Finite Distinction Maintenance in Fault-Tolerant Quantum Computation
Logical Distinction Ledgers, Error-Correction Infrastructure, and Architecture-Specific Resource Bounds

Yining Wu[1, *]

[1] *Independent Researcher*

(Dated: May 2026)

Fault-tolerant quantum computation suppresses logical error by maintaining fragile quantum distinctions under noise. This paper develops a threshold-compatible finite-maintenance audit for this process. It does not claim that quantum computation is impossible, does not refute the threshold theorem, and does not identify every syndrome measurement with Landauer erasure. A logical qubit is treated as a protected quantum quotient or code-algebra distinction, not as a simultaneously readable classical bit. Quantum error correction converts distinction loss into maintenance work: redundancy, syndrome acquisition, decoding, feedback, reset, cooling, routing, reservoir engineering, and control. The central object is a vector ledger, not a scalar cost: syndrome, decoding, reset, latency, power, cooling, and effective-error components must each fit their corresponding capacity window and physical location. Landauer's principle enters only as a lower bound on logically irreversible reset, overwrite, or garbage disposal under its standard assumptions; real hardware may relocate erasure and entropy export across cold stages, room-temperature electronics, optical channels, or autonomous reservoirs. This paper therefore frames scalability as an architecture-specific ledger problem: logical error can fall only if the physical infrastructure that maintains the logical distinction grows, relocates, or reorganizes without exceeding finite resource, latency, and dissipation budgets.

READER CONTRACT

This paper is not a no-go theorem for quantum computation. It accepts the threshold theorem as a conditional mathematical result: under suitable noise, locality, error-rate, decoding, and implementation assumptions, increasing code distance can suppress logical error [8–11]. Recent below-threshold surface-code demonstrations and low-overhead fault-tolerant memory proposals make this paper's audit framing especially timely: the issue is not whether logical error can be suppressed, but how the maintenance ledger scales when it is suppressed [30, 32]. This paper asks a different question: what finite physical ledger must be paid to instantiate those assumptions at scale?

This paper does not claim that quantum error correction cannot work, that Landauer cost is the only cost in quantum hardware, that every syndrome measurement dissipates $k_{\mathrm{B}}T \ln 2$ at the measurement location, or that all architectures share the same cooling bottleneck. Syndrome measurement creates correction information; Landauer's bound applies to logically irreversible reset, overwrite, or disposal of information under the standard physical bridge assumptions [5–7]. Real devices also pay positive control, readout, wiring, leakage, clocking, isolation, amplifier, transport, and refrigeration costs. This paper does not assume that every extracted syndrome bit is erased at the same temperature stage where it is measured. Syndrome information can be exported, buffered, decoded elsewhere, processed reversibly for a time, or dissipated in an engineered reservoir. The audit requirement is that the reset, garbage, heat, latency, and entropy-export path be specified rather than hidden. The claim of this paper is narrower:

Logical error suppression requires finite distinction-maintenance infrastructure.

A claimed escape from Q2 must identify where the correction entropy, reset burden, latency, heat, or failure probability has been relocated.

I. CLAIM-STATUS SUMMARY

II. INTRODUCTION

A. Quantum computation as distinction preservation

Quantum computation is often described in terms of Hilbert-space dimension, entanglement, interference, or algorithmic speedup. Those descriptions are correct but incomplete from an FDS perspective. A useful quantum computation must preserve task-relevant alternatives long enough to initialize, transform, correct, decode, read out, and verify them. The logical states $|0_{\mathrm{L}}\rangle$ and $|1_{\mathrm{L}}\rangle$, a stabilizer sector, a phase relation used by an algorithm, or a nonlocal topological degree of freedom are not free distinctions. They are fragile logical distinctions that must be maintained under noise.

This paper's central question is therefore:

What must be paid to keep $Q_{\mathrm{L}} = q(\rho_{\mathrm{phys}})$ stable under noise?

* yining.wu@alumni.upenn.edu

TABLE I. Central claims of this paper, status, and demotion conditions. This paper is formulated as a finite-maintenance audit, not as a universal impossibility theorem.

Claim	Status	Demotion or failure condition
Logical qubits are protected quantum distinctions	FDS/QI bridge	Logical information cannot be represented as a protected code quotient or logical algebra.
QEC is active finite-distinction maintenance	Main interpretation of this paper	Logical distinctions are preserved under noise with no redundancy, syndrome information, feedback, passive protection, or maintenance channel.
Threshold theorem is accepted as conditional baseline	Scope firewall	The paper is read as refuting the threshold theorem rather than auditing its physical assumptions.
Correction demand is a vector ledger	Engineering bridge	Syndrome, decoding, reset, latency, cooling, and control demands cannot be collapsed into one scalar without losing bottleneck information.
Irreversible reset has a Landauer lower bound	Physical bridge	Reliable logically irreversible reset below $k_{\mathrm{B}}T\ln 2$ per erased bit under standard Landauer conditions.
Cryogenic solid-state systems face cold-stage ledger constraints	Architecture-specific claim	Large-scale cryogenic processors maintain falling logical error while all cold-stage power, routing, latency, and reset ledgers remain within budget.
Topological/passive protection can reduce active load	Escape-channel bridge	Passive protection removes preparation, finite-temperature, readout, braiding, leakage, and residual correction costs entirely.
Q2 failure does not falsify FDS Core or Q1	Failure propagation rule	Architecture-specific failure is treated as failure of the formal finite-system core or quantum mechanics.

Here q is a logical quotient from the physical state to the computational degree of freedom that the algorithm requires. The maintenance problem is not identical to final measurement. A logical qubit is not a stable classical record; it is a protected quantum quotient whose final readout may later become a record. This distinction inherits the discipline of Q1, where coherence preservation, internal record stability, external record availability, and erasure history were separated rather than conflated [2].

B. From Q1 record boundaries to maintenance cost analysis

Q1 treated Wigner-friend scenarios as finite record-boundary promotion problems. A friend-relative record cannot be promoted into Wigner's accessible algebra unless a physical record channel supplies sufficient mutual information. Q1 also introduced a record-availability horizon and emphasized that device diagnostics should separate coherence time, internal record stability, external availability, and erasure history [2]. This paper applies the same finite-boundary discipline to quantum computation. In one line: Q1 asks when a quantum interaction becomes an accessible record; this paper asks what maintains a logical quantum distinction before readout. The relevant object is no longer only a measurement record Z, but a logical quantum distinction Q_{L} maintained by a code, controller, decoder, reservoir, or protected sector.

C. Maintenance constraints, not no-go theorems

Finite-maintenance constraints are ledger constraints, not mathematical impossibility claims. The threshold theorem says that logical error can be suppressed by increasing encoded structure when its assumptions hold. This paper asks whether the physical system can keep increasing, relocating, and organizing that structure without violating finite capacity, latency, cooling, and free-energy budgets.

Thus the thesis of this paper can be summarized as:

> Fault tolerance suppresses logical error by increasing maintained structure;
> Q2 audits whether the maintenance ledger scales with it.

III. FDS INPUTS

A. Active finite distinction systems

The FDS Core defines an active finite distinction system as a tuple

$$S=(X,E,B,M,Y,A,U,\pi,\ell,\Phi,\mathcal{P},\tau), \qquad (1)$$

where X is the system state space, E is the environment, B is the boundary, M is internal memory or model space, Y is the observation channel, A is the action or control space, U is the update rule, π is a finite distinction projection, ℓ is a boundary-maintenance loss, Φ is a finite resource budget, $\mathcal{P}$ is an admissible perturbation family, and τ is the update timescale [1]. This paper maps these components onto quantum hardware: physical qubits are carriers, the code is a quotient, noise is the perturbation family, syndrome extraction is the observation channel, recovery is the action/update channel, and the maintenance loss is the logical error probability.

B. Capacity deficit

The FDS capacity-deficit theorem states that if the task-relevant rate-distortion demand exceeds the internal capacity,

$$R_{\min}^{(\tau)}(\epsilon) > C_S, \tag{2}$$

then no purely internal model of capacity C_S can encode the least demanding admissible task statistic to tolerance ϵ over window τ; the system must approximate, externalize, relax the task, or fail [1].

