

Diffusion-Operator Qubit Bounds and an Error-Burst Qubit Ceiling for Oracle-Based Quantum Search: A Conditional Resource Estimate

Ying Liu

Department of Engineering Technology, Savannah State University, Savannah, GA, USA

Email: liuy@savannahstate.edu

How to cite this paper: Liu, Y. (2026) Diffusion-Operator Qubit Bounds and an Error-Burst Qubit Ceiling for Oracle-Based Quantum Search: A Conditional Resource Estimate. *Journal of Quantum Information Science*, 16, 494-513.

<https://doi.org/10.4236/jqis.2026.164018>

Received: September 4, 2026

Accepted: October 20, 2026

Published: October 23, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Grover's algorithm searches an unstructured database of $N = 2^n$ items in $\mathcal{O}(\sqrt{N})$ sequential oracle-and-diffusion rounds, a bound already proven optimal. This paper asks a different question: given realistic fault-tolerant hardware constraints, how large an unstructured search can actually be carried to completion? Using a spacetime-volume resource model—accounting for circuit depth, active and idle qubit exposure, and surface-code error suppression—we first derive an oracle-independent upper bound: the diffusion operator alone, together with the round count fixed by the $\sqrt{N}$ lower bound, bounds the number of index qubits that a completed search can use to roughly 22 to 53, depending on code distance, a bound that no oracle implementation can loosen, only tighten. We then identify a second, distinct constraint: error bursts, rare correlated fault events with a specific physical origin in cosmic-ray and ambient-radiation exposure, measured directly on current superconducting hardware, are not suppressed by increasing code distance the way ordinary logical error is. This produces an asymptotic ceiling of approximately 50 index qubits that persists regardless of code distance, since the limiting factor becomes a fault mechanism that ordinary error correction does not address. This finding is scoped deliberately: it is a conditional resource estimate tied to one class of hardware's currently measured, largely unmitigated behavior, not a fundamental limit on quantum computing or a demonstration that this failure mode is physically irreducible—active mitigation research and recent theoretical work indicate the ceiling may relax as hardware and decoding strategies mature. Finally, we show the oracle-independent bound degrades gracefully rather than catastrophically as realistic oracle cost is introduced, via a single-parameter sensitivity analysis.

Keywords

Quantum Computing, Grover's Algorithm, Oracle, Surface Code, Quantum Error Correction, Error Bursts, Spacetime Volume, Resource Estimation, Fault-Tolerant Quantum Computing

1. Introduction

Grover's algorithm searches an unstructured database of N items using $\mathcal{O}(\sqrt{N})$ sequential oracle-and-diffusion rounds, a quadratic speedup over classical search that is provably optimal [1]-[3]. That optimality is not in question here. What this paper asks instead is a resource question: given realistic constraints on fault-tolerant quantum hardware, how large a search problem can such a deep, sequential algorithm actually be run to completion on?

This question does not reduce to counting logical qubits. A quantum computer with many available logical qubits is not thereby able to run a deep computation to completion—circuit depth, not qubit count, is what a fault-tolerant error budget actually constrains, and the two are only coupled for algorithms whose required depth scales with the size of the problem instance encoded in the qubit register. Unstructured search is the clearest such case: doubling the number of index qubits roughly squares the number of sequential rounds required. This coupling is what makes Grover's algorithm a useful lens for asking how deep a fault-tolerant computation can practically go, and it is the reason a bound on achievable depth translates, for this algorithm specifically, into a bound on the number of index qubits a completed search can use.

Two separate resource questions arise once this framing is in place, and this paper treats them as genuinely separate. The first is how expensive the search algorithm's own structure is, independent of any particular oracle: Sections 2 and 3 show that the diffusion operator alone, together with the fixed number of rounds required by the $\sqrt{N}$ lower bound, sets a resource bound that no oracle implementation can loosen, only tighten—an oracle-independent upper bound that holds regardless of how the oracle is built. Section 7 then asks how much a real oracle's cost narrows this bound, without committing to any single oracle's specific implementation.

The second question is what “achievable depth” means once realistic error correction is accounted for, and this is where this paper's central finding emerges. Standard fault-tolerant resource estimates assume that logical faults occur independently at each qubit, each cycle—an assumption that error-corrected quantum memories are specifically designed to make safe. But this assumption has a measured exception. Superconducting quantum processors are subject to error bursts: rare, transient events, physically traced to cosmic-ray muons and ambient radioactivity striking the chip substrate, that elevate the fault rate across many qubits simultaneously [4] [5]. Sections 4 through 6 show that this failure mode, measured

directly on current hardware, is not suppressed by increasing code distance the way ordinary logical error is, and that this asymmetry produces an asymptotic ceiling on the number of index qubits a completed search can use—a ceiling that persists no matter how good the underlying error-correcting code is chosen to be.

It is important to state precisely what kind of claim it is, since the distinction matters throughout. It is not a fundamental limit on quantum computing, nor a bound on how many logical qubits fault-tolerant hardware can support—current and near-term devices already exceed the qubit counts this paper discusses, running algorithms whose depth does not scale the way Grover’s does. It is also not a settled, irreducible physical limit: active mitigation research targets the underlying error-burst rate directly, and separate recent theoretical work indicates that appropriately designed decoding and mitigation strategies may restore favorable code-distance scaling even in the presence of these events [6]. What this paper reports is a conditional resource estimate: given the error-burst rate currently measured on one class of superconducting hardware, and without additional burst-specific mitigation, this is the largest unstructured search Grover’s algorithm could complete. Section 4 discusses this scope explicitly, including which parts of the finding are specific to the cited hardware and which reflect a more general concern for fault-tolerant computation.

The remainder of the paper proceeds as follows. Section 2 develops the resource-accounting model—spacetime volume rather than gate count alone, and the diffusion operator construction this paper adopts. Section 3 establishes the oracle-independent upper bound and proves that no real oracle can loosen it. Sections 4 through 6 introduce error bursts, incorporate them into the resource model, and derive the resulting asymptotic ceiling. Section 7 discusses how a real oracle’s cost tightens the bound established in Section 3. Section 8 explains the reasons behind the bounds and ceiling. Section 9 concludes.

2. Model and Upper Bound Equation

To compute a resource bound on the number of logical qubits usable by a given algorithm, several assumptions about the algorithm’s structure are required. An algorithm generally depends on both an oracle operator and a diffusion operator. The diffusion operator alone, however, imposes a bound on the number of logical qubits usable in a completed search, independent of the oracle’s implementation. We derive this bound first (Section 3) and introduce the oracle’s effect afterward (Section 7). Throughout, we depend on a number of explicit assumptions, listed in full at the end of this section (§2.8); several of these may shift as hardware continues to improve.

