Resilience of quantum random access memory to generic noise

Connor T. Hann,[1,2] Gideon Lee,[1,2,3] S. M. Girvin,[1,2] and Liang Jiang[1,2,4]

[1] *Departments of Applied Physics and Physics, Yale University, New Haven, Connecticut 06520, USA*
[2] *Yale Quantum Institute, New Haven, CT 06520, USA*
[3] *Yale-NUS College, 16 College Avenue West, 138527, Singapore*
[4] *Pritzker School of Molecular Engineering, The University of Chicago, Chicago, Illinois 60637, USA*

Quantum random access memory (QRAM)—memory which stores classical data but allows queries to be performed in superposition—is required for the implementation of numerous quantum algorithms. While naive implementations of QRAM are highly susceptible to decoherence and hence not scalable, it has been argued that the bucket brigade QRAM architecture [Giovannetti *et al.*, Phys. Rev. Lett. 100 160501 (2008)] is highly resilient to noise, with the infidelity of a query scaling only logarithmically with the memory size. In prior analyses, however, this favorable scaling followed directly from the use of contrived noise models, thus leaving open the question of whether experimental implementations would actually enjoy the purported scaling advantage. In this work, we study the effects of decoherence on QRAM in full generality. Our main result is a proof that this favorable infidelity scaling holds for *arbitrary* error channels (including, e.g., depolarizing noise and coherent errors). Our proof identifies the origin of this noise resilience as the limited entanglement among the memory's components, and it also reveals that significant architectural simplifications can be made while preserving the noise resilience. We verify these results numerically using a novel classical algorithm for the efficient simulation of noisy QRAM circuits. Our findings indicate that QRAM can be implemented with existing hardware in realistically noisy devices, and that high-fidelity queries are possible without quantum error correction. Furthermore, we also prove that the benefits of the bucket-brigade architecture persist when quantum error correction is used, in which case the scheme offers improved hardware efficiency and resilience to logical errors.

I. INTRODUCTION

Numerous quantum algorithms have been proposed that claim speedups over their classical counterparts. Such algorithms typically require that classical data—constituting a classical description of the problem instance—be made available to a quantum processor. Frequently, theoretical constructions called oracles (or black boxes) are invoked to provide this access [1]. For example, oracles can provide quantum access to classical descriptions of Hamiltonians in quantum simulation algorithms [2–7], and they are used to encode classical datasets into quantum states in quantum machine learning algorithms [8–12]. In practice, however, providing quantum access to classical data can be nontrivial, and in order to claim a genuine quantum speedup it is crucial that the details of how such oracles are implemented be specified [13].

Quantum random access memory (QRAM) [14–19] is a general-purpose architecture for the implementation of quantum oracles. QRAM can be understood as a generalization of classical RAM; the classical addressing scheme in the latter is replaced by a quantum addressing scheme in the former. More precisely, in the case of classical RAM, an address i is provided as input, and the RAM returns the memory element x_i stored at that address. Analogously, in the case of QRAM, a quantum superposition of different addresses $|\psi_{\text{in}}\rangle$ is provided as input, and the QRAM returns an entangled state $|\psi_{\text{out}}\rangle$ where each address is correlated with the corresponding memory element,

$$|\psi_{\text{in}}\rangle = \sum_{i=0}^{N-1} \alpha_i \, |i\rangle^A \, |0\rangle^B$$
$$\xrightarrow{\text{QRAM}} |\psi_{\text{out}}\rangle = \sum_{i=0}^{N-1} \alpha_i \, |i\rangle^A \, |x_i\rangle^B \,, \tag{1}$$

where N is the size of the memory [20], and the superscripts A and B respectively denote the input and output qubit registers. (In this work, we restrict our attention to the case where the memory elements are classical, though in principle QRAM can also be used to query quantum data.) Remarkably, QRAM can perform operation (1) in only $O(\log N)$ time, albeit at the cost of $O(N)$ ancillary qubits. The short query time, together with the generality of operation (1), makes QRAM appealing for use in many quantum algorithms, especially those that require $O(\log N)$ query times in order to claim exponential speedups. Further, QRAM can serve as an oracle implementation in quantum algorithms for machine learning [8–12, 21–23], chemistry [7, 24], and a host of other areas [25–33].

The idea of QRAM has faced skepticism, however, and the question of whether QRAM can be used to facilitate quantum speedups, either in principle or in practice, has not been definitively settled (see, e.g., Refs. [13, 34], or the excellent summary in Ref. [12]). A central practical concern is the seemingly high susceptibility of QRAM to decoherence [14, 17]. As we discuss below, naive implementations of QRAM perform operation (1) with an infidelity that scales linearly with the size of the memory. Such implementations are not scalable. As the memory

size increases, the infidelity grows rapidly without quantum error correction, yet the overhead associated with error correction can quickly become prohibitive because all $O(N)$ ancillary qubits need to be corrected [18].

Refs. [14, 15] proposed the so-called "bucket-brigade" QRAM architecture as a potential solution to this decoherence problem, though this solution has also faced skepticism. Proponents argue that the bucket-brigade QRAM is highly resilient to noise, in that it can perform operation (1) with an infidelity that scales only polylogarithmically with the size of the memory. This favorable scaling could allow for high-fidelity queries of large memories without the need for quantum error correction, thereby mitigating the aforementioned scalability problem. This noise resilience, however, has only been derived for contrived noise models that place severe constraints on the quantum hardware [14, 15, 17], thus casting doubt on the viability of the bucket-brigade architecture. Indeed, while several proposals for experimental implementations of QRAM have been put forth [15, 16, 35–37], to our knowledge there has yet to be an experimental demonstration of even a small-scale QRAM [38]. Absent from this debate has been a fully general and rigorous analysis of how decoherence affects the bucket-brigade architecture.

In this work, we study the effects of generic noise on the bucket-brigade QRAM architecture. Our main result is that the architecture is far more resilient to noise than was previously thought (our main scaling results are summarized in Table I). We rigorously prove that the infidelity scales only *polylogarithmically* with the memory size even when all components are subject to arbitrary noise channels, and we verify this scaling numerically. Remarkably (and perhaps counter-intuitively), this scaling holds even for noise channels where the expected number of errors scales *linearly* with the memory size. Our analysis reveals that this remarkable noise resilience is a consequence of the limited entanglement among the memory's components. We leverage this result to show that significant architectural simplifications can be made to the bucket-brigade QRAM, and that so-called "hybrid" architectures [18, 19, 33, 39], which implement (1) with fewer qubits but longer query times, can also be made partially noise resilient. We also show that these benefits persist when quantum error correction is used. Importantly, the present work shows that a noise-resilient QRAM can be constructed from realistically noisy devices, paving the way for small-scale, near-term experimental demonstrations of QRAM.

This paper is organized as follows. In Section II, we give a detailed review of QRAM architectures, and an intuitive explanation for the noise-resilience of the bucket-brigade scheme is provided. Our main result is presented in Section III: we prove that the query infidelity of the bucket-brigade architecture scales only polylogarithmically when its components are subject to generic mixed-unitary error channels (the full proof for arbitrary error channels is given in Appendix D). Importantly, these proofs assume that *all* components of the QRAM (both active and inactive) are susceptible to decoherence, in contrast to prior works. The remaining sections provide corollaries, generalizations, and numerical demonstrations of this main result. In Section IV, we propose and implement an efficient classical algorithm for the simulation of noisy QRAM circuits, and we use this algorithm to confirm that the bucket-brigade QRAM is resilient to realistic errors. Next, in Section V, we show that the use of three-level memory elements in the original bucket-brigade architecture is superfluous and that the architecture can be significantly simplified (while maintaining noise resilience) by instead using two-level memory elements. In Section VI, we show that the bucket-brigade architecture can also be employed to imbue hybrid architectures with partial noise resilience. In Section VII, we prove that error-corrected implementations of the bucket-brigade architecture are resilient to logical errors, and we discuss the practical utility of error-corrected QRAM. Finally, in Section VIII we conclude by discussing potential applications.

Architecture	**Infidelity scaling**
Fanout QRAM (Sec. II)	$N \log N$
Standard BB QRAM (Secs. II,III)	$\log^2 N$
Two-level BB QRAM (Sec. V)	$\log^3 N$
Hybrid fanout (Sec. VI)	$N \log N + M \log^2 N$
Hybrid BB (Sec. VI)	$M \log^2 N$

TABLE I. Infidelity scalings of QRAM architectures. N denotes the size of the classical memory being queried, and bucket-brigade is abbreviated as BB. The first three architectures have circuit depth $O(\log N)$ and require $O(N)$ qubits. For the hybrid architectures, $M \leq N$ is a tunable parameter that determines the circuit depth, $O(M \log N)$, and the number of qubits, $O(N/M + \log N)$.

II. QUANTUM RANDOM ACCESS MEMORY

In both classical and quantum random accesses memories, each location in memory is indexed by a unique binary address. To read from the memory, an address is provided as input, and the memory element located at that address is returned at the output. In the classical case, transistors are the physical building blocks of the addressing scheme: they act as classical routers, directing electrical signals to the memory location specified by the address bits. Analogously, in the quantum case, quantum routers are the fundamental building blocks of the addressing scheme. As shown in Fig. 1(a), a quantum router is a device that directs incident signals along different paths in coherent superposition, conditioned on the state of a routing qubit. For example, if the routing qubit is in state $|0\rangle$ ($|1\rangle$), then a qubit incident on the

router is routed to the left (right). If the routing qubit is in a superposition, then the incident qubit is routed in both directions in superposition, becoming entangled with the routing qubit in the process. Quantum routers can also be understood through the language of quantum circuits [Fig. 1(b)]; the routing operation is a unitary that can be implemented via a sequence of controlled-SWAP gates (Fredkin gates).

In this section, we review two QRAM architectures based on quantum routers: the fanout architecture [40] and the bucket-brigade architecture [14, 15]. The fanout architecture is highly susceptible to noise, and we discuss it in order to illustrate how the lack of noise-resilience fundamentally limits QRAM scalability. The noise-resilience of the bucket-brigade architecture is the main focus of this work.

A. Fanout QRAM architecture

A QRAM can be constructed out of quantum routers as shown in Fig 1(c) (see Ref. [40], Chapter 6). A collection of routers is arranged in a binary tree, with the outputs of routers at one level of the tree acting as inputs to the routers at the next level down. The memory is located at the bottom of the tree, with each of the N memory cells connected to a router at the bottom level. To query the memory, all routing qubits are initialized in $|0\rangle$, and a register of $\log N$ address qubits is prepared in the desired state (in this work, all logarithms are base 2). All routing qubits at level ℓ of the tree are then flipped from $|0\rangle$ to $|1\rangle$ conditioned on the ℓ-th address qubit. To retrieve the memory contents, a so-called bus qubit is prepared in the state $|0\rangle$ and injected into the tree at the top node. The bus follows the path indicated by the routers down to the memory. Upon reaching a memory cell, the contents of that memory cell are copied into the state of the bus (see Appendix B for details). Note that because we consider *classical* data, the data can be copied without violating the no-cloning theorem. For simplicity, we assume that each memory element x_i is a single bit, in which case a single bus qubit suffices to store the memory element (higher-dimensional data can be retrieved using multiple bus qubits). Finally, the bus is routed back out of the tree via the same path, and all routers are flipped back to $|0\rangle$ in order to disentangle them from the rest of the system.

Importantly, because the routers operate coherently, the above procedure allows one to query multiple memory elements in superposition, as in (1). If the address qubits are prepared in a superposition of different computational basis states, the bus is routed to a superposition of different memory locations.

In this architecture, the total time required to perform a query (or, equivalently, the circuit depth) is only $O(\log N)$. The ability to perform queries in logarithmic time can be crucial for algorithms that invoke QRAM in order to claim exponential speedups over their classical counterparts. However, this speed comes at the price of a high hardware cost. To perform operation (1), both the fanout and bucket-bridgade architectures require $O(N)$ ancillary qubits to serve as routers. We discuss practical concerns associated with the high hardware cost more thoroughly in Sections VII and VIII. For context, we note that there is a space-time trade-off in implementing operation (1). At the other extreme, there are circuits which implement (1) in $O(N)$ time using only $O(\log N)$ qubits [5, 41, 42], and several implementations that leverage this trade-off have also been proposed [18, 19, 33, 39]. We discuss the effect of decoherence on these implementations in detail in Section VI.

The fanout architecture is impractical due to its high susceptibility to decoherence. Each address qubit is maximally entangled with all routers at the respective level of the tree (similar to a GHZ state), so the decoherence of any individual router is liable to ruin the query. As an example, suppose that the routers are subject to amplitude damping errors. The loss of an excitation from any router at level ℓ collapses all other level-ℓ routers and the ℓ-th address qubit to the $|1\rangle$ state. Any terms in the superposition where the ℓ-th address qubit was in the $|0\rangle$ state prior to the error are thus projected out, thereby reducing the fidelity by a factor of 2 on average.

More generally, suppose that each router suffers an error with probability ε at each time step during the query. The final state Ω of the full system (address, bus, and routers) can then be written as a statistical mixture

$$\Omega = (1-\varepsilon)^{T(N-1)}\,\Omega_{\text{ideal}} + \dots \tag{2}$$

where Ω_{ideal} is the error-free state, $T = O(\log N)$ is the number of time steps required to perform a query, and "..." denotes all states in the mixture where at least one of the $N-1$ routers has suffered an error. We define the *query fidelity* as

$$F = \langle\psi_{\text{out}}|\text{Tr}_R\left(\Omega\right)|\psi_{\text{out}}\rangle\,, \tag{3}$$

where Tr_R indicates the partial trace over the routers. The routers are traced out because only the address and bus registers are passed on to whatever algorithm has queried the QRAM; the routers are ancillae whose only purpose is to facilitate the implementation of operation (1).

As illustrated by the amplitude-damping example, the problem with the fanout implementation is that the no-error state Ω_{ideal} is generally the only state in the mixture (2) with high fidelity. Neglecting the low-fidelity states, the query infidelity scales as

$$1 - F \sim \varepsilon N T, \tag{4}$$

to leading order in ε. We refer to this linear scaling of the infidelity with the memory size as *unfavorable* because error probabilities $\varepsilon \ll 1/NT$ are required to perform queries with near-unit fidelity. This stringent requirement severely constrains the size of fanout QRAMs.

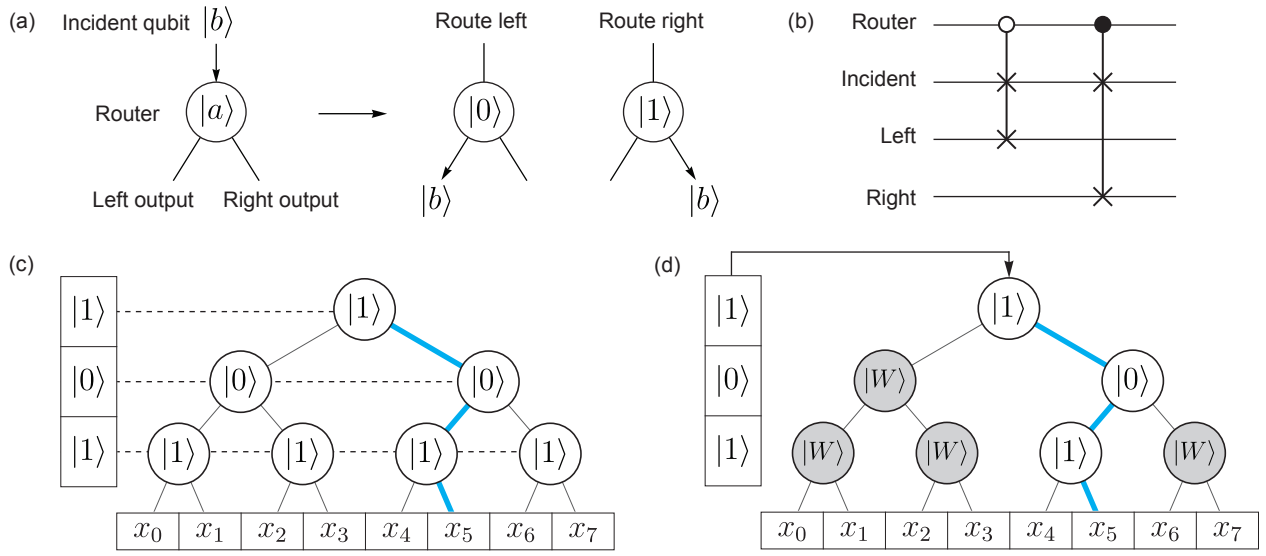

FIG. 1. QRAM implementations. (a) Quantum router. The router directs an incident qubit $|b\rangle$ at its top port out of either the left or right output ports conditioned on the state $|a\rangle$ of the router. When $|a\rangle = |0\rangle\,(|1\rangle)$, the incident qubit leaves out of the left (right) port. (b) Example of a quantum circuit that implements the routing operation using two controlled-SWAP gates, one conditioned on the control being $|0\rangle$ (open circle) and the other conditioned on the control being $|1\rangle$ (filled circle). (c) Fanout QRAM. Each address qubit controls the states of all routers within the corresponding level of the binary tree. A bus qubit injected at the top node then follows the path (blue) to the specified memory element. (d) Bucket-brigade QRAM, utilizing routers with three sates: wait $|W\rangle$, route left $|0\rangle$, and route right $|1\rangle$. The address qubits themselves are routed into the tree, carving out a path to the memory.