For QEC, define a correction demand ledger

$$\mathcal{R}_{\mathrm{corr}}^{(\tau)}(\epsilon_L; \mathcal{N}, \mathcal{C}) = \big(B_{\mathrm{syn}}, B_{\mathrm{dec}}, B_{\mathrm{reset}}, L_{\mathrm{latency}}, P_{\mathrm{maint}}\big), \tag{3}$$

where $\mathcal{N}$ is the noise family and $\mathcal{C}$ is the code/control architecture. The corresponding maintenance capacity ledger is

$$\mathcal{C}_{\mathrm{maint}}^{(\tau)} = \big(C_{\mathrm{read}}, C_{\mathrm{dec}}, C_{\mathrm{reset}}, C_{\mathrm{feed}}, C_{\mathrm{cool}}, C_{\mathrm{ctrl}}, C_{\mathrm{latency}}\big). \tag{4}$$

The QEC deficit condition is vectorial:

$$\Delta_{\mathrm{QEC}}^{(\tau)} > 0 \quad \Longleftrightarrow \quad \mathcal{R}_{\mathrm{corr}}^{(\tau)} \not\preceq \mathcal{C}_{\mathrm{maint}}^{(\tau)} \tag{5}$$

for at least one required ledger component. Writing the problem this way prevents a common mistake: a system can have enough syndrome bits but insufficient cooling, enough cooling but insufficient decoding latency, or adequate readout but excessive routing-induced idle error.

C. Componentwise ledger convention

The order symbol $\preceq$ is not a scalar comparison across incommensurable units. Each ledger component is compared only after conversion to the relevant window, location, and unit. Bit demands are measured per correction window or per second; decoding load is measured in operations or bits per latency window; reset is measured as irreversible bits per cycle and by the temperature stage where they are erased; power is measured in watts at the relevant stage; latency is measured in seconds relative to τ_{cyc}; and error probabilities are compared to the code model's threshold assumptions. Thus

$$\mathcal{R}_{\mathrm{corr}} \preceq \mathcal{C}_{\mathrm{maint}} \tag{6}$$

means a componentwise feasibility audit, not a universal conversion of bits, watts, seconds, and probabilities into one currency.

D. Landauer bridge and relocation

The FDS physical bridge uses Landauer's principle for logically irreversible updates: under its standard physical assumptions, erasing one bit dissipates at least $k_{\mathrm{B}}T \ln 2$ of heat [5–7]. The FDS Core writes logical erasure per update as a preimage entropy,

$$b_t = H(M_t \mid M_{t+1}, Y_t), \tag{7}$$

and the informational heat rate obeys

$$\dot{Q}_{\mathrm{info}} \geq \frac{k_{\mathrm{B}}T \ln 2}{\tau} H(M_t \mid M_{t+1}, Y_t), \tag{8}$$

when the Landauer bridge assumptions are satisfied [1]. This paper imports only this lower-bound logic. Syndrome acquisition is not automatically Landauer erasure. The erasure floor attaches to reset, overwrite, garbage disposal, and other logically irreversible reuse operations.

A central principle of this paper follows:

Maintenance cost can be relocated, but not made unaccounted.

Moving a decoder to room temperature, replacing digital measurement with autonomous dissipation, using photonic feed-forward, or using topological protection changes the ledger location. It does not remove the need to account for entropy export, power, latency, finite-temperature events, readout, preparation, or failure probability. Modern machine-learning decoders improve logical performance by learning from noisy syndrome data, but they also make decoding capacity, training data, latency, and deployment location explicit audit variables [31].

IV. QUANTUM-COMPUTING BASELINE

A. Logical qubits and code spaces

A quantum error-correcting code embeds a logical Hilbert space in a larger physical Hilbert space,

$$\mathcal{H}_{\mathrm{code}} \subset \mathcal{H}_{\mathrm{phys}}^{\otimes n}. \tag{9}$$

Noise acts as a channel $\mathcal{N}_t$ and recovery attempts to restore the logical sector:

$$\mathcal{R} \circ \mathcal{N}_t(\rho_{\mathrm{phys}}) \approx \rho_{\mathrm{code}}. \tag{10}$$

For stabilizer codes, the code space is the simultaneous $+1$ eigenspace of stabilizer generators S_i,

$$\mathcal{H}_{\mathrm{code}} = \{\, |\psi\rangle : S_i|\psi\rangle = |\psi\rangle \ \forall i \,\}. \tag{11}$$

The logical operators form an algebra acting on equivalence classes of physical states modulo correctable errors. In FDS language, this is an invariant or quotient-supported distinction.

B. Threshold theorem as accepted baseline

The threshold theorem is not challenged here. It is the mathematical baseline that makes this paper interesting. If physical errors are below threshold and the model assumptions hold, increasing code distance can suppress logical error [8–11]. This paper does not ask whether the theorem is true. It asks how much physical machinery is required to keep its assumptions true at useful scale.

C. Surface-code normal form

For a surface-code-like architecture [12–14], a common normal form is

$$N_{\mathrm{phys}} \sim \alpha n_L d^2, \tag{12}$$

where n_L is the number of logical qubits and d is code distance. A schematic logical error expression is

$$p_L(d,p) \sim A\left(\frac{p}{p_{\mathrm{th}}}\right)^{(d+1)/2}, \quad p < p_{\mathrm{th}}. \tag{13}$$

Syndrome load per cycle scales similarly,

$$B_{\mathrm{syn/cycle}} \sim \beta n_L d^2. \tag{14}$$

For a target logical error $p_L^\star$, the same normal form gives the required distance

$$d_{\mathrm{req}} \approx 2\frac{\ln(A/p_L^\star)}{\ln(p_{\mathrm{th}}/p)} - 1, \qquad p < p_{\mathrm{th}}. \tag{15}$$

Thus $p_{\mathrm{eff}} \uparrow$ implies $d_{\mathrm{req}} \uparrow$, which in turn increases N_{phys}, B_{syn}, routing, decoding, and reset demand. These equations are not universal hardware laws. They supply a normal form for illustrating the core tradeoff described here: logical error decreases while maintained distinction infrastructure grows.

V. DEFINITIONS

Definition 1 (Logical distinction)**.** *A logical distinction is a task-relevant quantum-computational alternative encoded as a protected quotient or logical algebra,*

$$Q_{\mathrm{L}} = q(\rho_{\mathrm{phys}}), \tag{16}$$

where q maps physical states to a logical code sector, stabilizer equivalence class, logical operator algebra, or protected topological sector. A logical distinction is not a simultaneously readable classical alternative and is not identical to a classical measurement record. It is a protected operational difference inside a code algebra, accessible through allowed logical operations and final readout.

Definition 2 (Distinction-maintenance task)**.** *A quantum processor performs finite distinction maintenance over window τ if it keeps the logical quotient stable to tolerance ϵ_L under an admissible noise family $\mathcal{N}$:*

$$P[\widehat{Q_{\mathrm{L}}}(t+\tau) = Q_{\mathrm{L}}(t)] \geq 1 - \epsilon_L, \tag{17}$$

where equality is understood operationally relative to the logical algebra and the accepted decoding criterion.

Definition 3 (Correction ledger)**.** *The QEC correction ledger is*

$$\begin{aligned}\mathcal{L}_{\mathrm{QEC}} = \big(&B_{\mathrm{syn}}, B_{\mathrm{dec}}, B_{\mathrm{reset}}, B_{\mathrm{leak}},\\ &P_{\mathrm{ctrl}}, P_{\mathrm{read}}, P_{\mathrm{wire}}, P_{\mathrm{cool}},\\ &\tau_{\mathrm{cyc}}, d, N_{\mathrm{phys}}\big).\end{aligned} \tag{18}$$

It records syndrome bits, decoding bits, reset or garbage bits, leakage events, control power, readout power, wiring power, cooling budget, cycle time, code distance, and physical carrier count.

Definition 4 (Maintenance capacity)**.** *The finite operational maintenance capacity over a correction window is the vector*

$$\begin{aligned}\mathcal{C}^{(\tau)}_{\mathrm{maint}} = \big(&C_{\mathrm{read}}, C_{\mathrm{dec}}, C_{\mathrm{feed}}, C_{\mathrm{reset}},\\ &C_{\mathrm{cool}}, C_{\mathrm{ctrl}}, C_{\mathrm{latency}}\big).\end{aligned} \tag{19}$$

It is not a Hilbert-space dimension. It is the finite capacity of the correction, reset, readout, cooling, feedback, control, and timing infrastructure.