2.1. Logical Error Budget

Under active quantum error correction, a logical qubit’s usable lifetime [7] [8] is not set by the raw physical decoherence time T_2 —continuous syndrome extraction and correction is designed precisely to outlast T_2 . The relevant constraint in-

stead is how much computation (gate count, depth, or volume) can be performed before the accumulated probability of an uncorrected logical fault exceeds an acceptable threshold for the whole computation, which will be discussed in the next subsection.

2.2. Resource Definitions

We distinguish three resource quantities:

- **Circuit depth D** : the length of the critical path—the number of sequential time layers, assuming maximal parallelism within each layer.
- **Total logical gate count G** : the sum of all gate operations across all layers. A layer with w parallel gates contributes 1 to D but w to G .
- **Total logical spacetime volume V** : the number of (logical qubit) $\times$ (QEC cycle) pairs exposed to error over the computation,

$$V \approx n_{\text{active}} \times D \quad (1)$$

where D is the circuit depth, n_{active} includes every logical qubit present—index qubits and ancilla—since under active surface-code QEC, idle qubits still require syndrome extraction every cycle and accrue logical error whether or not a gate acts on them that cycle.

Failure probability accumulates with spacetime volume, not logical gate count alone—a qubit sitting idle for many cycles is exposed to error just as a qubit being actively gated is. All resource accounting below is therefore done in terms of V .

Fault-tolerant quantum computation works by continuously monitoring every logical qubit, every cycle, whether or not that qubit is being gated at that moment. The whole computation can be pictured as a two-dimensional grid: one axis is *which logical qubit* (index qubits, plus any ancilla present), the other is *which QEC cycle*. Every cell in this grid—one qubit, for one cycle—carries some small, independent probability of an uncorrected logical fault. This grid is exactly what “spacetime volume” means: V is the number of cells in it.

2.3. Diffusion Operator Construction

The diffusion operator

$$D = 2|\psi\rangle\langle\psi| - I \quad (2)$$

can be realized with either a minimal-ancilla linear construction (depth $O(n)$, $O(1)$ ancilla) or an ancilla-tree construction (depth $O(\log n)$, $O(n)$ ancilla)—see Appendix A for both constructions and the derivation below. Comparing spacetime volume across Grover’s $K(n)$ rounds shows the ancilla-tree construction has lower total volume for $n \gtrsim 16$, which covers the entire problem-size range this paper addresses. We adopt it throughout:

$$D_{\text{diff}}(n) \approx 2 \cdot \log_2 n \quad (3)$$

$$n_{\text{active,diff}}(n) \approx 2n \quad (4)$$

conditional on sufficient physical qubits being available for the $O(n)$ diffusion ancilla.

2.4. Memory and Active Logical Error Assumption

Within the diffusion operator's window, each logical qubit at each cycle is either idle or being gated, and the two cases carry different logical error rates. The memory rate, P_{mem} , applies to idle cycles and follows the standard surface-code suppression formula (§2.5), sourced from quantum-memory experiments [5]. Any cycle in which a qubit participates in a gate incurs the higher rate, P_{gate} :

$$P_{\text{gate}} = \kappa \cdot P_{\text{mem}} \quad (5)$$

with $\kappa \approx 1.25 - 1.4$ drawn from circuit-level lattice-surgery studies of logical CNOT and merge/split operations relative to memory baselines; we adopt $\kappa = 1.3$ as a representative point estimate.

Rather than tracking exactly which cycles are idle and which are gated within the diffusion operator's window, we consider two simple bracketing choices: pricing every counted qubit-cycle at the lower rate, P_{mem} (equivalent to $\kappa = 1$), or at the higher rate, P_{gate} . Because the ancilla-tree construction (§2.3) is densely gated for nearly its entire $O(\log n)$ depth, we adopt the latter, more conservative choice throughout:

$$P_{\text{eff}} \equiv P_{\text{gate}} \quad (6)$$

for the effective per-cycle rate (effective fault probability per logical-qubit/QEC-cycle exposure) used in all spacetime-volume calculations from here on. Because κ is close to 1, these two bracketing choices are close together: solving for the maximum index-qubit count under $P_{\text{eff}} = P_{\text{mem}}$ versus $P_{\text{eff}} = P_{\text{gate}}$ shifts the result by less than one qubit across the entire range this paper considers. The simpler, uniform, and conservative choice therefore costs little.

2.5. Surface-Code Logical Error Rate

P_{mem} follows the standard surface-code suppression formula [7] [8]:

$$P_{\text{mem}} = A \cdot \left(\frac{p}{p_{\text{th}}} \right)^{\frac{d+1}{2}} \quad (7)$$

where

- p is the physical two-qubit gate error rate [7] [8],
- $p_{\text{th}} \approx 0.01$ is the code threshold [9]-[11],
- d is the code distance, and
- $A \approx 0.1 - 1$ is an empirical prefactor.

Physical qubits per logical qubit scale as $b \approx 2d^2$ for a standard surface code—some 2025-2026 architectures already beat this ratio substantially (e.g., ~2:1 physical-to-logical overhead reported for some trapped-ion and neutral-atom codes) [5] [12], so $2d^2$ is a conservative, not universal, estimate.

Google's Willow processor has demonstrated a below-threshold, distance-7

surface code quantum memory with per-cycle logical error rate 0.143% [5]; IBM heavy-hex architectures have separately demonstrated subthreshold scaling to anisotropic distance- $(5,3)/(3,5)$ codes [12], on a native connectivity that does not natively match the surface code lattice. Both are broadly consistent with $p \approx 10^{-3}$ as a representative near-term rate. The code distances used in this paper's sensitivity analysis ($d = 11 - 21$) exceed what has been experimentally demonstrated to date ($d \leq 7$); the resulting estimates should be read as near-term projections under continued scaling, not results already achieved.