For example, error probabilities $\varepsilon \sim 10^{-3}$ would restrict the maximum size of a high-fidelity fanout QRAM to less than $N \sim 100$ memory cells. While quantum error correction can be used to suppress the error rates in principle, the additional hardware overhead can be prohibitive [18] because all $O(N)$ routers must be error corrected (see Section VII for a more detailed discussion of error correction and the associated overhead). Thus, because of its high susceptibility to decoherence, the fanout architecture is not regarded as scalable.

B. Bucket-brigade QRAM architecture

In Ref. [14], two modifications to the fanout architecture were proposed, and it was argued that the modified architecture, termed the "bucket-brigade" [Fig. 1(d)], is highly resilient to noise. The first modification is that the two-level routing qubits are replaced with three-level routing qutrits. In addition to the $|0\rangle$ (route left) and $|1\rangle$ (route right) states, each router also has a third state, $|W\rangle$ (wait). We refer to the states $|0\rangle, |1\rangle$ as *active*, and the state $|W\rangle$ as *inactive*. We assume that all routers are initialized in the $|W\rangle$ state, and that the action of the routing operation [Fig. 1(b)] is trivial when the routing qutrit is in the $|W\rangle$ state. (In Section V, we show that these assumptions may be relaxed, but we make them here for concreteness.) Each router's incident and output modes are also now taken to be physical three-level systems, and each address qubit is encoded within a two-level subspace of a physical three-level system.

We have borrowed the terminology of *active* and *inactive* routers from earlier works on QRAM [14, 17]. While these prior works assumed inactive routers to be free from decoherence, we stress that we make no such assumption here. As is discussed further in Section II C, we assume that routers are prone to decoherence regardless of whether they are active or inactive. In this work, we define the terms "active" and "inactive" only as labels for the different subspaces of a router's Hilbert space: active $\equiv \mathrm{span}(|0\rangle, |1\rangle)$ and inactive $\equiv \mathrm{span}(|W\rangle)$. We do not assume that either of these subspaces has any special properties (e.g., different decoherence rates).

The second modification is that the address qubits are themselves routed into the tree during a query. When an address qubit encounters a router in the $|0\rangle$ ($|1\rangle$) state, it is routed to the left (right) as usual. When an address qubit encounters a router in the $|W\rangle$ state, the states of the router and incident mode are swapped, so that the router's state becomes $|0\rangle$ ($|1\rangle$) when the incident address was $|0\rangle$ ($|1\rangle$). The physical implementation described in Ref. [14] provides a helpful example to visualize how these operations could be realized: the authors envisage the routers as three-level atomic systems, with the address qubits encoded in the polarization states of flying photons. (Note that the two polarization states constitute a two-level subspace of a physical three-level system, since the photonic mode may also be in the vacuum state.) When a photon encounters an atom in the $|W\rangle$ state, it is absorbed, and in the process it excites

the atom to the $|0\rangle$ or $|1\rangle$ state conditioned on its polarization. When subsequent photons encounter the excited atom, they are routed accordingly. These operations can also be described using the conventional quantum circuit model, and in Appendix A we provide a full circuit diagram.

To query the memory, the address qubits are sequentially injected into the tree at the root node. The first address qubit is absorbed by the router at the root node, exciting it from $|W\rangle$ to the $\{|0\rangle, |1\rangle\}$ subspace in the process. The second address qubit is routed left or right, conditioned on the state of the router at the root node. The state of the first address qubit thereby dictates the routing of the second. The second address is subsequently absorbed by one of the routers at the second level of the tree. The process is repeated, with the earlier addresses controlling the routing of later ones, carving out a path of active routers from the root node to the specified memory element. Once all address qubits have been routed into the tree, the bus qubit is routed down to the memory and the data is copied as before (see Appendix B). Finally, the bus and all address qubits are routed back out of the tree in reverse order to disentangle the routers. Here again, we emphasize that multiple memory elements can be queried in superposition, as in (1), because all routing operations are performed coherently.

C. Noise resilience: overview and conceptual explanation

The bucket-brigade architecture is clearly resilient to certain types of noise. For example, Ref. [17] studied the bucket-brigade QRAM with routers subject to $|0\rangle \leftrightarrow |1\rangle$ bit-flip errors, with the $|W\rangle$ states assumed to be error free. In this case, the expected number of errors is only $\varepsilon \log N$, because only the $\log N$ active routers are prone to errors [43]. The expected number of errors also scales with $\log N$ for the error model considered in Refs. [14–16], where gates involving inactive routers are assumed to be error free.

For these error models, the query infidelity is

$$1 - F \sim \varepsilon T \log^{\alpha} N. \tag{5}$$

to leading order in ε, where α is some constant, and we recall that $T = O(\log N)$ is the number of time steps. We refer to this logarithmic scaling of the infidelity with the memory size as *favorable* because queries can be performed with near-unit fidelity so long as the error rate satisfies $\varepsilon \ll 1/T \log^{\alpha} N$. This is a much more forgiving requirement; memories of exponentially larger size can be queried relative to the fanout architecture. Indeed, the exponential improvement in scalability suggests that quantum error correction is not required to query large memories with high fidelity, provided physical error rates are sufficiently low.

Unfortunately, the above error models can be poor approximations of the noise in actual quantum hardware. In these contrived models, inactive routers are assumed to be completely free from decoherence. More realistically, all routers will be prone to decoherence, independent of whether they are active or inactive. For example, though several proposals for experimental implementations of the bucket-brigade scheme have been put forth [15, 16, 35–37], none have proposed a method of engineering routers that are free from decoherence when inactive. While one can conceive of implementations in which inactive routers have decoherence rates which are nonzero but far smaller than those of active routers, it is not obvious whether such implementations would enjoy the favorable infidelity scaling. Indeed, Ref. [14] conjectured that decoherence of inactive routers could significantly increase the infidelity in this case, owing to the exponentially larger number of inactive routers. Furthermore, Refs. [14, 17] portray the favorable infidelity scaling as a direct consequence of the assumption that inactive routers are decoherence-free.

It is thus natural to ask whether the favorable scaling still holds when inactive routers are not assumed to be decoherence-free. Relaxing this assumption causes the expected number of errors to increase *exponentially*, from $O(\log N)$ to $O(N)$. Because the expected number of errors in the fanout architecture is also $O(N)$, one might naively expect that the favorable infidelity scaling no longer holds. However, in the next section we prove that this is not the case. Perhaps surprisingly, the infidelity of the bucket-brigade architecture still scales favorably despite the exponential increase in the expected number of errors. Moreover, the favorable scaling holds for *arbitrary* error channels.

The noise-resilience of the bucket-brigade architecture can be understood intuitively as a consequence of the minimal entanglement among the routers, see Fig. 2. Suppose one queries all memory locations in equal superposition. Then in both the fanout and bucket-brigade architectures, all of the routers are entangled. However, the degree to which each router is entangled with the rest of the system is quite different between the two architectures. This difference can be quantified by computing the entanglement entropy for a given router

$$S(\rho) = -\mathrm{Tr}\left[\rho \log \rho\right] \tag{6}$$

where ρ is the reduced density matrix of the router, obtained by tracing out the rest of the system. In the fanout architecture, each router is maximally entangled with the rest of the system; the reduced density matrix is the maximally mixed state $\rho = I/2$ (recall the fanout architecture employs two-level routers), for which $S(\rho) = 1$. In contrast, in the bucket-brigade architecture, the entanglement entropy of a router depends on its location within the tree. A router at level ℓ (0-indexed) of the tree is only active in $N2^{-\ell}$ of the N different branches of the superposition. As a result, the entanglement entropy decreases exponentially with depth, $S(\rho) \sim 2^{-\ell}$. Routers deeper down in the tree are nearly disentangled from the system, and their decoherence only reduces the query

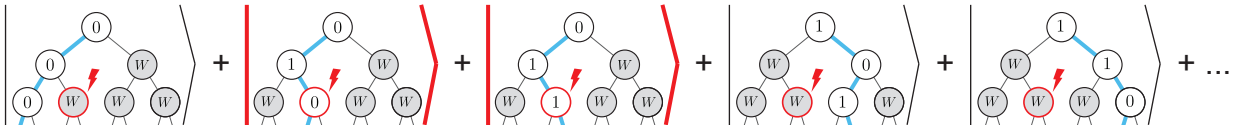

FIG. 2. Conceptual picture of noise resilience. Each ket represents the state of the QRAM when a different memory element is queried, with the superposition of kets representing a superposition of queries to different elements. When a router r suffers an error (red lightning bolt), it corrupts only the subset of queries where r is active (indicated by thick red kets); other queries in the superposition succeed regardless. Because most routers are only active in a small fraction of queries, most queries succeed and the total infidelity is low.

fidelity by an exponentially decreasing amount. Thus, despite the fact that exponentially many such errors typically occur, the overall fidelity can remain high. More precisely, if we posit that the infidelity associated with an error in a router at level ℓ scales as $\sim 2^{-\ell}$ due to the limited entanglement, and that $\varepsilon T\, 2^{\ell}$ such routers suffer errors on average, then the total infidelity scales as

$$1 - F \sim \sum_{\ell=1}^{\log N} \left(2^{-\ell}\right)\left(\varepsilon T 2^{\ell}\right) = \varepsilon T \log N. \tag{7}$$

The infidelity scales only logarithmically with N because the exponential increase in the expected number of errors with ℓ is precisely cancelled by the exponential decrease in the infidelity associated with each. We rigorously justify these claims in the next section.

III. PROOF OF NOISE RESILIENCE

In this section, we prove that the query infidelity of the bucket brigade architecture is upperbounded by

$$1 - F \leq A\varepsilon T \log N, \tag{8}$$

where $T = O(\log N)$ is the time required to perform a query, ε is the probability of error per time step, and A is a constant of order 1. This bound holds even when all N memory elements are queried in superposition, and it holds for arbitrary error channels, including, e.g., depolarizing errors and coherent errors. Moreover, throughout this paper we assume no special structure in the classical data x_i, so our bounds hold independent of the data.

Our proof is based on a careful analysis of how errors can propagate throughout the QRAM. Accordingly, we begin by defining our error model. We suppose that each routing qutrit is subject to an error channel in the form of a generic completely-positive trace-preserving map,

$$\rho \to \mathcal{E}(\rho) = \sum_i K_m \rho K_m^\dagger, \tag{9}$$

where the Kraus operators K_m obey the completeness relation $\sum_m K_m^\dagger K_m = I$. The error channel is applied simultaneously to all routers at discrete time steps throughout the query (see Eq. (15) below). In Appendix D, we prove that the bound (8) holds for arbitrary error channels of the form (9). For the sake of brevity and simplicity, however, here we restrict our attention to channels where (i) there is a no-error Kraus operator, K_0, that is proportional to the identity, and (ii) the remaining Kraus operators are proportional to unitaries, $K_m^\dagger K_m \propto I$. Under these restrictions,

$$\mathcal{E}(\rho) = (1-\varepsilon)\rho + \sum_{m>0} K_m \rho K_m^\dagger, \tag{10}$$

for some $\varepsilon \in [0,1]$. An operational interpretation of this channel is that one of the errors $K_{m>0}$ occurs with probability ε, and no error occurs with probability $1-\varepsilon$. Experimentally relevant examples include bit-flip, dephasing, and depolarizing channels. The restriction to this form of mixed-unitary channel allows us to make two assumptions that greatly simplify the proof: (i) the probability that an error occurs is independent of the router state, and (ii) the no-error backaction $K_0 \propto I$ is trivial. We make no further assumptions about the Kraus operators, and we stress that they may act non-trivially on the inactive state $|W\rangle$, meaning that inactive routers can decohere.

It is important to note that this error model only describes decoherence of the routing qutrits; a router's incident and output modes may also decohere, and there may be errors in the gates that implement the routing operation. At the end of this section, we prove that the bound (8) still holds when including these other errors, but we neglect them for now to simplify the discussion.

The proof proceeds by direct calculation. To bound the infidelity, we first write the final state Ω as a sum over different *error configurations*,

$$\Omega = \sum_c p(c)\Omega(c), \tag{11}$$

where an error configuration c specifies which Kraus operator is applied to each router at each time step. Here, $p(c)$ is the probability of configuration c, and the pure state $\Omega(c) = |\Omega(c)\rangle\langle\Omega(c)|$ is the corresponding final state of the system (both quantities are defined more formally below). The fidelity is thus given by,

$$F = \sum_c p(c)F(c), \tag{12}$$

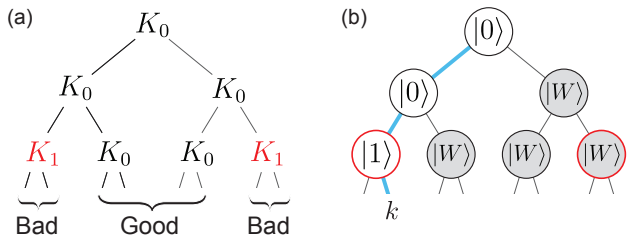

FIG. 3. Error configurations. (a) Example composite Kraus operator $K_{c(t)}$. The single-router Kraus operators $K_{c(r,t)}$ comprising the tensor product $K_{c(t)}$ are arranged geometrically according to the routers on which they act. Branches of the tree are classified as either good or bad according to the locations of the errors $K_{m>0}$. (b) Query to an element $k \notin g(c)$. Routers are labelled with their ideal, error-free states, and routers outlined in red suffer errors. Because one of the active routers suffers an error, the query is liable to fail.

where

$$F(c) = \langle\psi_{\text{out}}|\text{Tr}_R\Omega(c)|\psi_{\text{out}}\rangle \tag{13}$$

is the query fidelity of the state $\Omega(c)$. Our approach is to place an upper bound on the infidelity by deriving an upper bound on $1 - F(c)$.

Let us formally define $\Omega(c)$ and $p(c)$. A QRAM query consists of $O(N)$ routing operations [Fig. 1(b)] performed in a predetermined sequence. By design, many of these operations commute and can be performed in parallel, so that the entire operation can be written as a quantum circuit with depth $T = O(\log N)$ (see Appendix A for circuit diagram). More precisely, operation (1) can be written as,

$$|\psi_{\text{out}}\rangle\,|\mathcal{W}\rangle = U_T \ldots U_2 U_1\,|\psi_{\text{in}}\rangle\,|\mathcal{W}\rangle\,, \tag{14}$$

where $|\mathcal{W}\rangle = |W\rangle^{\otimes(N-1)}$ is the initial state of the routers, and U_t is a constant-depth circuit. Now, let $K_{c(r,t)}$ denote the Kraus operator applied to router r at time step t, and define the composite Kraus operator $K_{c(t)} = \bigotimes_{r=1}^{N-1} K_{c(r,t)}$ [see Fig. 3(a)]. The final state $|\Omega(c)\rangle$ is

$$|\Omega(c)\rangle = \frac{1}{\sqrt{p(c)}}\left[U_T K_{c(T)} \ldots U_1 K_{c(1)}\right]|\psi_{\text{in}}\rangle\,|\mathcal{W}\rangle\,, \tag{15}$$

The requirement that $|\Omega(c)\rangle$ is normalized defines the probability $p(c)$ of obtaining state $\Omega(c)$ in the mixture (11). Note that $\sum_c p(c) = 1$ follows from the Kraus operators' completeness relation.

For a given error configuration c, it is convenient to classify branches of the tree as either *good* or *bad*, depending on whether errors $K_{m>0}$ are ever applied to the routers in the branch [Fig. 3(a)]. More precisely, let $\mathbf{i}$ denote the set of all routers in the i-th branch of the tree (corresponding to address i), and let $\mathbf{c}$ denote the set of all routers which have an error $K_{m>0}$ applied to them at some time step. A branch i is defined to be good if $\mathbf{i} \cap \mathbf{c} = \varnothing$, and bad otherwise. To keep the notation simple, we use $g(c)$ to denote set of good branches. As illustrated in Fig. 3(b), queries to addresses $i \notin g(c)$ are liable to fail because they rely on routers that suffer errors.

