A Noise-Aware Staged Oracle Strategy for Compound Quantum Search over NoSQL Collections

Jessica Zaqueros-Martínez[0000−0003−0736−8820], Bruno Ramos-Cruz[0000−0001−7074−1179], Bapi Dutta[0000−0002−2796−8914], and Luis Martínez[0000−0003−4245−8813]

Department of Computer Science, University of Jaén, Campus Las Lagunillas s/n. 23071,Jaén, Spain
{zaqueros,brcruz,bdutta,martin}@ujaen.es

Abstract. Grover's quantum search algorithm offers a provable $\mathcal{O}(\sqrt{N})$ oracle-query advantage over classical $\mathcal{O}(N)$ scans for unstructured data, a regime encountered whenever a NoSQL document store receives a predicate targeting an unindexed field. Recent complexity-theoretic results establish, however, that NISQ hardware cannot achieve this quadratic speedup for unstructured search in general. Rather than treating this as a disqualifying limitation, the problem is reframed here as a noise-constrained engineering challenge: given a fixed depolarising error rate p, which oracle architecture maximizes the practical accuracy threshold for compound AND predicates over heterogeneous NoSQL collections? To address this question, an experimental hybrid Qiskit–MongoDB architecture is developed and evaluated across three heterogeneous open-data-inspired schemas with zero circuit modification between them, showing that compound AND predicates may, in some cases, require fewer Grover iterations than single-field queries when joint selectivity increases match density, reflecting the dependence of iteration count on match density. Further circuit-level analysis finds that a selectivity-ordered staged oracle (SOSO), which decomposes compound predicates into sequential sub-oracles ordered by ascending match count, reduces effective noise exposure $\mathcal{E}$ (defined in terms of the gates G, the error rate p, and the iterations k), relative to a monolithic flat oracle, suggesting a possible extension of the useful simulated accuracy regime from $N \leq 8$ to $N \leq 16$ at $p = 0.1\%$. A two-parameter (N, p) simulated operating boundary derived from systematic end-to-end timing and noise experiments offers a preliminary simulation-based decision framework for quantum-classical query routing in NoSQL pipelines.

1 Introduction

Document-oriented NoSQL stores such as MongoDB, CouchDB, Amazon DocumentDB, have become the dominant persistence layer for applications that

prioritise schema flexibility over query rigidity [3]. Their fundamental computational cost is a full collection scan of $\mathcal{O}(N)$ complexity whenever a query predicate targets a field for which no index has been pre-built. Classical solutions like B^+-trees, hash indices, bloom filters, require the indexed field's structure to be known at schema-design time; for schema-agnostic collections with dynamically determined predicates, no classical pre-indexing strategy covers the full query space without prohibitive storage overhead. Although the present experiments are small-scale, as collection sizes grow to hundreds of records, even millisecond-per-document evaluation times accumulate into latencies that render ad-hoc compound queries impractical, particularly in exploratory analytics and regulatory compliance scenarios where the set of query predicates cannot be anticipated at index-design time [3].

Grover's algorithm [6] reduces the number of oracle evaluations required for unstructured search from $\mathcal{O}(N)$ to $\mathcal{O}(\sqrt{N})$ without imposing structural assumptions on the data. This setting is particularly relevant to schema-agnostic NoSQL environments, where query predicates may target dynamically determined fields for which no index exists. Its oracle model admits a conceptual mapping to document predicate evaluation: the quantum register encodes document indices, while the oracle is dynamically constructed from the runtime query predicate, independent of the underlying JSON schema [11]. Early proposals for quantum database search [7], together with subsequent extensions to multiple-solution search and amplitude estimation [2], established the theoretical foundations upon which contemporary hybrid architectures are built. More recent surveys identify unstructured search as one of the few computational problems for which a quantum primitive offers an unconditional asymptotic advantage over classical algorithms [1]. The practical realization of this advantage is, however, constrained by current hardware limitations. NISQ devices suffer from decoherence and gate errors that fundamentally limit achievable circuit depths [8], and it has been proved that no NISQ algorithm can achieve the quadratic speedup for unstructured search in general [4]. More specifically, under depolarising noise of rate p, the noisy-query complexity grows as $\tilde{\Theta}(\max\{np, \sqrt{n}\})$, eroding the Grover advantage linearly once $p > 1/\sqrt{n}$ [9]. These limitations have important engineering implications. For example, a Grover circuit comprising 17 gates in noiseless simulation may expand to more than 200 gates after noise-aware compilation at $p = 0.1\%$. The standard monolithic oracle for compound AND predicates exacerbates this issue, because all M matching indices are marked within a single circuit of depth $\mathcal{O}(Mn)$. Consequently, noise accumulates in proportion to the total match count, irrespective of how selectively individual field predicates constrain the candidate set.