2.6. Grover's Resource Requirement

Searching $N = 2^n$ items requires

$$K(n) = (\pi/4) \cdot 2^{n/2} \quad (8)$$

oracle-and-diffusion rounds [2]. In this section, oracle depth and oracle ancilla count are both set to zero—a “no-oracle” idealization, addressed explicitly in Section 7. Per-round depth and active-qubit count are then given entirely by the diffusion operator (Equation (3) and Equation (4)), and total spacetime volume across all $K(n)$ rounds is:

$$V(n) = K(n) \cdot n_{\text{active,diff}}(n) \cdot D_{\text{diff}}(n) \quad (9)$$

From Equation (3) and Equation (4),

$$V(n) = \pi \cdot n \cdot \log_2(n) \cdot 2^{n/2} \quad (10)$$

2.7. Upper Bound Equation for the Maximum Index-Qubit Count

For a fault-tolerant quantum computation, the appropriate exponent (Circuit depth D , Total logical gate count G , Total logical spacetime volume V) in the fault-free probability model depends on the unit associated with the effective fault probability P_{eff} . When P_{eff} denotes the probability of a logical fault per logical-qubit exposure during one QEC cycle, the relevant number of independent fault opportunities is the total logical spacetime volume V , defined as the number of (logical-qubit, QEC-cycle) pairs over the computation.

The computation succeeds only if every cell in this grid comes through clean. With V independent cells, each with fault probability P_{eff} (the effective fault probability per logical-qubit/QEC-cycle exposure), the probability that all of them are fault-free is

$$1 - \varepsilon = (1 - P_{\text{eff}})^V \quad (11)$$

So the probability of at least one fault anywhere—a computational failure—is

$$\varepsilon = 1 - (1 - P_{\text{eff}})^{V(n)} \quad (12)$$

For small P_{eff} (P_{eff} ranges from $\sim 10^{-7}$ to $\sim 10^{-12}$ across our code distances), this is closely approximated by the first term of its expansion:

$$\varepsilon \approx V(n) \cdot P_{\text{eff}} \quad (13)$$

A simpler alternative would set a budget on total gate count alone, ignoring qubits that are present but not being directly gated at a given cycle. This implicitly treats an idle qubit as carrying no risk—which is not physically accurate (§2.4). Under active surface-code QEC, an idle qubit is still tracked by continuous syndrome extraction every cycle, and still carries its own, smaller only by a factor of κ , chance of an uncorrected fault each cycle; that continuous monitoring is precisely why quantum error correction works at all.

In Grover’s algorithm, this matters a great deal: the n index qubits must remain coherent and error-corrected across the full $K(n)$ -round computation, but within any single round—and even within a single diffusion operator’s own $O(\log n)$ -depth construction, where most qubits sit waiting at any given layer while only a subset are combined—they are directly gated for only a small fraction of the elapsed cycles. $K(n)$ itself is enormous across the range this paper considers, see Equation (8). Counting gate operations alone assigns zero risk to all the remaining idle cycles; counting spacetime volume assigns each of them their proper share. This is an important contributing factor to the low value of the bound proposed in this paper.

Setting the total spacetime volume against the failure budget therefore gives the Upper Bound Equation:

$$\varepsilon \approx \pi \cdot n \cdot \log_2(n) \cdot 2^{n/2} \cdot \kappa \cdot A \cdot \left(\frac{p}{p_{\text{th}}} \right)^{\frac{d+1}{2}} \quad (14)$$

Equation (14) is transcendental in n for the upper bound of logical qubits. Section 3 solves it numerically for representative code distances.

2.8. Assumptions

The bound derived from Equation (14) depends on the following explicit assumptions, several of which may shift as hardware and error-correction methods improve:

- 1) A single marked item, black-box oracle access, one run without repetition, target success probability 1/2.
- 2) Grover round count $K(n)$ per Equation (8).
- 3) Diffusion operator realized via the ancilla-tree construction (Appendix A), valid for $n \gtrsim 16$.
- 4) Oracle depth and ancilla set to zero in this section (relaxed in Section 7).
- 5) Surface-code memory error model, Equation (7), with $p \approx 10^{-3}$, $p_{\text{th}} = 0.01$, $A = 0.1$, drawn from near-term hardware demonstrations at $d \leq 7$; code distances $d = 11 - 21$ used in Section 3 exceed what has been demonstrated.
- 6) Two-rate active/idle error model, $\kappa \approx 1.3$ (§2.4), applied uniformly and conservatively.
- 7) Overall failure budget $\varepsilon = 0.1$, a chosen target rather than a derived quantity.
- 8) Per-cycle faults assumed independent, except where explicitly revisited in Section 4.

9) Physical-qubit overhead per logical qubit modeled as $b \approx 2d^2$, a conservative estimate relative to some current architectures.

3. Oracle-Independent Upper Bounds

Section 2 built the entire resource model under one explicit idealization: that the oracle contributes zero additional depth and requires no additional ancilla qubits. That's not a realistic assumption—every real oracle does some work. This is what makes this bound useful rather than merely illustrative, it applies to any oracle.

The bound on logical qubits in Section 2 can be solved once, without ever having to pin down what a real oracle costs, and the resulting number is not an approximation of the true bound—it is provably at least as large as the true bound for any real algorithm. Any real oracle adds depth and ancilla, which can only increase the spacetime volume, and the number of rounds $K(n)$ is fixed by the $\sqrt{N}$ lower bound [2], [3]; the bound computed here is therefore at least as large as the true bound for any oracle. Section 7 discusses what happens once a real oracle's cost is added back in; everything in this section describes the best case that no real implementation can exceed.

Using the numerical values of Section 2.8 and solving Upper Bound Equation (Equation (14)) numerically for representative code distances gives **Table 1**.

Table 1. Oracle-independent upper bound for logical qubit at each code distance.

Code distance, d	$n_{\max}$ (oracle-free)
11	22.5
15	34.2
21	52.6

These are the largest problem sizes the diffusion operator alone, at each code distance, can carry to completion within the stated failure budget—and, by the argument above, no real oracle can push any of these numbers higher.

A remark on the κ simplification: Section 2.4 priced every qubit-cycle in the diffusion operator's window at the higher of two possible rates, $P_{\text{gate}} = \kappa \cdot P_{\text{mem}}$ with $\kappa \approx 1.3$ (Equation (5)), and claimed this shifts the result by less than one qubit relative to using the lower rate, P_{mem} , throughout. Solving Equation (14) both ways confirms this: at $d = 11, 15$, and 21 , the two choices differ by 0.65, 0.68, and 0.71 qubits respectively—under one qubit at every code distance considered, and the gap grows only slowly with d . The simplification made in Section 2.4 costs very little in exchange for not having to track which specific cycles are idle and which are gated.

4. Error Burst

The correlated fault this paper's asymptotic ceiling depends on has a specific, well-characterized physical origin—not leakage, not calibration drift, and not the static

crosstalk familiar from superconducting two-qubit gates. When a cosmic-ray muon or gamma radiation from ambient radioactivity strikes the chip substrate, it deposits substantial energy—100 keV to 1 MeV, far above the $\sim 350 \mu\text{eV}$ needed to break a single Cooper pair [4]. This energy radiates outward as high-energy phonons that propagate through the substrate and break Cooper pairs across a wide area of the chip, producing a transient avalanche of quasiparticles. These quasiparticles tunnel across the Josephson junctions at the heart of each affected qubit, driving simultaneous energy decay across many qubits at once—precisely the correlated, code-distance-resistant failure mode this paper’s model in Section 5 depends on.