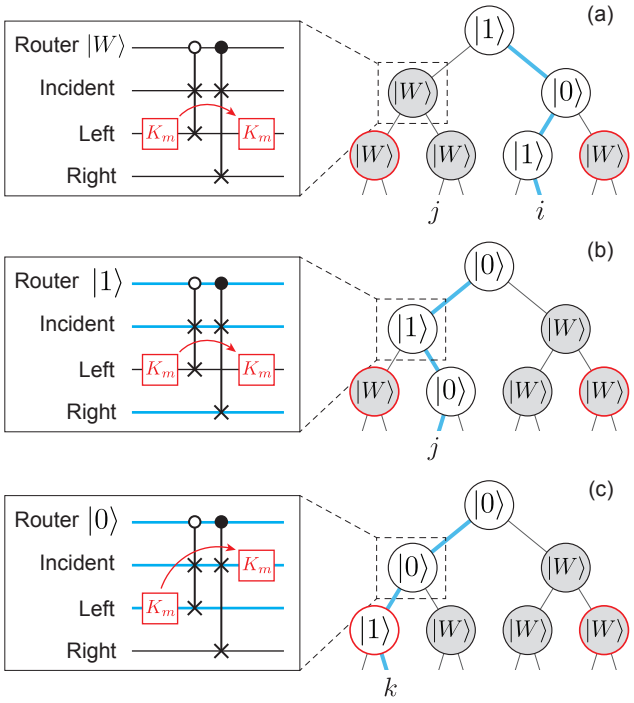

FIG. 4. Error propagation. (a,b) Constrained propagation during queries to elements $\in g(c)$. The error in the leftmost router can propagate upward into the left output of the router indicated by the dashed box. The circuits on the left show that the error does not propagate further, regardless of whether the router is inactive (a) or active (b). In the circuit diagrams, red boxes denote errors $K_{m>0}$, and the red arrows indicate how the error propagates (i.e. how the error transforms under conjugation by the routing operation). (c) Error propagation is not constrained during queries to elements $\notin g(c)$. Note that the state of the router dictates how the error propagates in these examples.

The main observation underlying our proof is that the propagation of errors is constrained when memory elements $\in g(c)$ are queried. Roughly speaking, errors do not propagate from bad branches into good branches. More precisely, for any $i, j \in g(c)$, errors do not propagate into branch j during a query to element i. We illustrate this fact with two examples, shown in Figs. 4(a,b). In general, errors in the bad branches can propagate. They can even propagate into an output mode of a router r in branch $j \in g(c)$, but they can never propagate into branch j. Fig. 4(a) shows an example of how such an error propagates through r's routing operation in the case where a memory element $i \neq j$ is queried. Because $j \in g(c)$, r's routing qutrit suffers no errors and is thus in $|W\rangle$. The action of the routing operation is trivial for a router in $|W\rangle$, so the error does not propa-

gate to other modes. (We reiterate that we are assuming error-free gates; gate errors are discussed at the end of this section.) Similarly, Fig. 4(b) shows an example of how errors propagate in the case where j is queried. The error-free routing qutrit is in $|1\rangle$, so the routing operation acts non-trivially on only the incident and right output modes. The error in the left output mode does not propagate upward. For comparison, in Fig. 4(c) we illustrate that the propagation of errors is not constrained in this way when memory elements $k \notin g(c)$ are queried. As an aside, we note that the constrained error propagation can be understood as a sort of error transparency [44–46]: when elements $\in$ g(c) are queried, the errors in the bad branches commute with the routing operations in the good branches.

The constrained propagation of errors has two important consequences. The first is that a query to memory element $i \in g(c)$ always succeeds, meaning that the address and bus registers are in the desired state $|i\rangle^A |x_i\rangle^B$ at the end of the query. This follows from the fact that errors cannot propagate to any of the routers in branch i. The second consequence is that, if multiple memory elements $i, j, \ldots \in g(c)$ are queried in superposition, the address and bus registers are disentangled from the routers at the end of the query. This follows from the fact that errors are restricted to propagate within the bad branches, and their propagation is unaffected by routers outside these branches. Figs. 4(a,b) provide an example. As a result, even though errors can propagate non-trivially among the bad branches during the query, the final state of the routers is independent of which memory element in $g(c)$ is queried.

It follows that the final state $|\Omega(c)\rangle$ can be written as

$$|\Omega(c)\rangle = |\text{good}(c)\rangle + |\text{bad}(c)\rangle\,, \tag{16}$$

with

$$|\text{good}(c)\rangle = \left(\sum_{i\in g(c)} \alpha_i\, |i\rangle^A |x_i\rangle^B\right) |f(c)\rangle^R\,. \tag{17}$$

Here, $|f(c)\rangle^R$ denotes the final state of the routers with respect to the good branches, and $|\text{bad}(c)\rangle$ contains the $i \notin g(c)$ terms. We now use the expression (16) to place a lower bound on $F(c)$. First notice that

$$F(c) \geq |\langle \psi_{\text{out}}, f(c)|\Omega(c)\rangle|^2\,, \tag{18}$$

which can be obtained by performing the partial trace in Eq. (13) using a basis that contains the state $|f(c)\rangle$ and neglecting the contributions from other states. Then, defining $\Lambda(c)$ as the weighted fraction of good branches,

$$\Lambda(c) = \langle \text{good}(c)|\text{good}(c)\rangle = \sum_{i\in g(c)} |\alpha_i|^2 \tag{19}$$

we have that

$$\langle \psi_{\text{out}}, f(c)|\text{good}(c)\rangle = \Lambda(c) \tag{20}$$

$$|\langle \psi_{\text{out}}, f(c)|\text{bad}(c)\rangle| \leq 1 - \Lambda(c). \tag{21}$$

To obtain the inequality (21) we have used the fact that $|\Omega(c)\rangle$ is normalized and that $\langle \text{good}(c)|\text{bad}(c)\rangle = 0$. The latter follows from the orthogonality of different initial address states, $\langle i|j\rangle^A = 0$ for $i \neq j$, and the fact that all subsequent operations, including the Kraus operators, are unitary and thus preserve inner products (this follows from our earlier restriction to mixed-unitary error channels; general channels are covered by the proof in Appendix D). Plugging Eqs. (16), (20) and (21) into the bound (18) and applying the reverse triangle inequality allows us to bound the infidelity as a function of $\Lambda(c)$,

$$F(c) \geq \begin{cases} (2\Lambda(c)-1)^2, & \Lambda(c) \geq 1/2, \\ 0, & \Lambda(c) < 1/2. \end{cases} \tag{22}$$

To proceed further, we compute the expected fraction of good branches, $\mathbb{E}(\Lambda)$, where the expectation value is taken with respect to the distribution of error configurations, i.e. $\mathbb{E}(f) = \sum_c p(c) f(c)$. This expectation value can be computed recursively for trees of increasing depth. Let $\mathbb{E}_d(\Lambda)$ denote the expected fraction of good branches for a depth-d tree. For a depth-1 tree, expected fraction is equivalent to the probability that the lone router never suffers an error, $\mathbb{E}_1(\Lambda) = (1-\varepsilon)^T$. For deeper trees, the expected fraction of error-free routers at each level is $(1-\varepsilon)^T$, so we have the recursive rule

$$\mathbb{E}_{d+1}(\Lambda) = (1-\varepsilon)^T \mathbb{E}_d(\Lambda). \tag{23}$$

Applying this rule to the initial condition $\mathbb{E}_1(\Lambda)$, we obtain

$$\mathbb{E}_{\log N}(\Lambda) = (1-\varepsilon)^{T \log N}. \tag{24}$$

We can now combine the above results to bound the infidelity. We have that

$$F = \mathbb{E}(F) \geq \mathbb{E}(\sqrt{F})^2 \tag{25}$$

$$\geq \left[2\mathbb{E}_{\log N}(\Lambda) - 1\right]^2 \tag{26}$$

$$= \left[2(1-\varepsilon)^{T\log N} - 1\right]^2, \tag{27}$$

where the second inequality follows from (22) under the assumption that $\mathbb{E}(\Lambda_{\log N}) \geq 1/2$. Applying Bernoulli's inequality yields the desired result,

$$1 - F \leq 4\varepsilon T \log N, \tag{28}$$

which holds for $\varepsilon T \log N \leq 1/4$. This bound is our main result, and we stress that it holds even when all N elements are queried in superposition, and that it was derived under the assumption that all routers are susceptible to decoherence, regardless of whether they are active or inactive.

We offer two additional remarks on the proof. First, we reiterate that while the above proof holds only for mixed-unitary error channels, in fact the favorable infidelity scaling holds for arbitrary error channels, which we prove in Appendix D. Second, the favorable scaling can

be interpreted as a consequence of the limited entanglement among the routers, as discussed in Section II. This limited entanglement manifests in Eqs. (16) and (17). The fact that a router at level ℓ is active in only $N2^{-\ell}$ of the N branches implies both that the router's entanglement entropy decreases exponentially with ℓ, and that only $N2^{-\ell}$ branches are corrupted when it suffers an error.

We conclude this section by describing four simple extensions of the proof that cover other cases of interest:

1. Initialization errors. Suppose that each router has some probability ε of not being initialized to $|W\rangle$ prior to the query. Such errors can be viewed as router errors of the form (9) that occur during the 0-th time step. As such, they are also covered by the proof provided one replaces $T \to T+1$ in the equations above. In Section V, we show that, in fact, one can make an even stronger statement: the infidelity scales favorably even when the QRAM is initialized in an arbitrary state.

2. Gate errors. Faulty implementation of the routing operation can be described without loss of generality as a composition $\mathcal{D}\circ\mathcal{R}$, of some error channel $\mathcal{D}$ followed by the ideal routing operation $\mathcal{R}$. Provided that $\mathcal{D}$'s Kraus operators are proportional to unitaries, and that there is a no-error Kraus operator proportional to the identity, then $\mathcal{D}$ can also be written in the form (10), and the proof proceeds as above. Note that the propagation of errors is still constrained in the case of gate errors because all routing gates in good branches are error-free by construction.

3. Alternate gate sets. We have defined the routing operation as a sequence of two controlled-SWAP gates [Fig. 1(b)], but this same operation could also be decomposed into other types of gates, e.g. into Toffolis, or Clifford + T gates. The bound (28) holds for any choice of gate decomposition. To see that the bound holds, consider that any error that propagates non-trivially through a given routing operation can be categorized as occurring either *before* or *during* that operation. The propagation of errors that occur before the operation is determined solely by the conjugation of the error with the entire routing operation (Fig. 4), which is unaffected by the choice of decomposition. In contrast, the propagation of errors that occur during the operation will generally depend on the choice of the decomposition. However, such errors can equivalently be described as a faulty implementation of the routing operation itself, so they do not spoil the favorable error scaling by the argument in the previous paragraph.

4. Correlated errors. The noise resilience also persists in the presence of correlated errors that afflict a constant number of adjacent routers in the tree. The proof assumes that if any error (correlated or otherwise) occurs in a branch, then that branch does not contribute to the fidelity. As such, whether an error afflicts only a given router r or also some of r's child routers lower in the tree is irrelevant to the proof. The effects of correlated errors can thus be incorporated simply by augmenting ε to also include the probability that a router is among those afflicted by a correlated error. For correlated errors afflicting only a constant number of adjacent routers, the resulting increase in ε is independent of N, so the query infidelity still scales only polylogarithmically with N.

IV. CLASSICAL SIMULATION OF NOISY QRAM CIRCUITS

In this section, we verify the bound (28) through numerical simulation of noisy QRAM circuits. While full state vector simulations require $\exp(N)$ memory and quickly become intractable as the QRAM size grows, our simulations are enabled by a novel classical algorithm with space and time complexity $\mathrm{poly}(N)$.

The main observation underlying the algorithm is that any quantum circuit consisting of the following elements can be simulated efficiently classically: state preparation in the computational basis, and gates from the set {SWAP, controlled-SWAP}. Such circuits are essentially classical—the system begins in a definite computational basis state, and the SWAP-type gates act only as permutations so that the system remains in a computational basis state through every step of the circuit. The simulation proceeds simply by tracking the (classical) state of the system. Furthermore, for initial states that are a superposition of polynomially-many different computational basis states, it follows from linearity that the action of any circuit composed of these SWAP-type gates can also be efficiently simulated. QRAM circuits can thus be efficiently simulated because they consist of SWAP-type gates acting on $O(N)$ qubits or qutrits, and the system is initialized in a superposition of only $O(N)$ computational basis states (one for each address). In fact, QRAM circuits are examples of so-called efficiently computable sparse (ECS) operations, whose efficient classical simulation is described in Ref. [47].

For context, we note that this approach is similar in spirit to the Gottesman-Knill theorem [48], which states that any Clifford circuit with preparation and measurement in the computational basis can be simulated classically in polynomial time. Because QRAM circuits necessarily employ non-Clifford gates (controlled-SWAP), however, the theorem does not directly apply. Still, the similarities are apparent: restricting the allowed gates and state preparations enables an efficient classical description of the system, making efficient simulation possible.

In addition, for a wide variety of error models, noisy QRAM circuits can be simulated efficiently using Monte Carlo methods. To simulate noisy circuits, the space of error configurations is randomly sampled according to the distribution $p(c)$. For each sampled configuration c from a set of samples S, we compute the final system state $|\Omega(c)\rangle$, and we obtain the fidelity by averaging $F = \frac{1}{|S|}\sum_{c\in S} F(c)$. This sampling procedure is efficient provided that two criteria are satisfied: first, that the

state $|\Omega(c)\rangle$ is efficiently computable, and second, that sampling from $p(c)$ is efficient. A sufficient condition for satisfying these two criteria is that the error channel maps computational basis states to other computational states, i.e., the channel's Kraus operators K_m satisfy

$$K_m |i\rangle \propto |i'\rangle , \tag{29}$$

for all m, and where $|i\rangle, |i'\rangle \in \{|0\rangle, |1\rangle, |W\rangle\}$ are computational basis states. The first criterion is satisfied because Eq. (29) guarantees that a QRAM circuit interspersed with applications of the Kraus operators K_m is still ECS. The second criterion is satisfied because the distribution $p(c)$ can be sampled efficiently by applying errors independently to each router (with appropriate probability) at each time step as the simulation proceeds. In detail, suppose that at time t the system is in a state $|\psi(t)\rangle$ that is a superposition of polynomially-many computational basis states,

$$|\psi(t)\rangle = \sum_{\{i_1,\ldots i_{N-1}\}\in C} \alpha_i |i_1, i_2, \ldots, i_{N-1}\rangle , \tag{30}$$

where $|i_r\rangle$ denotes the state of router r, and the cardinality of the set C is $O(\text{poly}N)$. The probability that a Kraus operator K_m is applied to router r is

$$\text{Tr}\left[K_m^\dagger K_m \rho_r\right] , \tag{31}$$

where $\rho_r(t) = \text{Tr}_{\bar{r}}(|\psi(t)\rangle\langle\psi(t)|)$ is the reduced density matrix of router r, with $\text{Tr}_{\bar{r}}$ denoting the partial trace over the rest of the system. Eq. (29) guarantees that this probability is efficiently computable, so sampling from the possible errors at time t is also efficient. This sampling procedure is repeated at each time step in order to sample from the full error configuration.

We apply this algorithm in order to compute the query infidelity for QRAM circuits with routers subject to a variety of noise channels. The results (Fig. 5) confirm that the QRAM query infidelity scales favorably in the presence of realistic noise channels acting on all of the memory's components. We stress that, for such channels, the expected number of errors generally scales linearly with N. Results for qutrit depolarizing, bit-flip, and dephasing channels are shown in panels (a), (b), and (c), respectively (see Appendix C for Kraus decompositions for each channel). These channels are all of the form (10), so the query fidelity is subject to the bound (28). The numerical results are all clearly consistent with this bound, and the expected $1 - F \propto \log^2 N$ scaling is evident on the log-log scale. In panels (d) and (e), we show numerical results for qutrit decay and heating channels (Appendix C). We find that the query fidelities for these channels also satisfy the bound (28). Note, however, that the decay and heating channels are not mixed-unitary channels, so the query fidelities are subject to the general bound derived in Appendix D, rather than Eq. (28).

V. NOISE RESILIENCE WITH TWO-LEVEL ROUTERS

In Section III, we proved that the query infidelity of the bucket-brigade QRAM scales favorably, even when inactive routers are subject to decoherence. It is thus natural to ask whether distinguishing between active and inactive routers is useful, and in fact whether the use of three-level routers is necessary in the first place. In this section, we show that the answer is no—the query infidelity still scales only polylogarithmically for QRAMs constructed from noisy two-level routers. As in Section III, the argument presented to justify this claim is based on a careful analysis of how errors propagate. Furthermore, we show that this same argument also reveals that noise resilience persists when the QRAM is initialized in an arbitrary state, and when the routing circuit [Fig. 1(b)] is modified. Taken together, the results in this section show that the noise resilience of the bucket-brigade scheme is a robust property that is insensitive to implementation details. They also show that existing experimental proposals [16, 35] employing two-level routers are noise-resilient.

