

Retention Choices for Quantum CHAM Key Search under a Global Qubit Budget

Minseo Kim¹[0009-0008-1979-5778], Seungwon Lee¹[0009-0009-9583-1387],
Subeen Cho¹[0009-0004-4352-8021], and Hwajeong Seo¹[0000-0003-0069-9061]

¹Department of Convergence Security,
Hansung University, Seoul (02876), South Korea,
{msqoo0212, dkajdfhd1, chosubin1208, hwajeong84}@gmail.com

Abstract. Retaining intermediate values can shorten a quantum cipher oracle but increase the logical qubits required by each parallel search worker. We propose a joint selection procedure for 80-round and revised 112-round CHAM-128/128 under a fixed global logical-qubit budget. It selects retained round addends, feasible integer allocations of search workers, and execution plans to minimize maximum scheduled logical depth subject to a target probability of recovering and accepting the original key. We evaluate libraries of 83 and 99 retention sets against a baseline whose retention choices are restricted to no retention and full retention. Both comparison groups use the same implementation and execution rules. Across 654 budget, success-target, and Toffoli-decomposition conditions per round count, the mean depth reductions are 11.12% and 8.39%, with strict improvements in 606 and 588 conditions, respectively. The selected retention count need not increase with the budget, and a fixed-circuit capacity discontinuity can disappear after reselection. The findings concern the measured libraries under an ideal-cipher success model and logical-layer costs.

Keywords: Quantum cryptanalysis · CHAM · Grover search · Reversible computation · Resource estimation

1 Introduction

Which intermediate values should a quantum key-search oracle retain when all search processors share a fixed qubit budget? A *processor* here is a logical quantum search worker assigned to one key-space partition during a search stage. Its allocation includes the key register and all workspace needed by the oracle and diffusion operations. This definition concerns logical resource allocation, without assuming a separate physical quantum computer for each worker. Extra workspace can shorten an individual circuit, but a wider circuit can admit fewer parallel processors. The resulting search then assigns more keys to each processor. Choosing a retention scheme therefore requires the cost of the complete search, including its probability of recovering the original key.

Table 1. Generic two-pair comparison for 112-round CHAM at $B = 4096$, target 99.5%, and depth-8 CCX. Here k counts retained addends, Q is processor width, P is processor count, and $C_{\max}$ is maximum search depth. Both choices use the same two-pair fanout layout, key program, compiler, and Grover plan; $D_0 = D(128) = D(126)$.

Choice	Q	D_0	$D(126)$	P	$C_{\max}/10^{23}$
QD_0^2 selector ($k = 0$)	773	104,611	104,611	4	7.236625
Deployment selector ($k = 3$)	965	99,662	99,662	4	6.894270

The general time-space tradeoff is established in quantum symmetric-key cryptanalysis [4,3]. Quantum CHAM studies provide reduced-width key schedules and depth-optimized implementations that retain intermediate addends [8,2]. These give the ingredients for a more specific question: how much should be retained, and where, once integer processor allocations and verified-key success enter the selection objective? We address this question for original 80-round and revised 112-round CHAM-128/128, keeping the round arithmetic and varying the retained addends [6].

We jointly select retained addends, feasible integer processor allocations, and execution plans satisfying a common accepted-original-key success target. The baseline restricts retention choices to the existing no-retention and full-retention constructions, while both comparison groups receive the same implementation and execution rules. Admitting the measured partial-retention sets then isolates their benefit for complete-search cost.

An observed generic two-pair comparison for 112-round CHAM makes the selection effect concrete (Table 1). At a budget of 4096 logical qubits, both a 773-qubit unretained oracle and a 965-qubit oracle retaining three addends admit four processors. The latter reduces iteration depth from 104,611 to 99,662. Because its greater width leaves the processor count unchanged, the complete search is 4.73% shallower at the same 99.5% success target. Here Q is the per-processor logical-qubit count and $D_0 = D(128)$ is the complete Grover-iteration depth before key-space partitioning. A QD_0^2 selector chooses the unretained circuit: its continuous width penalty misses the unused capacity in this particular deployment.

Our contributions are:

1. *A success-constrained joint selection procedure.* We formulate retention as a deployment choice under a global qubit budget. Within the measured menu, the procedure selects retained round locations, processor allocations, and execution plans to minimize maximum scheduled logical depth. One- and two-pair oracles meet the same original-key recovery and acceptance requirement.
2. *Measured improvement over an endpoint-restricted baseline.* Using 83 retention sets for 80 rounds and 99 for revised 112 rounds, joint selection reduces depth by an average of 11.12% and 8.39% over that baseline across 654 conditions per round count. The reductions are strict in 606 and 588 conditions, respectively.

Paired deployments identify which retention and allocation choices produce the savings.

3. *Deployment-dependent design insights.* The selected retention count can be nonmonotone in the qubit budget, and circuit-level width–depth scores can miss lower-depth searches. A conditional result for homogeneous prefix partitions and measured reselection cases explain the observed capacity effects.

1.1 Relation to prior work

Kim et al. study original-key success and quantum time–space tradeoffs [4]; Jaques et al. connect concrete oracles, spurious keys, and resource-constrained attack costs [3]. Our contribution is a controlled retention-selection study within this framework. The proxy diagnostics isolate the choice made by depth and QD_0^2 scores; they do not reproduce those works’ complete attack optimizers.

For CHAM, Yang et al. study optimized implementations, including key-schedule width reduction [8]. Jang et al. shorten circuit depth with retained intermediate values and explicitly motivate depth reduction through parallel search [2]. Our baseline evaluates the no-retention and full-retention CHAM constructions within the common complete-oracle and search model; the library comparison additionally admits partial retention. Measuring revised CHAM supplies a second round count under the same contract. Appendix A.1 relates the published encryption-only resource tables to our complete-oracle baseline.

General space-constrained uncomputation and automated cipher benchmarking address broader synthesis problems [5,1]. Here the varied design choice is narrower: which addends to retain within the existing CHAM construction. This scope makes it possible to trace the effects of retention through circuit scheduling, deployment, and accepted-key success without changing the cipher or its adder.

2 Retention Circuits and the Measured Library

2.1 How retention changes the circuit

Write the four 32-bit state words before round i as (x_0, x_1, x_2, x_3) and its round-key word as k_i . With addition modulo 2^{32} , the round computes

$$\begin{aligned} a_i &= \text{ROL}_{\alpha_i}(x_1) \oplus k_i, \\ (x'_0, x'_1, x'_2, x'_3) &= (x_1, x_2, x_3, \text{ROL}_{\beta_i}((x_0 \oplus i) + a_i)), \end{aligned} \quad (1)$$

where $(\alpha_i, \beta_i) = (1, 8)$ for even zero-based i and $(8, 1)$ for odd i [6,8]. Fixed rotations and the state-word permutation use wire relabeling.

The in-place round temporarily XORs the rotated x_1 into the round-key register, uses that register as the adder input, and restores k_i with the same XOR. The retained round computes a_i into a fresh 32-qubit word and leaves it live after the addition [2]. Subsequent operations on k_i and x_1 can then proceed without waiting for the in-place routine’s post-addition XOR. Both routines

Table 2. Implementation choices shared by every retention set. Compiler weights guide gate ordering; measured depth uses the chosen exact CCX decomposition and unit-duration logical layers.