Definition 5 (Reset-erasure rate)**.** *Let $b_{\mathrm{erase/cycle}}$ denote the number of logically irreversible reset, overwrite, or garbage-disposal bits per correction cycle. Then*

$$\dot{b}_{\mathrm{erase}} = \frac{b_{\mathrm{erase/cycle}}}{\tau_{\mathrm{cyc}}}. \tag{20}$$

Under Landauer bridge assumptions, the informational heat floor is

$$\dot{Q}_{\mathrm{erase}} \geq k_{\mathrm{B}} T \ln 2\, \dot{b}_{\mathrm{erase}}. \tag{21}$$

Definition 6 (Ledger relocation)**.** *A maintenance ledger is relocated when correction entropy, heat, latency, memory, decoding, reset, or failure probability is moved from one subsystem or temperature stage to another. Formally,*

$$\begin{aligned}\mathcal{L}_{\mathrm{QEC}} = &\mathcal{L}_{\mathrm{cold}} + \mathcal{L}_{\mathrm{room}} + \mathcal{L}_{\mathrm{optical}}\\ &+ \mathcal{L}_{\mathrm{control}} + \mathcal{L}_{\mathrm{time}} + \mathcal{L}_{\mathrm{failure}}.\end{aligned} \tag{22}$$

Relocation is a valid engineering strategy. It is not disappearance of the ledger.

Definition 7 (Record availability horizon for QEC)**.** *Using the Q1 diagnostic, define*

$$\tau_{\mathrm{RAH}} = \inf\{t : I_2(Z_{\mathrm{internal}}; Z_{\mathrm{external}}(t)) \geq H_2(Z_{\mathrm{internal}}) - \epsilon\}. \tag{23}$$

In this paper this is a hardware diagnostic for when an internal logical, syndrome, or readout record has leaked, been exported, or become externally available. It is not a new fundamental timescale.

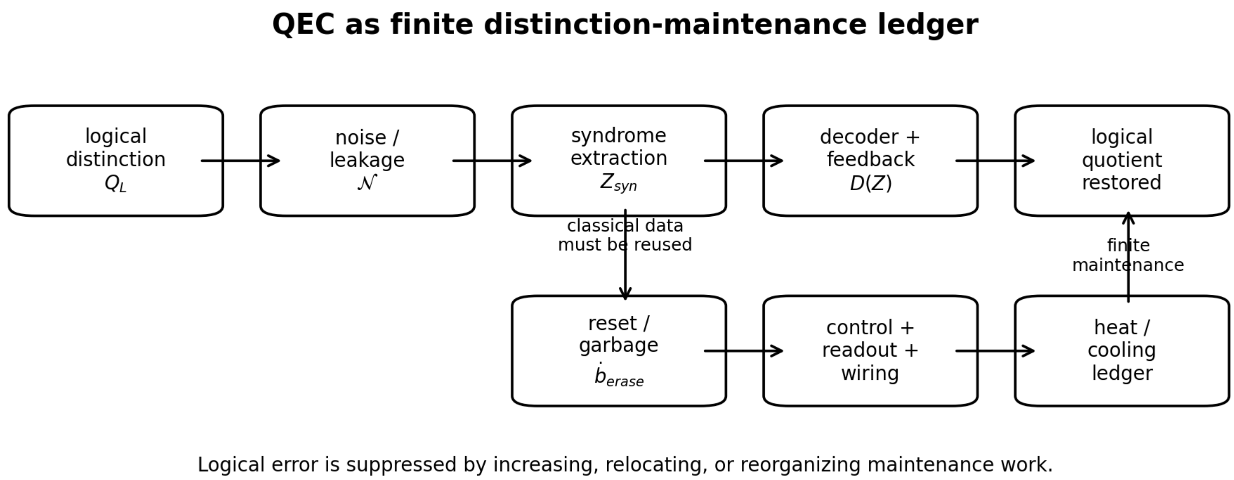

FIG. 1. QEC as a finite distinction-maintenance ledger. Logical error is suppressed by extracting syndrome information, decoding, applying feedback, resetting reusable carriers, and paying control/readout/cooling costs. This is a schematic normal form, not a hardware model.

VI. GENERAL RESULTS

Proposition 1 (QEC as active finite-distinction maintenance)**.** *Quantum error correction is active finite-distinction maintenance: it preserves a logical quotient* $Q_{\mathrm{L}} = q(\rho_{\mathrm{phys}})$ *under a noise family* $\mathcal{N}$ *by using redundancy, syndrome information, recovery operations, passive protection, or engineered dissipation.*

Proof. A code maps many physical states to a smaller logical sector. Correctable errors alter the physical representative while preserving the logical quotient after recovery. Syndrome extraction, recovery, or autonomous dissipation supplies the update channel that maintains the quotient under perturbations. Therefore the code implements the FDS pattern of active boundary maintenance: it does not eliminate perturbations; it converts them into correction work or passive-protection requirements. □

Audit Principle 1 (No-free logical distinction maintenance)**.** *Let a logical distinction* Q_{L} *be exposed to an admissible noise family* $\mathcal{N}$ *with nonzero entropy injection, leakage, or drift relative to the target logical algebra. If passive protection does not fully suppress* $\mathcal{N}$*, then maintaining* Q_{L} *below error tolerance* ϵ_L *over window* τ *requires at least one of: redundancy, syndrome information, feedback, reset or export, engineered dissipation, latency increase, resource growth, task relaxation, or failure.*

This is an audit principle rather than a formal no-go theorem. Its content is the classification of possible maintenance exits. If $\mathcal{N}$ introduces task-relevant uncertainty about Q_{L} and passive protection does not fully remove it, the architecture must either acquire and process information about the error, encode the logical distinction redundantly, dissipate entropy into a reservoir, slow the task to reduce error accumulation, relax the tolerance, or fail to maintain the target distinction. This is the quantum-computing instance of the FDS capacity-deficit exit logic: when task demand exceeds maintained capacity, approximation, externalization, task relaxation, resource growth, or failure is forced [1].

Proposition 2 (Correction-ledger feasibility)**.** *A finite QEC architecture can maintain a target logical error rate only if the correction demand fits the available maintenance ledger. A useful audit form is*

$$\begin{aligned} B_{\mathrm{syn}} &\le C_{\mathrm{read}}, \quad B_{\mathrm{dec}} \le C_{\mathrm{dec}}, \\ B_{\mathrm{reset}} &\le C_{\mathrm{reset}}, \quad \tau_{\mathrm{read}} + \tau_{\mathrm{dec}} + \tau_{\mathrm{feed}} \le \tau_{\mathrm{cyc}}, \\ P_{\mathrm{maint}} &\le P_{\mathrm{avail}}, \quad p_{\mathrm{eff}} < p_{\mathrm{th}}. \end{aligned} \tag{24}$$

Failure of any component creates a maintenance bottleneck even if other components have spare capacity.

Proof. Each inequality is required to instantiate a correction cycle at the specified tolerance. Syndrome information must be read, decoded, fed back, and reset or exported before accumulated errors exceed the code's correction capability. Power must fit within the available cooling and energy budget. Effective physical error must remain below the threshold assumed by the code model. A violation of any component invalidates the assumed maintenance cycle. □

Proposition 3 (Reset-erasure lower bound)**.** *If a QEC architecture performs* $b_{\mathrm{erase/cycle}}$ *logically irreversible reset, overwrite, or garbage-disposal operations per correction cycle at temperature* T*, then under Landauer bridge* *assumptions,*

$$P_{\text{erase}} \geq \frac{k_B T \ln 2}{\tau_{\text{cyc}}} b_{\text{erase/cycle}}. \tag{25}$$

The total maintenance power is

$$\begin{aligned} P_{\text{maint}} = & P_{\text{erase}} + P_{\text{read}} + P_{\text{ctrl}} + P_{\text{dec}} \\ & + P_{\text{wire}} + P_{\text{leak}} + P_{\text{cool}} + P_{\text{parasitic}}. \end{aligned} \tag{26}$$

Remark 1. *The first term is a lower bound, not a full hardware power model. In practical hardware, control, readout, wiring, amplifiers, refrigeration overhead, leakage recovery, clocking, and routing may dominate the Landauer floor by many orders of magnitude. This paper uses Landauer's principle as an irreducible accounting anchor, not as a complete power prediction.*

Audit Principle 2 (Ledger relocation principle)**.** *If an architecture claims to evade a QEC maintenance bottleneck, it must identify where the relevant ledger has been relocated: colder or warmer reset, reversible buffering, external decoding, optical loss and multiplexing, engineered dissipation, topological gap, preparation overhead, readout overhead, latency, or increased failure probability.*

The logical distinction must remain protected under the same noise task. If one maintenance channel is removed, some other physical channel must supply the required protection, entropy export, or tolerated failure. Otherwise the no-free maintenance principle is violated. Thus relocation is possible, but unaccounted disappearance is not.