Consider a QRAM constructed from routers with only two states: $|0\rangle$ (route left) and $|1\rangle$ (route right). Routers are thus always active. For concreteness, we suppose that the routing operation is implemented using the circuit in Fig. 1(b), and that all routers are initialized in $|0\rangle$, though these assumptions can be relaxed. Unfortunately, the proof from Section III cannot be directly applied to show that the query fidelity also scales favorably in this case. The proof fails in the case of two-level routers because the propagation of errors is no longer so highly constrained. Recall that in the case of three-level routers, errors do not propagate from bad branches into good branches. More precisely, for any $i, j \in g(c)$, errors do not propagate into branch j when branch i is queried. This is not the case for two-level routers: while errors do not propagate into branch i when branch i is queried, they can propagate into other branches j, as illustrated in Fig. 6. Because of this difference, when multiple memory elements $i, j, \ldots \in g(c)$ are queried in superposition, it is not guaranteed that the address and bus registers will be disentangled from the routers at the end of the query. Thus, Eqs. (16) and (17) no longer hold. Instead, the final state $|\Omega(c)\rangle$ is given by

$$|\Omega(c)\rangle = \sum_{i\in g(c)} \alpha_i |i\rangle^A |x_i\rangle^B |f_i(c)\rangle^R + |\text{bad}(c)\rangle , \tag{32}$$

where $|f_i(c)\rangle$ denotes the now address-dependent final state of the routers, and $|f_i(c)\rangle \neq |f_j(c)\rangle$ in general. As a result, the $i, j \in g(c)$ terms are no longer guaranteed to be in coherent superposition after tracing out the routers. Rather, the final state of the address-bus system is liable to contain an incoherent mixture of these terms. That is, the final density matrix can contain terms of the form $|i, x_i\rangle\langle i, x_i|$ and $|j, x_j\rangle\langle j, x_j|$ without $|i, x_i\rangle\langle j, x_j|$

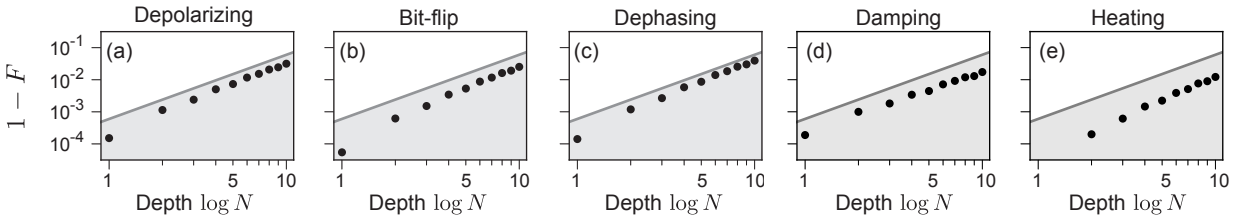

FIG. 5. Favorable error scaling. For a variety of error channels, the query infidelity (black dots) is calculated numerically and plotted as a function of the tree depth $\log N$ (note the logarithmic scaling on both axes). The region defined by the upper bound (28) is shown in gray in each plot. Plotted infidelities are averages over many randomly generated binary data sets $\{x_0, \ldots x_{N-1}\}$. Each such data set is generated by randomly choosing each x_i to be 0 or 1 with equal probability. Error bars are smaller than the dot size. The error rate for all plots is $\varepsilon = 10^{-4}$.

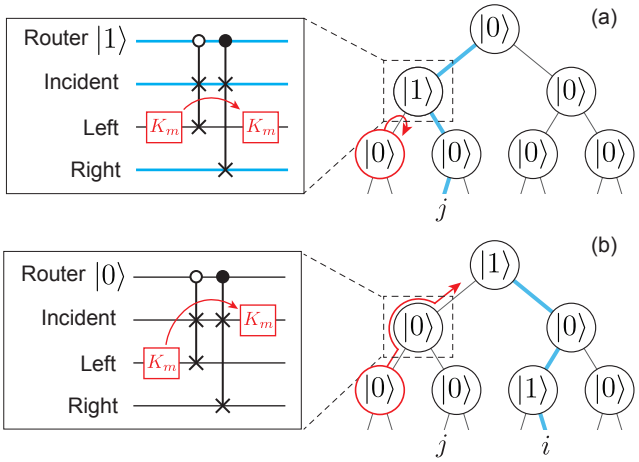

FIG. 6. Error propagation with two-level routers. (a) A query to memory element $j \in g(c)$, with an error $K_{m>0}$ applied to the red-outlined router. The circuit on the left shows how the error propagates through the router indicated by the dashed box. In this case, the error does not propagate into branch j. (b) A query to a different memory element $i \in g(c)$. In this case, the error propagates upward into branch j, in contrast to the situation in (a).

or $|j, x_j\rangle\langle i, x_i|$ terms. This loss of coherence reduces the fidelity.

We now proceed to estimate this reduction in fidelity. We find that the reduction is mild, such that the infidelity still scales only polylogarithmically with the memory size. Our approach is to isolate the subset of branches in $g(c)$ for which the sort of damaging error propagation described above does not occur. Explicitly, we define the subset $\tilde{g}(c) \subseteq g(c)$ as the largest subset such that for any $i, j \in \tilde{g}(c)$ errors do not propagate into branch j during a query to element i. We then have that $|f_i(c)\rangle = |f_j(c)\rangle$ by the same argument as given in Section III. It follows that, if multiple memory elements in $\tilde{g}(c)$ are queried in superposition, the address and bus registers will be disentangled from the routers at the end of the query.

Having defined $\tilde{g}(c)$ as the subset of good branches without damaging error propagation, we are free to define all other branches as bad and then proceed exactly as in Section III. In particular, we analogously define

$$\tilde{\Lambda}(c) = \sum_{i \in \tilde{g}(c)} |\alpha_i|^2 \tag{33}$$

as the weighted fraction of good branches, and

$$F \geq [2\mathbb{E}(\tilde{\Lambda}) - 1]^2, \tag{34}$$

follows as the analog of Eq. (26). Because $\tilde{g}(c) \subseteq g(c)$, we have that

$$\mathbb{E}(\tilde{\Lambda}) = (1 - \delta)\mathbb{E}(\Lambda), \tag{35}$$

for some $\delta \in [0, 1]$ to be determined. Proceeding as in Section III, it follows that the infidelity satisfies the bound

$$1 - F \leq 4\varepsilon T \log N + 4\delta \tag{36}$$

assuming $\varepsilon T \log N + \delta \leq 1/4$.

We can estimate δ by computing the average probability that errors propagate from bad branches into good branches. More specifically, we compute the probability that an error propagates into a branch $i \in g(c)$ when some other branch $j \in g(c)$ is queried. Suppose that a router r suffers an error at time step t, and let $P_{r \to i}(t)$ denote the probability of this error propagating into branch i. Then to leading order in ε,

$$\delta = \varepsilon \sum_{r,t} P_{r \to i}(t) + O(\varepsilon^2), \tag{37}$$

which can be understood as the total probability that an error occurs and propagates into branch i. To compute $\sum_{r,t} P_{r \to i}(t)$ to leading order, we observe that errors are generally free to propagate from a router's left output to its input, as illustrated in Fig. 6(b). This is because, by default, all routers are initialized in $|0\rangle$, for which the routing operation swaps the states at the incident and left ports. In contrast, for an error to propagate upward from a router's right output, an additional error would be required to flip the router from $|0\rangle$ to $|1\rangle$. Thus, only the errors which can reach branch i by propagating upward exclusively through the left outputs of routers contribute to $\sum_{r,t} P_{r \to i}(t)$ to leading order in ε. A conservative overestimate is thus obtained by first enumerating

all routers r that are connected to i through the left ports of other routers, then pessimistically taking $P_{r\to i}(t) = 1$ for each. There are at most $\log^2 N$ such routers, so

$$\delta \leq \varepsilon T \log^2 N + O(\epsilon^2). \tag{38}$$

Substituting this expression into Eq. (36), we obtain

$$1 - F \lesssim 4\varepsilon T \left(\log N + \log^2 N\right). \tag{39}$$

Here we use the symbol $\lesssim$ to contrast this bound with Eq. (28); we proved the bound (28) rigorously, while we have obtained Eq. (39) through a scaling argument. As such, it is appropriate to focus only on the scaling of Eq. (39). We see that the infidelity still scales only polylogarithmically with the memory size, indicating that a bucket-brigade QRAM constructed from noisy two-level routers also exhibits noise resilience. Note, however, that the infidelity here scales with $\log^3 N$ [recall $T = O(\log N)$], as opposed to $\log^2 N$ in the case of three-level routers. Both scalings are still favorable according to our definition, but the discrepancy indicates that three-level routers impart better noise resilience than two-level routers.

We simulate noisy QRAM circuits with two-level routers in order to verify this noise resilience. Simulation results are shown in Fig. 7. For all noise channels simulated, the query infidelity is observed to scale polylogarithmically with the memory size, as expected. Moreover, the observed scaling exponents are ≤ 3 in all cases, consistent with the pessimistic $1 - F \sim \log^3 N$ scaling given above.

It is interesting to note that two-level routers are more resilient to certain noise channels than others, as quantified by the observed differences in scaling exponents. For example, the infidelity under the dephasing channel is observed to scale approximately as $1 - F \sim \log^2 N$. This relatively mild scaling can be explained as follows. When the dephasing errors are propagated through the QRAM circuit, they may act non-trivially on the final state of the address and bus registers, but they act trivially on the final state of the routers (the all-$|0\rangle$ state). As a result, the final state of the routers is the same for every address: $|f_i(c)\rangle = |f_j(c)\rangle$ for all i, j. Hence, $\tilde{g}(c) = g(c)$, and the bound from Section III applies. For the other channels, $\tilde{g}(c) \neq g(c)$ in general, consistent with observed scaling exponents > 2. The case of amplitude damping is also interesting to consider: the expected number of errors for this channel is only $\varepsilon T \log N$ because only $\log N$ excitations are injected into the tree. Because $T = O(\log N)$, one expects the infidelity to scale with $\log^2 N$. The observed slope of 1.86 is somewhat smaller owing to the fact that, in our simulations, excitations are only susceptible to damping while they reside in the tree.

The scaling argument presented in this section also suffices to show that the noise resilience persists in two other interesting situations: when the QRAM is initialized in an arbitrary state, and when the routing circuit is modified. Regarding initialization, observe that the above argument is straightforwardly modified to cover the case where all routers are initialized in $|1\rangle$ rather than $|0\rangle$. Indeed, such an argument holds regardless of whether a given router is initialized in $|0\rangle$ or $|1\rangle$. It follows that the query infidelity scales favorably when the QRAM is initialized in an arbitrary state [49] (though some additional care must be taken when copying data to the bus—see Appendix B for details). This observation has great practical utility, as it means that QRAM can be constructed even from physical components that cannot reliably be initialized to a particular state.

Regarding modifications to the routing circuit, it is helpful to consider an example. In Ref. [35] a modified routing circuit was proposed in which one of the controlled-SWAP gates in Fig. 1(b) is replaced by a SWAP gate. This modification has nontrivial effects on how errors propagate. With the modified circuit, errors can propagate from bad branches into good branches even when three-level routers are used. However, this is the same sort of damaging error propagation as is illustrated in Fig. 6. Indeed, from the perspective of error propagation, the effect of this modification to the routing circuit is equivalent to replacing three-level routers with two-level routers. Accordingly, the argument above can be directly applied to show that the favorable scaling persists with the modified circuit. This example demonstrates that noise resilience is not a specific feature of the routing circuit [Fig. 1(b)].

Taken together, the results from this section demonstrate that the noise resilience of the bucket brigade architecture is a robust property that is insensitive to implementation details. This observation affords a great deal of freedom to experimentalists in deciding how the routers and routing operations could be implemented in practice.

VI. HYBRID ARCHITECTURES

The bucket-brigade architecture allows one to perform queries in $O(\log N)$ time using $O(N)$ qubits. This allocation of resources represents one extreme; at the other extreme are architectures [5, 41, 42] that perform queries in $O(N \log N)$ time using $O(\log N)$ qubits. In fact, there exists a family of architectures that interpolate between these two extremes to leverage this space-time tradeoff [18, 19, 33, 39]. We refer to these as *hybrid architectures*, and in this section we study their noise resilience. We find that hybrid architectures can be imbued with a partial noise resilience when they employ the bucket-brigade QRAM as a subroutine. As a result, these hybrid bucket-brigade architectures can have significantly higher query fidelities than other architectures that require the same resources.

One of the primary benefits of the bucket-brigade architecture is that queries can be performed in only $O(\log N)$ time. These fast query times are essential for algorithms that must rapidly load large classical data

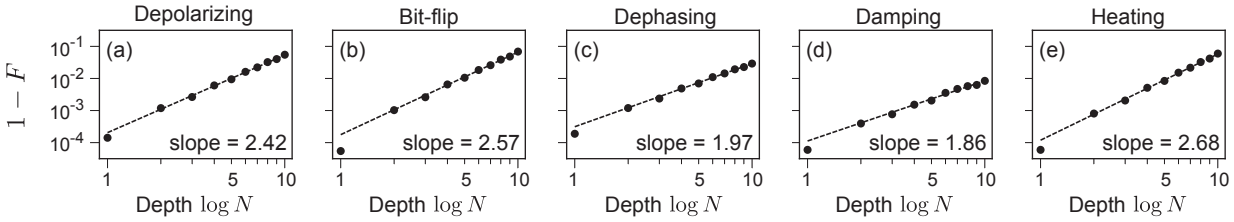

FIG. 7. Favorable error scaling with two-level routers. For a variety of error channels, the query infidelity (black dots) is calculated numerically and plotted as a function of the tree depth $\log N$. Linear fits for each data set are shown as dashed lines, with the corresponding slopes given on each plot. Fits are performed only on data points with $\log N \geq 3$ so that the slopes are not skewed by finite-size effects at small $\log N$. Slopes ≤ 3 are consistent with the scaling argument in the text. The error rate for all plots is $\varepsilon = 10^{-4}$.

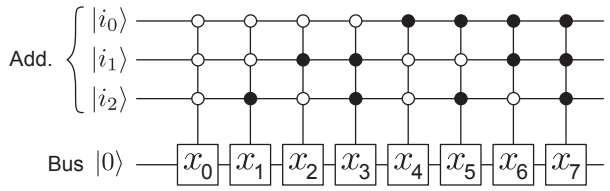

FIG. 8. QROM circuit. The circuit implements operation (1) by iterating over all N possible states of the address register. The j-th gate flips the state of the bus qubit if the address register (Add.) is in state $|j\rangle$ and $x_j = 1$, otherwise the gate acts trivially.

sets in order to claim exponential speedups over their classical counterparts, e.g., quantum machine learning algorithms [8–12]. However, the $O(N)$ hardware overhead that enables such fast queries is practically daunting, and not all algorithms require such fast queries in the first place. In algorithms that only require comparatively small data sets to be loaded, e.g. simulating local Hamiltonians [2–7], slower query times can be sufficient. Circuits that use fewer qubits at the price of longer query times are better suited for such algorithms.

Fig. 8 provides a straightforward example of such a circuit. To query a memory of size N, a sequence of N multiply-controlled Toffoli gates is applied, where each gate has $\log N$ controls (the address qubits) and one target (the bus qubit). The circuit sequentially iterates over all N possible addresses, flipping the bus qubit conditioned on the corresponding classical data. The circuit requires only $O(\log N)$ qubits, but it has depth $O(N \log N)$, since each multiply-controlled Toffoli gate can be performed in depth $O(\log N)$ [50]. Adopting the nomenclature introduced in Ref. [5], we refer to such circuits as Quantum Read-Only Memory (QROM). We note that this circuit can be further optimized to reduce the depth, as shown in Ref. [5]. We omit these optimizations for simplicity, as they do not affect our main conclusions concerning the effects of noise.

More generally, circuits can be constructed that trade longer query times for fewer qubits by combining QROM and QRAM, as shown in Fig. 9. We introduce a tunable parameter $M \leq N$, defined to be a power of 2. That is, $M = 2^m$, with m an integer in the interval $[0, \log N]$. The idea is to divide the full classical memory into M blocks, each with N/M entries. These blocks are queried one-by-one using a QRAM of size N/M concatenated with a QROM-like iteration scheme. The total hardware cost of the scheme is $O(\log N + N/M)$, comprising $O(\log N)$ qubits for the address and bus registers and $O(N/M)$ ancillary qubits for the QRAM. The total circuit depth is $O(M \log N)$ because each of the M iterations in the circuit can be performed in depth $O(\log N)$. Therefore, by tuning the parameter M, one can interpolate between large-width, small-depth circuits like QRAM, and small-width, large-depth circuits like QROM. The hybrid circuit reduces to QRAM for $M = 1$, and QROM for $M = N$. We note that the circuits we introduce in Fig. 9 are very similar to those in Refs. [18, 19, 33, 39]. The main difference is that our circuits explicitly invoke QRAM as a subroutine, which makes the analysis of their noise resilience more straightforward.