Dimension	Measured choices
Key schedule	Original ancilla circuit; Gaussian elimination; regenerated FSE-style linear synthesis
Pair layout	One pair with held output; two pairs with held outputs; two pairs with coherent key fanout; two pairs with serial flags
Compiler	Unoptimized; Qiskit commutation optimization; commutation ordering with CCX weights 1, 8, or 11
CCX decomposition	Published depth-8 circuit; Qiskit depth-11 circuit
Free key bits	$n = 117, \dots, 128$, corresponding to $P = 2048, \dots, 1$
Complete iteration	Phase oracle followed by diffusion, scheduled continuously across their boundary

use two word-wide XORs and the same modular adder; the benefit comes from changed dependencies and scheduling.

For r rounds, a set $S \subseteq \{0, \dots, r - 1\}$ selects the addends to retain, and $k = |S|$ is the retention count. Other rounds use the in-place routine. Retained values remain coherent until inverse encryption restores their workspace inside the phase oracle, so their width is charged throughout. The endpoint $S = \emptyset$ is OLD, and $S = \{0, \dots, r - 1\}$ is FULL.

Both round counts use the same round arithmetic, adders, and carry assignment. The phase oracle computes a ciphertext-matching predicate, marks the matching keys by a sign change, and uncomputes. This restores all non-key workspace. Encryption is checked against independent classical implementations and published known-answer vectors.

2.2 The common implementation menu

An implementation configuration fixes a key-schedule circuit, the arrangement of one or two plaintext–ciphertext pairs, and a compiler. Every S , including both endpoints, receives the $3 \times 4 \times 5 = 60$ configurations in Table 2. Each has 24 resource profiles: 12 free-key sizes and two Toffoli (CCX) decompositions.

A held-output layout keeps each computed ciphertext until a joint comparison marks keys matching every pair. The two-pair version shares the key register. Key fanout instead supplies separate key registers through coherent CNOT copies of basis labels, permitting more independent scheduling; the copies are uncomputed afterwards. Serial flags reuse one encryption workspace, retaining only one match flag per pair before marking their conjunction. All layouts restore their workspaces. Held-output and fanout comparisons use a clean AND tree. Serial layouts use clean AND ladders to compute the individual match flags before marking their conjunction.

Gaussian elimination and FSE-style synthesis generate CNOT circuits for the linear key maps. The latter is regenerated with the method of Xiang et al. [7]; all retention sets share the resulting key program. The compiler’s commutation ordering runs before and after the Qiskit pass, using a deterministic scheduling heuristic with the stated CCX weight. Every rewrite preserves the circuit’s function; compiler scores are distinct from the final measured depth.

2.3 Candidate counts and placements

The library contains 83 sets for 80 rounds and 99 for 112 rounds: 182 round/set pairs and 262,080 profiles. Candidate counts concentrate near processor-capacity thresholds. For construction, the one- and two-pair width estimates $386 + 32k$ and $773 + 64k$ identify the largest feasible k at selected budget/processor combinations; the two preceding counts are also included. Final selection uses the measured widths for all configurations.

Placements include approximately uniform round locations with two offsets and, when the count permits, locations congruent to 2 modulo 4 with two offsets. Additional reference patterns and both endpoints complete the library. Appendix A.5 completely specifies the library, including the construction budgets, integer rounding, offsets, additional sets, and set identifiers. This provides structured variation in count and placement while keeping the comparison finite. The candidates were developed through exploratory comparisons, so the results describe this library rather than an unseen test set.

2.4 Generic and prefix-specialized circuits

A *generic* circuit retains the full key register. Assigning a fixed key prefix changes the search domain and diffusion without resynthesizing encryption. A *prefix-specialized* circuit additionally propagates the known prefix values through the gates and removes redundant constant wires. It counts any required constant preparation and erasure and constructs diffusion on the resulting free-key indices. At prefix depth zero ($P = 1$), only public-input and workspace constants are propagated.

A uniform prefix family contains one circuit for every prefix of a fixed length. Its total width is the sum of the member widths, and its common-plan iteration depth is their maximum. We measure such families using the regenerated FSE program, one-pair hold or two-pair fanout, and the three compiler choices without CCX reweighting. Complete families are available at every budget admitting their allocation. Bounded nonuniform partitions use the measured nodes described in Appendix A.3.

For input sensitivity, 11 representative retention sets receive all 60 configurations on four public input instances. Matched prefix comparisons use only families fully measured across all four instances. We distinguish the cost change of a fixed deployment from the result of selecting its configuration and retention set again.

3 Proposed Method: Success-Constrained Joint Selection

Given measured circuit profiles, a total logical-qubit budget B , a success target τ , and a CCX decomposition, our method selects a retention configuration, feasible processor allocation, and execution plan minimizing maximum scheduled logical depth. It evaluates feasible deployments with plans meeting the same accepted-original-key target, then compares their complete-search costs. The selected plan’s success bound is independently certified.

Section 3.1 specifies the common resource and success model; Section 3.2 gives the finite-menu selection procedure and partition controls.

3.1 Cost and success contract

Resources and the recovered-key event. Let B be the total available logical qubits. A generic processor searches n free key bits using Q allocated logical qubits, including the key register and all oracle and diffusion workspace. Its measured iteration depth $D(n)$ is the depth of the compiled oracle *followed by* the compiled diffusion operator. The second schedule starts from the first schedule’s per-wire completion times. We do not substitute the sum of two independently minimized depths. Allocated unused wires count towards Q .

Both Clifford and T gates have unit logical layer duration, with all-to-all connectivity. We report two exact CCX decompositions under this convention. The model does not include routing, error correction, magic-state factories, or unequal physical gate durations. Thus $D(n)$ and the attack cost below are logical depths, not elapsed execution times. The implementation uses no approximate arbitrary-rotation-to- T conversion.

Fixing $128 - n$ leading key bits defines a prefix domain of 2^n keys. A uniform dyadic partition uses $P = 2^{128-n}$ such domains and assigns one processor to each, covering the full key space without overlap. The deployment is feasible when

$$PQ \leq B, \quad P 2^n = 2^{128}. \quad (2)$$

We compare all feasible measured values of P , rather than assuming that the largest one is always the best. For a family of prefix-specialized processors, the allocation is $Q_\Sigma = \sum_p Q_p$, and the synchronized iteration depth is $\max_p D_p$ when they use the same query and repetition plan.

A measured candidate key is classically checked against two distinct public plaintext–ciphertext pairs. Success means that the *original* key has been recovered and accepted. Merely measuring some key marked by a one-pair oracle is a different event. Classical verification calls are counted separately from logical circuit depth. Early acceptance can shorten an execution, but the objective is its maximum scheduled depth; expected completion time is not minimized.

The success target averages quantum measurement outcomes and the ideal-cipher collision distribution defined below. It is imposed for each possible original-key partition. The depth objective takes the worst execution branch. Thus the model combines a probabilistic success requirement with a worst-path resource

bound; it does not assert the target for every particular cipher’s realized collision count.

Fixed-instance false-key collisions. Consider the partition containing the original key and let $N = 2^n$. Under the ideal-cipher model, a one-pair predicate has $F \sim \text{Bin}(N - 1, 2^{-128})$ false keys in that partition. The marked-key count is $M = 1 + F$. One Grover attempt with j iterations returns the original key with probability

$$s_M(j) = \frac{\sin^2((2j + 1)\theta_M)}{M}, \quad \theta_M = \arcsin \sqrt{M/N}. \quad (3)$$

The divisor M is essential: amplitude amplification favors all marked keys, while the required event identifies one of them.