VII. SPECIAL CASE: SURFACE-CODE NORMAL FORMS

A. Logical error versus maintenance load

For a surface-code-like architecture, the normal forms

$$\begin{aligned} p_L(d,p) &\sim A\left(\frac{p}{p_{\text{th}}}\right)^{(d+1)/2}, \\ N_{\text{phys}} &\sim \alpha n_L d^2. \end{aligned} \tag{27}$$

show the key signature described here. For a target $p_L^\star$, the required distance obeys the same inverse relation $d_{\text{req}} \approx 2\ln(A/p_L^\star)/\ln(p_{\text{th}}/p) - 1$. Increasing d reduces logical error when $p < p_{\text{th}}$, but it increases physical carriers, syndrome load, routing, decoding, reset, and latency demands. Figure 2 visualizes this normal form.

B. Cryogenic solid-state reset ledger

For cryogenic solid-state architectures such as superconducting circuits and semiconductor spin qubits, a useful special-case ledger is the cold-stage reset fraction. Let χ_{reset} be the fraction of syndrome or auxiliary carriers whose logically irreversible reset is effectively paid at the cold stage. Then

$$b^{\text{cold}}_{\text{reset/cycle}} \sim \chi_{\text{reset}} \beta n_L d^2, \tag{28}$$

so that

$$P^{\text{cold}}_{\text{reset}} \gtrsim \frac{k_B T_{\text{cold}} \ln 2}{\tau_{\text{cyc}}} \chi_{\text{reset}} \beta n_L d^2. \tag{29}$$

The full cold-stage ledger is larger:

$$\begin{aligned} P^{\text{QEC}}_{\text{cold}} = & P^{\text{cold}}_{\text{reset}} + P^{\text{cold}}_{\text{readout}} + P^{\text{cold}}_{\text{control}} \\ & + P^{\text{cold}}_{\text{wiring}} + P_{\text{parasitic}} + \cdots . \end{aligned} \tag{30}$$

The feasibility condition is

$$P^{\text{QEC}}_{\text{cold}} \leq P^{\text{avail}}_{\text{cool}}(T_{\text{cold}}). \tag{31}$$

If the condition fails, the architecture must slow down, reduce load, export heat, increase cooling, change the control stack, adopt a different code, or leave the target fault-tolerant regime. Commercial dilution refrigerators publish finite cooling powers that depend strongly on model, wiring, load, and operating temperature; for example, a current Bluefors comparison table lists guaranteed 100 mK cooling powers ranging from $> 250\,\mu$W to $> 1000\,\mu$W across systems, and 20 mK powers from about $> 10\,\mu$W to $> 30\,\mu$W for listed dilution units [28]. This paper uses such numbers only as scaling-wall audit references, not as universal limits.

C. Scaling-wall audit

From the cold-stage reset expression of Eq. (29), a reference scaling-wall diagnostic for the maximum logical-qubit count under a given cooling budget is

$$n_L^{\max} \sim \frac{P^{\text{avail}}_{\text{cool}} \tau_{\text{cyc}}}{\eta\, \chi_{\text{reset}}\, \beta\, k_B T_{\text{cold}} \ln 2\, d^2}, \tag{32}$$

where η is an engineering amplification sensitivity factor capturing the ratio of real cold-stage power to the Landauer reset floor. This is a scaling-wall diagnostic, not a platform-independent maximum logical-qubit count. The audit identifies where the budget is exhausted; it does not predict that a specific hardware platform cannot exceed the reference estimate.

D. Ledger-relocation latency tax

Relocating the decoder or syndrome processing to a warmer stage can reduce cold-stage power but adds feedback-loop latency:

$$\tau_{\text{loop}} = \tau_{\text{meas}} + \tau_{\text{export}} + \tau_{\text{decode}} + \tau_{\text{feed}}. \tag{33}$$

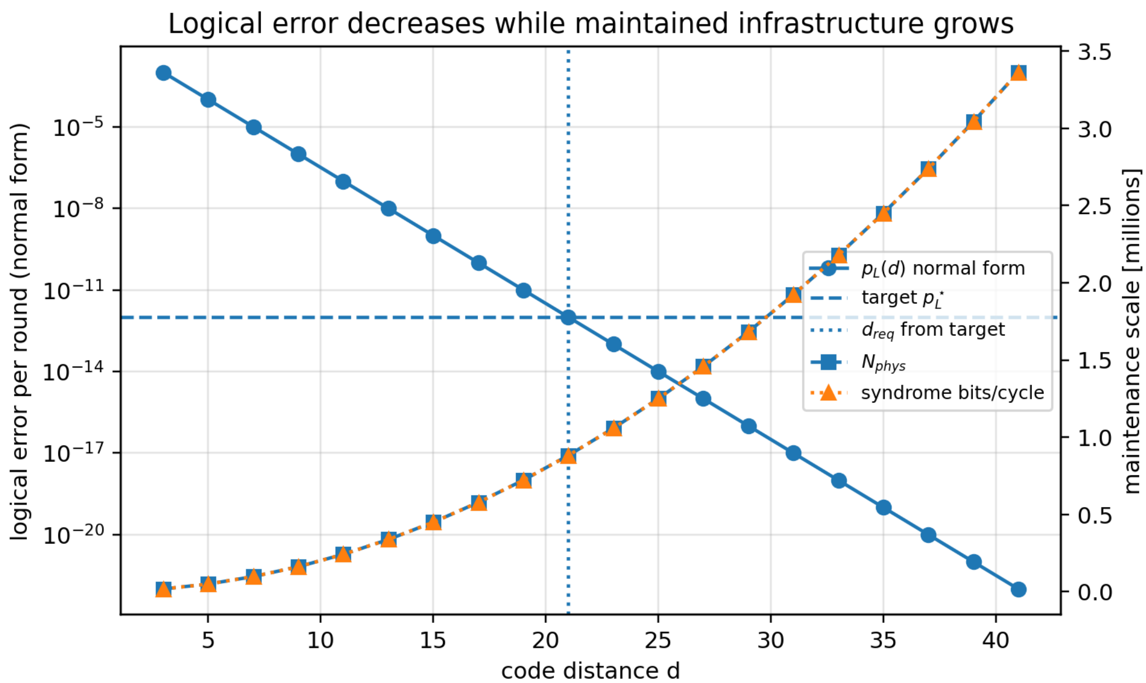

FIG. 2. Surface-code normal form. Logical error decreases exponentially in code distance when below threshold, while physical carriers and syndrome bits per cycle grow approximately as $n_L d^2$. The figure illustrates a scaling relation, not a hardware prediction.

The idle error accumulated during the loop is approximately

$$p_{\rm idle}(\tau_{\rm loop}) \approx 1 - e^{-\tau_{\rm loop}/T_2}, \tag{34}$$

so that increased loop latency raises effective physical error, requiring larger code distance and therefore more physical carriers, syndrome bandwidth, and maintenance power:

$$\tau_{\rm loop} \uparrow \Rightarrow \; p_{\rm eff} \uparrow \Rightarrow \; d_{\rm req} \uparrow \Rightarrow \; N_{\rm phys}, B_{\rm syn}, P_{\rm maint} \uparrow . \tag{35}$$

This nonlinear feedback is characteristic of ledger relocation: moving the ledger does not eliminate the maintenance burden; it transforms it into a different cost channel. The cold-stage and control-electronics ledger is also an active engineering topic: recent work on cryogenic control and surface-code energy-power scaling emphasizes that wiring, control electronics, temperature stages, and room-temperature dissipation must be included in resource estimates [37, 38].