It is worth being precise about what this paper’s “ceiling” is and is not. It is not a directly observed computational limit: no computation of the length this paper considers has been run and shown to saturate at this point. Nor is it derived from first-principles physics alone. Physics explains the mechanism—that cosmic-ray muons and ambient gamma radiation will occasionally strike the chip substrate and generate a burst of quasiparticles—but the specific rate at which this happens is an empirically measured quantity, not one physics alone predicts a value for. The measured rate used here, roughly one event per hour, comes from dedicated, much shorter burst-detection experiments [4] [5], not from running a search algorithm to its predicted limit. What this paper reports is a projection: a real, physically measured per-event rate carried through the resource-accounting model of Sections 2 - 3, via the same reasoning used throughout. It is a ceiling produced by a specific mathematical model, resting on a genuine physical mechanism together with an empirically measured input.

1) Efforts that improve the rate itself

Mitigation research targeting the burst rate directly is active and has already demonstrated partial success: physical die separation—fabricating the qubit array across two adjacent, galvanically isolated substrates—substantially reduces, though does not eliminate, correlated events spanning both dies. Phonon downconversion using islands of lower-gap superconducting material [13], quasiparticle traps, and Josephson-junction gap engineering have each shown partial suppression of burst severity or frequency. Each of these approaches works by lowering how often, or how severely, a burst event occurs in the first place.

2) The rate as currently modeled is independent of code distance

The burst rate itself does not depend on the code distance d chosen for error correction: unlike ordinary logical error, which shrinks exponentially with d , nothing about increasing d changes how often a cosmic-ray event strikes the chip. But this does not mean the code’s resilience to that fixed rate is similarly fixed. A dedicated recent study of exactly this question [6] shows that given appropriate mitigation limiting burst duration and standard decoding adapted to handle it, surface-code memory time can grow arbitrarily long as code size increases even in the presence of error bursts, in the regimes they study—and at the architectural level, distributed error-correction schemes have been proposed specifically to tol-

erate chip-wide events by spreading a logical qubit's physical qubits across separated dies [14].

The rate itself is a hardware-specific input, not a universal constant. Equation (16) applies this single-chip rate to the modeled machine regardless of its size; a machine spanning many chips would be exposed to a higher overall rate, which lowers the ceiling—it reflects the measured cosmic-ray and radioactivity flux at one device's location and packaging, scaled by chip area; a different device, shielding arrangement, or underground facility would measure a materially different rate.

This leaves a two-part answer to whether this paper's result is general or hardware-specific, and the two parts should not be conflated. The qualitative phenomenon—a rare, transient, spatially correlated spike in error rate defeating the independent-fault assumption standard quantum error correction relies on—is not unique to superconducting hardware or to cosmic rays. The same 2026 study frames this as a general class and notes that trapped-ion and neutral-atom platforms face an analogous risk from large, correlated deviations in shared global control fields—an entirely different physical mechanism, but a structurally similar failure mode. In that sense, the concern this paper raises is plausibly general to fault-tolerant quantum computing. The specific numbers are not: the rate in Equation (16) below, and the resulting $n \approx 50$ ceiling in Section 6, are tied to one measured device's currently reported, largely unmitigated behavior, and should be read as an illustration of what today's hardware implies under this paper's model, not a claim about fault-tolerant computing's ultimate limits.

5. The Upper Bound Equation with Error Burst

Acharya *et al.* [5] report the error burst rate as roughly once per hour, on the same device whose 1.1-microsecond QEC cycle time is used throughout this paper. Converting between the two: one hour contains

$$3600 \text{ s} / (1.1 \times 10^{-6} \text{ s}) \approx 3.27 \times 10^9 \text{ cycles}$$

which we round to 3×10^9 for Equation (16) below. This rounding is deliberately conservative: using the more precise 3.27×10^9 figure would very slightly raise the asymptotic ceiling in Section 6, from $n \approx 50.0$ to $n \approx 50.3$ —an 8% adjustment to the rate that moves the headline result by under a third of a qubit, well within the precision this paper claims anywhere else.

We can fold this into the same reasoning used to derive Equations (11)-(13). The total number of cycles the diffusion-only computation requires is:

$$D_{\text{total}}(n) = K(n) \cdot D_{\text{diff}}(n) \quad (15)$$

Treating error bursts as occurring at a small, roughly constant per-cycle rate,

$$p_{\text{burst}} \approx 1 / (3 \times 10^9) \quad (16)$$

the chance of encountering at least one during a computation of $D_{\text{total}}(n)$ cycles is, by the same reasoning that produced Equation (13):

$$\varepsilon_{\text{burst}}(n) \approx D_{\text{total}}(n) \cdot p_{\text{burst}} \quad (17)$$

Inserting Equation (15) and Equation (16),

$$\varepsilon_{\text{burst}}(n) = K(n) \cdot D_{\text{diff}}(n) / (3 \times 10^9) \quad (18)$$

Inserting Equation (3) and Equation (8),

$$\varepsilon_{\text{burst}}(n) = (\pi/2) \cdot \log_2(n) \cdot 2^{n/2} / (3 \times 10^9) \quad (19)$$

Because this failure mode arises from a different physical mechanism than the ordinary per-cycle error already modeled in Equation (14), the two risks can be combined the same way ordinary faults were combined across many qubit-cycles: by adding their probabilities.

$$\varepsilon_{\text{total}}(n) \approx \varepsilon_{\text{gate}}(n) + \varepsilon_{\text{burst}}(n) \quad (20)$$

where $\varepsilon_{\text{gate}}(n)$ is exactly the quantity solved via Equation (14) earlier. Inserting Equation (14) and Equation (19) into Equation (20) gives the Upper Bound Equation with Error Burst:

$$\varepsilon_{\text{total}}(n) \approx \pi \cdot \log_2(n) \cdot 2^{n/2} \cdot \left[n \cdot \kappa \cdot A \cdot \left(\frac{p}{p_{\text{th}}} \right)^{\frac{d+1}{2}} + \frac{1}{6 \times 10^9} \right] \quad (21)$$

The second term is genuinely illuminating. Both terms share the common factor $\pi \cdot \log_2(n) \cdot 2^{n/2}$, and factoring it out separates the two failure mechanisms cleanly inside the brackets: the first term, $n \cdot \kappa \cdot A \cdot (p/p_{\text{th}})^{(d+1)/2}$, is the gate-error contribution—it shrinks toward zero as d grows, since larger code distance suppresses it exponentially. The second term, $1/(6 \times 10^9)$, is the error-burst contribution—a fixed constant, completely unaffected by d .