The public pairs remain fixed throughout an execution. Measurements and reinitializations create fresh attempts conditional on F , but do not draw a new F . For stages with query counts j_a and repetition counts R_a using that same one-pair predicate, the true-key success probability before the acceptance correction is

$$p_{\text{true}} = \mathbb{E}_F \left[1 - \prod_a (1 - s_{1+F}(j_a))^{R_a} \right]. \quad (4)$$

Multiplying marginal failure probabilities instead of conditional ones would remove this fixed-instance correlation and evaluate a different experiment. Rejected keys are not removed from subsequent predicates.

For the large domains, we approximate the binomial weights with Poisson weights of mean $(N - 1)2^{-128}$ and retain an explicit omitted-tail bound plus a Le Cam bound. Tiny verification instances use exact binomial weights. The search evaluates probabilities at 100 decimal digits. Selected plans are then certified with outward-rounded integer intervals, including the collision mixture, tail, and approximation error (Appendix A.4).

The probability that any incorrect key in the global key space passes both verification pairs is bounded by $\varepsilon = 2^{-128}$ under the stated ideal-cipher model. Subtracting this global bound also covers an incorrect accepted key preempting a later true-key result in another processor. This correction applies to the accepted-original-key event, not just to the oracle predicate in the original key’s partition. The collision model is an assumption for resource analysis; the experiments do not measure a collision distribution for CHAM itself.

Maximum depth and oracle transitions. We use each measured Grover iteration as a complete scheduled block, with the next iteration starting after that block finishes. For fixed-query stages using the same processor circuit, we charge

$$C_{\text{max}} = D(n) \sum_a j_a R_a + 3 \sum_a R_a - 1. \quad (5)$$

The $3 \sum_a R_a - 1$ term charges one layer each for preparation and measurement per attempt and one reset layer between attempts. Public-prefix X gates and initial Hadamards can share the preparation layer. Constant materialization and erasure needed inside a specialized oracle are instead present in its measured gate sequence. Maximum classical verification calls are $2P \sum_a R_a$ for a uniform deployment; this count is not converted into a quantum layer cost.

We also allow a one-pair stage (j_1, R_1) followed, only after all attempts are rejected, by a two-pair stage (j_2, R_2) . Each stage may select its own retention set and implementation. In the same- P control the processor count stays fixed. In the reallocation control, P_1 and P_2 are chosen independently from the feasible dyadic levels. The first stage ends with measured keys and clean oracle workspace; all used wires are reset before the second stage prepares a fresh full-space partition. No quantum state or exclusion list is transferred between stages.

The reallocation model uses a global all-to-all logical pool. Classical wire relabeling after reset has no additional logical depth. Let $Q_{\Sigma,a} = \sum_{p=1}^{P_a} Q_{a,p}$ be the actual allocation of stage a . Then

$$Q_{\text{peak}} = \max(Q_{\Sigma,1}, Q_{\Sigma,2}), \quad (6)$$

$$C_{\text{max}} = D_1 j_1 R_1 + D_2 j_2 R_2 + 3(R_1 + R_2) - 1. \quad (7)$$

For homogeneous generic stages, $Q_{\Sigma,a} = P_a Q_a$; synchronized prefix families use $D_a = \max_p D_{a,p}$. This is distinct from a model with permanently assigned per-processor memory. Coverage $P_i 2^{n_i} = 2^{128}$ is checked separately in both stages. The maximum verification-call count becomes $2(P_1 R_1 + P_2 R_2)$.

To evaluate the transition, couple the two-pair predicate to an oracle that marks only the original key. Let $A(P_1)$ be an upper bound on first-stage failure, calculated with the fixed one-pair collision mixture. In the coupled experiment the fresh second-stage failure $U(P_2)$ depends on its domain and plan, and not on the first-stage collision count. Thus a lower bound on accepted-original-key success is

$$p_{\text{acc}} \geq 1 - A(P_1)U(P_2) - \varepsilon. \quad (8)$$

The same global rare event covers a difference between the actual and coupled two-pair predicates and false-acceptance preemption. It is not subtracted once per retry. The coupling remains valid when $P_1 \neq P_2$ because the second stage covers the entire key space afresh. Small exact models test both directions of processor-count changes, including the actual acceptance ordering.

3.2 Joint selection and partition controls

Execution plans. For $N = 2^n$, the query grid with m divisions contains

$$\mathcal{J}_m(n) = \left\{ \left\lceil \frac{\pi i \sqrt{2^n}}{4m} \right\rceil : i = 1, \dots, m \right\} \cup \mathcal{A}(n), \quad (9)$$

where $\mathcal{A}(n)$ contains the first rising-branch query counts reaching unique-key probabilities 0.9, 0.95, 0.975, 0.99, 0.9925, 0.995, 0.9975, 0.999, 0.9995, and

0.9999. The tested grids are nested and cover the first unique-key Grover peak. We compare $m = 1024, 4096$, and 16384 throughout the budget grid.

The no-switch choices are fixed-query retries with at most 128 attempts, or two query lengths on the same one-pair circuit. Oracle transitions use a one-pair stage followed by a two-pair stage. We test total-attempt caps 16, 32, 64, and 128 for the two-query and transition plans. Both stages have positive query and repetition counts, and single-stage plans remain available in every comparison.

At each P , a fixed-query stage is represented by total queries jR , attempts R , and an upper failure bound. A point can be removed if another point is no larger in all three coordinates. Keeping attempts as a separate coordinate ensures that the reduction is valid under every smaller total attempt cap. For a one-pair to two-pair plan, each retained first stage is paired with the cheapest admissible second stage satisfying Eq. (8) and the remaining attempt allowance.

For two query lengths on one unchanged one-pair circuit, we retain the conditional failure vector over F before applying Eq. (4). The selected plan must satisfy the accepted-key target, and its cost is evaluated by Eq. (5) or (7). Appendix A describes the independent selection checks.

Circuit and deployment choice. A comparison condition is $(r, B, \tau, d_{\text{CCX}})$, where r is the round count, τ is the accepted-original-key success target, and d_{CCX} identifies the decomposition. For each condition we retain all feasible measured generic profiles and complete measured prefix families. Each complete prefix family is available at every budget that admits its actual allocation. The same rule applies to the endpoint and library groups. Within a fixed $(P, \text{pair count})$, a smaller depth with no larger feasible allocation weakly improves every plan in the logical-pool model. This reduces the subsequent plan search without discarding a feasible improvement in maximum depth.

For each group of circuits, we compare (i) no switch, (ii) a one-pair to two-pair switch with the same P , and (iii) a switch allowing independent P_1, P_2 . The endpoint group restricts each stage to OLD or FULL; the library group admits all measured retention sets. A stage may change implementation and retention set. Feasibility is checked using its actual allocation, and the two-stage peak uses Eq. (6). The result is the minimum over the declared finite menu, with deterministic tie breaking. Because the available menu is nested as B increases, the selected cost cannot increase with the budget. We check that invariant explicitly.