Let us consider the effects of noise on these circuits. We first consider QROM, then turn to the hybrid circuits. One can easily observe that QROM does not possess any intrinsic noise resilience. For example, when all memory elements are queried in equal superposition [$\alpha_i = 1/\sqrt{N}$ in Eq. (1)], a single dephasing error at any location in the QROM circuit reduces the query fidelity to 0. The effects of bit flips are similarly detrimental, assuming there is no contrived redundancy in the classical data. More generally, we can follow the approach of Section III and express the QROM query fidelity as $F = \sum_c p(c)F(c)$, where the error configuration c specifies which Kraus operators are applied at each location in the circuit, and $F(c)$ is the final state fidelity of the address and bus registers given configuration c. In the case of QROM, only the error configuration with no errors is guaranteed to have unit or near-unit fidelity in general [51]. There are $O(N \log^2 N)$ possible error locations, so it follows that the QROM query infidelity scales as

$$1 - F_{\text{QROM}} \sim \varepsilon N \log^2 N, \tag{40}$$

to leading order. Therefore, QROM is not noise resilient, since near-unit query fidelities generally require

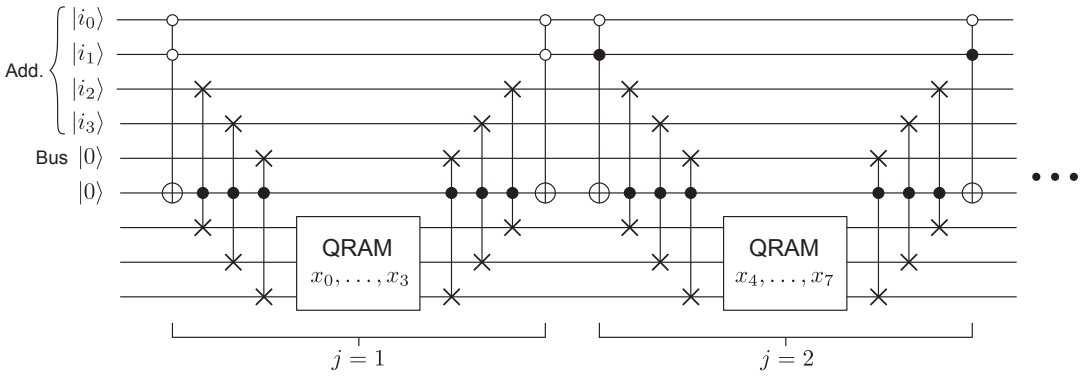

FIG. 9. Hybrid circuit. All $M = 2^m$ possible states of the first m address qubits are iterated over sequentially, as in QROM. Conditioned on these qubits, the remaining address qubits are used to query an (N/M)-cell classical memory via QRAM. In the circuit shown, $\log N = 4$ and $m = 2$. The boxes labelled QRAM implement (1), using either the fanout or bucket-brigade architecture. At the j-th iteration ($j \in [1, M]$), the data elements $\{x_{[(j-1)N/M]}, \ldots, x_{[j(N/M)-1]}\}$ are queried by the QRAM. Only the first two iterations are shown. The circuit depth is $O(M \log N)$, and the circuit uses $O(N/M + \log N)$ qubits, which includes the $O(N/M)$ ancillary qubits required by the QRAM (not shown). The initial state of the QRAM can be arbitrary (see Section V).

$\varepsilon \ll 1/N$, neglecting logarithmic factors.

Similarly, the hybrid circuits do not exhibit noise resilience when the QRAM subroutines are implemented with the fanout architecture. Recall from Section II that the fanout architecture is not noise resilient; only the fanout's no-error configuration is guaranteed to have high fidelity in general. Because neither QROM nor the fanout QRAM are noise resilient, only the no-error configuration of the hybrid fanout circuit is guaranteed to have high fidelity. Since the number of possible error locations is $O(M \log N(\log N + N/M))$, the query fidelity scales as

$$1 - F_{\text{hybrid,fanout}} \sim \varepsilon(N \log N + M \log^2 N), \tag{41}$$

to leading order. Here again, error rates $\varepsilon \ll 1/N$ are required for near-unit query fidelity, neglecting logarithmic factors.

In contrast, the hybrid circuits do exhibit partial noise resilience when the QRAM subroutines are implemented with the bucket-brigade architecture. Because the bucket-brigade QRAM is resilient to noise, error configurations with errors occurring exclusively in the QRAM subroutines can still have high fidelities. We can obtain a lower bound on the query fidelity by neglecting all other configurations. Doing so allows us to bound the query fidelity by a product of two factors

$$F_{\text{hybrid,BB}} \gtrsim (1-\varepsilon)^{O(M \log^2 N)} \times (1-\varepsilon)^{O(M \log N \log N/M)} \tag{42}$$

The first factor is simply the probability that no errors occur outside the QRAM. The second factor is the expected fraction of error-free branches within the QRAM (each branch contains $\log N/M$ routers, and there are $T = O(M \log N)$ possible time steps at which errors may occur). We have related this expected fraction to $F_{\text{hybrid-BB}}$ by the same argument as in Section III. Thus, to leading order,

$$1 - F_{\text{hybrid,BB}} \lesssim \varepsilon M \log N(\log N + \log N/M), \tag{43}$$

$$\sim \varepsilon M \log^2 N. \tag{44}$$

Note that we have not kept track of prefactors since we are only interested in how the infidelity scales; a strict upper bound could be rigorously derived following the approach of Section III. Near-unit query fidelities only require error rates $\varepsilon \ll 1/M$, neglecting logarithmic factors (cf. the $\varepsilon \ll 1/N$ requirement for the other cases). Because $M \leq N$, the infidelity of the hybrid bucket-brigade architecture scales more favorably than both QROM and the hybrid fanout architecture. Of course, the extent of the scaling advantage depends on M. For example, if one chooses $M = \sqrt{N}$, so that the number of qubits and circuit depth are comparable, then the hybrid bucket-brigade architecture yields a quadratic improvement in the infidelity scaling. Note that we assume three-level routers above for simplicity; for two-level routers, one should replace $\log N/M \to \log^2 N/M$ in the above expressions, in accordance with the argument from Section V.

VII. ERROR-CORRECTED QRAM

In this section, we show that the benefits of the bucket brigade scheme persist when quantum error correction is used. When the bucket-brigade QRAM is implemented using error-corrected routers and fault-tolerant routing operations [40, 52], the logical query infidelity scales only polylogarithmically with the memory size. Thus, error-corrected implementations of the bucket-brigade scheme can offer improved fidelity or reduced overhead relative to other implementations. In practice, these improvements may be tempered by the overhead associated with

the fault-tolerant implementation of the routing operations, and we discuss the utility of the bucket-brigade architecture in light of such considerations.

While we have shown that the query infidelity of the bucket-brigade scheme scales favorably with the memory size, strategies to further suppress the infidelity are desirable, and quantum error correction provides one possible approach. Indeed, error correction may be required in cases where the physical error rate cannot be made sufficiently small, or when many queries must be performed in sequence. For example, Ref. [17] argued that error correction is likely to be needed for any algorithm that requires a number of QRAM queries that scales superpolynomially in $\log N$, e.g., Grover's algorithm [25].

It is thus natural to ask whether an error-corrected bucket brigade QRAM offers any advantages over other architectures. Indeed, this question was previously considered in Ref. [17], where the authors argue in the negative. Their argument is based on the canonical attribution [10, 12, 14–17] of the bucket brigade's noise resilience to the limited number of active routers. Error-corrected routers must be considered active, they argue, and so the number of active routers is the same in both the fanout and bucket brigade schemes. Hence, the bucket brigade scheme was not believed to provide any advantage if error correction were used.

As we have shown, however, the noise resilience of the bucket brigade scheme is not a function of the number of active routers, but rather a function of the limited entanglement among the routers. As a direct corollary of this result, we find that, in fact, the benefits of the bucket brigade scheme do persist when error correction is used. The proof from Section III is agnostic to whether the routers are composed of uncorrected physical qubits or error-corrected logical qubits, provided that uncorrectable logical errors occur independently with some probability ε_L (which can be guaranteed by implementing the routing operations fault tolerantly). Physical errors occurring with probability ε can simply be replaced by logical errors occurring with probability ε_L, and one obtains the corresponding bound

$$1 - F_L \leq 4\varepsilon_L T_L \log N, \tag{45}$$

where F_L is the query fidelity of the logical QRAM circuit, and T_L is the circuit depth. Thus, when implemented fault-tolerantly, the logical bucket-brigade circuits possess an intrinsic resilience to logical errors, in that the logical infidelity scales only polylogarithmically with the the size of the memory (This scaling assumes $T_L = O(\log N)$; see further discussion at the end of this section).

To provide further exposition, we give a concrete example of an error-corrected quantum router. Consider a quantum error correcting-code, with logical codewords $|0_L\rangle$ and $|1_L\rangle$ satisfying the Knill-Laflamme conditions [40, 53],

$$PK_i^\dagger K_j P = h_{ij} P, \tag{46}$$

where P is the projector onto the code space, the $\{K_i\}$ are the set of correctable errors, and h is a Hermitian matrix. A logical two-level quantum router then constitutes a single logical qubit (similarly, a logical three-level quantum router can be constructed from a pair of logical qubits, for example). Crucially, the logical routers comprising the QRAM can be corrected without revealing any information about which memory elements are being accessed. This is because the conditions (46) guarantee that errors can be corrected without revealing any information about the encoded state. Even when the logical router is in a superposition of different states, or entangled with other routers, syndrome measurements do not reveal information about the router state. Note that the conditions (46) also guarantee that information is not leaked to the environment; the states $|0_L\rangle$ and $|1_L\rangle$ necessarily have equal probability of suffering errors.

Because of the favorable logical error scaling Eq. (45), error-corrected implementations of the bucket-brigade scheme can offer improved fidelity or reduced overhead relative to other implementations. For instance, if the same error-correcting code is used in fault-tolerant implementations of the bucket-brigade and fanout QRAMs, the logical infidelity of bucket brigade QRAM will be lower than the logical infidelity of the fanout QRAM by a factor of $\sim 1/N$ in general. Alternatively, if a given application requires that QRAM have a logical infidelity below some threshold, the error-correction overhead required to realize such high-fidelity queries can be significantly smaller for the bucket-brigade scheme relative to the fanout scheme. Indeed, even if the reduction in error-correction overhead is fairly small for each router, the total overhead reduction considering all N routers can be significant. Such reductions could be of significant practical benefit. For context, we note that detailed overhead estimates for fault-tolerant QRAM using the surface code were made in Ref. [18]; these overheads can potentially be improved by exploiting the bucket-brigade's noise resilience.

We conclude this section with an important caveat concerning fault-tolerant QRAM.

VIII. DISCUSSION

We have shown that the bucket-brigade QRAM architecture possesses a remarkable resilience to noise. Even when all $O(N)$ components comprising the QRAM are subject to arbitrary error channels, the query infidelity scales only polylogarithmically with the memory size. As a result, the bucket-brigade architecture can be used to perform high-fidelity queries of large memories without the need for quantum error correction, provided physical error rates are low. Importantly, we prove that this noise resilience holds for *arbitrary* error channels, demonstrating that a noise-resilient QRAM can be implemented with realistically noisy devices.

In the near-term, this noise resilience could facili-

tate experimental demonstrations and benchmarking of numerous quantum algorithms. We are presently in the Noisy, Intermediate-Scale Quantum (NISQ) era [60], when making more qubits is easier than making better qubits. The same is likely to be true even in the era of early fault-tolerance. In these eras, the bucket-brigade architecture—with its larger overhead and noise resilience—could actually prove to be more practical than alternatives like QROM (see Section VI) that have a lower overhead but are less tolerant to noise. The bucket-brigade architecture thus more readily enables small-scale, near-term implementations of algorithms, and important practical insights are likely to be gained from such demonstrations. Schemes to further suppress the query fidelity without resorting to full error correction [61] could prove useful in this effort.

In the long-term, this noise resilience may prove useful in facilitating speedups for certain quantum algorithms, but it is important that the required resources be carefully assessed before a speedup via QRAM is claimed. Consider an oracle-based algorithm that requires n qubits (not including ancillary qubits needed to implement the oracle). As we show in Table II, such algorithms can be conveniently classified according to how the size of the classical memory being queried, N, and the total number of queries, Q, scale with n. Assuming $\mathrm{poly}(n)$ query times are required, the memory size N dictates whether QRAM (as opposed to QROM or a hybrid architecture) is required to implement the oracle. The number of queries Q dictates whether error correction is necessarily required [17]. The noise resilience of the bucket-brigade has the biggest potential impact in case of $N = \exp(n)$ and $Q = \mathrm{poly}(n)$. In this case, QRAM is required, and the noise-resilience of the bucket-brigade architecture, together with the comparatively small number of queries, allows for the possibility that the QRAM could be implemented without error correction. Of course, the noise resilience can also be advantageous in the other cases, where hybrid architectures may be employed (Section VI) or when error correction is used (Section VII).

Finally, it is worth emphasizing that the results in this paper constitute general statements about the bucket-brigade architecture, independent of its application to particular algorithms. In fact, the architecture may prove useful in applications other than facilitating algorithmic speedups. For example, Ref. [29] employs the bucket-brigade architecture in a quantum cryptographic protocol. The architecture may similarly prove useful for quantum communication or metrology. Exploring applications of the bucket-brigade architecture—and the utility of its noise resilience—in these other contexts represents an interesting direction for future research. In particular, applications involving quantum queries of quantum data remain largely unexplored.

IX. ACKNOWLEDGEMENTS

We thank Shruti Puri and Xiaodi Wu helpful discussions. We acknowledge the Yale Center for Research Computing for use of its high-performance computing clusters. CTH acknowledges support from the NSF Graduate Research Fellowship Program (DGE1752134). We acknowledge support from the ARO (W911NF-18-1-0020, W911NF-18-1-0212), ARO MURI (W911NF-16-1-0349), AFOSR MURI (FA9550-19-1-0399), DOE (DE-SC0019406), NSF (EFMA-1640959, OMA-1936118, EEC-1941583), NTT Research, and the Packard Foundation (2013-39273).

[1] A. Ambainis, Quantum query algorithms and lower bounds, in *Classical and New Paradigms of Computation and Their Complexity Hierarchies* (Springer, Dordrecht, 2004) pp. 15–32.

[2] D. W. Berry and A. M. Childs, Black-box Hamiltonian simulation and unitary implementation, QIC **12**, 10.26421/QIC12.1-2 (2012).

[3] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian Dynamics with a Truncated Taylor Series, Phys. Rev. Lett. **114**, 090502 (2015).

[4] D. W. Berry, A. M. Childs, and R. Kothari, Hamiltonian Simulation with Nearly Optimal Dependence on all Parameters, in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science* (2015) pp. 792–809.

[5] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity, Phys. Rev. X **8**, 041015 (2018).

[6] G. H. Low and I. L. Chuang, Hamiltonian Simulation by Qubitization, Quantum **3**, 163 (2019).

[7] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, Quantum algorithms for quantum chemistry and quantum materials science (2020), arXiv:2001.03685.

[8] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning (2013), arXiv:1307.0411.

[9] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic Press, 2014).

[10] J. Adcock, E. Allen, M. Day, S. Frick, J. Hinchliff, M. Johnson, S. Morley-Short, S. Pallister, A. Price, and S. Stanisic, Advances in quantum machine learning, (2015), arXiv:1512.02900.

[11] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature **549**, 195 (2017).

[12] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Quantum machine learning: A classical perspective, Proc. R. Soc. A **474**, 20170551 (2018).

[13] S. Aaronson, Read the fine print, Nat. Phys. 10.1038/nphys3272 (2015).

N	Q	Applicable architectures	QEC required?	Paradigmatic example
$\exp(n)$	$\exp(n)$	QRAM	Yes	Searching an unstructured database [25]
$\exp(n)$	$\text{poly}(n)$	QRAM	Maybe not	Solving linear systems of equations [30] (sparse, well-conditioned systems)
$\text{poly}(n)$	$\text{poly}(n)$	QRAM, QROM, Hybrid	Maybe not	Simulating local Hamiltonians [62]

TABLE II. Algorithm categorization. Algorithms are sorted based on how the size of the classical memory, N, and the number of queries, Q, scale with the number of qubits, n. When $N = \exp(n)$, QRAM is the only suitable architecture, assuming $\text{poly}(n)$ query times are required. When $Q = \text{poly}(n)$ quantum error correction may not be required, depending on the physical error rates. For the examples in the last two rows, Q also depends on the particular algorithm used and the desired precision; we assume these are chosen such that $Q = \text{poly(n)}$. We omit the case of $N = \text{poly}(n)$ and $Q = \exp(n)$, for which the query complexity is exponential in the problem size.

[14] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum Random Access Memory, Phys. Rev. Lett. **100**, 160501 (2008).

[15] V. Giovannetti, S. Lloyd, and L. Maccone, Architectures for a quantum random access memory, Phys. Rev. A **78**, 052310 (2008).

[16] F.-Y. Hong, Y. Xiang, Z.-Y. Zhu, L.-Z. Jiang, and L.-N. Wu, Robust quantum random access memory, Phys. Rev. A **86**, 010306(R) (2012).

[17] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, and P. V. Srinivasan, On the robustness of bucket brigade quantum RAM, New J. Phys. **17**, 123010 (2015).

[18] O. Di Matteo, V. Gheorghiu, and M. Mosca, Fault-Tolerant Resource Estimation of Quantum Random-Access Memories, IEEE Trans. Quantum Eng. **1**, 1 (2020).