E. Latency-density double bind

Spacing qubits farther apart can reduce heat density, crosstalk, and wiring congestion, but may increase routing delay:

$$\tau_{\rm cyc} \geq \tau_0 + \frac{L_{\rm route}}{v_{\rm sig}}. \tag{36}$$

A correction cycle must also fit within the relevant coherence and scheduling window. A conservative audit writes

$$\tau_{\rm cyc} \leq \zeta T_2, \tag{37}$$

with $0 < \zeta < 1$ a platform-dependent safety factor. Longer cycle time increases idle exposure,

$$p_{\rm eff} = p_{\rm gate} + p_{\rm meas} + p_{\rm idle}(\tau_{\rm cyc}), \tag{38}$$

which may require larger d, which increases $N_{\rm phys}$, syndrome load, wiring, and heat. Figure 4 shows this as a normal-form feedback rather than a universal hardware law.

VIII. ARCHITECTURE CLASSES

This paper must be architecture-aware. The same finite-maintenance principle applies broadly, but the dominant ledger differs.

A. Cryogenic solid-state processors

This paper's strongest special-case claim concerns cryogenic solid-state processors, where useful qubits, readout, control, and reset may place nontrivial load on the cold stage. The claim is not that superconducting

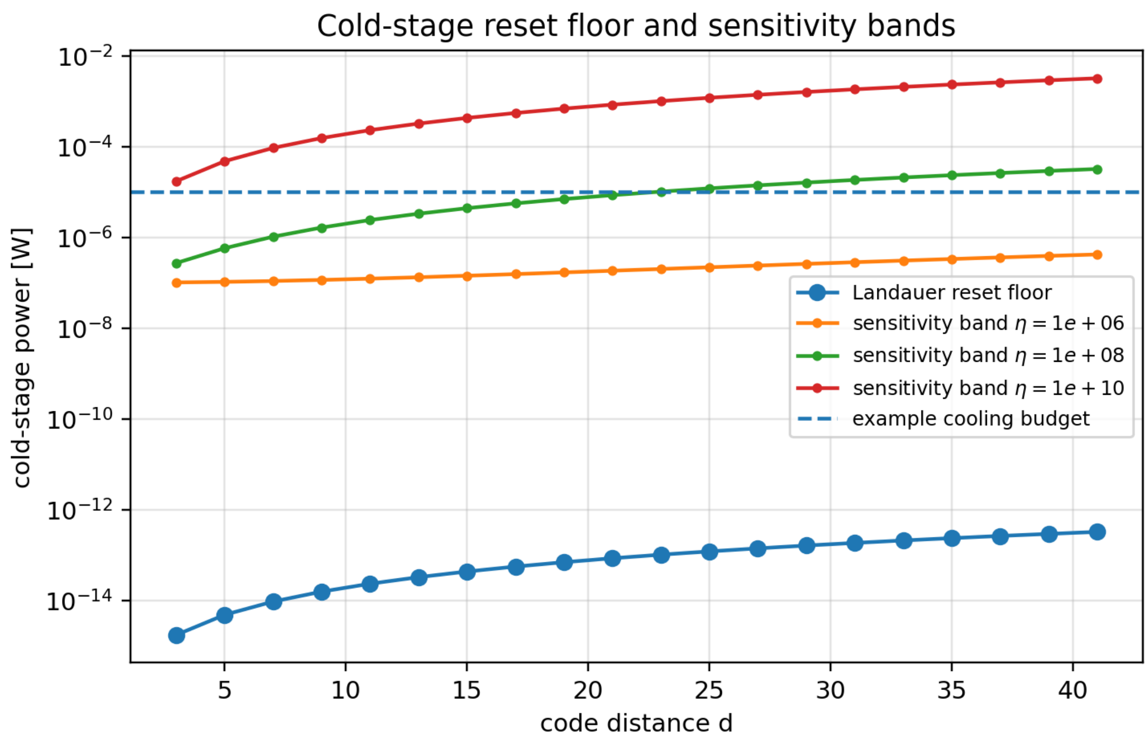

FIG. 3. Cold-stage reset floor and sensitivity bands. The Landauer reset floor is only the bottom curve. The additional curves multiply the floor by deliberately non-fitted sensitivity factors $\eta = 10^6, 10^8, 10^{10}$ and add a small parasitic term to show how real hardware ledgers can cross a cooling budget long before the formal information floor itself. The sensitivity bands are not claims about actual overhead factors for any specific platform. This is a conceptual normal form, not a refrigerator model.

TABLE II. Architecture audit template. This paper does not impose one bottleneck on all platforms; it tracks where maintenance cost is paid.

Architecture	Protected distinction	Main leakage or error	Maintenance channel	Dominant ledger
Superconducting	Stabilizer/logical sector	Decoherence, leakage, crosstalk	Syndrome readout, reset, feedback	Cold power, wiring, readout, latency
Spin qubit	Spin or encoded sector	Charge noise, hyperfine noise, readout error	Control, shuttling, readout, reset	Cryogenic control, routing, calibration
Trapped ion	Internal/motional modes	Motional heating, laser noise, transport error	Laser gates, sympathetic cooling, measurement	Speed, laser power, transport latency
Neutral atom	Hyperfine/Rydberg sector	Atom loss, motion, blockade errors	Rearrangement, gates, measurement	Loss replacement, laser power, parallelism
Photonic	Path/time-bin/cluster distinction	Loss, source failure, detector inefficiency	Multiplexing, feed-forward, detection	Source rate, detector load, latency
Topological	Nonlocal invariant sector	Finite-temperature anyons, poisoning, readout error	Gap protection plus residual correction	Preparation, gap, braiding/readout, residual QEC
Autonomous QEC	Stabilized bosonic/logical manifold	Reservoir mismatch, leakage, pump noise	Engineered dissipation	Pump power, entropy export, reservoir design

or spin-qubit fault tolerance is impossible. It is that any utility-scale architecture must satisfy a cold-stage correction ledger:

$$P_{\text{cold}}^{\text{QEC}} < P_{\text{cool}}^{\text{avail}}, \quad \tau_{\text{dec}} + \tau_{\text{feed}} < \tau_{\text{cyc}}, \quad p_{\text{eff}} < p_{\text{th}}. \quad (39)$$

B. Ions, neutral atoms, and photonics

Trapped ions and neutral atoms are not governed by the same millikelvin cooling bottleneck. Their ledgers include laser power, motional heating, transport, atom loss, rearrangement, parallelism, and gate speed. Photonic architectures relocate the ledger into loss, source brightness, detector efficiency, multiplexing, feed-forward, and cluster-state generation. A photonic system can avoid a cryogenic reset bottleneck while still facing finite distinction maintenance through photon loss and feed-forward latency.