Let $C_{\text{endpoints}}(c)$ and $C_{\text{all}}(c)$ denote the minimum C_{max} over these endpoint and library menus at condition c . We report the relative saving

$$\Delta(c) = \frac{C_{\text{endpoints}}(c) - C_{\text{all}}(c)}{C_{\text{endpoints}}(c)}. \quad (10)$$

Because the library group includes both endpoints, $\Delta(c) \geq 0$ by construction. A fixed partial-retention design does not have that guarantee. Our descriptive means average the condition-wise ratios in Eq. (10); they are not ratios of aggregate depths or estimates under a random distribution of hardware budgets.

The circuit-level proxy QD_0^2 , where $D_0 = D(128)$, has a useful interpretation. It multiplies the per-processor logical-qubit count by the squared full-key Grover-iteration depth. If allocation is relaxed to $P \simeq B/Q$, iteration depth is approximated by D_0 , and a common unique-key query constant κ applies, then $C_{\max} \simeq \kappa 2^{64} D_0 \sqrt{Q/B}$, so minimizing its square ranks circuits by QD_0^2 . The diagnostic quantifies the effect of replacing this relaxation with integer allocations, measured $D(n)$, and success-constrained execution plans.

Homogeneous prefix partitions. A processor count between two powers of two can be implemented with a complete binary-prefix tree. For example, 31 leaves can consist of one domain of size 2^{124} and thirty domains of size 2^{123} . This uses all 31 processors, but the largest domain remains as large as a domain in a 16-processor uniform deployment. Our success requirement applies to every possible true-key leaf, and our depth is the maximum branch depth.

Proposition 1. *Fix a generic circuit configuration with common width Q on every leaf and a common allowed execution menu depending only on prefix depth. Let C_d be that menu's minimum branch cost at prefix depth d . Every feasible complete binary-prefix tree is weakly dominated, in total width and maximum cost, by one of its feasible uniform dyadic levels.*

Proof. Let d be the depth of the shallowest leaf. Since there are no leaves above depth d , all 2^d nodes at depth d exist. The tree therefore has at least 2^d leaves and uses at least $2^d Q$ qubits. A shallowest leaf costs at least C_d , so the tree's maximum cost is at least C_d . The uniform level at depth d uses exactly $2^d Q$ qubits and has cost C_d , which establishes the claim.

The proposition does not assume $C_{d+1} \leq C_d$. It justifies considering all feasible dyadic levels for this homogeneous generic model; it does not prove that the maximum feasible P is always optimal. Its hypotheses fail when prefix values change the circuit resources, leaves use different configurations, or arbitrary intervals require different preparation and reflection circuits. It is also a worst-branch result, not an average-runtime result.

Specialized and heterogeneous prefix trees. For specialized circuits, a measured prefix node v can search its entire domain or split into two children. Only the leaves of the resulting static partition execute. Each implementation family has a maximum measured prefix depth L , the number of fixed leading key bits. In the homogeneous control a deployment fixes S , pair layout, and key schedule; nodes choose their fastest measured compiler among alternatives of equal width. In the heterogeneous control each leaf can also choose its own measured retention set and pair layout, including pair count. All families use the same regenerated FSE key program and the depth bounds in Appendix A.3. Each leaf receives the no-switch plan for its domain and pair count. Combining child states (Q_0, C_0) and (Q_1, C_1) gives

$$(Q_0 + Q_1, \max(C_0, C_1)). \quad (11)$$

The union of direct leaf choices and child combinations is Pareto-pruned at each node. An independent deadline-based recurrence checks the frontier, and explicit witnesses check coverage, width, cost, and leaf success (Appendix A.3).

Every leaf meets the same accepted-original-key target. The true key lies in exactly one leaf, and the common two-pair verification event supplies the same global false-acceptance bound. Thus mixing retention and layout does not change the success event. Both comparison groups receive this extension. The measured node bank and family-specific depth bounds define the extent of the tree search.

4 Experimental Results

Admitting partial-retention sets reduces complete-search cost beyond the baseline restricted to no retention and full retention. Under the final combined menu, mean condition-wise reductions in maximum scheduled logical depth are 11.12% for 80 rounds and 8.39% for 112 rounds. Improvements are strict in 606 and 588 of the 654 conditions, respectively; the remaining 48 and 66 are exact cost ties (Table 3). The later partition and execution controls test whether further choices improve these already selected costs.

Table 3. Condition-wise reductions in maximum logical depth over the matched no-retention/full-retention baseline. Each row uses the same 654 conditions. Strict improvements and ties compare exact integer costs. Both groups are reoptimized within each menu, with partial retention additionally allowed for the library group; relative savings need not grow with menu size.

Rounds	Mean (%)	Max (%)	Improved	Tied
			conditions / 654	conditions / 654
<i>Main: generic + measured prefix, with transitions</i>				
80	11.123	24.553	606	48
112	8.391	19.528	588	66
<i>Control: generic, no switch</i>				
80	11.505	24.553	606	48
112	8.485	19.528	588	66

4.1 Comparison conditions

The main grid contains 109 budgets between 2^{12} and 2^{20} logical qubits, success targets 0.99, 0.9925, and 0.995, and two CCX decompositions: 654 conditions per round count. Of the budgets, 65 form a broad grid and 44 probe deployment boundaries. The broad grid divides each doubling interval into eight equal steps and includes the final endpoint. Appendix A.7 lists the boundary probes. All groups use the same conditions.

The final comparison includes all 83 and 99 generic sets for the respective round counts and complete FSE prefix families for the reference public input wherever their actual allocation fits. The specialized bank comes from focal families; the common generic menu covers the full library. Results use the 16,384-division query grid and 128-attempt execution menu. Generic-only comparisons and coarser execution menus provide controls.

Both round counts are measured directly using Python 3.12.10 and Qiskit 2.5.2.

4.2 Selected deployments and budget effects

Figure 1 shows where the reductions occur. The broad-grid means are 11.60% and 8.62%; the complete-grid means also include the boundary probes.

Table 4 pairs the selected baseline and joint-selection deployments. At $B = 4096$, the 80-round joint choice uses one processor instead of four, but its much shorter iteration still reduces complete search depth by 16.34%. For 112 rounds at that budget, both choices use four processors. The selected specialized depths are 103,671 and 98,722, giving a 4.774% reduction; the generic-only example in Table 1 uses different depths and gives 4.731%.

These deployments also show that retaining more is not a monotone response to a larger budget. For 80 rounds, these examples select 50 retained addends at $B = 4096$, three at $B = 65536$, and 19 at $B = 2^{20}$. For 112 rounds, all four examples select three.

Table 4 uses the partial sets 112-012: $\{6, 42, 78\}$, 80-016: $\{14, 38, 66\}$, 80-060: $\{2, 6, \dots, 74\}$, 80-078: $\{\lfloor 80i/50 \rfloor : 0 \leq i < 50\}$. All round indices are zero-based. Appendix A.6 gives the reference public input, implementation choices, and exact query plans.