[19] A. Paler, O. Oumarou, and R. Basmadjian, Parallelizing the queries in a bucket-brigade quantum random access memory, Phys. Rev. A **102**, 032608 (2020).

[20] Alternatively, one could denote the size of the memory by 2^n, where $n \equiv \log_2 N$ is the number of qubits in the address register A. Our main result, the polylogarithmic (in N) scaling of the QRAM query infidelity, could then be equivalently stated as a polynomial scaling (in n).

[21] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum Deep Learning (2014), arXiv:1412.3489.

[22] I. Kerenidis and A. Prakash, Quantum Recommendation Systems, (2016), arXiv:1603.08675.

[23] I. Kerenidis and A. Prakash, Quantum gradient descent for linear systems and least squares, Phys. Rev. A **101**, 022316 (2020).

[24] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, Quantum Chemistry in the Age of Quantum Computing, Chem. Rev. **119**, 10856 (2019).

[25] L. K. Grover, A Fast Quantum Mechanical Algorithm for Database Search, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96 (ACM, New York, NY, USA, 1996) pp. 212–219.

[26] R. Schützhold, Pattern recognition on a quantum computer, Phys. Rev. A **67**, 062311 (2003).

[27] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, Exponential algorithmic speedup by quantum walk, Proc. Thirty-Fifth ACM Symp. Theory Comput. - STOC 03 , 59 (2003).

[28] G. Schaller and R. Schützhold, Quantum algorithm for optical-template recognition with noise filtering, Phys. Rev. A **74**, 012303 (2006).

[29] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum Private Queries, Phys. Rev. Lett. **100**, 230502 (2008).

[30] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**, 150502 (2009).

[31] N. Wiebe, D. Braun, and S. Lloyd, Quantum algorithm for data fitting, Phys. Rev. Lett. **109**, 050505 (2012).

[32] S. Lloyd, S. Garnerone, and P. Zanardi, Quantum algorithms for topological and geometric analysis of data, Nat. Commun. **7**, 10138 (2016).

[33] G. H. Low, V. Kliuchnikov, and L. Schaeffer, Trading T-gates for dirty qubits in state preparation and unitary synthesis, (2018), arXiv:1812.00954.

[34] D. Steiger, Racing in parallel: Quantum versus Classical (2016).

[35] C. T. Hann, C.-L. Zou, Y. Zhang, Y. Chu, R. J. Schoelkopf, S. M. Girvin, and L. Jiang, Hardware-Efficient Quantum Random Access Memory with Hybrid Quantum Acoustic Systems, Phys. Rev. Lett. **123**, 250501 (2019).

[36] T. H. Kyaw, S. Felicetti, G. Romero, E. Solano, and L.-C. Kwek, Scalable quantum memory in the ultrastrong coupling regime, Sci. Rep. **5**, 8621 (2015).

[37] A. Cadellans, *A Transmon-Based Quantum Switch for a Quantum Random Access Memory*, Ph.D. thesis, Leiden University (2015).

[38] Note that the "random access quantum memories" demonstrated in Refs. [63–65] are distinct from QRAM; these experiments do not demonstrate the quantum addressing needed to perform operation (1).

[39] D. W. Berry, C. Gidney, M. Motta, J. R. McClean, and R. Babbush, Qubitization of Arbitrary Basis Quantum Chemistry Leveraging Sparsity and Low Rank Factorization, Quantum **3**, 208 (2019).

[40] M. A. Nielsen and I. L. Chuang, *Quantum Information and Quantum Computation* (Cambridge University Press, 2000).

[41] D. K. Park, F. Petruccione, and J.-K. K. Rhee, Circuit-Based Quantum Random Access Memory for Classical Data, Sci Rep **9**, 3949 (2019).

[42] T. M. L. Veras, I. C. S. De Araujo, K. D. Park, and A. J. Dasilva, Circuit-based quantum random access memory for classical data with continuous amplitudes, IEEE Transactions on Computers , 1 (2020).

[43] This is true even when all memory elements are queried in superposition; each router is in a superposition of active and inactive states, but there are only $\log N$ active routers in each branch of the superposition corresponding to a definite address.

[44] O. Vy, X. Wang, and K. Jacobs, Error-transparent evolution: The ability of multi-body interactions to bypass decoherence, New J. Phys. **15**, 053002 (2013).

[45] E. Kapit, Error-Transparent Quantum Gates for Small Logical Qubit Architectures, Phys. Rev. Lett. **120**, 050503 (2018).

[46] W.-L. Ma, M. Zhang, Y. Wong, K. Noh, S. Rosenblum, P. Reinhold, R. J. Schoelkopf, and L. Jiang, Path-Independent Quantum Gates with Noisy Ancilla, Phys. Rev. Lett. **125**, 110503 (2020).

[47] M. V. den Nest, Simulating quantum computers with probabilistic methods, (2010), arXiv:0911.1624.

[48] D. Gottesman, The Heisenberg Representation of Quantum Computers, (1998), arXiv:quant-ph/9807006.

[49] This observation is distinct from the observation of Refs. [33, 39] that the ancillary qubits used to perform a query can be "dirty." See Appendix B.

[50] M. Saeedi and M. Pedram, Linear-depth quantum circuits for n-qubit Toffoli gates with no ancilla, Phys. Rev. A **87**, 062318 (2013).

[51] Some other error configurations may have high fidelity for specific choices of the error channel, the initial address state, or the classical data, but we ignore this possibility to keep the analysis general and pessimistic.

[52] D. Gottesman, An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation, (2009), arXiv:0904.2557.

[53] E. Knill and R. Laflamme, Theory of quantum error-correcting codes, Phys. Rev. A **55**, 900 (1997).

[54] A. G. Fowler, S. J. Devitt, and C. Jones, Surface code implementation of block code state distillation, Sci. Rep. **3**, 1939 (2013).

[55] J. O'Gorman and E. T. Campbell, Quantum computation with realistic magic-state factories, Phys. Rev. A **95**, 032338 (2017).

[56] T. J. Yoder, R. Takagi, and I. L. Chuang, Universal fault-tolerant gates on concatenated stabilizer codes, Phys. Rev. X **6**, 031039 (2016).

[57] J. Guillaud and M. Mirrahimi, Repetition Cat Qubits for Fault-Tolerant Quantum Computation, Phys. Rev. X **9**, 041053 (2019).

[58] S. Puri, L. St-Jean, J. A. Gross, A. Grimm, N. E. Frattini, P. S. Iyer, A. Krishna, S. Touzard, L. Jiang, A. Blais, S. T. Flammia, and S. M. Girvin, Bias-preserving gates with stabilized cat qubits, Sci. Adv. **6**, eaay5901 (2020).

[59] C. Chamberland, K. Noh, P. Arrangoiz-Arriola, E. T. Campbell, C. T. Hann, J. Iverson, H. Putterman, C. Bohdanowicz, S. T. Flammia, A. Keller, G. Refael, J. Preskill, L. Jiang, A. H. Safavi-Naeini, and O. Painter, Building a fault-tolerant quantum computer using concatenated cat codes, (2020), arXiv:2012.04108.

[60] J. Preskill, Quantum Computing in the NISQ era and beyond, (2018), arXiv:1801.00862.

[61] G. Lee, C. T. Hann, S. Puri, S. M. Girvin, and L. Jiang, (in preparation).

[62] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Toward the first quantum simulation with quantum speedup, PNAS **115**, 9456 (2018).

[63] R. K. Naik, N. Leung, S. Chakram, P. Groszkowski, Y. Lu, N. Earnest, D. C. McKay, J. Koch, and D. I. Schuster, Random access quantum information processors using multimode circuit quantum electrodynamics, Nat. Commun. **8**, 1904 (2017).

[64] N. Jiang, Y.-F. Pu, W. Chang, C. Li, S. Zhang, and L.-M. Duan, Experimental realization of 105-qubit random access quantum memory, Npj Quantum Inf. **5**, 28 (2019).

[65] S. Langenfeld, O. Morin, M. Körber, and G. Rempe, A network-ready random-access qubits memory, Npj Quantum Inf. **6**, 1 (2020).

[66] M. Ramzan and M. K. Khan, Decoherence and entanglement degradation of a qubit-qutrit system in non-inertial frames, Quantum Inf Process **11**, 443 (2012).

[67] H.-R. Wei, B.-C. Ren, and F.-G. Deng, Geometric measure of quantum discord for a two-parameter class of states in a qubit–qutrit system under various dissipative channels, Quantum Inf Process **12**, 1109 (2013).

Appendix A: Bucket-brigade QRAM circuit

Fig. 10 shows a circuit diagram for the bucket-brigade QRAM in the case of an $N = 8$ cell memory. In contrast to the $T = O(\log^2 N)$ depth circuits described implicitly in Refs. [14, 15, 17], the circuit depth here is only $T = O(\log N)$. The quadratic speedup is due to additional parallelization of the routing operations that we now describe.

In Refs. [14, 15, 17], address qubits are routed into the tree one at a time; the $(\ell + 1)$-th address qubit is only injected into the tree once the ℓ-th address qubit has taken its position at level ℓ of the tree. If each routing operation takes one time step, then one waits ℓ time steps between the injection of the ℓ-th and $(\ell + 1)$-th address qubits. The total circuit depth is obtained by summing the number of time steps that it takes for each address to reach the corresponding level,

$$T \sim \sum_{\ell=0}^{\log N-1} \ell = O(\log^2 N). \tag{A1}$$

However, it is not necessary to wait for an address qubit to reach its destination before subsequent address qubits are sent into the tree, and this realization enables the circuit depth to be reduced to $O(\log N)$. Notice that the routing operations for routers located at even levels of the tree act on mutually disjoint qubits and hence mutually commute (the same is true for the odd levels). Thus, all routing operations at either even or odd levels can be performed in parallel. In practice, then, once an address qubit has reached level $\ell = 2$, the next address qubit can be sent into the tree at level $\ell = 0$, and the two can be routed in parallel. This way, the wait time between the injection of subsequent address qubits into the tree is constant (cf. the $O(\ell)$ wait time above). Exploiting this additional parallelism, the total circuit depth is reduced to $T = O(\log N)$.

The circuit diagram in Fig. 10 illustrates how such $T = O(\log N)$ bucket-brigade QRAM circuits are structured. The circuit is a sequence of T constant-depth circuits U_t. During each block U_t, a subset of the possible routing operations is performed (orange boxes in the figure), with routing operations at even levels performed before routing operations at odd levels. Importantly, because the operations at even levels can be performed in parallel (sim. for odd levels), each orange box constitutes only two layers of parallel gates. In principle, new address qubit can then be injected into the tree at the beginning of each orange box, with routing performed in parallel whenever multiple address qubits are being routed through the tree. The case of $N = 8$ shown in the figure is too small to demonstrate this parallelism, but to see how the parallelism would manifest at larger memory sizes, notice that it would be possible to inject another qubit into the tree during U_5 and route this qubit (at level $\ell = 0$) in parallel with one at level $\ell = 2$.

Appendix B: Copying data to the bus

In this section, we explicitly describe how classical data can be copied into the state of the bus. Slightly different procedures are required depending on whether the QRAM is implemented with two-level or three-level systems, and whether the QRAM is initialized in a known state or in some arbitrary state (see Section V).

We begin with the case where the QRAM is implemented with two-level routers, as described in Section V. Each router's incident and output modes are also taken to be physical two-level systems. All routers and their respective modes are initialized to $|0\rangle$. For reasons that will become apparent shortly, we suppose that the bus qubit is initialized to $|+\rangle \equiv (|0\rangle + |1\rangle)/\sqrt{2}$ prior to the query. During the query, this bus qubit is routed down the tree, to an output mode of some router at the bottom level. At this point, classical data is encoded into the state of the bus qubit by applying classically controlled Z gates, as illustrated in Fig. 11(a). If the memory element being queried is 1, a Z gate is applied, and the state of the bus is flipped from $|+\rangle$ to $|-\rangle \equiv (|0\rangle - |1\rangle)/\sqrt{2}$. If the memory element queried is 0, no Z gate is applied, and the bus remains in $|+\rangle$. In this way, the classical bit is encoded in the $|\pm\rangle$ basis of the bus qubit. Note, however, that because the location of the bus is not known, classically controlled Z gates must be applied to the output modes of all routers at the bottom level of the tree.

This data copying operation has a crucial property, which we call *no extra copying*: in the absence of errors, the copying operation acts trivially on all modes that do not contain the bus qubit. In the above case, all modes that do not contain the bus are in $|0\rangle$, so they are unaffected by the Z gates, hence why we use the $|\pm\rangle$ basis for the bus [35]. The no extra copying property is crucial because it guarantees that the final state of the tree is the same across all good (error-free) branches, as required by the arguments in the main text. Were this property not to hold, the final state of the tree would depend on which element was queried, so the bus would remain entangled with the routers after the query, even in the absence of errors.

Now let us consider the case where the QRAM is implemented with three-level routers, as described in Sections II and III. Each router's incident and output modes are taken to be physical three-level systems, whose basis states we also label as $|0\rangle$, $|1\rangle$, and $|W\rangle$. The address and bus qubits are encoded within the $|0, 1\rangle$ subspace of such three-level systems. Prior to the query, all routers, as well as their incident and output modes, are initialized to $|W\rangle$, and the bus is initialized to $|0\rangle$. During the query, the bus is routed to an output mode of some router at the bottom level of the tree. Data is copied into the bus by applying classically controlled $\tilde{X}$ gates to the output modes [Fig. 11(b)], where

$$\tilde{X} = |1\rangle\langle 0| + |0\rangle\langle 1| + |W\rangle\langle W|. \tag{B1}$$

If the memory element being queried is 1, the $\tilde{X}$ gate is

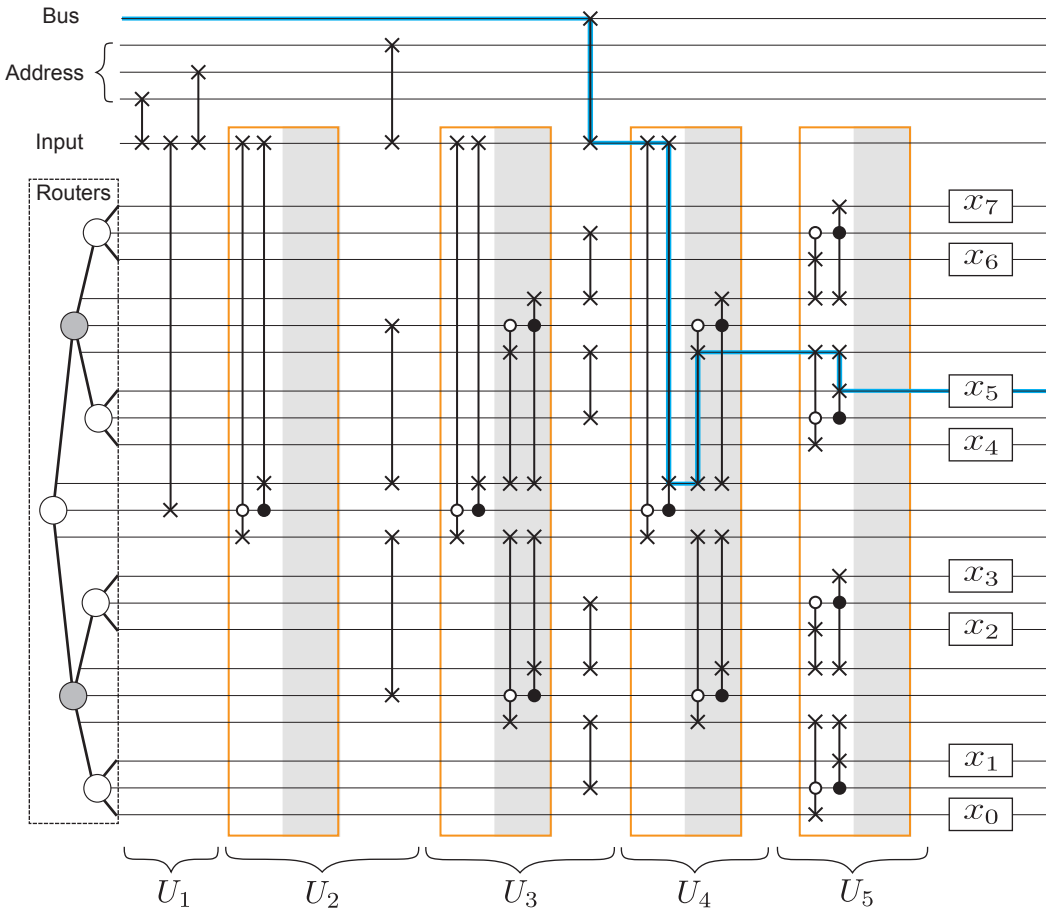

FIG. 10. Bucket-brigade QRAM circuit for $N = 8$. The bus and address register are indicated by rails at the top of the diagram, and the routers are indicated by the rails below. The "input" rail is an extra ancilla, included only to simplify the circuit diagram. For each router shown on the left, there are three rails: one for the router's internal state, and two for the router's two output modes. All rails represent qubits (qutrits) when the variant of the bucket-brigade architecture with two-level (three-level) routers is used; the circuit is the same in either case. To simplify the circuit we use a shorthand notation for the routing operation (top right). Each orange box contains routing operations for a subset of routers in the tree, with routing operations for the even levels first (white background), followed by the odd levels (gray background). Though only a subset of the possible routing operations are actually applied at each time step, in principle all routing operations at even (odd) levels can be applied in parallel. The first part of the circuit, $U_3U_2U_1$, routes the addresses into the tree, as in Fig. 1(d). The next part, U_5U_4, routes the bus to the appropriate memory cell. The path of the bus is highlighted in blue for the case where the three address qubits are initialized to $|010\rangle$. The last layer of gates copy data into the state of the bus (see Appendix B). To route the addresses and bus out of the tree and complete the query, the inverse operation, $(U_5U_4U_3U_2U_1)^\dagger$, must subsequently be applied (not shown).

applied, and the state of the bus is flipped from $|0\rangle$ to $|1\rangle$. If the memory element queried is 0, no $\tilde{X}$ gate is applied, and the bus remains in $|0\rangle$. In this way, the classical bit is encoded in the $|0, 1\rangle$ basis of the bus qubit (one could also choose to encode the information in the $|\pm\rangle$ basis by constructing an analogous $\tilde{Z}$ gate). Here again, the classically controlled gates must be applied to the output modes of all routers at the bottom of the tree. This operation satisfies the no extra copying property because, in the absence of errors, all modes not containing the bus are in $|W\rangle$, on which $\tilde{X}$ acts trivially.

