lambda + \log_2\frac{1}{p}\right) \\
+ \frac{\tau^3}{N} + \frac{(B+1)R\tau^4}{N^2}\Big).
\end{aligned} \tag{64}$$

Proof. The $2B$ additional queries in τ evaluate the two output messages and are used only for verification. Choose

$$k := \min\left\{R, \left\lceil \frac{8eR^{3/2}}{\sqrt{N}} + \frac{1}{2}\log_2\frac{1}{p} + 20\Lambda \right\rceil\right\}.$$

If $k = R$, the structural tail $\varepsilon_{R,k}$ from Lemma 5.3 is zero. Otherwise, writing $\kappa := k+1$ and using $\binom{R}{\kappa} \le (eR/\kappa)^\kappa$, the definition of k gives $\varepsilon_{R,k} \le \sqrt{p}\,2^{-20\Lambda}$. In either case, $k = O(R^{3/2}/\sqrt{N} + \Lambda + \log_2(1/p))$.

Apply the conditional projection comparison Lemma 3.11 with $G = \Pi_{\mathcal{G}_{k,B}}$ and online budget τ. The resulting projected initial vector has norm at most one, is supported on databases in $\mathcal{G}_{k,B}$ of size at most Q, and has empty salt row. It therefore satisfies the assumptions of Lemma 5.10. Since the real collision projector is bounded by $\mathsf{Col}^{\le B,+}$, the common output-recording bound, Lemma 3.10, gives

$$\sqrt{\Pr[\text{win} \mid \mathcal{W}]} \le \frac{(\tau+1)\varepsilon_{R,k}}{\sqrt{p}} + \sqrt{\frac{Q}{N}} + \left\|\mathsf{Col}^{\le B,+}\,|v_\tau\rangle\right\| + \sqrt{\frac{2B}{N}}.$$

As seen in Lemma 3.11, the term $\sqrt{Q/N}$ is the cost of removing the component on which the fresh salt row was already touched during the offline phase. It is not divided by $\sqrt{p}$ because the salt is sampled only after conditioning.

Square the preceding inequality and apply (63). Substituting the bound on k proves (64), the squared structural tail is negligible compared with $(Q+B)/N$. □

Applying the quantum-advice reduction.

Proof of Theorem 5.1. Assume first that $S \ge 1$. Apply Theorem 3.1 with $Q = 2S\tau$, so $R = (2S+1)\tau$. Every conditioning event used by that reduction has probability at least N^{-2S}, and hence $\log_2(1/p) \le 2S\log_2 N$. Substituting these parameters into Theorem 5.11 gives

$$\begin{aligned}\mathrm{Adv}^{\mathsf{MD}\text{-}B\mathsf{col}\text{-}\mathsf{q}}_{N,M}(S,T) = O\Bigg(&\frac{(B+1)\tau^2}{N}\left(\frac{R^{3/2}}{\sqrt{N}} + \Lambda + S\log_2 N\right)\\ &+ \frac{\tau^3}{N} + \frac{(B+1)R\tau^4}{N^2}\Bigg).\end{aligned} \tag{65}$$

The terms $(Q+B)/N$ and the $1/N$ alternative in the advice-removal theorem are absorbed by the displayed expression.

We next simplify the last term without making any assumption involving $S\tau^3$. If $\tau^3 \le N$, then, since $R \ge \tau$,

$$\frac{(B+1)R\tau^4/N^2}{(B+1)R^{3/2}\tau^2/N^{3/2}} = \frac{\tau^2}{\sqrt{RN}} \le \sqrt{\frac{\tau^3}{N}} \le 1.$$

Hence the last term of (65) is absorbed by the $R^{3/2}$ term. If $\tau^3 > N$, then $\tau^3/N > 1$ and the trivial success bound one is already stronger. After capping the probability at one, this proves the term $\min\{1, \mathcal{F}_B\}$ in (53).