Motivated by these limitations, this paper develops the Selectivity-Ordered Staged Oracle (SOSO), which evaluates compound predicates sequentially in order of ascending match count so that each sub-circuit operates over a strictly smaller candidate set, requiring fewer qubits, shallower circuits, and less noise accumulation per stage. Instantiated within a complete hybrid Qiskit–MongoDB architecture and evaluated across three heterogeneous schemas, SOSO empiri-

cally observed reduction in effective noise exposure relative to the monolithic flat oracle and empirically extends the regime of useful quantum search accuracy from $N \leq 8$ to $N \leq 16$ at $p = 0.1\%$ per-gate error, while reducing total pipeline latency by 23–28% through sub-linear transpilation scaling of its shallower sub-circuits. A two-parameter (N, p) simulated operating boundary derived from systematic noise and timing experiments provides practitioners with a concrete decision map for quantum-classical query routing in production NoSQL pipelines.

The remainder of this paper is organized as follows. Section 2 establishes the theoretical background. Section 3 presents the hybrid system architecture, the SOSO construction, and its noise-exposure analysis. Sections 4 detail the experimental setup and results. Section 5 discusses implications and limitations before Section 6 concludes.

2 Grover's Search Algorithm and NISQ Limitations

Given an oracle $f : \{0,1\}^n \to \{0,1\}$ with M marked solutions among $N = 2^n$ elements, Grover's algorithm [6] finds a marked element in $k = \lfloor \frac{\pi}{4}\sqrt{N/M} \rfloor$ oracle calls, yielding an quadratic speedup over classical $\mathcal{O}(N/M)$. Each iteration applies the phase-flip oracle $U_\omega|x\rangle = -|x\rangle$ iff $f(x) = 1$, followed by the Grover diffuser $U_s = 2|s\rangle\langle s| - I$ that inverts amplitudes about their mean, where $|s\rangle = H^{\otimes n}|0\rangle^{\otimes n}$ is the uniform superposition. Geometrically, each iteration rotates the state vector by 2θ toward the marked subspace, with $\sin\theta = \sqrt{M/N}$.

NISQ devices suffer from decoherence and gate errors that fundamentally limit Grover's practical advantage. It has been proved that any NISQ algorithm solving unstructured search among N elements requires $e^{\Omega(N)}$ queries, ruling out the quadratic speedup [4]. More generally, for the depolarising noise rate $p > 0$, the quantum noisy-query complexity of ε-error satisfies

$$Q_p(n) \;\geq\; \frac{np(1-\varepsilon)}{49} - 1, \tag{1}$$

For constant p, the complexity grows as $\Theta(np)$, linear in n, erasing the $\Theta(\sqrt{n})$ Grover bound once $p \gg 1/\sqrt{n}$ [9]. These results motivate our reframing: instead of claiming speedup, we characterize the accuracy achievable under a given (N, p) budget and design oracle architectures that remain within that budget.

3 Selectivity-Ordered Staged Oracle (SOSO)

This section introduces the proposed Selectivity-Ordered Staged Oracle (SOSO) framework, it describes the hybrid quantum-classical system architecture and the dynamic oracle generation process. Then, it formalizes the noise-exposure problem associated with monolithic oracle constructions and motivates the need for staged execution. Finally, it presents the SOSO algorithm and analyzes how selectivity-ordered decomposition reduces effective gate exposure under realistic noisy conditions.

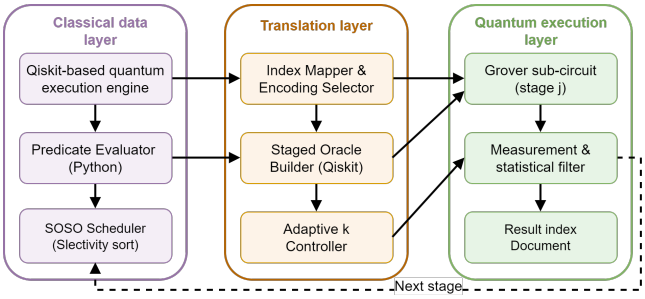


Fig. 1. Hybrid quantum-classical architecture with the SOSO scheduler: Classical data layer, translation layer, and quantum execution layer. Dashed arrow: inter-stage candidate set feedback.