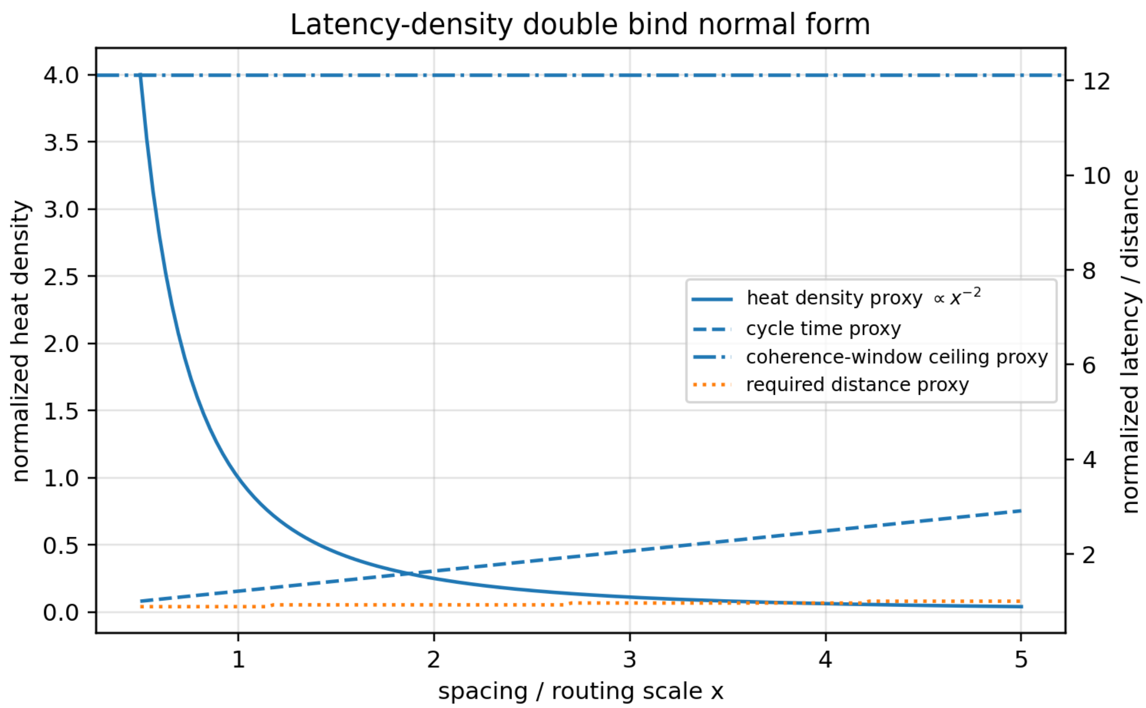

FIG. 4. Latency-density double-bind normal form. Increasing spacing reduces heat density but increases routing and cycle-time proxies. The dash-dot line marks an illustrative coherence-window ceiling proxy, not a measured hardware value for any specific platform.

C. Topological and passive protection

Topological protection reduces active correction load by encoding logical distinctions in nonlocal or invariant sectors [12, 13]. This is conceptually adjacent to FDS invariant persistence. However, topological protection does not make computation free. Preparation, finite-temperature stability, quasiparticle poisoning, braiding, measurement, decoding, and residual correction remain in the ledger.

D. Autonomous and dissipative QEC

Autonomous QEC replaces repeated digital syndrome measurement and reset with engineered dissipation or reservoir coupling [22–24]. Q2 treats this as a legitimate ledger relocation. The relevant cost is not necessarily $b_{\mathrm{reset}}/\tau_{\mathrm{cyc}}$, but

$$P_{\mathrm{auton}} = P_{\mathrm{pump}} + P_{\mathrm{reservoir}} + P_{\mathrm{stabilization}} + P_{\mathrm{leakage}} + P_{\mathrm{entropy\ export}}. \quad (40)$$

If the reservoir actively removes error entropy, the entropy leaves the logical subsystem through a different channel. That is a success for engineering, not a violation of this paper's audit. Recent autonomous and bosonic-code experiments show that explicit syndrome-measurement cycles can be reduced or replaced by engineered dissipation and protected manifolds; in this paper's language, this is ledger relocation rather than ledger elimination [33–35]. Low-overhead qLDPC and LDPC-cat approaches change the shape of the ledger by reducing qubit overhead or exploiting biased noise, but they still require explicit accounting of connectivity, syndrome extraction, logical gates, readout, and residual correction [32, 36].

Autonomous QEC relocates the ledger from digital syndrome extraction and feedback to engineered reservoirs. The hidden ledger components include pump purity, nonlinear drive stability, reservoir entropy export, and thermal noise suppression. Autonomous QEC is a ledger-relocation strategy, not a ledger-elimination strategy.

IX. RELATION TO Q1 AND OTHER FDS PAPERS

Q1 established that quantum records are boundary-indexed. It separated ideal global coherence from operational access and introduced record promotion, false promotion, and a record availability horizon [2]. This paper uses those diagnostics but shifts the focus from measurement records to logical quantum distinctions.

The relation is:

Q1 asks when a quantum interaction becomes an accessible record.
This paper asks what maintains a fragile logical quantum distinction before readout.

P6 supplies speed-precision-dissipation intuition; P7 supplies the idea that invariant or topological protection can reduce active forgetting; the FDS Core supplies capacity deficit, Landauer bridge, and maintenance exits [1, 3, 4].

X. NUMERICAL NORMAL-FORM DEMONSTRATIONS

The accompanying Python script in the code directory generates all figures, CSV tables, and model summaries. The outputs are deterministic normal forms, not empirical fits and not quantum dynamical simulations.

The model uses four demonstration blocks:

1. surface-code logical-error and load scaling;
2. cold-stage reset floor plus illustrative overhead relocation;
3. latency-density feedback;
4. architecture maintenance regimes.

XI. OPERATIONAL TESTS AND ENGINEERING DIAGNOSTICS

Protocol 1 (Syndrome reset ledger). *Measure* $b_{\mathrm{erase/cycle}}$, τ_{cyc}, *the effective reset temperature, and reset/export location. Compare irreversible reset or overwrite operations to*

$$P_{\mathrm{erase}} \geq k_{\mathrm{B}} T \ln 2\, b_{\mathrm{erase/cycle}}/\tau_{\mathrm{cyc}}. \tag{41}$$

Observable: reset-erasure floor and the difference between the floor and actual reset/readout power.

Protocol 2 (Logical error versus maintenance power). *Track* $p_L(d)$, $N_{\mathrm{phys}}(d)$, $B_{\mathrm{syn}}(d)$, *and* $P_{\mathrm{maint}}(d)$ *as code distance changes. This paper expects decreasing logical error to require growing or relocated maintenance infrastructure unless passive protection changes the scaling.*

Protocol 3 (Cooling-margin audit). *Define*

$$\mathcal{M}_{\mathrm{cool}} = P_{\mathrm{cool}}^{\mathrm{avail}} - P_{\mathrm{cold}}^{\mathrm{QEC}}. \tag{42}$$

If $\mathcal{M}_{\mathrm{cool}} < 0$, *the same cold-stage correction regime cannot be sustained without relocation or redesign.*

Protocol 4 (Latency-induced error audit). *Measure the dependence of idle error on correction-cycle time:*

$$p_{\mathrm{idle}}(\tau_{\mathrm{cyc}}), \quad p_{\mathrm{eff}}, \quad d_{\mathrm{req}}(p_{\mathrm{eff}}, p_L^{\star}). \tag{43}$$

Observable: whether wiring or cooling changes increase latency enough to raise correction demand.

Protocol 5 (Record leakage and RAH audit). *Use the Q1 metric* τ_{RAH} *to identify when internal syndrome or logical records become externally available or environmentally objectified. Observable:* $I_2(Z_{\mathrm{internal}}; Z_{\mathrm{external}}(t))$ *versus time and architecture boundary.*

XII. MINIMUM REPORTING TEMPLATE AND SUPPORT PATTERNS

A Q2 audit report should specify at least

$$(n_L, d, N_{\mathrm{phys}}, p_{\mathrm{eff}}, p_L^{\star}, \tau_{\mathrm{cyc}}, B_{\mathrm{syn}}, B_{\mathrm{reset}}, P_{\mathrm{cold}}, P_{\mathrm{room}}, P_{\mathrm{total}}), \tag{44}$$

together with decoder location, reset location, cooling stage, latency bottleneck, and entropy-export path. These fields are not a complete resource estimate; they are the minimum needed to make the maintenance ledger auditable.

What would count as support for Q2? First, a family of devices may show p_L decreasing with d while B_{syn}, N_{phys}, and measured maintenance ledgers grow. Second, an architecture may successfully reduce a cold-stage term by moving correction work to photonic feed-forward, room-temperature decoding, or an autonomous reservoir, while a corresponding source, detector, reservoir, latency, or entropy-export ledger becomes measurable. Third, a topological or passive design may reduce active syndrome burden while making a gap, preparation, readout, or finite-temperature ledger explicit. These are positive Q2 signatures: the ledger is not denied; it is measured or relocated.

XIII. FALSIFICATION AND DEMOTION CONDITIONS

This paper is weakened or demoted if a large-scale architecture demonstrates sustained logical-error suppression while all correction, reset, cooling, routing, latency, and entropy-export ledgers remain bounded within the stated resources and no relocation, passive-protection, or task-restriction account is needed; if irreversible reset, overwrite, or garbage disposal is reliably implemented below the Landauer floor under the bridge assumptions; if, for a fixed useful computational task family with growing logical volume, logical error is suppressed below target while all relevant maintenance components remain bounded without relocation or passive protection; if topological or passive protection eliminates not merely reduces active maintenance, preparation, readout, and finite-temperature costs; or if the finite-maintenance ledger has no predictive relation to hardware scaling.