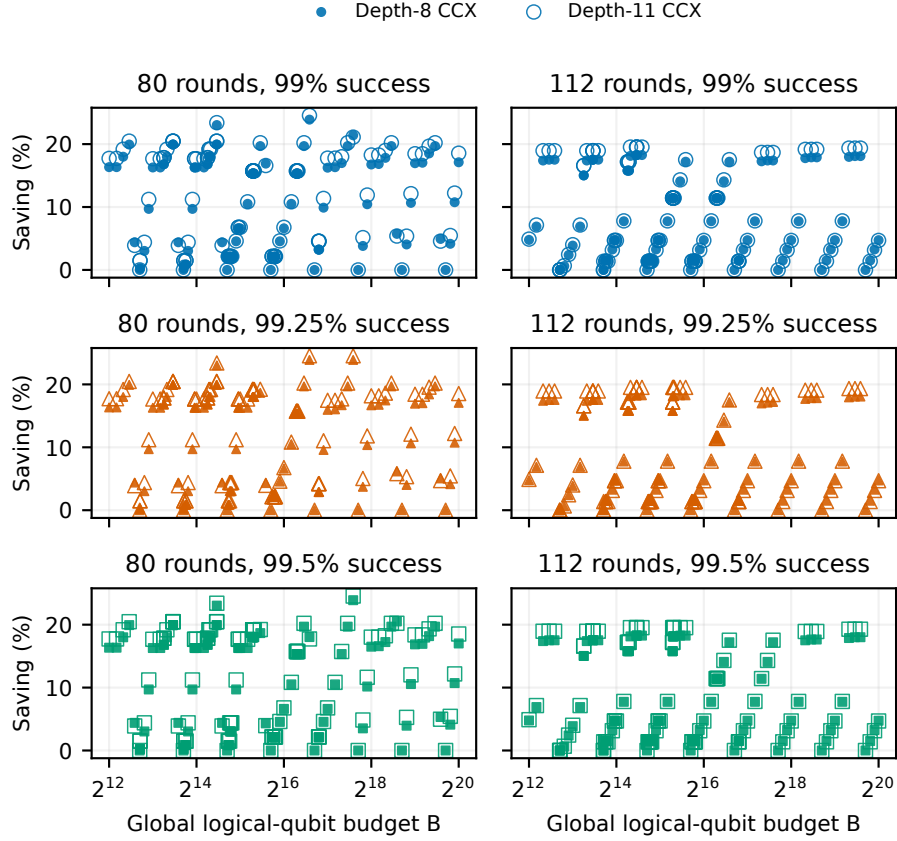


Fig. 1. Savings under the final generic-plus-prefix menu, with oracle transitions permitted in both groups, relative to the matched endpoint baseline (Table 3). Columns separate round counts and rows separate success targets. Filled and open markers indicate depth-8 and depth-11 CCX, respectively; color is supplementary. Each series covers all 109 budgets.

Table 4. Paired deployments at target 99.5% and depth-8 CCX. Base restricts retention to OLD/FULL; Joint admits all measured sets. Saving is relative to Base and computed from exact costs. Parentheses give k ; superscript p denotes prefix specialization. Q_Σ is total allocation and $n = 128 - \log_2 P$. Two-pair rows use fanout; one-pair rows hold the output. Every plan uses one attempt, except $P = 128$, which uses two query lengths once each.

	B	Menu	Set (k)	P	Pairs	Q_Σ	$D(n)$	$C_{\max}/10^{23}$	Saving (%)
<i>80 rounds</i>									
4,096	Base	OLD	(0)	4^p	2	3,092	73,914	5.113113	–
	Joint	80-078	(50)	1^p	2	3,973	30,918	4.277599	16.341
16,384	Base	OLD	(0)	16^p	2	12,368	73,914	2.556557	–
	Joint	80-078	(50)	4	2	15,892	30,938	2.140183	16.286
65,536	Base	OLD	(0)	128	1	49,408	74,834	1.108976	–
	Joint	80-016	(3)	128	1	61,696	69,885	1.035636	6.613
1,048,576	Base	OLD	(0)	2048	1	790,528	74,834	0.229136	–
	Joint	80-060	(19)	1024	1	1,017,856	43,845	0.190149	17.015
<i>112 rounds</i>									
4,096	Base	OLD	(0)	4^p	2	3,092	103,671	7.171599	–
	Joint	112-012	(3)	4^p	2	3,860	98,722	6.829244	4.774
16,384	Base	OLD	(0)	16	2	12,368	104,611	3.618312	–
	Joint	112-012	(3)	16	2	15,440	99,662	3.447135	4.731
65,536	Base	OLD	(0)	128	1	49,408	104,593	1.549978	–
	Joint	112-012	(3)	128	1	61,696	99,644	1.476638	4.732
1,048,576	Base	OLD	(0)	2048	1	790,528	104,593	0.320255	–
	Joint	112-012	(3)	2048	1	987,136	99,644	0.305102	4.732

4.3 Why count, placement, and the selector matter

Figure 2 shows the generic width–depth frontiers at $n = 128$, separately by round and pair count. Deployment uses the measured $D(n)$ and plan for the partition that actually fits.

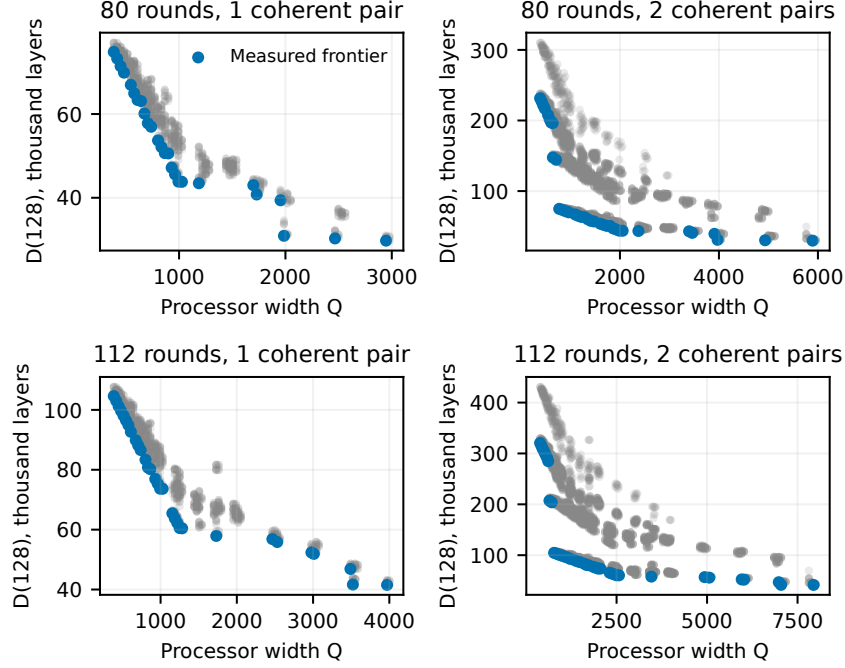


Fig. 2. Generic width–depth choices at $n = 128$ with depth-8 CCX, highlighting non-dominated points. Each panel fixes round and coherent pair count, using the applicable configurations from the common menu: 15 per set for one pair and 45 for two pairs.

Holding the retention count fixed, we give every placement its best feasible generic configuration and no-switch plan. The largest observed cost excess of the worst placement over the best is 24.69% for 80 rounds and 38.40% for 112 rounds. Table 5 identifies these extrema and their selected deployments; Appendix A.5 specifies the sets. In the 80-round pair, width, processor count, query plan, and gate counts agree, so the depth difference comes from dependencies and scheduling (Section 2). These maxima show placement sensitivity within the measured library, rather than a typical penalty or a universally preferred round pattern.