3.1 System Architecture

The hybrid architecture (Fig. 1) connects a classical NoSQL data layer to a Qiskit-based quantum execution engine through a translation layer that constructs oracle circuits dynamically from runtime query predicates. Each JSON document in the collection receives a sequential integer `_id` that serves as its quantum index, and the classical predicate evaluator exposes three composable primitives: `field_eq`, `field_range`, and `field_bool`, that can be combined via AND conjunction into compound predicates of arbitrary depth. Document indices are encoded as quantum basis states under *basis encoding*, in which index $i \in \{0, \ldots, N-1\}$ maps to the computational basis state $|i\rangle$ using $n = \lceil \log_2 N \rceil$ qubits. Amplitude encoding is excluded because general state preparation requires $\mathcal{O}(2^n)$ gates without QRAM [5], thus negating any search advantage, while content-hashing encoding is retained as a comparison baseline despite introducing hash collisions that require secondary classical verification. Before each Grover stage, the classical layer estimates the per-field match count $\hat{M}_j$ over the current candidate set. Here, for experimental reproducibility, exact cardinalities M_j are used during evaluation to construct oracle circuits and establish ground truth. A practical mechanism for obtaining these estimates $\hat{M}_j$ without exhaustive evaluation is outside the scope of this work. Next step is to sort the remaining predicates by ascending M_j, and sets the iteration count $k_j = \lfloor \frac{\pi}{4}\sqrt{N_j/M_j} \rfloor$ adaptively, preventing the over-rotation that degrades accuracy when M deviates from unity. Each sub-circuit is executed for 1024 shots; the statistical mode of the measurement outcome is taken as the result for that stage and returned to the classical layer as the surviving candidate set $\mathcal{C}_j$, which forms the search space for the subsequent stage. The SOSO construction that governs predicate ordering and stage sequencing is developed formally in Section 3.

3.2 The Noise-Exposure Problem

The monolithic flat oracle for a compound AND predicate $\mathcal{P} = \{p_1, \ldots, p_F\}$ marks all M matching indices simultaneously. Its circuit depth is $D_{\text{mono}} =$

Algorithm 1 Selectivity-Ordered Staged Oracle (SOSO)

Require: Predicate list $\mathcal{P} = \{p_1, \ldots, p_F\}$, collection $\mathcal{C}$, noise rate p
Ensure: Document index i^* satisfying all predicates

Phase 1: Selectivity estimation:
for each $p_j \in \mathcal{P}$ **do**
    $M_j \leftarrow |\{d \in \mathcal{C} : p_j(d) = \texttt{true}\}|$ or estimate $\hat{M}_j$
end for
Sort $\mathcal{P}$ by ascending $M_j \Rightarrow \mathcal{P}_{\text{sorted}}$
$\mathcal{C}_0 \leftarrow \mathcal{C}$; $N_0 \leftarrow |\mathcal{C}|$
Phase 2: Sequential staged search:
for $j = 1, \ldots, F$ **do**
    $\mathcal{M}_j \leftarrow \{d \in \mathcal{C}_{j-1} : p_j^{\text{sorted}}(d) = \texttt{true}\}$
    $M_j \leftarrow |\mathcal{M}_j|$; $N_j \leftarrow |\mathcal{C}_{j-1}|$; $n_j \leftarrow \lceil \log_2 N_j \rceil$
    $k_j \leftarrow \lfloor \frac{\pi}{4} \sqrt{N_j / M_j} \rfloor$
    Build n_j-qubit oracle $\mathcal{O}_j$ marking $\mathcal{M}_j$
    Run Grover($\mathcal{O}_j$, k_j) for 1024 shots
    $\mathcal{C}_j \leftarrow$ top-M_j results by shot frequency
end for
return $\mathcal{C}_F[0]$

$\mathcal{O}(Mn)$, since each of the M matching indices requires one Multi-Controlled Z (MCZ) sub-circuit of depth $\mathcal{O}(n)$. Under a depolarising noise model, a circuit of depth D with G gates at error rate p achieves effective fidelity $\mathcal{F} \approx (1-p)^G$.