For the second term in the minimum, replace arbitrary S-qubit advice by the maximally mixed state using Lemma 3.12. The latter state is independent of the oracle. With $k = 0$, a database in $\mathcal{G}_{k,B}$ has indegree at most one and no directed cycle of length at most B, so Lemma 5.4 rules out a recorded collision. Applying Lemma 5.3 to an oracle-independent execution with at most τ queries and then using Lemma 3.10 gives success $O(\tau^3/N)$. The positive-operator comparison therefore gives $O(2^S\tau^3/N)$ for arbitrary advice. The same argument with $S = 0$ gives the claimed oracle-independent bound.

Finally suppose $(2S+1)\tau^3 \le N$. Since $R = (2S+1)\tau$, this is $R\tau^2 \le N$, and hence $R^{3/2}/\sqrt{N} \le R/\tau = 2S+1$ and $\Lambda = O(\log N)$. Substituting these inequalities into $\mathcal{F}_B$ proves (54). If $B \le T$, then $\tau \le 3T$, giving the stated $\widetilde{O}((BST^2+T^3)/N)$ form. □

Comparison with the two-block result. For $B = 2$, the sharper argument in Section 4 distinguishes between different kinds of recorded equal-output structure and removes the $R^{3/2}$ contribution without a parameter restriction. The present graph argument does not do that. When $B > 2$, one new edge may expose an entire recorded continuation rather than only one additional row, and our proof controls this by the structural restriction $\mathcal{G}_{k,B}$ and the size of the reachable set.

Acknowledgements

This work grew out of discussions during the authors' visit to ICTS-TIFR in Bengaluru. The authors thank ICTS for its hospitality and support, Jaikumar Radhakrishnan for hosting them and for insightful discussions, and Mohammad Saif Khan for his contributions to the early discussions.

The work of Ashrujit Ghoshal was supported by an ANRF Early Career Research Grant (Grant No ANRF/ECRG/2025/000556/ENS), a New Faculty Initiation Grant at IIT Madras, the Arvind Raghunathan Center for Theoretical CS (aRtCS), the Mphasis F1 Foundation, the National Quantum Mission initiative of Dept of Science & Technology, India and CyStar, IIT Madras. The work of Venkata Koppula was funded by the HUJI-IITD Joint Global Research Partnership Program (IRD Project No. IR00016G FS), ANRF MATRICS Research Grant (Grant No. ANRF/ARGM/2025/002278/MTR) and a grant from the National Science Foundation to the Simons Institute for the Theory of Computing. This work was done in part while Venkata was visiting the Simons Institute for the Theory of Computing.

AI Disclosure

We used ChatGPT 5.6 Sol and ChatGPT-6 Astra exclusively for editorial assistance with the presentation and structural organization of the paper. In particular, Table 2, Table 3, and the TikZ code for Figures 1 to 4 were generated using ChatGPT. All ideas, results, and proofs were human-developed.

References

[Aar05] Scott Aaronson. Limitations of quantum advice and one-way communication. *Theory Comput.*, 1:1–28, 2005.

[ACDW20] Akshima, David Cash, Andrew Drucker, and Hoeteck Wee. Time-space tradeoffs and short collisions in Merkle–Damgård hash functions. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology—CRYPTO 2020, Part I*, volume 12170 of *Lecture Notes in Computer Science*, pages 157–186. Springer, 2020.

[AGL22] Akshima, Siyao Guo, and Qipeng Liu. Time-space lower bounds for finding collisions in Merkle–Damgård hash functions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology—CRYPTO 2022, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 192–221. Springer, 2022.

[AGL24] Akshima, Siyao Guo, and Qipeng Liu. Time-space lower bounds for finding collisions in merkle–damgård hash functions. *Journal of Cryptology*, 37(2):10, 2024.

[AS04] 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.

[BHT98] 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.

[CDGS18] Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John P. Steinberger. Random oracles and non-uniformity. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology—EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 227–258. Springer, 2018.

[CFHL21] 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 *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 598–629. Springer, 2021.

[CGLQ20] Kai-Min Chung, Siyao Guo, Qipeng Liu, and Luowen Qian. Tight quantum time-space tradeoffs for function inversion. In *61st IEEE Annual Symposium on Foundations of Computer Science (FOCS 2020)*, pages 673–684. IEEE, 2020.