This indicates that increasing code distance stops helping at some point: the bracket can never shrink below $1/(6 \times 10^9)$ no matter how large d gets. As shown in Section 6, it is this second term that produces the $n \approx 50$ asymptotic ceiling.

Solving this combined equation for n , at the same budget $\varepsilon_{\text{total}} = 0.1$, gives a materially different picture at the high end of the range from Section 3 in **Table 2**.

Table 2. Oracle-independent upper bound for logical qubits at each code distance, considering the error burst effect.

d	n_{max} (Gate error only)	n_{max} (gate error + error bursts)
11	22.5	22.5 (unaffected)
15	34.2	34.2 (unaffected)
21	52.6	49.1 (7% change)

At the two smaller code distances, error-burst risk is negligible next to ordinary gate error, and the gate-only bound from Section 3 is essentially unchanged. At $d = 21$, it is not: nearly all of the failure budget at the joint solution is being spent on error-burst risk rather than ordinary error. This raises a natural question, ad-

dressed in Section 6: what happens as code distance keeps increasing—does the bound keep improving indefinitely, or does something else take over? The next section discusses an important concept, the asymptotic ceiling of logical qubits.

6. Asymptotic Ceiling of Logical Qubits

Ordinary gate error can, in principle, be suppressed arbitrarily by choosing a sufficiently large code distance—the surface-code formula (Equation 7) shrinks exponentially in d . Error-burst risk cannot: Equation (19) shows $\varepsilon_{\text{burst}}(n)$ has no dependence on d at all. This asymmetry is the source of Asymptotic Ceiling Equation in this section.

It is visible directly in the factored form of Equation (21):

$$\varepsilon_{\text{total}}(n) \approx \pi \cdot \log_2(n) \cdot 2^{n/2} \cdot \left[n \cdot \kappa \cdot A \cdot \left(\frac{p}{p_{\text{th}}} \right)^{\frac{d+1}{2}} + \frac{1}{6 \times 10^9} \right]$$

As d grows arbitrarily large, the first term inside the brackets shrinks toward zero, but the second term, $1/(6 \times 10^9)$, stays fixed. The bracket can never fall below this floor no matter how large d becomes—meaning $\varepsilon_{\text{total}}(n)$ can never be driven below $\pi \cdot \log_2(n) \cdot 2^{n/2} / (6 \times 10^9)$ by improving the code distance alone.

Taking the $d \rightarrow \infty$ limit, in which the gate-error term vanishes, and setting the entire failure budget equal to error-burst risk alone, gives:

$$\varepsilon_{\text{total}}(n) = \pi \cdot \log_2(n) \cdot 2^{n/2} / (6 \times 10^9) \quad (22)$$

Solving the Asymptotic Ceiling Equation (22) for n gives $n \approx 50.0$. This is the asymptotic ceiling: no matter how large a code distance is chosen, this computation cannot exceed roughly 50 index qubits at the stated budget. Past this point, the limiting factor is no longer ordinary error correction at all—it is a fault mechanism that error correction, as currently understood for this hardware, does not suppress.

7. Discussion of Oracles

Section 3 established that any real oracle can only lower the bounds derived in Sections 3 - 5, never raise them: a real oracle's added depth and ancilla can only add to total resource cost, never subtract from it. This section asks how much lower, without committing to any single oracle's specific cost, since an unstructured black-box oracle has no universal implementation and no single number could speak for all of them.

Rather than model a particular oracle circuit, we introduce one parameter, $\gamma \geq 1$, representing the ratio of the round's total resource cost—diffusion operator and oracle combined—to the diffusion operator's cost alone:

$$D_{\text{round}}(n) = D_{\text{oracle}}(n) + D_{\text{diff}}(n) = \gamma \cdot D_{\text{diff}}(n) \quad (23)$$

$\gamma = 1$ recovers the zero-additional-cost baseline of Sections 3 - 5 exactly. $\gamma = 2$ represents an oracle whose own contribution roughly doubles the round's total

cost, compared to diffusion alone; or oracle cost = diffusion cost. A larger γ represents progressively more expensive oracles. For simplicity, the total active-qubit footprint is scaled by the same factor as the round's total depth:

$$n_{\text{active,round}}(n) = \gamma \cdot n_{\text{active,diff}}(n) \quad (24)$$

This is a deliberate simplification, not a claim about how real oracle circuits behave—a real oracle's depth and ancilla requirements need not scale identically. It is chosen because it makes the resulting sensitivity transparent and traceable to a single, explicitly stated assumption, rather than to an unstated or unjustified circuit model.

Since spacetime volume is the product of active-qubit count and depth, scaling both by γ scales volume by γ^2 :

$$V_{\text{total}}(n, \gamma) = K(n) \cdot n_{\text{active,round}}(n) \cdot D_{\text{round}}(n) = \gamma^2 \cdot V(n) \quad (25)$$

where $V(n)$ is the diffusion-only volume from Section 2. The corresponding gate-error contribution scales the same way,

$$\mathcal{E}_{\text{gate,total}}(n, \gamma) = \gamma^2 \cdot \mathcal{E}_{\text{gate}}(n) \quad (26)$$

Error-burst exposure, by contrast, depends only on total depth (Equation (15)), so it scales linearly rather than quadratically:

$$\mathcal{E}_{\text{burst,total}}(n, \gamma) = \gamma \cdot \mathcal{E}_{\text{burst}}(n) \quad (27)$$

Combining both terms gives the generalized failure budget:

$$\mathcal{E}_{\text{total}}(n, \gamma) \approx \gamma^2 \cdot \mathcal{E}_{\text{gate}}(n) + \gamma \cdot \mathcal{E}_{\text{burst}}(n) \quad (28)$$

The Upper Bound Equation (Equation (14)) extends:

$$\mathcal{E} \approx \pi \cdot \gamma^2 \cdot n \cdot \log_2(n) \cdot 2^{n/2} \cdot \kappa \cdot A \cdot \left(\frac{P}{P_{\text{th}}} \right)^{\frac{d+1}{2}} \quad (29)$$

The Upper Bound Equation (Equation (21)) with Error Burst extends:

$$\mathcal{E}_{\text{total}}(n, \gamma) \approx \pi \cdot \log_2(n) \cdot 2^{n/2} \cdot \left[\gamma^2 \cdot n \cdot \kappa \cdot A \cdot \left(\frac{P}{P_{\text{th}}} \right)^{\frac{d+1}{2}} + \frac{\gamma}{6 \times 10^9} \right] \quad (30)$$

The Asymptotic Ceiling Equation (Equation (22)) extends

$$\mathcal{E}_{\text{total}}(n, \gamma) = \pi \cdot \gamma \cdot \log_2(n) \cdot 2^{n/2} / (6 \times 10^9) \quad (31)$$

Solving Equation (30) for n at fixed γ , across the same code distances used throughout, gives **Table 3**.