Architecture-specific demotion should not be overgeneralized. A failure of the cryogenic solid-state subclaim does not falsify photonic or trapped-ion ledgers. A successful autonomous-QEC architecture does not falsify this paper if it identifies an engineered reservoir that ex-

TABLE III. Q1-to-Q2 diagnostic translation. Q2 keeps coherence, record formation, leakage, and reset history separate.

Variable or diagnostic	Quantum-computing meaning
T_1, T_2	Physical coherence metrics for carriers, not full logical maintenance guarantees.
τ_{rec}	Time until syndrome or readout record stabilizes to error tolerance.
τ_{RAH}	Time until internal syndrome/logical information becomes externally available or environmentally objectified.
C_{rec}	Stable distinguishable readout capacity of the measurement chain.
$I_2(Z_{\mathrm{internal}}; Z_{\mathrm{external}})$	Cross-boundary leakage or export of syndrome/logical record information.
b_{erase}	Irreversible reset, overwrite, or decoder-garbage burden per cycle.
P_{QEC}	Total finite-maintenance power, including reset, readout, control, decoding, routing, and cooling.

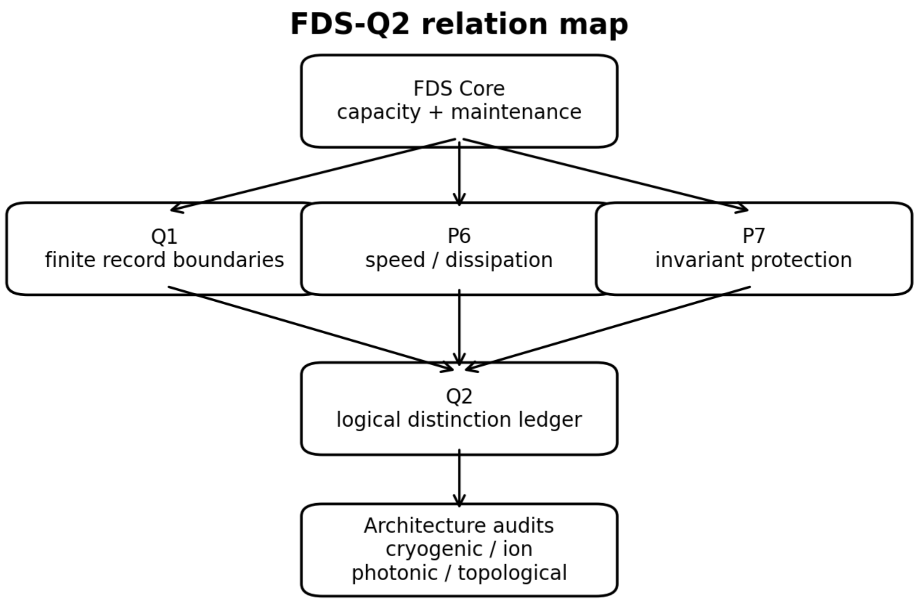

FIG. 5. FDS-Q2 relation map. Q2 inherits finite capacity and maintenance logic from the FDS Core, record-boundary discipline from Q1, speed/precision/dissipation constraints from P6, and invariant-protection language from P7.

ports entropy and maintains the logical quotient. Failure of this paper would demote this quantum-computing bridge; it would not falsify quantum mechanics, QEC theory, Q1, or the FDS formal core.

XIV. LIMITATIONS

This paper is not a full hardware resource-estimation framework. It does not include device-specific microwave control models, cryostat engineering, exact decoder complexity, fabrication yield, packaging, crosstalk, leakage models, or correlated-noise modeling. The numerical figures are normal forms. A platform-specific Q2 audit must specify the architecture, code, noise model, temperature stages, reset mechanism, decoder placement, wiring topology, and cooling curve.

This paper also does not claim that all correction information must be erased at the cold stage. Reversible buffering, higher-temperature processing, optical export, autonomous reservoirs, and architectural redesign can relocate cost. This paper's requirement is that the relocation be explicit.

XV. CONCLUSION

FDS-Q2 frames fault-tolerant quantum computation as finite distinction maintenance. Logical qubits are protected quantum quotients, not free abstract alternatives. Error correction suppresses logical error by increasing, relocating, or reorganizing the physical ledger that maintains those distinctions. The threshold theorem describes how logical error can fall under suitable assumptions; Q2 audits the finite physical cost of instantiating those assumptions at scale.

The core result is not a universal no-go theorem. It is

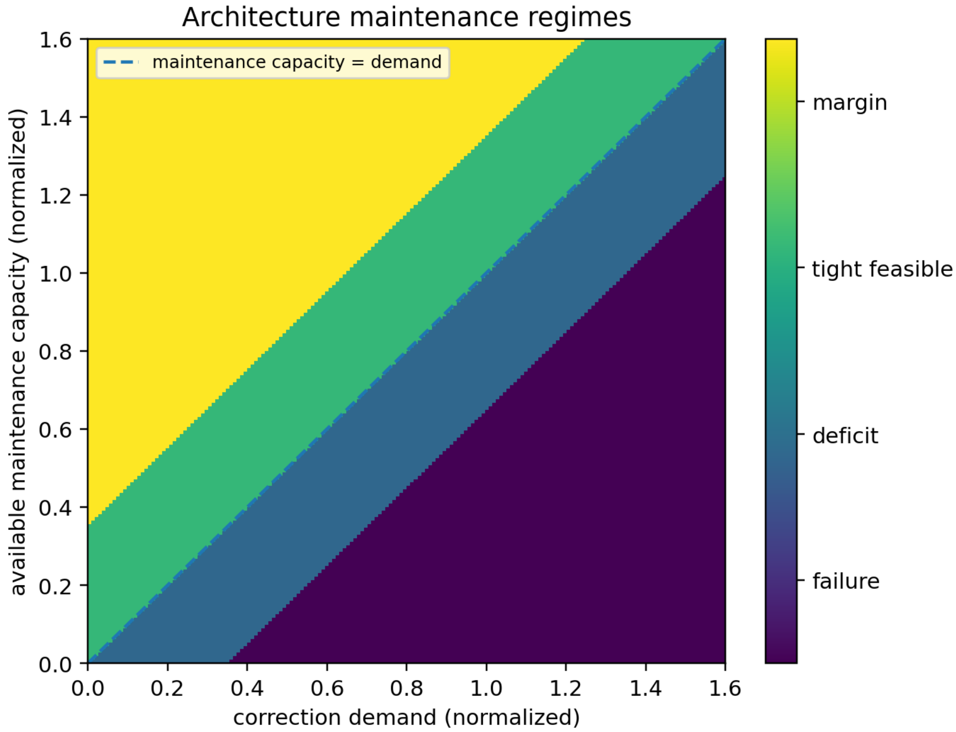

FIG. 6. Architecture maintenance regime map. The diagonal line marks equality of normalized correction demand and maintenance capacity. Below it, the architecture enters deficit or failure regimes unless the ledger is reduced, relocated, or the task is changed.

TABLE IV. Minimum Q2 reporting template for architecture audits.

Field	Unit or domain	Purpose
$n_L, d, N_{\rm phys}$	counts	logical scale, code distance, and maintained physical carriers.
$p_{\rm eff}, p_L^\star$	probabilities	effective physical error and target logical error.
$\tau_{\rm cyc}$	seconds	correction-cycle time relative to coherence and scheduling windows.
$B_{\rm syn}, B_{\rm reset}$	bits/cycle or bits/s	syndrome acquisition and irreversible reset or garbage-disposal burden.
$P_{\rm cold}, P_{\rm room}, P_{\rm total}$	watts by stage	where heat and entropy-export costs are actually paid.
decoder/reset locations	architecture labels	prevents hiding cost by moving it across the boundary.

a ledger discipline:

$$\boxed{\mathcal{R}^{(\tau)}_{\rm corr}(\epsilon_L) \preceq \mathcal{C}^{(\tau)}_{\rm maint}, \quad P_{\rm maint} \le P_{\rm avail}.} \tag{45}$$

When this fails, the architecture must slow down, relocate cost, increase resources, change code, change hardware, relax the computational task, or lose fault tolerance. Quantum computation remains possible; scalable quantum computation must be physically maintained.