[DGK17] Yevgeniy Dodis, Siyao Guo, and Jonathan Katz. Fixing cracks in the concrete: Random oracles with auxiliary input, revisited. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology—EUROCRYPT 2017, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 473–495. Springer, 2017.

[DLW24] Fangqi Dong, Qipeng Liu, and Kewen Wu. Tight characterizations for preprocessing against cryptographic salting. In *Annual International Cryptology Conference*, pages 377–411. Springer, 2024.

[GK22] Ashrujit Ghoshal and Ilan Komargodski. On time-space tradeoffs for bounded-length collisions in Merkle–Damgård hashing. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology—CRYPTO 2022, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 161–191. Springer, 2022.

[GLLZ21] Siyao Guo, Qian Li, Qipeng Liu, and Jiapeng Zhang. Unifying presampling via concentration bounds. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography—TCC 2021, Part I*, volume 13042 of *Lecture Notes in Computer Science*, pages 177–208. Springer, 2021.

[Goo25] Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. *Nature*, 638:920–926, 2025.

[Hel80] Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, 26(4):401–406, July 1980.

[HM23] Yassine Hamoudi and Frédéric Magniez. Quantum time-space tradeoff for finding multiple collision pairs. *ACM Transactions on Computation Theory*, 15(1–2):1–22, 2023.

[IK10] Russell Impagliazzo and Valentine Kabanets. Constructive proofs of concentration bounds. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 617–631. Springer, 2010.

[Liu23] Qipeng Liu. Non-uniformity and quantum advice in the quantum random oracle model. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology—EUROCRYPT 2023, Part I*, volume 14004 of *Lecture Notes in Computer Science*, pages 117–143. Springer, 2023.

[MW05] Chris Marriott and John Watrous. Quantum Arthur–Merlin games. *Computational Complexity*, 14(2):122–152, 2005.

[NABT15] Aran Nayebi, Scott Aaronson, Aleksandrs Belovs, and Luca Trevisan. Quantum lower bound for inverting a permutation with advice. *Quantum Information and Computation*, 15(11–12):901–913, 2015.

[Unr07] Dominique Unruh. Random oracles and auxiliary input. In Alfred Menezes, editor, *Advances in Cryptology—CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 205–223. Springer, 2007.

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology—CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 239–268. Springer, 2019.

A Notations for Section 4 and Section 5

Table 2 and Table 3 summarize the notations used in the two-block analysis and general-B analysis, respectively.

Table 2: Notation for the two-block analysis. An edge is relevant if its source belongs to $\mathcal{R}_a(D)$.