Table 3. Maximum index qubits as a function of oracle cost multiplier γ .

d	$\gamma = 1$	$\gamma = 2$	$\gamma = 5$	$\gamma = 10$
11	22.5	19.1	14.8	11.8
15	34.2	30.6	25.9	22.5
21	49.1	46.5	42.7	39.5

The bound degrades gracefully as oracle cost increases rather than collapsing sharply: even a tenfold increase in per-round cost ($\gamma = 10$, a deliberately pessimistic scenario) reduces the $d = 21$ bound from 49.1 to 39.5—a real effect, but one that leaves the paper’s central conclusion intact rather than overturning it. This is the sense in which the qubit bound is oracle-independent in the strong sense established in Section 3: it does not require knowing the oracle’s cost to be useful, and remains informative across a wide range of possible oracle costs.

One resource dimension this section’s model does not capture is worth naming explicitly rather than glossing over: real oracles typically require non-Clifford operations (Toffoli gates decomposed into T-gates, for instance), which in a fault-tolerant surface-code implementation are supplied by dedicated magic-state distillation circuits with their own resource footprint, separate from the depth and ancilla accounting above. Whether this cost is dominant is itself an unsettled question in the literature: it was long treated as the primary bottleneck of fault-tolerant computation following its original proposal [15], but more recent optimized distillation protocols have substantially reduced this overhead—in one detailed worked example, to roughly 2% of total physical qubits [16]. Given this is contested and architecture-dependent, we do not attempt to fold a specific magic-state cost into γ here, since doing so would require committing to a particular oracle’s non-Clifford gate count—precisely the kind of oracle-specific assumption this section has aimed to avoid throughout. A fully worked treatment of non-Clifford resource cost is left as an explicit direction for future work, rather than folded into the present estimate.

8. Discussion

8.1. Depth Wall

The achievable circuit depth of a quantum computation is fundamentally limited by the overhead of Quantum Error Correction (QEC) [7] [8], regardless of how many logical qubits the underlying hardware provides. We call this fixed, assumption-dependent limit on achievable depth the Depth Wall.

For some quantum algorithms, this depth wall does not translate directly into a limit on the number of qubits the algorithm can use, because depth and qubit count are largely independent design choices. But for a specific and important class of algorithms—those whose required depth scales with the size of the problem instance encoded in the qubit register—the Depth Wall translates directly into a resource bound on qubit count.

Grover’s algorithm [1] [2] requires $\Theta(\sqrt{N})$ sequential oracle-and-diffusion rounds to search a database of $N = 2^n$ items, so the number of index qubits, n , and the required circuit depth are locked together; a limit on circuit depth therefore limits the number of qubits, n , as well.

Beyond a certain depth, an independent constraint enters. As shown in Sections 4 - 6, error bursts impose their own limit on achievable depth—one that, unlike ordinary logical error, is not relieved by choosing a larger code distance.

The factors that fix the Depth Wall—the $\Theta(\sqrt{N})$ amplitude-amplification requirement, the per-round resource cost of the diffusion operator (and, per Section 7, the oracle), the per-gate error suppression of surface-code QEC, and error bursts—together place a practical limit on the largest unstructured search problem, in terms of index qubits n , that can be run to completion within a fixed computational error budget.

8.2. A Tighter Constraint than Hardware Qubit Count

Because this limit follows from the depth requirement of the algorithm itself, it applies independently of how many logical qubits the underlying hardware provides. Larger quantum computers, in terms of more logical qubits, will not help in these cases, making this resource bound a tighter restriction than the underlying hardware’s qubit count.

Recent public hardware reports already describe systems with several dozen logical qubits and roadmaps targeting hundreds by the end of the decade. This is fully consistent with the estimates above, because none of those demonstrations are running deep $\Theta(\sqrt{N})$ Grover searches—they are running shallow entangling circuits [17], Bell-state [17] and GHZ-state preparations [18] [19], Bernstein-Vazirani-type algorithms [20], or error-correction memory experiments [5], all of which have depth requirements that grow slowly, or not at all, with qubit count.

Hardware logical qubit count and the depth-derived index-qubit bound are separate axes of scaling, and they are currently improving at very different rates. The practical implication is that near-term and early-fault-tolerant machines will likely have enough logical qubits to index very large unstructured search spaces well before they have enough usable depth to actually search them—Grover’s algorithm, and any other algorithm with genuinely problem-size-dependent sequential depth requirements, will lag behind raw qubit-count progress for exactly this reason.

9. Conclusions

This paper asked a resource question about unstructured search rather than a query-complexity one: given realistic fault-tolerant hardware constraints, how large a Grover search can actually be carried to completion in terms of the number of logical qubits? The answer developed across the paper has two independent parts, and neither reduces to a simple qubit count.

The first part concerns the algorithm’s own structure. Section 2 and 3 showed that the diffusion operator alone, together with the round count fixed by the $\sqrt{N}$ lower bound, establishes a resource bound that no oracle implementation can loosen—an oracle-independent upper bound of roughly 22 to 53 index qubits across the code distances considered, depending on the surface code’s suppression of ordinary gate error. The Upper Bound Equation is derived and solving this equation will produce this oracle-independent upper bound as a function of code distance. Section 7 showed this bound degrades gracefully, not catastrophically,

as real oracle cost is introduced: even a tenfold increase in per-round cost relative to the diffusion operator alone left the paper's central conclusions intact. The Upper Bound Equation with Oracle is derived and solving this equation will produce the upper bound as function of code distance with oracle effects.

The second, more consequential part concerns a failure mode that standard resource estimates do not typically account for. Sections 4 through 6 showed that error bursts—rare, transient, spatially correlated fault events with a specific, well-characterized physical origin in cosmic-ray and ambient-radiation exposure—are not suppressed by increasing code distance the way ordinary logical error is, at least under currently reported, largely unmitigated hardware operation. This asymmetry produces an asymptotic ceiling: no matter how large a code distance is chosen, the computation modeled in this paper cannot exceed roughly 50 index qubits at the stated error budget, because the limiting factor stops being ordinary error correction entirely. The Asymptotic Ceiling Equation is derived and solving this equation will produce this ceiling, 50.