CODE AVAILABILITY

The Python script `code/generate_results.py` regenerates all figures, CSV tables, and the model summary JSON included with this release.

AI ASSISTANCE DISCLOSURE

AI-assisted tools were used for language polishing, structural feedback, LaTeX drafting support, and code-debugging assistance. The author reviewed and edited all content and remains responsible for all claims, references, simulations, and conclusions. No AI system is listed as an author.

[1] Y. Wu, "Active Finite Distinction Systems: A Formal Core for Boundary Maintenance under Finite Capacity," Zenodo (2026), doi:10.5281/zenodo.20158923.

[2] Y. Wu, "Finite Record Boundaries in Wigner's Friend Scenarios: Observer-Relative Distinguishability and Quantum Record Availability in Finite Distinction Systems," Zenodo (2026), doi:10.5281/zenodo.20289215.

[3] Y. Wu, "Speed, Precision, and Dissipation Bounds for Boundary Maintenance in Finite Distinction Systems," Zenodo (2026), doi:10.5281/zenodo.20269733.

[4] Y. Wu, "Topological Obstruction to Forgetting in Finite Distinction Systems," Zenodo (2026), doi:10.5281/zenodo.20265386.

[5] R. Landauer, "Irreversibility and heat generation in the computing process," IBM J. Res. Dev. **5**, 183-191 (1961), doi:10.1147/rd.53.0183.

[6] C. H. Bennett, "The thermodynamics of computation - a review," Int. J. Theor. Phys. **21**, 905-940 (1982), doi:10.1007/BF02084158.

[7] M. B. Plenio and V. Vitelli, "The physics of forgetting: Landauer's erasure principle and information theory," Contemp. Phys. **42**, 25-60 (2001), doi:10.1080/00107510010018916.

[8] D. Aharonov and M. Ben-Or, "Fault-tolerant quantum computation with constant error," in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, 176-188 (1997), doi:10.1145/258533.258579.

[9] J. Preskill, "Reliable quantum computers," Proc. R. Soc. Lond. A **454**, 385-410 (1998), doi:10.1098/rspa.1998.0167.

[10] E. Knill, R. Laflamme, and W. H. Zurek, "Resilient quantum computation," Science **279**, 342-345 (1998), doi:10.1126/science.279.5349.342.

[11] P. Aliferis, D. Gottesman, and J. Preskill, "Quantum accuracy threshold for concatenated distance-3 codes," Quantum Inf. Comput. **6**, 97-165 (2006), arXiv:quant-ph/0504218.

[12] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," Ann. Phys. **303**, 2-30 (2003), doi:10.1016/S0003-4916(02)00018-0.

[13] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," J. Math. Phys. **43**, 4452-4505 (2002), doi:10.1063/1.1499754.

[14] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," Phys. Rev. A **86**, 032324 (2012), doi:10.1103/PhysRevA.86.032324.

[15] S. J. Devitt, W. J. Munro, and K. Nemoto, "Quantum error correction for beginners," Rep. Prog. Phys. **76**, 076001 (2013), doi:10.1088/0034-4885/76/7/076001.

[16] B. M. Terhal, "Quantum error correction for quantum memories," Rev. Mod. Phys. **87**, 307-346 (2015), doi:10.1103/RevModPhys.87.307.

[17] D. A. Lidar and T. A. Brun, eds., *Quantum Error Correction* (Cambridge University Press, 2013), doi:10.1017/CBO9781139034807.

[18] C. Gidney and M. Ekerå, "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits," Quantum **5**, 433 (2021), doi:10.22331/q-2021-04-15-433.

[19] C. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter, "Surface code quantum computing by lattice surgery," New J. Phys. **14**, 123011 (2012), doi:10.1088/1367-2630/14/12/123011.

[20] A. Blais, A. L. Grimsmo, S. M. Girvin, and A. Wallraff, "Circuit quantum electrodynamics," Rev. Mod. Phys. **93**, 025005 (2021), doi:10.1103/RevModPhys.93.025005.

[21] D. Ristè, M. Dukalski, C. A. Watson, G. de Lange, M. J. Tiggelman, Y. M. Blanter, K. W. Lehnert, R. N. Schouten, and L. DiCarlo, "Deterministic entanglement of superconducting qubits by parity measurement and feedback," Nature **502**, 350-354 (2013), doi:10.1038/nature12513.

[22] M. Mirrahimi, Z. Leghtas, V. V. Albert, S. Touzard, R. J. Schoelkopf, L. Jiang, and M. H. Devoret, "Dynamically protected cat-qubits: a new paradigm for universal quantum computation," New J. Phys. **16**, 045014 (2014), doi:10.1088/1367-2630/16/4/045014.

[23] Z. Leghtas *et al.*, "Confining the state of light to a quantum manifold by engineered two-photon loss," Science **347**, 853-857 (2015), doi:10.1126/science.aaa2085.

[24] N. Ofek *et al.*, "Extending the lifetime of a quantum bit with error correction in superconducting circuits," Nature **536**, 441-445 (2016), doi:10.1038/nature18949.

[25] R. Raussendorf and H. J. Briegel, "A one-way quantum computer," Phys. Rev. Lett. **86**, 5188-5191 (2001), doi:10.1103/PhysRevLett.86.5188.

[26] M. A. Nielsen, "Optical quantum computation using cluster states," Phys. Rev. Lett. **93**, 040503 (2004), doi:10.1103/PhysRevLett.93.040503.

[27] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest, "Measurement-based quantum computation," Nat. Phys. **5**, 19-26 (2009), doi:10.1038/nphys1157.

[28] Bluefors, "Dilution Refrigerator Measurement Systems: System Comparison," https://bluefors.com/products/dilution-refrigerator-measurement-systems/ (accessed May 2026). Manufacturer cooling powers are illustrative reference values, not platform-independent physical limits.

[29] H. M. Wiseman and G. J. Milburn, *Quantum Measurement and Control* (Cambridge University Press, 2010), doi:10.1017/CBO9780511813948.

[30] Google Quantum AI and Collaborators, "Quantum error correction below the surface code threshold," Nature **638**, 920–926 (2025), doi:10.1038/s41586-024-08449-y.

[31] J. Bausch, A. W. Senior, F. J. H. Heras, T. Edlich, A. Davies, M. Newman, C. Jones, K. Satzinger, M. Y. Niu, et al., "Learning high-accuracy error decoding for quantum processors," Nature **635**, 834–840 (2024), doi:10.1038/s41586-024-08148-8.

[32] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, "High-threshold and low-overhead fault-tolerant quantum memory," Nature **627**, 778–782 (2024), doi:10.1038/s41586-024-07107-7.

[33] H. Putterman, K. Noh, C. T. Hann, et al., "Hardware-efficient quantum error correction via concatenated bosonic qubits," Nature **638**, 927–934 (2025), doi:10.1038/s41586-025-08642-7.

[34] U. Reglade, A. Bocquet, R. Gautier, J. Cohen, A. Marquet, et al., "Quantum control of a cat qubit with bit-flip times exceeding ten seconds," Nature **629**, 778–783

(2024), doi:10.1038/s41586-024-07294-3.

[35] Z. Li, T. Roy, D. Rodriguez Perez, K.-H. Lee, E. Kapit, and D. I. Schuster, "Autonomous error correction of a single logical qubit using two transmons," Nat. Commun. **15**, 1681 (2024), doi:10.1038/s41467-024-45858-z.

[36] D. Ruiz, J. Guillaud, A. Leverrier, M. Mirrahimi, and C. Vuillot, "LDPC-cat codes for low-overhead quantum computing in 2D," Nat. Commun. **16**, 1040 (2025), doi:10.1038/s41467-025-56298-8.

[37] A. Potočnik, "How to scale the electronic control systems of a quantum computer," Nat. Electron. **8**, 3–4 (2025), doi:10.1038/s41928-024-01331-9.

[38] M. De Michielis and E. Ferraro, "Energy and power scaling in quantum computers based on rotated surface codes with silicon flip-flop qubits," EPJ Quantum Technology **12**, 64 (2025), doi:10.1140/epjqt/s40507-025-00351-4.