In order to enforce the no extra copying property, both of the above data copying operations rely on the fact that the routers, as well as their input and output modes, are initialized to some known state. When the QRAM is initialized in an arbitrary state (see Section V), however, additional care must be taken to ensure this property still holds. The challenge is that the mode that actually contains the bus must somehow be distinguished from all the other modes, which may have been initialized in the same state as the bus. This problem is solved by the circuit in Fig. 11(c). The QRAM is queried twice, and the no extra copying property is guaranteed by the fact that the entire QRAM unitary operation is idempotent. In

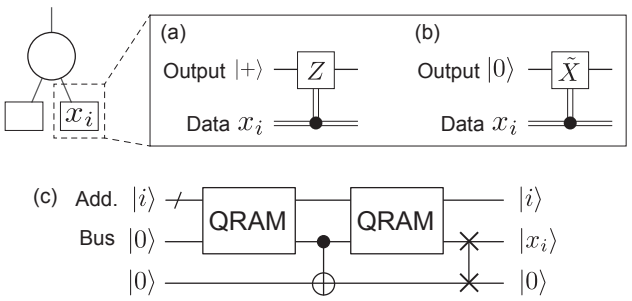

FIG. 11. Circuits for copying classical data. (a) Two-level circuit. The bus qubit is encoded within a physical two-level system and initialized in $|+\rangle$. A Z gate flips the bus to $|-\rangle$ conditioned on the classical data. (b) Three-level circuit. The bus qubit is encoded within a two-level subspace of a physical three-level system and initialized in $|0\rangle$. The $\tilde{X}$ gate (see text) flips the bus to $|1\rangle$ conditioned on the classical data. (c) Query circuit for QRAM initialized in an arbitrary state. The circuits assumes three-level routers, so the bus is initialized in $|0\rangle$ and circuit (b) is employed within each QRAM block to copy data to the bus. An analogous circuit can be constructed for two-level routers. The ancillary qubits comprising the QRAM's routers (not shown) can be initialized in an arbitrary state.

particular, even if the process of copying data during the first query acts non-trivially on modes not containing the bus, these modes are always reset to their initial states by the second query. In fact, even the bus is reset to its initial state by the second query. Thus, the information stored in the bus is copied to an ancillary qubit in between the two queries, then swapped back into the bus after the second query. We emphasize that the query fidelity of this circuit scales favorably, which can be shown by simply replacing $T \to 2T$ in the scaling argument from Section V to account for the fact that the QRAM is called twice.

As an aside, let us distinguish between our observation that QRAM is resilient to noise even when initialized in an arbitrary state (Section V), and the observation of Refs. [33, 39] that the ancillary qubits used to perform a query can be "dirty." The latter states that circuits can be designed such that, *in the absence of errors*, any ancillary qubits used during the query are returned to their initial state after the query, regardless of what the initial state was (note the circuit in Fig. 11(c) has this property). In contrast, our observation concerns what happens when errors occur during the query: the query infidelity of the circuit Fig. 11(c) scales favorably even when the QRAM is initialized in an arbitrary state.

Appendix C: Error channel Kraus decompositions

In this Appendix, we give the Kraus decompositions for the channels used in our simulations. We specify a generic channel $\mathcal{E}$ as via a list of its Kraus operators as

$$\mathcal{E} = \{K_0, K_1, K_2, \ldots\}. \tag{C1}$$

1. Qubit error channels

Let X, Y, Z denote the Pauli matrices. The decompositions of the qubit error channels are

$$\text{Depolarizing} = \left\{\sqrt{1-\varepsilon}I, \sqrt{\frac{\varepsilon}{3}}X, \sqrt{\frac{\varepsilon}{3}}Y, \sqrt{\frac{\varepsilon}{3}}Z\right\} \tag{C2}$$

$$\text{Bit-flip} = \left\{\sqrt{1-\varepsilon}I, \sqrt{\varepsilon}X\right\} \tag{C3}$$

$$\text{Dephasing} = \left\{\sqrt{1-\varepsilon}I, \sqrt{\varepsilon}Z\right\} \tag{C4}$$

$$\text{Damping} = \left\{|0\rangle\langle 0| + \sqrt{1-\varepsilon}\,|1\rangle\langle 1|, \sqrt{\varepsilon}\,|0\rangle\langle 1|\right\} \tag{C5}$$

$$\text{Heating} = \left\{|1\rangle\langle 1| + \sqrt{1-\varepsilon}\,|0\rangle\langle 0|, \sqrt{\varepsilon}\,|1\rangle\langle 0|\right\} \tag{C6}$$

2. Qutrit error channels

We follow the definitions for qutrit depolarizing, bit-flip, and dephasing channels given in Refs. [66], [17], and [67], respectively. Define the operators

$$A_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \; A_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \omega & 0 \\ 0 & 0 & \omega^2 \end{pmatrix}, \tag{C7}$$

where the matrices are written in the $\{|W\rangle, |0\rangle, |1\rangle\}$ basis, and $\omega = e^{i2\pi/3}$. The decompositions of the qutrit error channels are

$$\text{Depolarizing} = \left\{\sqrt{1-\varepsilon}I, \sqrt{\frac{\varepsilon}{8}}A_1, \sqrt{\frac{\varepsilon}{8}}A_2, \sqrt{\frac{\varepsilon}{8}}A_1^2, \sqrt{\frac{\varepsilon}{8}}A_2^2, \sqrt{\frac{\varepsilon}{8}}A_1A_2, \sqrt{\frac{\varepsilon}{8}}A_1^2A_2, \sqrt{\frac{\varepsilon}{8}}A_1A_2^2, \sqrt{\frac{\varepsilon}{8}}A_1^2A_2^2\right\} \tag{C8}$$

$$\text{Bit-flip} = \left\{\sqrt{1-\varepsilon}I, \sqrt{\varepsilon}\left(|0\rangle\langle 1| + |1\rangle\langle 0|\right)\right\} \tag{C9}$$

$$\text{Dephasing} = \left\{\sqrt{1-\varepsilon}I, \sqrt{\frac{\varepsilon}{2}}A_2, \sqrt{\frac{\varepsilon}{2}}A_2^2\right\} \tag{C10}$$

$$\text{Damping} = \left\{|W\rangle\langle W| + \sqrt{1-\varepsilon}\left(|0\rangle\langle 0| + |1\rangle\langle 1|\right), \sqrt{\varepsilon}|W\rangle\langle 0|, \sqrt{\varepsilon}|W\rangle\langle 1|\right\} \tag{C11}$$

$$\text{Heating} = \left\{|0\rangle\langle 0| + |1\rangle\langle 1| + \sqrt{1-\varepsilon}|W\rangle\langle W|, \sqrt{\frac{\varepsilon}{2}}|0\rangle\langle W|, \sqrt{\frac{\varepsilon}{2}}|1\rangle\langle W|\right\}. \tag{C12}$$

Appendix D: Proof of noise resilience for arbitrary error channels

In this Appendix, we prove that, for arbitrary error channels, the QRAM query infidelity satisfies the bound

$$1 - F \leq A\varepsilon T \log N, \tag{D1}$$

where A is a constant of order 1 (defined below). We begin by defining the error model and introducing some convenient notation.

1. Error model

We suppose that at each time step, every router in the QRAM is subject to an error channel $\mathcal{E}$ of the form

$$\rho \to \mathcal{E}(\rho) = \sum_m K_m \rho K_m^\dagger. \tag{D2}$$

One could also consider situations where different routers are subject to different error channels; the proof straightforwardly extends to such situations. For the moment, we make two additional assumptions that simplify the proof. First, we restrict our attention to channels with Kraus rank two, i.e. channels that can be expressed with only two non-zero Kraus operators, K_0 and K_1. Second, we assume that each router is subjected to the error channel $\mathcal{E}$ only once, at the time step t^* after all addresses have been routed into place and immediately before the bus enters the tree (this time step is the one illustrated in Figs. 1 and 2). We relax both of these assumptions later on.

Let us define the error rate, ε. For a generic state $|\psi\rangle$, we may write

$$K_0|\psi\rangle = a_\psi|\psi\rangle + b_\psi|\psi^\perp\rangle \tag{D3}$$

where $\langle\psi|\psi^\perp\rangle = 0$, and a_ψ,b_ψ are complex numbers. We define ε as the smallest positive real number such that, for any $|\psi\rangle$, we have

$$|b_\psi| \leq \sqrt{\varepsilon}, \tag{D4}$$

and for any collection of arbitrary states $|\psi_i\rangle$, we also have

$$\Re\left[\prod_{i=1}^m a_{\psi_i}\right] \geq (1-\varepsilon)^{m/2}, \text{ for any } m \leq \log N. \tag{D5}$$

This definition is unconventional, so let us emphasize that ε can be understood as quantifying the distance between $\mathcal{E}$ and the identity channel. Specifically, for any given $\varepsilon_0 \in [0,1]$, there always exists some $\delta \in [0,1]$ such that, whenever $|\mathcal{E} - I| < \delta$ according to some distance metric, then $\varepsilon < \varepsilon_0$.

We adopt this definition of ε because it turns out to be very convenient for the proof. For example, it follows that

$$\langle\psi|K_0^\dagger K_0|\psi\rangle \geq 1-\varepsilon \tag{D6}$$

$$\langle\psi|K_1^\dagger K_1|\psi\rangle \leq \varepsilon. \tag{D7}$$

and, for a generic product state, $|\xi_m\rangle = \bigotimes_{i=1}^m |\psi_i\rangle$,

$$K_0^{\otimes m}|\xi_m\rangle = a_{\xi_m}|\xi_m\rangle + b_{\xi_m}|\xi_m^\perp\rangle, \tag{D8}$$

where it follows from Eqs. (D4) and (D5) that

$$\Re(a_{\xi_m}) \geq (1-\varepsilon)^{m/2} \tag{D9}$$

$$|b_{\xi_m}| \leq \varepsilon^{m/2} \tag{D10}$$

for $m \leq \log N$.

2. Notation

We now define some convenient notation. As in the main text, we use the shorthand

$$K_c \equiv \bigotimes_{r=1}^{N-1} K_{c(r)} \tag{D11}$$

to denote the composite Kraus operator acting on all routers in the tree. Here, $K_{c(r)} \in \{K_0, K_1\}$ denotes the Kraus operator applied to router r at time t^* as specified

by the error configuration c. For the i-th branch of the tree, we also define the operator $K_{c\notin i}$ as that obtained by taking K_c and replacing $K_{c(r)} \to I$ for any router r in branch i:

$$K_{c\notin i} \equiv \bigotimes_{r=1}^{N-1} \begin{cases} I, & r \in i, \\ K_{c(r)}, & \text{otherwise.} \end{cases} \tag{D12}$$

Similarly, for a pair of branches, i and j, we analogously define

$$K_{c\notin i,j} \equiv \bigotimes_{r=1}^{N-1} \begin{cases} I, & r \in i \cup j, \\ K_{c(r)}, & \text{otherwise.} \end{cases} \tag{D13}$$

where $r \in i \cup j$ denotes the set of routers in either branch i or j.

Additionally, we define $|R_i\rangle$ to be the ideal state of the routers at time t^* assuming the i-th memory element is queried. Specifically, $|R_i\rangle$ is a computational basis state for which all routers $r \notin i$ are in the state $|W\rangle$, and routers $r \in i$ are in either $|0\rangle$ or $|1\rangle$ and carve out a path to the memory element i. As examples, $|R_{101}\rangle$ is illustrated in Fig. 1(d), and the superposition

$$|R_{000}\rangle + |R_{010}\rangle + |R_{011}\rangle + |R_{101}\rangle + |R_{110}\rangle$$

is illustrated in Fig. 2.

Finally, throughout the proof it will be convenient to work with unnormalized quantum states. We distinguish between normalized states $|\psi\rangle$ and unnormalized states $|\overline{\psi}\rangle$ with an overbar.

3. Proof

Our proof of QRAM's noise resilience for generic error channels follows the same outline as the proof for mixed-unitary error channels given in the main text. We again begin by defining the final state of the system Ω as an incoherent mixture over final states for different error configurations c,

$$\Omega = \sum_c \overline{\Omega}(c) \tag{D14}$$

in analogy to Eq. (11). Here, the final system state given error configuration c is $\overline{\Omega}(c) = |\overline{\Omega}(c)\rangle \langle\overline{\Omega}(c)|$, where $|\overline{\Omega}(c)\rangle$ is the unnormalized state

$$|\overline{\Omega}(c)\rangle = V \left[|0\rangle^A |0\rangle^B K_c \left(\sum_i \alpha_i |R_i\rangle^R \right) \right], \tag{D15}$$

where the superscripts A, B, R denote the addresses, bus, and routers, respectively. In accordance with the error model described in Appendix D 1, the quantity in square brackets is the ideal state of the system at time t^* with errors K_c subsequently applied to the routers. The unitary V comprises all operations performed after time t^*. Namely, V routes the bus to memory, copies data into the bus, then routes the bus and all address qubits out of the tree. Note that K_c is not unitary, and the states $|\overline{\Omega}(c)\rangle$ are not normalized. The probability $p(c)$ of error configuration c occurring is

$$p(c) = \mathrm{Tr}[\overline{\Omega}(c)]. \tag{D16}$$

The query fidelity is given by

$$F = \sum_c \overline{F}(c) \tag{D17}$$

where

$$\overline{F}(c) = \langle\psi_{\text{out}}|\mathrm{Tr}_R \overline{\Omega}(c)|\psi_{\text{out}}\rangle \tag{D18}$$

in analogy to Eqs. (12) and (13). As in the main text, we proceed by placing a lower bound on $\overline{F}(c)$. To do so, we write

$$|\overline{\Omega}(c)\rangle = |\overline{\text{good}}(c)\rangle + |\overline{\text{bad}}(c)\rangle\,, \tag{D19}$$

in analogy to Eq. (16), and where the definitions of $|\overline{\text{good}}(c)\rangle$ and $|\overline{\text{bad}}(c)\rangle$ will be given shortly.

Let us recall the definition of *good* and *bad* branches given in the main text. There, a branch i was defined to be good if and only if $\mathbf{i} \cap \mathbf{c} = \varnothing$, where $\mathbf{i}$ is the set of routers in the i-th branch of the tree, and $\mathbf{c}$ is the set of routers

which have $K_{m>0}$ applied to them according to error configuration c. We retain this definition here. Now, in main text we considered the case of mixed-unitary error channels, for which $K_0 \propto I$, and for these channels we have that

$$K_c \left|R_i\right\rangle = K_{c\notin i} \left|R_i\right\rangle, \text{ for } K_0 \propto I \tag{D20}$$

for all $i \in g(c)$, where $g(c)$ denotes the set of good branches. In words, Eq. (D20) says that effect of the of no-error operators on the active routers is trivial. However, in the present case of general channels, for which $K_0 \neq I$ in general, the no-error backaction associated with K_0 can alter the states of active routers. In particular,

$$K_c \left|R_i\right\rangle = K_{c\notin i} \left(a_{R_i} \left|R_i\right\rangle + b_{R_i} \left|R_i^{\perp}\right\rangle\right), \text{ for general } K_0 \tag{D21}$$

for all $i \in g(c)$, and where $\left|R_i^{\perp}\right\rangle$ denotes an orthogonal state, $\langle R_i | R_i^{\perp}\rangle = 0$. It follows from Eqs. (D8) to (D10) that the coefficients a_{R_i} and b_{R_i} satisfy

$$\Re\left(a_{R_i}\right) \geq (1-\varepsilon)^{\log N/2} \tag{D22}$$

$$\left|b_{R_i}\right| \leq \varepsilon^{\log N/2}. \tag{D23}$$

In words, Eq. (D21) says that effect of the of no-error operators on the active routers is nontrivial and places the active routers in a superposition of the ideal state and some orthogonal state. Eqs. (D22) and (D23) bound the coefficients in this superposition; in the relevant regime of $\varepsilon \log N \ll 1$, this nontrivial, no-error backaction is small, and the state $K_c \left|R_i\right\rangle$ is close to $K_{c\notin i} \left|R_i\right\rangle$.