This finding should be read for precisely what it is, not more. It is a conditional resource estimate, not a fundamental limit on quantum computing or on the number of logical qubits fault-tolerant hardware can support—current devices already exceed the qubit counts discussed here, running algorithms whose required depth does not couple to problem size the way Grover's does. It is not a proven, irreducible physical bound: mitigation research targeting the error-burst rate directly is active and has shown real partial success, and separate, dedicated theoretical work indicates that appropriately designed mitigation and decoding may restore favorable code-distance scaling even in the presence of these events. And it is only partly general: the qualitative concern—that rare, correlated failures can defeat the independent-fault assumption standard error correction relies on, in a way ordinary code-distance scaling does not address—plausibly extends across quantum computing platforms, but the specific numbers reported here are tied to one class of superconducting hardware's currently measured, largely unmitigated behavior.

Two limitations are worth stating explicitly as directions for future work rather than as gaps quietly left unaddressed. First, this paper's resource model accounts for depth, gate count, and idle-versus-active logical error, but does not cost the non-Clifford operations that any real oracle, and the diffusion operator's own ancilla-tree construction, are likely to require; magic-state distillation cost is a separate resource dimension this paper has deliberately left unresolved, since folding it in would require committing to a specific, unjustifiable oracle circuit. Second, the error-burst rate used throughout is a single hardware-specific measurement; as mitigation strategies mature and as other qubit modalities are characterized for their own correlated-failure mechanisms, the numerical ceiling reported here should be expected to shift, likely upward, without the underlying methodology changing.

The broader point this paper makes is not about Grover's algorithm specifically.

Any algorithm whose required circuit depth scales with the size of the problem it solves will eventually run long enough, in wall-clock QEC cycles, to become exposed to rare correlated failure modes that ordinary error-correction scaling does not address. As fault-tolerant hardware matures and deep algorithms become a realistic target rather than a theoretical one, resource estimates for such algorithms may need to account for this class of failure explicitly, alongside the gate-count and code-distance considerations that have so far dominated the literature.

Acknowledgements

We thank Claude (Anthropic AI assistant, Claude Sonnet 4.5) for extensive technical analysis, systematic review, and constructive feedback during manuscript preparation. We thank Gina Porter for proofreading this paper and valuable suggestions for improving clarity. We thank ChatGPT for many discussions.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Grover, L.K. (1996) A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of the Twenty-Eighth Annual ACM symposium on Theory of Computing*, Philadelphia, 22-24 May 1996, 212-219. <https://doi.org/10.1145/237814.237866>
- [2] Boyer, M., Brassard, G., Høyer, P. and Tapp, A. (1998) Tight Bounds on Quantum Searching. *Fortschritte der Physik*, **46**, 493-505. [https://doi.org/10.1002/\(sici\)1521-3978\(199806\)46:4<493::aid-prop493>3.0.co;2-p](https://doi.org/10.1002/(sici)1521-3978(199806)46:4<493::aid-prop493>3.0.co;2-p)
- [3] Zalka, C. (1999) Grover's Quantum Searching Algorithm Is Optimal. *Physical Review A*, **60**, 2746-2751. <https://doi.org/10.1103/physreva.60.2746>
- [4] McEwen, M., Faoro, L., Arya, K., Dunsworth, A., Huang, T., Kim, S., *et al.* (2022) Resolving Catastrophic Error Bursts from Cosmic Rays in Large Arrays of Superconducting Qubits. *Nature Physics*, **18**, 107-111. <https://doi.org/10.1038/s41567-021-01432-8>
- [5] Acharya, R., Abanin, D.A., Aghababaie-Beni, L., Aleiner, I., Andersen, T.I., Ansmann, M., *et al.* (2025) Quantum Error Correction Below the Surface Code Threshold. *Nature*, **638**, 920-926. <https://doi.org/10.1038/s41586-024-08449-y>
- [6] Tan, S.J.S., Pattison, C.A., McEwen, M. and Preskill, J. (2026) Resilience of the Surface Code to Error Bursts. *Physical Review A*, **113**, Article 022450. <https://doi.org/10.1103/vwtl-hwfx>
- [7] Fowler, A.G., Whiteside, A.C. and Hollenberg, L.C.L. (2012) Towards Practical Classical Processing for the Surface Code. *Physical Review Letters*, **108**, Article 180501. <https://doi.org/10.1103/physrevlett.108.180501>
- [8] Fowler, A.G., Mariantoni, M., Martinis, J.M. and Cleland, A.N. (2012) Surface Codes: Towards Practical Large-Scale Quantum Computation. *Physical Review A*, **86**, Article 032324. <https://doi.org/10.1103/physreva.86.032324>
- [9] Lin, W., Cho, H., Chen, Y., Vavilov, M.G., Wang, C. and Manucharyan, V.E. (2025) 24 Days-Stable CNOT Gate on Fluxonium Qubits with over 99.9% Fidelity. *PRX Quantum*, **6**, Article 010349. <https://doi.org/10.1103/prxquantum.6.010349>
- [10] Ransford, A., Allman, M.S., Arkinstall, J., Campora, J.P., Cooper, S.F., Delaney, R.D.,