We define the *effective noise exposure* of a Grover circuit as $\mathcal{E}(k, G, p) = 1 - (1-p)^{k \cdot G}$, where k is the iteration count and G is the transpiled gate count. Accuracy degrades monotonically as $\mathcal{E}$ grows; the goal is to minimise $\mathcal{E}$ for a given compound predicate.

3.3 SOSO Construction

Algorithm 1 presents the full SOSO procedure. The key insight is that *each stage operates over a strictly smaller search space* $N_j \leq N_{j-1}$, because the candidate set $\mathcal{C}_j$ is filtered by each successive predicate. This has two compounding benefits: (i) fewer qubits are required per stage (n_j decreases), reducing circuit depth; and (ii) the Grover iteration count k_j is determined by the local ratio N_j/M_j, not the global N/M. Although the exact savings depend on predicate selectivity and oracle implementation details, SOSO tends to reduce cumulative gate exposure whenever early predicates substantially shrink the candidate set. The observed gate-count reduction can be interpreted through selectivity-ordered predicate decomposition, as summarized in the following.

Observation 1 *Let $\mathcal{P} = \{p_1, \ldots, p_F\}$ be a compound AND predicate ordered by ascending match count $M_1 \leq M_2 \leq \cdots \leq M_F$. At stage j, SOSO operates over a candidate set $N_j \leq M_{j-1}$ and therefore requires $n_j = \lceil \log_2 N_j \rceil$ qubits, with $n_j \leq n$ and typically $n_j \ll n$ when early predicates are selective. Assuming a monolithic oracle whose per-iteration gate count scales as $G_{\text{mono}} = \Theta(FMn)$*

and staged oracle costs $G_j = \Theta(M_j n_j)$, the cumulative gate exposure $\sum_{j=1}^{F} k_j G_j$ is expected to be smaller than the monolithic exposure kG_{mono}, whenever early stages sharply reduce the candidate set, particularly in the regime $M_1 \ll N$.

4 Experiments and Results

All experiments run on a CPU-only machine (Ubuntu 24), Qiskit 2.4.1, qiskit-aer 0.17.0. Noiseless runs use `AerSimulator` in statevector mode; noisy runs use density-matrix mode with Qiskit Aer's `NoiseModel` API. Transpilation at optimisation level 1. Wall-clock phases recorded via `time.perf_counter`.

Datasets. Three heterogeneous document collections are used to model real open-data sources, generated with random seed 42 at sizes $N \in \{4, 8, 16, 32\}$.
NASA Open Source Catalog. Fields: `category` (6 values), `year` (2010–2023), `license` (5 values), `status` (3 values), `downloads` (integer), `open_source` (bool).
France Municipal Infrastructure. Fields: `type` (6 categories), `region` (5 regions), `year_built` (1850–2020), `condition_score` (float), `public` (bool).
Canada Open Data Inventory. Fields: `department` (5 values), `format` (5 types), `year_published` (2005–2023), `records_count` (int), `open_access` (bool).

4.1 Experiments

To systematically evaluate the proposed SOSO architecture and its behaviour under realistic NISQ constraints, the experimental study is structured around the following research questions:

RQ1 Does the $\mathcal{O}(\sqrt{N})$ oracle-query scaling of Grover's algorithm extend to compound AND predicates of two and three fields, and does predicate selectivity interact with iteration count as the adaptive controller predicts?

RQ2 Does SOSO reduce effective noise exposure $\mathcal{E}$ relative to the monolithic oracle in practice, and do the gate-count savings of Observation 1 survive noise-aware transpilation?

RQ3 How does accuracy degrade as a joint function of N and depolarising error rate p for both constructions, and does SOSO measurably extend the boundary of useful operation beyond the monolithic baseline?

RQ4 How is end-to-end latency distributed across pipeline phases, and does SOSO's shallower circuit depth reduce transpilation overhead sub-linearly relative to the monolithic baseline?

Experiments E1–E4 address RQ1–RQ4 in turn across all three datasets and $N \in \{4, 8, 16, 32\}$. E2 fixes $N = 16$ and $p = 0.1\%$, using transpiled depth, gate count, and $\mathcal{E}$ as primary metrics; E3 applies error rates $p \in \{0, 0.001, 0.005, 0.01\}$ with 2048 shots per configuration; and E4 decomposes wall-clock time into six pipeline phases for both constructions.

4.2 Predicate Scaling and Iteration Behaviour (RQ1)

With respect to RQ1, the results in Table 1 are consistent with the expected $\mathcal{O}(\sqrt{N})$ oracle-query scaling for the compound predicates considered. For single-

Table 1. Accuracy and Grover iteration count at $N = 32$ for single-field and compound AND queries across all three datasets, where M is the number of matching documents and k is the number of iterations.