Notation	Meaning
Query budgets and local notation	
$Q,\ t,\ R$	Offline, online, and total query bounds, with $R = Q + t$. In the conditioned bound, t includes verification.
τ	$T + 4$: the original online budget plus verification of two messages of at most two blocks.
$\mathcal{W},\ p$	Offline conditioning event and its probability $p = \Pr[\mathcal{W}]$.
$H = D^{-x}$	Database with the queried coordinate $x = (s, m)$ erased.
$\mathcal{O},\ R_u,\ \Phi_u$	Compressed-oracle query, its local matrix, and the diagonal phase operator, respectively.
Recorded graph and equal-output statistics	
$\mathcal{R}_a(D)$	$\mathrm{Reach}_D^{\le 1}(a)$: the salt and its recorded first-block outputs.
$\mathcal{L}(D),\ J(D)$	Rows containing a self-loop or two distinct edges with the same output; $J(D) = \lvert\mathcal{L}(D)\rvert$.
$n_z(D),\ r_z(D; a)$	Numbers of all recorded edges, and of relevant recorded edges, with output z, respectively.
$K_a(D)$	$\sum_z n_z(D) r_z(D; a)$: ordered equal-output edge pairs whose first edge is relevant, including self-pairs.
$O_a(D),\ P_a(D)$	$O_a(D) = \{z : r_z(D; a) > 0\}$; $P_a(D)$ is the set of rows with an edge whose output lies in $O_a(D)$.
$c_z(D; y),\ b_z(y)$	$c_z(D; y)$ counts recorded edges from row y to output z; locally, $b_z(y) = \mathbf{1}[y \notin \mathcal{R}_a(H)] c_z(H; y)$ counts those that become relevant.
$\Gamma(D),\ Z(D)$	$\Gamma(D) = \sum_z n_z(D)^2$ and $Z(D) = \Gamma(D) - \lvert D\rvert$: ordered equal-output pair counts with and without self-pairs.
Tail potentials and structural restriction	
$d_s(D),\ L$	Number of recorded entries in row s, and the row-occupancy threshold $L = \lceil N^{1/3} \rceil$.
$f_s(D),\ F(D)$	$f_s(D) = 1$ for $s \in \mathcal{L}(D)$, and $f_s(D) = \min\{d_s(D)/L, 1\}$ otherwise; $F(D) = \sum_s f_s(D) \ge J(D)$.
$\lvert\Psi_q\rangle,\ g_q(D)$	State after q queries, and the square-root potential $g_q(D) = \sqrt{q + Z(D)}$.
$\theta,\ k,\ \gamma$	Tail slack $\theta \ge 0$ and thresholds $k = \lceil 16R/N^{1/3} + \theta \rceil$, $\gamma = (\sqrt{R} + 8R/\sqrt{N} + \theta)^2$.
G_2	$\Pi_{\{J \le k,\, \Gamma \le \gamma\}}$: projector imposing both structural bounds.
Online progress and matching attack	
$\lvert v_j^{(2)}\rangle$	Online state after j queries with G_2 inserted at each step; it need not be normalized.
$E_j,\ b_j$	$E_j = \lVert\sqrt{K}\,\lvert v_j^{(2)}\rangle\rVert^2$ and $b_j = \lVert \mathrm{Col}^{\le 2} \lvert v_j^{(2)}\rangle\rVert$: squared-amplitude-weighted pair count and collision amplitude.
$\alpha^2,\ \overline{E}_j$	$\alpha^2 = 4\gamma/N$ and $\overline{E}_j = 256(j + \alpha^2 j^2)$: the numerical induction bound satisfying $E_j \le \overline{E}_j$.
$\mathcal{C}_{a,2},\ \mathrm{Col}^{\le 2}$	Recorded-collision event for distinct nonempty messages of at most two blocks, and its salt-controlled projector.
$Y_{a,x,H}$	For collision-free H, a set containing every collision-creating finite answer; $\lvert Y_{a,x,H}\rvert \le k + 1 + 2K_a(H)$ when $J(H) \le k$.
$K_0,\ w(y)$	Local baseline $K_0 = K_a(H)$ and the remaining increment after separating the new edge's self-pair contribution; $w(\perp) = 0$.
$r,\ v_i,\ (b_i, b_i')$	Number of public target rows, their states, and their stored colliding blocks: $h(v_i, b_i) = h(v_i, b_i')$, $b_i \ne b_i'$.

Table 3: Notation for the general-B analysis. The quantity $V_a(D)$ counts reachable vertices, whereas $K_a(D)$ in the two-block analysis counts edge pairs.