Table 5. Observed same-count placement extrema at target 99.25% and depth-11 CCX. Budgets are 113,743 (80 rounds) and 229,376 (112 rounds). All four choices use the same regenerated FSE program, one held pair, Qiskit commutation optimization, $P = 128$, and one attempt of 1 240 243 248 481 040 091 queries.

Rounds	k	Set ID	Q	$D(121)$	$C_{\max}/10^{22}$
80	15	80-050	866	57,100	7.081789
80	15	80-049	866	71,199	8.830408
112	42	112-081	1,730	65,154	8.080681
112	42	112-080	1,730	90,170	11.183273

The selector diagnostic fixes pair count and ranks configurations by either $D_0 = D(128)$ or QD_0^2 . Every tied minimum receives its best feasible deployment and no-switch plan; the best tie supplies the primary result. Table 6 compares this cost with the minimum in the same pair-count stratum. For two-pair oracles, the QD_0^2 choice has mean excess cost of 10.04% and 10.82% for 80 and 112 rounds. The 112-round example in Table 1 isolates one cause: both choices fit four processors, so the continuous width penalty changes the ranking without changing the actual parallelism.

Table 6. Final-grid generic proxy diagnostics with pair count fixed. Relative excess is measured against the minimum in that same stratum; ties use their best deployment.

Rounds	Pairs	Selector	Mean excess (%)	Max (%)	Proxy-worse
					conditions / 654
80	1	D_0	135.287	274.419	654
80	1	QD_0^2	85.082	227.354	652
80	2	D_0	25.868	45.623	636
80	2	QD_0^2	10.044	31.124	528
112	1	D_0	201.007	430.563	654
112	1	QD_0^2	8.472	24.017	600
112	2	D_0	41.103	64.241	654
112	2	QD_0^2	10.823	24.267	588

4.4 Do partition restrictions create the observed boundaries?

For homogeneous generic circuits, 84 comparison conditions give equal minimum costs for uniform dyadic and balanced prefix partitions, including all feasible processor counts (2 round counts, 7 budgets, 3 targets, and 2 CCX choices). A balanced partition uses leaves at the two adjacent depths needed for its processor count. For specialized circuits, bounded complete trees improve 0 of 48 group/condition costs over uniform families.

Allowing each leaf its own measured retention set, layout, and pair count also leaves these 48 costs unchanged. All 36 width–cost frontiers agree with the homogeneous-family envelope. Appendix A.3 specifies the families, prefix depths, and comparison axes behind these counts. The primary budgets are 4096 for both round counts and 22607, 22608 for 80 rounds; additional checks cover every integer budget from 4096 to each family’s largest primary budget.

The fixed 80-round configuration with $S = \{2, 6, \dots, 74\}$, two-pair fanout, and $Q = 1989$ illustrates why circuit reselection matters. At $B = 63,647$, 31 copies fit; the largest prefix domain still determines the tree’s maximum depth. At $B = 63,648$, 32 equal domains fit and the fixed-configuration cost drops by approximately $1 - 1/\sqrt{2}$. Re-evaluating both budgets over all 83 measured generic retention sets for 80-round CHAM and the final no-switch plans gives exactly the same selected cost at both budgets for depth-8 CCX and 99.5% success.

A separate boundary survives reselection. From $B = 22607$ to 22608, the 80-round, depth-8, 99.5% choice changes from nine to ten retained addends, with 16 processors in both cases, and cost falls by 3.87% (Figure 3). The two examples separate a circuit-specific capacity drop from one in the selected portfolio.

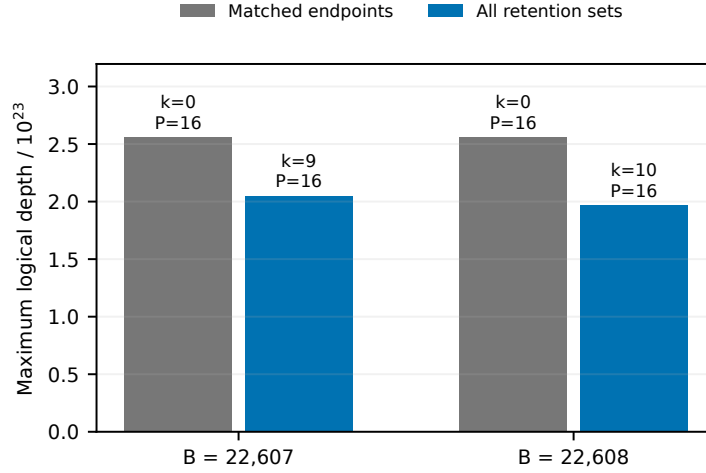


Fig. 3. Selected deployments around the 22,608-qubit boundary for 80 rounds, 99.5% success, and depth-8 CCX. Bars plot the exact costs; labels identify retention and processor counts. Adjacent integer budgets are separate points.

4.5 Do stronger execution menus change the choices?

Both groups receive fixed retries, two query lengths on one unchanged one-pair circuit, and one-pair to two-pair transitions. At the final query grid, neither

same- P transitions nor independent second-stage P reduce any selected cost relative to the no-switch menu. Each extension covers 2,616 comparisons: two round counts, 654 conditions, and two groups.

Refining the query grid from 4096 to 16,384 divisions changes the final selected costs by at most 0.0193%. Increasing the attempt cap for two-query and transition plans from 16 to 128, while retaining 128-attempt fixed-query plans throughout, does not reduce the selected costs. These checks distinguish sensitivity to query resolution from gains due to the larger execution menus.

Directed integer intervals certify that all selected plans and checked transition plans meet their success targets. Independent 200-digit trigonometric evaluation agrees with the corresponding enclosures (Appendix A.4).

4.6 Public inputs and circuit correctness

The input comparison applies all 60 generic configurations to the focal 11-set library and four public input instances, then reprices plans at the final query grid. Across 180 additional-input generic conditions, the largest absolute fixed-deployment cost change is 0.0413%; with the complete matched prefix families it is 0.0413%. A fixed deployment preserves its S , configuration, P , and plan, including each prefix’s compiler for specialized families. Table 7 separates fixed-deployment feasibility and losses from reselected choices.

Table 7. Public-input sensitivity: 5 budgets $\times$ 3 targets $\times$ 2 CCX choices $\times$ 3 additional inputs give 90 conditions per round count and menu (Appendix A.7). The combined menu includes generic and prefix-specialized circuits. A fixed loss means the original deployment costs more than the reselected endpoint baseline.

Rounds	Menu	S changes	Fixed infeasible	Fixed losses	Max abs. change (%)
80	generic	0	0	0	0.019408
80	combined	0	0	0	0.019408
112	generic	0	0	0	0.041348
112	combined	0	0	0	0.041348

Known-answer and independent classical checks validate encryption; phase checks verify marking and restored workspace. Compiler and prefix equivalence checks validate circuit transformations, while expanded gate-dependency graphs verify the reported depths. Appendix A describes these checks.

5 Interpretation and Scope

Retention is useful when its scheduling benefit outweighs its effect on processor capacity and the required search plan. Table 1 shows a simple case: greater width leaves the processor count unchanged, so a continuous width penalty rejects a

useful circuit. At other budgets, losing a processor level can outweigh a shorter iteration.