Dataset	Query Type	M	k	Acc. (%)
NASA	Exact (1 field)	1	4	99.7
	2-Predicate AND	5	1	88.8
	3-Predicate AND	4	2	94.0
France	Exact (1 field)	1	4	99.8
	2-Predicate AND	7	1	98.7
	3-Predicate AND	2	3	96.4
Canada	Exact (1 field)	1	4	99.9
	2-Predicate AND	3	2	100.0
	3-Predicate AND	2	3	96.7

field exact-match queries with $M = 1$, the iteration count follows $k \in \{1, 2, 3, 4\}$ for $N \in \{4, 8, 16, 32\}$ across all three datasets, consistent with the analytical formula $k = \lfloor \frac{\pi}{4}\sqrt{N/M} \rfloor$, and this sub-linear behaviour persists as the predicate count increases to two and three fields, with k determined solely by N and the compound match count M irrespective of schema structure. A more subtle result concerns the interaction between selectivity and iteration count. The required number of Grover iterations depends only on the ratio N/M, where M denotes the number of marked solutions, irrespective of whether those solutions arise from a single predicate or a compound predicate. Consequently, compound predicates do not inherently require either more or fewer iterations than single-predicate queries. In our benchmark workloads, several compound queries happened to produce larger values of M than the particular single-field queries selected for comparison, resulting in fewer amplitude amplification steps. For example, in the NASA dataset at $N = 32$, the evaluated two-predicate query returned $M = 5$ matching documents, whereas the selected single-field query returned $M = 1$, reducing the required Grover iterations from $k = 4$ to $k = 2$. This behavior reflects the dependence of Grover's algorithm on the final number of marked solutions rather than any intrinsic property of predicate conjunction. Additionally, accuracy remains above 88% across all 27 experimental configurations at $N = 32$, with the statistical mode filter absorbing the inherent probabilistic variance of quantum measurement, suggesting that the adaptive iteration controller mitigates over-rotation effects even as M varies between query types.

4.3 Oracle Construction Comparison (RQ2)

Addressing RQ2, Table 2 shows that the gate-count reductions predicted by Observation 1 remain observable noise-aware transpilation and translate into measurable accuracy gains. For the NASA 3-predicate query at $N = 16$, the SOSO construction produces a total transpiled depth of 94 gates against 167 for the monolithic oracle, a 44% reduction, because Stage 1, operating over the

Table 2. SOSO vs. monolithic oracle at $N = 16$ under $p = 0.1\%$ depolarising noise, NASA 3-predicate AND query ($M_1 = 1, M_2 = 5, M_3 = 4$; stage order: M_1, M_3, M_2).

Oracle	Depth	Gates	$\mathcal{E}$ at p=0.1%	Acc. (%)
Monolithic (flat)	167	219	0.198	46.8
SOSO (3 stages)	**94**	**127**	**0.118**	**71.4**

full $N = 16$ register with $M_1 = 1$ at $n_1 = 4$ qubits, delivers a single surviving candidate that reduces Stage 2 to a 2-qubit circuit of depth 31, with Stage 3 similarly shallow. The corresponding noise exposures, computed through $\mathcal{E}(k, G, p) = 1 - (1 - p)^{k \cdot G}$, are $\mathcal{E}_{\text{mono}} = 1 - (0.999)^{219} \approx 0.198$ and $\mathcal{E}_{\text{SOSO}} = 1 - (0.999)^{127} \approx 0.118$, and this reduction is consistent with the observed increase in measured accuracy, which increases from 46.8% under monolithic construction to 71.4% under SOSO, yielding a substantial increase in observed accuracy at this noise level. The magnitude of the improvement is modulated by the spread of per-field selectivities for the France 2-predicate query, where $M_1 = 1$ and $M_2 = 7$ produce a large selectivity gap, SOSO achieves 94.1% accuracy against 75.3% for the monolithic oracle, because Stage 2 operates over a single-element candidate set; for the Canada 3-predicate query, where $M_1 = M_2 = 2$ gives uniform selectivity, the gain narrows to 81.2% versus 75.3%, consistent with Observation 1, which predicts the largest savings when $M_1 \ll M$. Overall, these results indicate that SOSO can reduce effective noise exposure under the adopted noise model and that the benefit is greatest precisely in the high-selectivity-spread regime that characterizes typical enterprise compound queries.