- et al.* (2026) A 98-Qubit Trapped-Ion Quantum Computer with All-to-All Connectivity. *Nature*, **655**, 81-86. <https://doi.org/10.1038/s41586-026-10676-4>
- [11] Zhang, H., Ding, C., Weiss, D.K., Huang, Z., Ma, Y., Guinn, C., *et al.* (2024) Tunable Inductive Coupler for High-Fidelity Gates between Fluxonium Qubits. *PRX Quantum*, **5**, Article 020326. <https://doi.org/10.1103/prxquantum.5.020326>
- [12] Vezvae, A., Benito, C., Morford-Oberst, M., Bermudez, A. and Lidar, D.A. (2026) Surface Code Scaling on Heavy-Hex Superconducting Quantum Processors. *Nature Communications*, **17**, Article No. 9201. <https://doi.org/10.1038/s41467-026-76090-6>
- [13] Iaia, V., Ku, J., Ballard, A., Larson, C.P., Yelton, E., Liu, C.H., *et al.* (2022) Phonon Downconversion to Suppress Correlated Errors in Superconducting Qubits. *Nature Communications*, **13**, Article No. 6425. <https://doi.org/10.1038/s41467-022-33997-0>
- [14] Xu, Q., Seif, A., Yan, H., Mannucci, N., Sane, B.O., Van Meter, R., *et al.* (2022) Distributed Quantum Error Correction for Chip-Level Catastrophic Errors. *Physical Review Letters*, **129**, Article No. 240502. <https://doi.org/10.1103/physrevlett.129.240502>
- [15] Bravyi, S. and Kitaev, A. (2005) Universal Quantum Computation with Ideal Clifford Gates and Noisy Ancillas. *Physical Review A*, **71**, Article 022316. <https://doi.org/10.1103/physreva.71.022316>
- [16] Litinski, D. (2019) Magic State Distillation: Not as Costly as You Think. *Quantum*, **3**, Article 205. <https://doi.org/10.22331/q-2019-12-02-205>
- [17] Zhang, J., Chen, Z., Wang, Y., Lu, B., Zhang, H., Li, J., *et al.* (2025) Demonstrating a Universal Logical Gate Set in Error-Detecting Surface Codes on a Superconducting Quantum Processor. *npj Quantum Information*, **11**, Article No. 177. <https://doi.org/10.1038/s41534-025-01118-6>
- [18] Bluvstein, D., Evered, S.J., Geim, A.A., Li, S.H., Zhou, H., Manovitz, T., *et al.* (2024) Logical Quantum Processor Based on Reconfigurable Atom Arrays. *Nature*, **626**, 58-65. <https://doi.org/10.1038/s41586-023-06927-3>
- [19] Dasu, S., DeCross, M., Guo, A.Y., Lavasani, A., Behrends, J., *et al.* (2026) Computing with Many Encoded Logical Qubits beyond Break-Even. arXiv:2602.22211.
- [20] Reichardt, B.W., Paetznick, A., Aasen, D. *et al.* (2025) Fault-Tolerant Quantum Computation with a Neutral Atom Processor. arXiv:2411.11822.
- [21] Brassard, G., Høyer, P., Mosca, M. and Tapp, A. (2002) Quantum Amplitude Amplification and Estimation. *Contemporary Mathematics*, **305**, 53-74.
- [22] Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., *et al.* (1995) Elementary Gates for Quantum Computation. *Physical Review A*, **52**, 3457-3467. <https://doi.org/10.1103/physreva.52.3457>

Appendix A: Diffusion Operator Construction

Two constructions implement the diffusion operator, $D = 2|\psi\rangle\langle\psi| - I$, applied once per Grover round; only the circuit realizing it differs. The number of rounds, $K(n) = (\pi/4) \cdot 2^{n/2}$, is fixed by amplitude amplification and is identical for both constructions. This appendix isolates the diffusion operator alone, independent of the oracle operator.

Any reflection about a prepared state $|\psi\rangle$ can be implemented via the identity [21]:

$$R_\psi = 2|\psi\rangle\langle\psi| - I = U(2|0\rangle\langle 0| - I)U^\dagger \quad (\text{A.1})$$

where U prepares the state, $U|0\rangle = |\psi\rangle$. Rather than implementing $|\psi\rangle\langle\psi|$ directly, the reflection proceeds in three steps: 1) uncompute $|\psi\rangle$ to $|0\rangle$ via $U^\dagger$; 2) flip the phase, $|0^n\rangle \rightarrow -|0^n\rangle$; 3) recompute $|0\rangle$ back to $|\psi\rangle$ via U . So $R_\psi = U \cdot (\text{phase flip}) \cdot U^\dagger$. Steps 1 and 3 cost $(U^\dagger)$ and (U) respectively; step 2 is a multi-controlled-Z gate on n qubits, whose depth and ancilla requirements depend on which construction is chosen—the subject of this appendix.

A.1. Two Constructions

Barenco *et al.* [22] give two constructions for the n -fold controlled phase flip with different depth/ancilla trade-offs:

Option 1 (minimal ancilla). A single borrowed ancilla qubit and a chain of Toffoli-type gates, each depending structurally on the previous step's output. Depth scales linearly with the number of controls:

$$D_{\text{diff},1}(n) \approx n \quad (\text{A.2})$$

$$n_{\text{ancilla}} \approx O(1) \quad (\text{A.3})$$

$$n_{\text{active},1}(n) \approx n \quad (\text{A.4})$$

Option 2 (ancilla tree). An $(n-2)$ -ancilla balanced binary tree computes the n -fold AND in depth $\log_2 n$ by letting independent branches proceed concurrently, applies a single controlled-Z, then uncomputes the tree:

$$D_{\text{diff},2}(n) \approx 2 \cdot \log_2 n \quad (\text{A.5})$$

$$N_{\text{ancilla}} \approx n - 2 \quad (\text{A.6})$$

$$n_{\text{active},2}(n) \approx 2n \quad (\text{A.7})$$

These are genuinely different circuits, not two schedulings of the same one: Option 1's chain cannot be parallelized below $O(n)$ depth, since each link depends on the last; Option 2 trades $n-2$ additional qubits for the ability to compute independent branches at once.

A.2. Spacetime Volume

§2.2, total logical spacetime volume V is:

$$V = K(n) \cdot n_{\text{active}}(n) \cdot D_{\text{diff}}(n) \quad (\text{A.8})$$

From which,

$$V_1(n) = (\pi/4) \cdot 2^{n/2} \cdot n \cdot n = (\pi/4) \cdot n^2 \cdot 2^{n/2} \quad (\text{A.9})$$

$$V_2(n) = (\pi/4) \cdot 2^{n/2} \cdot 2n \cdot 2 \log_2 n = \pi \cdot n \cdot \log_2(n) \cdot 2^{n/2} \quad (\text{A.10})$$

A.3. Selection

To compute crossover,

$$V_2(n)/V_1(n) = 4 \log_2(n)/n \quad (\text{A.11})$$

Setting this ratio to 1 and solving $4 \log_2(n) = n$ gives an exact crossover at $n = 16$ ($4 \cdot \log_2 16 = 16$). For $n < 16$, Option 1 has lower spacetime volume; for $n > 16$, Option 2 dominates, with the advantage widening as n grows (see [Table A1](#)).

Table A1. V_2/V_1 at representative problem sizes.

n	$4 \log_2(n)/n$	Favored option
10	1.33	Option 1
16	1.00	crossover
25	0.74	Option 2
37	0.56	Option 2
55	0.42	Option 2

Because $n \geq 16$ spans the entire regime this paper addresses, Option 2 is adopted throughout the main text as the default diffusion-operator construction, conditional on sufficient ancilla being available.