Two further observations guide deployment decisions. A larger budget need not select a larger retention count: the 80-round examples move from 50 retained addends to three and then to 19. Likewise, a capacity boundary of one fixed circuit need not remain a boundary of the selected library. The 63,648-qubit discontinuity disappears after reselection, whereas the 22,608-qubit boundary survives. Circuit configurations should therefore be reselected at the budget of interest before interpreting either a retention trend or a capacity jump.

Oracle transitions leave the full-grid selected costs unchanged; heterogeneous prefix choices do so in the bounded partition controls. The evidence covers a finite CHAM library, bounded execution plans, and the measured prefix depths. Different circuit synthesis or arbitrary-interval search could change the frontier. Likewise, the four input instances support the reported sensitivity check, and the ideal-cipher probability model and unit logical-layer costs define the interpretation of the results. Physical runtime and input-independent performance require additional models.

6 Conclusion

Joint retention and deployment selection reduces maximum scheduled logical depth by 11.12% for 80-round CHAM and 8.39% for revised 112-round CHAM on average over the measured grid. The baseline restricts retention to no retention and full retention under the same implementation and execution rules. Improvements are strict in 606 and 588 of the 654 conditions, respectively.

The selected retention count need not grow with the budget, and a fixed-circuit capacity boundary can disappear after reselection. Retained locations should therefore be chosen jointly with feasible processor allocations and execution plans that meet the original-key recovery and acceptance target.

References

1. Choi, C., Oh, J., Lee, S., Cho, G., Choi, D.: An automated synthesis framework for benchmarking quantum resource costs of symmetric-key cryptography. *Mathematics* **14**(4), 719 (2026). <https://doi.org/10.3390/math14040719>, <https://www.mdpi.com/2227-7390/14/4/719>
2. Jang, K., Oh, Y., Seo, H.: Depth-optimized quantum implementation of CHAM. *Cryptology ePrint Archive*, Paper 2025/1188 (2025), <https://eprint.iacr.org/2025/1188>, revised 2025-08-04
3. Jaques, S., Naehrig, M., Roetteler, M., Virdia, F.: Implementing Grover oracles for quantum key search on AES and LowMC. In: *EUROCRYPT 2020* (2020). https://doi.org/10.1007/978-3-030-45724-2_10, <https://eprint.iacr.org/2019/1146>, ePrint version revised 2023-06-07
4. Kim, P., Jeong, K.C., Han, D.: Time-space complexity of quantum search algorithms in symmetric cryptanalysis. *arXiv:1805.05534v1* (2018), <https://arxiv.org/abs/1805.05534>

5. Paradis, A., Bichsel, B., Vechev, M.: Reqomp: Space-constrained uncomputation for quantum circuits. *Quantum* **8**, 1258 (2024). <https://doi.org/10.22331/q-2024-02-19-1258>, <https://arxiv.org/abs/2212.10395v2>
6. Roh, D., Koo, B., Jung, Y., Jeong, I.W., Lee, D.G., Kwon, D., Kim, W.H.: Revised version of block cipher CHAM. In: *Information Security and Cryptology – ICISC 2019*. LNCS, vol. 11975, pp. 1–19 (2020). https://doi.org/10.1007/978-3-030-40921-0_1, https://link.springer.com/chapter/10.1007/978-3-030-40921-0_1
7. Xiang, Z., Zeng, X., Lin, D., Bao, Z., Zhang, S.: Optimizing implementations of linear layers. *Cryptology ePrint Archive*, Paper 2020/903 (2020), <https://eprint.iacr.org/2020/903>
8. Yang, Y., Jang, K., Bakshi, A., Seo, H.: Optimized implementation and analysis of CHAM in quantum computing. *Applied Sciences* **13**(8), 5156 (2023). <https://doi.org/10.3390/app13085156>, <https://www.mdpi.com/2076-3417/13/8/5156>

A Implementation and Validation Details

A.1 Published baselines and key synthesis

Published 80-round encryption costs are $(Q, D) = (292, 37766)$ for the original circuit, $(259, 37768)$ for FSE, and $(2852, 14772)$ for full retention [2]. We reproduce the first and third from public code. The original published FSE program was unavailable, so we regenerate an FSE-style linear program using Xiang et al. [7]. Eight trials per 32-bit map with seed 20260910 select the smallest CNOT count, breaking ties by trial order. The main baseline then measures complete phase oracles and diffusion under the common contract.

A.2 Circuits and selection checks

Independent classical encryption and published known-answer vectors check the cipher function. Basis states and small coherent superpositions check the marked phase, key preservation, and restoration of workspace. Exact compiler checks verify allowed gate reorderings and rewrites; prefix checks verify equivalence on the fixed-prefix subspace. Expanded gate-dependency graphs independently check depth after each CCX decomposition, including scheduling across the oracle–diffusion boundary.

The plan search removes points dominated in total queries, attempts, and failure. For two query lengths on one circuit it preserves the conditional failure vector over F , as required by Eq. (4). Independent deadline enumeration checks the resulting minima, while small exhaustive instances check the cost and probability screens. Since measured iteration depths exceed 384 and at most 128 attempts are allowed, one additional query outweighs any possible difference in reset overhead. This permits shared query templates for an unchanged circuit; transitions between different circuits use both actual depths.

Table 8. Maximum measured prefix depth L by family. One-pair layouts hold the output; two-pair layouts use key fanout. A dash means the layout is outside that comparison.

Rounds	Primary budget(s)	Retention	One pair	Two pairs
80	4,096	OLD	4	3
80	4,096	S_{50}	1	0
80	4,096	FULL	0	0
112	4,096	OLD	4	3
112	4,096	S_3	3	2
112	4,096	FULL	0	0
80	22,607; 22,608	OLD	–	5
80	22,607; 22,608	S_9	–	4
80	22,607; 22,608	S_{10}	–	4
80	22,607; 22,608	FULL	–	2

Table 9. Composition of the partition-control counts. Both endpoint and library groups receive each specialized-tree extension.

Reported count	Comparison axes
84 generic conditions	2 round counts $\times$ 7 budgets $\times$ 3 targets $\times$ 2 CCX choices. Budgets: 4,096; 22,607; 22,608; 63,647; 63,648; 65,536; 1,048,576.
48 selected costs	4 round/budget pairs $\times$ 3 targets $\times$ 2 CCX choices $\times$ 2 groups. Pairs: (80, 4096), (112, 4096), (80, 22607), (80, 22608).
36 tree frontiers	3 measured family scopes (80-round low budget, 112-round low budget, 80-round boundary) $\times$ 3 targets $\times$ 2 CCX choices $\times$ 2 groups.

A.3 Prefix-tree comparison families

Table 8 specifies the measured depths used for the homogeneous and heterogeneous tree comparisons. Depth L fixes L key bits; all prefix values through that depth are represented. The partial sets are $S_3 = \{6, 42, 78\}$ for 112 rounds, $S_{50} = \{\lfloor 80i/50 \rfloor : 0 \leq i < 50\}$ for 80 rounds, $S_9 = \{6, 14, 22, 30, 38, 50, 58, 66, 74\}$, and $S_{10} = \{4, 12, \dots, 76\}$ for the 80-round boundary. Each family uses the regenerated FSE program; nodes can select unoptimized, Qiskit-only, or unit-weight commutation-ordered compilation.

Both endpoint and library groups retain their feasible generic alternatives. Table 9 expands the comparison counts.