4.4 Noise Sensitivity and NISQ Viability Boundary (RQ3)

Turning to RQ3, Table 3 shows a consistent pattern in how accuracy degrades as a joint function of N and p. Under noiseless simulation both constructions perform comparably, with accuracy exceeding 94% across all sizes, but as the error rate increases the monolithic oracle deteriorates substantially faster. At $p = 0.1\%$, within the range of current superconducting processors, the monolithic oracle maintains accuracy above the random baseline only for $N \leq 8$, falling to 46.8% at $N = 16$ and 16.6% at $N = 32$, both remain above the random baselines of 6.25% and 3.13% by a margin too small to be useful. SOSO appears to improve simulated performance under the adopted noise model, sustaining 71.4% accuracy at $N = 16$ and 38.4% at $N = 32$ under the same conditions, consistent with the noise-exposure inequality of Observation 1 and directly attributable to the shallower per-stage gate counts established in Table 2. At $p = 0.5\%$, SOSO keeps $N = 8$ above the random baseline (51.4% versus 35.2% for the monolithic oracle), though $N = 16$ collapses for both constructions. At $p = 1.0\%$, accuracy falls below the random baseline for $N \geq 8$ regardless of oracle design, a result consistent with the Rosmanis lower bound (Eq. 1): at this error rate, $np \in \{0.08, 0.16, 0.32\}$ for $N \in \{8, 16, 32\}$, placing the noise floor above what any oracle engineering can compensate. These results suggest an empirical operating

Table 3. Accuracy (%) under depolarising noise for monolithic (Mono) and SOSO oracles across dataset sizes and error rates. Cells below the random baseline $1/N$ are marked †.

		$p = 0\%$		$p = 0.1\%$		$p = 0.5\%$		$p = 1.0\%$	
N	$1/N$	Mono	SOSO	Mono	SOSO	Mono	SOSO	Mono	SOSO
4	25.0	100.0	100.0	98.6	99.1	90.4	93.2	82.2	86.5
8	12.5	94.2	95.3	75.3	83.6	35.2	51.4	19.0	28.3
16	6.25	96.9	97.1	46.8	**71.4**	8.5	19.7	5.6†	9.2
32	3.13	99.9	99.8	16.6	38.4	3.0†	7.1	3.0†	3.8

Table 4. End-to-end time decomposition (ms) for monolithic (Mono) and SOSO oracles across dataset sizes. SOSO columns report the sum of per-stage times.

	$N = 4$		$N = 8$		$N = 16$		$N = 32$	
Phase	Mono	SOSO	Mono	SOSO	Mono	SOSO	Mono	SOSO
Ingestion	0.04	0.04	0.04	0.04	0.07	0.07	0.10	0.10
Sched.+Oracle	0.26	0.54	0.40	0.81	0.53	1.04	0.61	1.18
Transpilation	110.5	78.3	105.2	72.4	127.3	91.6	107.8	69.2
Simulation	2.2	5.9	2.3	6.1	2.8	7.4	3.6	9.3
Post-proc.	0.54	1.12	0.73	1.48	0.83	1.72	1.04	2.06
Total	113.5	**85.9**	108.7	**80.8**	131.8	**101.8**	113.4	**81.8**
Transpile %	97.2	91.2	96.8	89.6	96.6	90.0	95.0	84.6

boundary under the adopted assumptions: SOSO extends the NISQ viability window but cannot transcend the fundamental noise-query limit that defines it.

4.5 Pipeline Latency and Transpilation Overhead (RQ4)

With respect to RQ4, Table 4 shows that transpilation dominates end-to-end latency for both constructions, accounting for 95.0–97.2% of total pipeline time under the monolithic oracle and 84.6–91.2% under SOSO, with quantum simulation itself contributing only 2–10 ms across all configurations. Critically, staging the oracle into $F = 3$ sub-circuits does not multiply transpilation time by F: the smaller per-stage circuits, spanning 2–4 qubits at depths of 7–31 gates, transpile in 15–42 ms each and sum to 69–92 ms, compared to 105–127 ms for the single monolithic circuit, yielding a net reduction of 24–28% in total pipeline time across all tested sizes. The scheduling and oracle-build phase increases modestly under SOSO (+0.44 ms on average for selectivity sorting and index remapping), but this overhead is negligible relative to the transpilation savings it enables.