Notation	Meaning
Query budgets and structural tails	
$Q,\ t,\ R,\ \tau$	Offline and online query bounds; $R = Q + t$ and $\tau = T + 2B$. In the advice reduction, $t = \tau$ and $Q = 2S\tau$.
$\mathcal{W},\ p,\ \Lambda$	Offline event, its probability $p = \Pr[\mathcal{W}]$, and $\Lambda = \lceil \log_2(2NR) \rceil$.
$n_y(D),\ c(D)$	Output multiplicity and $c(D) = \sum_y \lfloor n_y(D)/2 \rfloor$, the maximum number of disjoint equal-output edge pairs.
$z_B(D)$	Maximum number of pairwise edge-disjoint directed cycles of length at most B, including self-loops.
$\mathcal{G}_{k,B},\ \Pi_{\mathcal{G}_{k,B}}$	$\mathcal{G}_{k,B} = \{D : c(D) \le k,\ z_B(D) \le k\}$ and its database projector; $k \ge 0$ is the structural threshold.
$\kappa,\ \varepsilon_{R,k}$	$\kappa = k + 1$. The uniform norm bound outside $\mathcal{G}_{k,B}$ is $\varepsilon_{R,k} = \min\{1, 2\binom{R}{\kappa}(4\sqrt{R/N})^\kappa\}$ if $\kappa \le R$, and 0 otherwise.
$\Pi^f_{\ge h},\ \Delta_{q,h}$	Tail projector onto $f(D) \ge h$, and $\Delta_{q,h} = \|\Pi^f_{\ge h}\|\Phi_q\rangle\|$, where $\|\Phi_q\rangle$ is the state after q queries.
$z(q),\ z_*$	Bound on the number of answers increasing f when $\|D^{-x}\| \le q$, and a uniform upper bound on $z(q)$.
Short paths and the pruned graph	
$\mathcal{C}^+_{a,B},\ \mathrm{Col}^{\le B,+}$	Recorded-collision event for distinct messages of at most B blocks, allowing the empty message, and its projector; $\mathcal{C}_{a,B} \subseteq \mathcal{C}^+_{a,B}$.
$H,\ E,\ C$	$H = D^{-x}$; E contains at most $3k$ deleted coordinates. The pruned graph $C = H \setminus E$ preserves short paths from a, has indegree at most one, and has no cycle of length at most B.
$\mathcal{R},\ \mathrm{src}(E)$	$\mathcal{R} = \mathrm{Reach}^{\le B}_H(a) = \mathrm{Reach}^{\le B}_C(a)$; $\mathrm{src}(E) = \{s : \exists m,\ (s,m) \in E\}$.
Y_H	$\{y \in [N] : \mathrm{Reach}^{\le B}_C(y) \cap \mathrm{src}(E) \ne \varnothing\}$: exceptional answers; $\|Y_H\| \le 3(B+1)k$.
$V_a(D)$	$\|\mathrm{Reach}^{\le B}_D(a)\| - 1$: number of reachable vertices other than the salt.
Projected and auxiliary online evolutions	
$\mathcal{P}_j,\ P_j$	$\mathcal{P}_j = \{(a, D) : D \in \mathcal{G}_{k,B},\ \|D\| \le Q + j,\ D \notin \mathcal{C}^+_{a,B}\}$ and $P_j = \Pi_{\mathcal{P}_j}$.
$\|v_j\rangle$	Online state with the structural projector $\Pi_{\mathcal{G}_{k,B}}$ inserted at every step.
$A_j,\ \beta$	Operator retaining off-diagonal transitions into Y_H, with input restricted by P_j; $\|A_j\| \le \beta = 4\sqrt{3(B+1)k/N}$.
$\widehat{\mathcal{O}}_j,\ \chi$	$\widehat{\mathcal{O}}_j = \Pi_{\mathcal{G}_{k,B}}(\mathcal{O}P_j - A_j)$; $\chi = 1 + \beta$ is the scaling factor ensuring $\|\chi^{-1}\widehat{\mathcal{O}}_j\| \le 1$.
F_j	$(I - P_{j+1})\widehat{\mathcal{O}}_j = \mathrm{Col}^{\le B,+}\widehat{\mathcal{O}}_j$: the remaining collision-creating part of a query.
$\|\widehat{v}_j\rangle$	Auxiliary state defined by $\|\widehat{v}_0\rangle = \|v_0\rangle$ and $\|\widehat{v}_{j+1}\rangle = \chi^{-1}U_{j+1}P_{j+1}\widehat{\mathcal{O}}_j\|\widehat{v}_j\rangle$.
$e_j,\ \alpha^2$	$e_j = \|\sqrt{V}\,\|\widehat{v}_j\rangle\|^2$: squared-amplitude-weighted reachable-vertex count; $\alpha^2 = (B+1)R/N$.
$\overline{e}_j$	$256(j + \alpha^2 j^2)$: the numerical induction bound satisfying $e_j \le \overline{e}_j$.
$V_0,\ w(y)$	Local baseline $V_0 = V_a(H)$ and the number of additionally reached vertices beyond y for retained, collision-free outputs; $w(\perp) = 0$.
$\mathcal{F}_B$	Success-bound expression $\frac{(B+1)\tau^2}{N}\big(R^{3/2}/\sqrt{N} + \Lambda + S\log_2 N\big) + \tau^3/N$.