For an independent check, fix a deadline T and compute

$$W(v, T) = \min \left\{ \min_{\ell \in \mathcal{L}_v: C_\ell \leq T} Q_\ell, W(v0, T) + W(v1, T) \right\}, \quad (12)$$

where $\mathcal{L}_v$ contains the measured leaf choices at prefix v . A missing choice has infinite width. Checking every leaf-cost deadline reconstructs the same frontier as

the Pareto composition. Explicit tree checks verify disjoint prefix coverage, total width, and maximum cost. Exhaustive small trees through depth three also check these recurrences, including an example where heterogeneous leaves improve the result.

A.4 Certified selected probabilities

The interval certifier stores endpoints as integers scaled by 2^{512} . Products, quotients, and integer square roots round outwards. For $x = M/N$, write $c = 1 - 2x$ and $s = 2\sqrt{x(1-x)}$. If $(c + is)^j = u + iv$, the original-key probability is

$$s_M(j) = \frac{(v\sqrt{1-x} + u\sqrt{x})^2}{M}. \quad (13)$$

Binary powering evaluates this expression in $O(\log j)$ interval operations. Exact-rational amplitude recurrences on small domains independently check the enclosure.

For one pair, odd/even alternating Taylor sums enclose the Poisson weight $e^{-\lambda}$ for $\lambda = (N-1)2^{-128} \leq 1$. The mixture retains $F = 0, \dots, 80$. With w_{80} the final weight, the omitted tail is bounded by

$$\frac{w_{80}\lambda/81}{1 - \lambda/82}. \quad (14)$$

We also include the conservative Le Cam error $2(N-1)2^{-256}$. Retry failures are multiplied before the mixture, and the global 2^{-128} acceptance correction is applied once per complete plan. All selected lower bounds meet their targets. A separate 200-digit sine/arcsine evaluation checks containment in the interval for the conservative corrected expression; the lower endpoint supplies the success certificate.

A.5 Exact retention-library specification

The construction budgets are $\mathcal{B} = \{m2^e/4 : e = 12, \dots, 19; m = 4, 5, 6, 7\} \cup \{2^{20}\}$. For each $r \in \{80, 112\}$, consider every $b \in \mathcal{B}$, $j = 0, \dots, 11$, and $(q, w) \in \{(386, 32), (773, 64)\}$. Compute $t = \lfloor ([b/2^j] - q)/w \rfloor$ and retain the values with $1 \leq t < r$ in T_r . The candidate counts are

$$K_r = \{t - \delta : t \in T_r, \delta \in \{0, 1, 2\}\} \cap \{1, \dots, r-1\}.$$

For every $k \in K_r$ and $p \in \{0, 1\}$, include

$$\begin{aligned} E_{r,k,p} &= \{(\lfloor ir/k \rfloor + \lfloor pr/(2k) \rfloor) \bmod r : 0 \leq i < k\}, \\ L_{r,k,p} &= \{2 + 4(\lfloor ih/k \rfloor + \lfloor ph/(2k) \rfloor) \bmod h : 0 \leq i < k\}, \quad h = r/4, \end{aligned}$$

where the lattice family L is included only when $k \leq h$. Add the empty and full sets and the twelve reference sets in Table 10, then remove duplicates. This gives exactly 83 sets for 80 rounds and 99 for 112 rounds. Set IDs sort each round's sets by $(|S|, \text{lexicographic } S)$, starting from one; for example, 80-001 is the empty 80-round set.

Table 10. The twelve additional reference sets. The hexadecimal integer is $H(S) = \sum_{i \in S} 2^i$: bit zero represents round zero. Together with the deterministic spacing families, these specify all 182 measured sets.

Set ID	k	$H(S)$ (hexadecimal)
80-008	2	00000000100000000010
80-009	2	00000000400000000040
80-014	3	00000400000040000040
80-029	8	00401004010040100401
80-040	11	08104001040810408104
80-042	12	08104081040810408104
80-055	17	044444444440444444440
112-012	3	0000000040000000040000000040
112-015	4	0004000000400000040000004000
112-026	7	0010001000100010001000100010
112-061	24	04444444044444440444444404444444
112-065	25	04444444444444440444444444444440

A.6 Reference input and selected execution plans

The reference plaintexts m_i and ciphertexts c_i are listed as four hexadecimal 32-bit words in state order (x_0, x_1, x_2, x_3) :

Rounds	Value	Words
80/112 m_1	33221100	77665544 bbaa9988 ffeeddcc
80/112 m_2	00000000	00000000 00000000 00000000
80 c_1	c3746034	b55700c5 8d64ec32 489332f7
80 c_2	1058ac84	4a742526 21e931db 6d085447
112 c_1	d05419ee	9f118f4c 99e36469 1c885ec1
112 c_2	2ebbc73f	4035b45a feb83aa8 1c647c69

One-pair oracles use m_1 and its ciphertext; verification uses both pairs.

Every deployment in Table 4 uses the regenerated FSE program. Compilation uses unit-weight commutation ordering, except for the 80-round Joint rows at $B = 4096, 16384$, and 1048576 , which use Qiskit-only optimization. Each specialized family includes every $\log_2 P$ -bit prefix, with the same compiler choice for all its members. Table 11 gives the per-processor plans for both groups.

Table 11. Exact per-processor query counts for Table 4. Each listed count is used once. A second count means a fresh attempt on the same one-pair circuit; all processors follow the same plan.

Pairs	P	First attempt j_1	Second attempt j_2
2	1	13 835 303 305 968 408 443	—
2	4	6 917 651 652 984 204 222	—
2	16	3 458 825 826 492 102 111	—
1	128	204 388 460 724 597 923	1 277 525 579 557 764 071
1	1,024	433 683 934 688 806 023	—
1	2,048	306 191 890 970 788 667	—

Together with the measured $D(n)$ in Table 4 and Eq. (5), these counts determine the unrounded search costs.

A.7 Evaluation budgets and input-sensitivity conditions

The 44 boundary probes are $\{h - 1, h : h \in \mathcal{H}\}$, where

$$\mathcal{H} = \{6688, 9760, 11296, 13392, 13856, 15904, \\ 19536, 20000, 22608, 27168, 27728, 28192, \\ 31824, 39968, 40016, 40480, 54352, 56400, \\ 56864, 79952, 80976, 113744\}.$$

Their union with the 65 broad-grid values in Section 4.1 gives exactly the 109 evaluation budgets.

The input-sensitivity study uses seven 80-round sets: 80-001, 80-016, 80-033, 80-036, 80-060, 80-078, 80-083. The four 112-round sets are 112-001, 112-012, 112-070, 112-099. Set IDs follow Appendix A.5.

In addition to the reference input in Appendix A.6, use seeds 20260925, 20260926, and 20260927. Initialize Python 3.12.10’s `random.Random` with each seed and make twelve successive calls to `getrandbits(32)`. The first four are the key words, the next four form m_1 , and the last four form m_2 , in the word order of Appendix A.6. The same key and plaintexts are used for both round counts; encrypting them gives the corresponding ciphertext pairs.

Table 7 crosses budgets 4096, 22607, 22608, 65536, and 1048576; targets 0.99, 0.9925, and 0.995; both CCX decompositions; and the three additional inputs: $5 \times 3 \times 2 \times 3 = 90$ conditions per round count and menu. The reference input supplies the original deployment for each comparison; it is not an additional condition in the 90.