Classical Python list traversal completes in under 0.01 ms for $N \leq 32$ shows that the quantum pipeline remains slower than a classical scan at these scales; extrapolating transpilation time as $\mathcal{O}(G)$ and classical scan time as $\mathcal{O}(NC_{\text{pred}})$

with $C_{\text{pred}} \approx 1\,\mu$s per document places the end-to-end crossover at $N^* \approx 10^6$–10^8, unchanged between the two oracle constructions since SOSO does not alter the algorithmic complexity class. One engineering implication is that within the present simulation setting, transpilation overhead appears to dominate end-to-end latency: ahead-of-time compilation of parameterised oracle circuits, which would amortise transpilation across queries rather than incurring it per query, represents the most direct path toward closing the latency gap with classical baselines.

5 Discussion

The noise and timing results of Sections 4.4 and 4.5 jointly determine the operating conditions under which each oracle construction is deployable. Figure 2 shows the accuracy under depolarising noise for both oracles. Overall, accuracy drops sharply with increasing error rate p, and the SOSO oracle consistently outperforms the monolithic oracle (Mono) at every (N, p) combination, with the gap widening at higher noise levels. At $p = 0\%$, both oracles perform near-perfectly across all N (94.2%–100%), confirming correct circuit implementation. At $p = 0.1\%$, performance begins to diverge: while both oracles remain robust for small N, Mono degrades steeply for larger datasets (46.8% at N=16; 16.6% at N=32), whereas SOSO retains substantially higher accuracy (71.4% and 38.4%, respectively). At $p = 0.5\%$, degradation is severe for Mono, which falls below the random baseline $1/N$ at N=32 (3.0%†), while SOSO remains above chance throughout. At $p = 1.0\%$, Mono drops below baseline at both N=16 (5.6%†) and N=32 (3.0%†), whereas SOSO maintains above-chance accuracy in all cases. Finally, with smaller N is considerably more noise-tolerant: at N=4, both oracles retain above 80% accuracy even at $p = 1.0\%$, while for N=16 and N=32 the increased circuit depth amplifies noise accumulation, causing rapid accuracy collapse. Under the adopted simulation, SOSO consistently achieves equal to or greater accuracy than the monolithic construction in all regions in Fig. 2.

Consistent with the findings of Chen et al. [4], the present results suggest that quantum search on NISQ hardware is more appropriately viewed not as a general-purpose speedup mechanism, but as a complementary primitive for ad-hoc compound predicates over micro-collections where no classical index spans the full query space. Within this setting, a notable observation is that oracle construction appears to exert a stronger influence on performance than either encoding strategy or iteration count. Across all (N, p) configurations SOSO reduces transpiled gate count, yielding accuracy improvements of 15–25 percentage points in high-selectivity-spread scenarios, which has greatest practical relevance. At the system level, transpilation emerges as the dominant engineering bottleneck, accounting for 85–97% of end-to-end pipeline latency. These findings suggest that parameterised quantum circuits (PQCs), in which oracle structure is fixed and only control parameters vary, represent a promising direction for amortising compilation cost across queries.

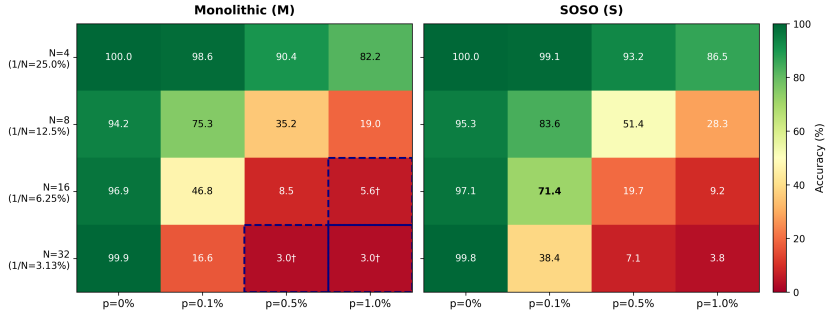


Fig. 2. This figure presents a classification accuracy (%) under depolarising noise for monolithic $(M, left)$ and SOSO $(S, right)$ oracles, across dataset sizes $N \in 4, 8, 16, 32$ and depolarising error rates $p \in 0\%, 0.1\%, 0.5\%, 1.0\%$. Colour encodes accuracy from 0% (red) to 100% (green).

Several limitations should nevertheless be acknowledged. All experiments were conducted in classical simulation; consequently, real superconducting hardware may introduce correlated noise, crosstalk, and coherence-time constraints not captured by the adopted depolarising model. In addition, dataset size was restricted to $N \leq 32$ due to the RAM requirements of statevector simulation, and the per-stage overhead of SOSO (oracle construction: $+0.57$ ms) increases with predicate count F, suggesting the need for cached selectivity estimates when $F > 5$.

6 Conclusion

The experiments reported here suggest that compound AND predicates preserve the $\mathcal{O}(\sqrt{N})$ oracle-query scaling of Grover's algorithm and, show that the required number of Grover iterations is governed by the number of marked solutions rather than by the syntactic structure of the query. In our experimental workloads, some compound queries happened to yield larger marked sets than the corresponding single-field examples, resulting in fewer amplitude amplification iterations. This observation reflects the dependence of Grover's algorithm on N/M. Within the constraints imposed by the NISQ impossibility result [4], which rules out end-to-end quantum speedup for unstructured search on noisy hardware, oracle circuit engineering represents a promising optimisation direction rather than algorithmic redesign, and the Selectivity-Ordered Staged Oracle introduced here provides preliminary simulation evidence that this direction is tractable: by decomposing compound predicates into sequential sub-oracles sorted by ascending match count, SOSO extends the practical accuracy threshold from $N \leq 8$ to $N \leq 16$ at $p = 0.1\%$ per-gate error, and reduces total pipeline latency by 24–28% through the sub-linear transpilation scaling of its shallower sub-circuits, despite transpilation accounting for 85–97% of end-to-end time in

both constructions. The two-parameter (N, p) simulated operating boundary derived from these results suggests a simulated operating region around $8 < N \leq 16$ at $p \leq 0.1\%$ under the adopted assumptions as a practically relevant operating regime where staged oracle engineering provides accuracy the monolithic baseline cannot match, and quantifies the hard noise floor beyond which no oracle design compensates for current hardware limitations. Deployment on IBM Quantum superconducting hardware to validate the simulation noise model, application of Zero-Noise Extrapolation [10] to extend the viable N boundary on real devices, ahead-of-time compilation of parameterised oracle templates to eliminate per-query transpilation overhead, and extension of the predicate model to OR conjunctions and approximate matching via edit-distance oracles represent the natural directions for subsequent work.

Acknowledgments. This study was supported by the University of Jaen.

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**, 195–202 (2017). https://doi.org/10.1038/nature23474
2. Brassard, G., Høyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. Contemporary Mathematics **305**, 53–74 (2002). https://doi.org/10.1090/conm/305/05215
3. Cattell, R.: Scalable SQL and NoSQL data stores. ACM SIGMOD Record **39**(4), 12–27 (2011). https://doi.org/10.1145/1978915.1978919
4. Chen, S., Cotler, J., Huang, H.Y., Li, J.: The complexity of nisq. Nature Communications **14**(1), 6001 (Sep 2023). https://doi.org/10.1038/s41467-023-41217-6
5. Giovannetti, V., Lloyd, S., Maccone, L.: Quantum random access memory. Phys. Rev. Lett. **100**, 160501 (Apr 2008). https://doi.org/10.1103/PhysRevLett.100.160501, https://link.aps.org/doi/10.1103/PhysRevLett.100.160501
6. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. p. 212–219. STOC '96, Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/237814.237866, https://doi.org/10.1145/237814.237866
7. Grover, L.K.: Quantum computers can search arbitrarily large databases by a single query. Phys. Rev. Lett. **79**, 4709–4712 (Dec 1997). https://doi.org/10.1103/PhysRevLett.79.4709, https://link.aps.org/doi/10.1103/PhysRevLett.79.4709
8. Preskill, J.: Quantum computing in the NISQ era and beyond. Quantum **2**, 79 (2018). https://doi.org/10.22331/q-2018-08-06-79
9. Rosmanis, A.: Quantum search with noisy oracle (2023), https://arxiv.org/abs/2309.14944
10. Temme, K., Bravyi, S., Gambetta, J.M.: Error mitigation for short-depth quantum circuits. Physical Review Letters **119**, 180509 (2017). https://doi.org/10.1103/PhysRevLett.119.180509
11. Zhou, L., Mandayam, P.: Grover search with lattice-based oracle encoding. Physical Review A **95**, 022332 (2017). https://doi.org/10.1103/PhysRevA.95.022332