Now, let us define $\left|\overline{\text{good}}(c)\right\rangle$ and $\left|\overline{\text{bad}}(c)\right\rangle$ in Eq. (D19). Given Eq. (D21), it will be convenient to separate the ideal and non-ideal components of $K_c \left|R_i\right\rangle$, retaining only the ideal components in $\left|\overline{\text{good}}(c)\right\rangle$. We thus define

$$\left|\overline{\text{good}}(c)\right\rangle \equiv V \left[\left|0\right\rangle^A \left|0\right\rangle^B \sum_{i\in g(c)} \alpha_i a_{R_i} K_{c\notin i} \left|R_i\right\rangle^R\right] \tag{D24}$$

in analogy to Eq. (17), and

$$\left|\overline{\text{bad}}(c)\right\rangle \equiv \left|\overline{\Omega}(c)\right\rangle - \left|\overline{\text{good}}(c)\right\rangle = V \left[\left|0\right\rangle^A \left|0\right\rangle^B \left(\sum_{i\in g(c)} \alpha_i b_{R_i} K_{c\notin i} \left|R_i^{\perp}\right\rangle^R + \sum_{j\notin g(c)} \alpha_j K_c \left|R_i\right\rangle^R\right)\right]. \tag{D25}$$

Note that $\left|\overline{\text{bad}}(c)\right\rangle$ not only contains a contribution from the bad branches, $j \notin g(c)$, but also from the no-error backaction in the good branches, $i \in g(c)$.

We now seek to prove analogous statements to Eqs. (20) and (21) (given by Lemmas 2 and 3 below). To do so, we first show that the overlap of $\left|\overline{\text{good}}(c)\right\rangle$ and the ideal state is large in the following sense.

Lemma 1. *The overlap between $\left|\overline{good}(c)\right\rangle$ and the ideal final state, $\left|\psi_{\text{out}}, f(c)\right\rangle$, satisfies the bound*

$$\Re\left[\langle\psi_{\text{out}}, f(c)|\overline{\text{good}}(c)\rangle\right] \geq \sqrt{q(c)}\,\Lambda(c) \left(\frac{1-\varepsilon}{1-\varepsilon_W}\right)^{\frac{3}{2}\log N} \tag{D26}$$

where $\Lambda(c) = \sum_{i\in g(c)} |\alpha_i|^2$ is the weighted fraction of good branches, $\left|f(c)\right\rangle$ is some fixed final state of the routers (defined below), and

$$q(c) \equiv (1-\varepsilon_W)^{(N-1)-|c|} \varepsilon_W^{|c|}, \tag{D27}$$

with $\varepsilon_W \equiv 1 - \langle W|K_0^\dagger K_0|W\rangle$, and $|c|$ denotes the number of Kraus operators $K_{m>0}$ in the composite Kraus operator K_c, i.e. $|c|$ is the number of errors.

Proof. Consider a good branch $i \in g(c)$. We have that

$$V\left[\left|0\right\rangle^A \left|0\right\rangle^B K_{c\notin i} \left|R_i\right\rangle^R\right] = \left|i\right\rangle^A \left|x_i\right\rangle^B \left|\overline{f}_i(c)\right\rangle^R, \tag{D28}$$

because all active routers in $K_{c\notin i} \left|R_i\right\rangle^R$ are in their ideal states, so the query succeeds. Here,

$$\left|\overline{f}_i(c)\right\rangle = \left\langle i\right|^A \left\langle x_i\right|^B V \left[\left|0\right\rangle^A \left|0\right\rangle^B K_{c\notin i} \left|R_i\right\rangle^R\right] \tag{D29}$$

is the unnormalized final state of the routers associated with branch i. Consider a second good branch, $j \in g(c)$, with $j \neq i$. In general, the final state of the routers with respect to branch j may be different from the final state with respect to branch i, i.e. $|\overline{f}_i(c)\rangle \neq |\overline{f}_j(c)\rangle$. As a result, the routers may be entangled with the address and bus in $|\overline{\text{good}}(c)\rangle$.

However, we now show that the overlap $\langle \overline{f}_i(c)|\overline{f}_j(c)\rangle$ is large, such that the routers are *nearly disentangled* from the address and bus in $|\overline{\text{good}}(c)\rangle$. Recall the definition of $K_{c\notin i,j}$ [Eq. (D13)] as the operator obtained by taking K_c and replacing all Kraus operators acting on routers in branches i and j with the identity. Observe that

$$K_{c\notin i}\,|R_i\rangle = K_{c\notin i,j}\left(a'_{R_i}\,|R_i\rangle + b'_{R_i}\,|R_i^{\perp}\rangle\right) \tag{D30}$$

where it follows from Eqs. (D8) to (D10) that

$$\Re\left(a'_{R_i}\right) \geq (1-\varepsilon)^{\log N/2} \tag{D31}$$

$$\left|b'_{R_i}\right| \leq \varepsilon^{\log N/2}. \tag{D32}$$

Now, the overlap is given by

$$\langle \overline{f}_i(c)|\overline{f}_i(j)\rangle = \left[\left(\langle 0|^A\,\langle 0|^B\,\langle R_i|^R\,K_{c\notin i}^{\dagger}\right)V^{\dagger}\,|i\rangle^A\,|x_i\rangle^B\right]\left[\langle j|^A\,\langle x_j|^B\,V\left(|0\rangle^A\,|0\rangle^B\,K_{c\notin j}\,|R_j\rangle\right)\right] \tag{D33}$$

$$= (a'_{R_i})^* a'_{R_j}\left[\left(\langle 0|^A\,\langle 0|^B\,\langle R_i|^R\,K_{c\notin i,j}^{\dagger}\right)V^{\dagger}\,|i\rangle^A\,|x_i\rangle^B\right]\left[\langle j|^A\,\langle x_j|^B\,V\left(|0\rangle^A\,|0\rangle^B\,K_{c\notin i,j}\,|R_j\rangle\right)\right], \tag{D34}$$

$$= (a'_{R_i})^* a'_{R_j}\left|K_{c\notin i,j}\,|R_i\rangle\right|^2. \tag{D35}$$

To obtain Eq. (D34) we have used the fact that

$$\langle i|^A\,\langle x_i|^B\,V\left(|0\rangle^A\,|0\rangle^B\,K_{c\notin i,j}\,|R_i^{\perp}\rangle^R\right) = 0 \tag{D36}$$

because by definition $K_{c\notin i,j}\,|R_i^{\perp}\rangle$ has at least one active router in an orthogonal state relative to $|R_i\rangle$, so that after the application of V the final address state is necessarily perpendicular to $|i\rangle$. To obtain Eq. (D35) we have used the fact that the two states in brackets in Eq. (D34) are in fact the same. This is because both $K_{c\notin i,j}\,|R_i\rangle$ and $K_{c\notin i,j}\,|R_j\rangle$ have routers in branches i and j in their ideal states. As articulated in the main text and illustrated in Fig. 5, it follows that the propagation of errors is constrained in such a way that the final states of the routers (the states in brackets) are the same.

Continuing with our calculation of $\langle \overline{f}_i(c)|\overline{f}_j(c)\rangle$, note that the norm $\left|K_{c\notin i,j}\,|R_i\rangle\right|^2$ can be computed straightforwardly using the fact both that $K_{c\notin i,j}$ and $|R_i\rangle$ are tensor products over the different routers. We obtain,

$$\left|K_{c\notin i,j}\,|R_i\rangle\right|^2 = \varepsilon_W^{|c|}(1-\varepsilon_W)^{(N-1)-2\log N-|c|}, \tag{D37}$$

where $|c|$ is the number of Kraus operators $K_{m>0}$ in the composite Kraus operator K_c, i.e. $|c|$ is the number of errors. We have used the definition $\langle W|K_1^{\dagger}K_1|W\rangle = \varepsilon_W$, which implies $\langle W|K_0^{\dagger}K_0|W\rangle = 1-\varepsilon_W$. Thus, combining Eqs. (D31), (D35) and (D37) the real part of the overlap can be bounded as

$$\Re\left[\langle \overline{f}_i(c)|\overline{f}_j(c)\rangle\right] \geq (1-\varepsilon)^{\log N}\varepsilon_W^{|c|}(1-\varepsilon_W)^{(N-1)-2\log N-|c|} = \frac{(1-\varepsilon)^{\log N}}{(1-\varepsilon_W)^{2\log N}}q(c), \tag{D38}$$

where $q(c)$ is defined in Eq. (D27). More convenient for our purposes is the equivalent bound

$$\Re\left[\langle f_i(c)|\overline{f}_j(c)\rangle\right] \geq \frac{(1-\varepsilon)^{\log N}}{(1-\varepsilon_W)^{\frac{3}{2}\log N}}\sqrt{q(c)}, \tag{D39}$$

where $|f_i(c)\rangle$ is the normalized version of $|\overline{f}_i(c)\rangle$, and we have used the fact that $\langle \overline{f}_i(c)|\overline{f}_i(c)\rangle = q(c)/(1-\varepsilon_W)^{\log N}$.

Now, for any $i \in g(c)$, consider the overlap,

$$\langle \psi_{\text{out}}, f_i(c)|\overline{\text{good}}(c)\rangle = \left[\sum_j \alpha_j^*\,\langle j|^A\,\langle x_j|^B\,\langle f_i(c)|^R\right]V\left[\sum_{k\in g(c)}\alpha_k a_{R_k}\,|0\rangle^A\,|0\rangle^B\,K_{c\notin k}\,|R_k\rangle^R\right] \tag{D40}$$

$$= \left[\sum_j \alpha_j^*\,\langle j|^A\,\langle x_j|^B\,\langle f_i(c)|^R\right]\left[\sum_{k\in g(c)}\alpha_k a_{R_k}\,|k\rangle^A\,|x_k\rangle^B\,|\overline{f}_k(c)\rangle^R\right] \tag{D41}$$

$$= \sum_{j\in g(c)}|\alpha_j|^2 a_{R_j}\,\langle f_i(c)|\overline{f}_j(c)\rangle \tag{D42}$$

where the second line follows from Eq. (D28). Using Eqs. (D22) and (D39), we can bound the real part of this overlap as

$$\Re\left[\langle\psi_{\text{out}}, f_i(c)|\overline{\text{good}}(c)\rangle\right] \geq \sqrt{q(c)}\,\Lambda(c)\left(\frac{1-\varepsilon}{1-\varepsilon_W}\right)^{\frac{3}{2}\log N}, \tag{D43}$$

completing the proof. □

The result of Lemma 1 is not precisely analogous to Eq. (20). Rather, the analogous statement is Lemma 2 (see below), which also incorporates detrimental effects of $|\overline{\text{bad}}(c)\rangle$. To incorporate these effects, it is convenient to write

$$|\overline{\text{bad}}(c)\rangle = |\overline{\text{bad}}(c)^{\parallel}\rangle + |\overline{\text{bad}}(c)^{\perp}\rangle \tag{D44}$$

where

$$|\overline{\text{bad}}(c)^{\perp}\rangle \equiv |\overline{\text{bad}}(c)\rangle - \frac{\langle\overline{\text{good}}(c)|\overline{\text{bad}}(c)\rangle}{\langle\overline{\text{good}}(c)|\overline{\text{good}}(c)\rangle}\,|\overline{\text{good}}(c)\rangle \tag{D45}$$

$$|\overline{\text{bad}}(c)^{\parallel}\rangle \equiv |\overline{\text{bad}}(c)\rangle - |\overline{\text{bad}}(c)^{\perp}\rangle \tag{D46}$$

are the components of $|\overline{\text{bad}}(c)\rangle$ perpendicular and parallel to $|\overline{\text{good}}(c)\rangle$, respectively. We proceed by first quantifying the detrimental effects of $|\overline{\text{bad}}(c)^{\parallel}\rangle$.

Lemma 2. *The overlap of* $|\overline{\text{good}}(c)\rangle + |\overline{\text{bad}}(c)^{\parallel}\rangle$ *with the ideal state can be bounded as*

$$\Re\left[\langle\psi_{\text{out}}, f(c)|\left(|\overline{\text{good}}(c)\rangle + |\overline{\text{bad}}(c)^{\parallel}\rangle\right)\right] \geq (1-B\varepsilon^2)\sqrt{q(c)}\,\Lambda(c)\left(\frac{1-\varepsilon}{1-\varepsilon_W}\right)^{\frac{3}{2}\log N}, \tag{D47}$$

with $B \approx 1$ *in the relevant limit of* $\varepsilon \log N \ll 1$.

Proof. We prove the result by showing that the overlap of $|\overline{\text{bad}}(c)\rangle$ and $|\overline{\text{good}}(c)\rangle$ is small relative to magnitude of $|\overline{\text{good}}(c)\rangle$. Specifically, we prove that

$$\frac{\left|\langle\overline{\text{good}}(c)|\overline{\text{bad}}(c)\rangle\right|}{\langle\overline{\text{good}}(c)|\overline{\text{good}}(c)\rangle} \leq B\varepsilon^2. \tag{D48}$$

Then, because $|\overline{\text{good}}(c)\rangle + |\overline{\text{bad}}(c)^{\parallel}\rangle$ is parallel to $|\overline{\text{good}}(c)\rangle$, and because its magnitude is bounded from below by $(1-B\varepsilon^2)||\,|\overline{\text{good}}(c)\rangle\,||$, the lemma follows.

First, we consider the magnitude of $|\overline{\text{good}}(c)\rangle$,

$$\langle\overline{\text{good}}(c)|\overline{\text{good}}(c)\rangle = \sum_{i,j\in g(c)} (\alpha_i a_{R_i})^* \alpha_j a_{R_j}\,\langle R_i|K_{c\notin i}^{\dagger}K_{c\notin j}|R_j\rangle \tag{D49}$$

$$= \sum_{i\in g(c)} |\alpha_i a_{R_i}|^2\,\langle R_i|K_{c\notin i}^{\dagger}K_{c\notin i}|R_i\rangle\,. \tag{D50}$$

Using Eq. (D22) together with the fact that $\langle R_i|K_{c\notin i}^{\dagger}K_{c\notin i}|R_i\rangle = q(c)/(1-\varepsilon_W)^{\log N}$, we can bound this overlap as

$$\langle\overline{\text{good}}(c)|\overline{\text{good}}(c)\rangle \geq q(c)\Lambda(c)\left(\frac{1-\varepsilon}{1-\varepsilon_W}\right)^{\log N}. \tag{D51}$$

Next, we consider the overlap of $|\overline{\text{good}}(c)\rangle$ and $|\overline{\text{bad}}(c)\rangle$, given by

$$\langle\overline{\text{good}}(c)|\overline{\text{bad}}(c)\rangle = \left[\sum_{i\in g(c)} (\alpha_i a_{R_i})^*\,\langle R_i|\,K_{c\notin i}^{\dagger}\right]\left[\sum_{j\in g(c)} \alpha_j b_{R_j} K_{c\notin i}\,|R_j^{\perp}\rangle + \sum_{k\notin g(c)} \alpha_k K_c\,|R_k\rangle\right] \tag{D52}$$

$$= \sum_{i\in g(c)}\sum_{j\in g(c), j\neq i} (\alpha_i a_{R_i})^*\alpha_j b_{R_j}\,\langle R_i|K_{c\notin i}^{\dagger}K_{c\notin j}|R_j^{\perp}\rangle + \sum_{i\in g(c)}\sum_{k\notin g(c)} (\alpha_i a_{R_i})^*\alpha_k\,\langle R_i|K_{c\notin i}^{\dagger}K_c|R_k\rangle\,, \tag{D53}$$

where we enforce $j\neq i$ in the second line by noting that $\langle R_i|K_{c\notin i}^{\dagger}K_{c\notin i}|R_i^{\perp}\rangle = 0$.

Remark 1. A subtle, but important point is that we can actually further enforce

$$j \neq i+1, \text{ for } i \text{ odd}, \tag{D54}$$
$$j \neq i-1, \text{ for } i \text{ even}, \tag{D55}$$

in summation indexed by j in Eq. (D53). This is because, for i even, branches i and $i+1$ share the same active routers, and can be grouped together when using Eq. (D21) to define of $|\overline{\text{good}}(c)\rangle$ and $|\overline{\text{bad}}(c)\rangle$. With these slightly modified definitions, $|\overline{\text{good}}(c)\rangle$ contains the state $K_{c\notin i}(\alpha_i |R_i\rangle + \alpha_{i+1}|R_{i+1}\rangle)$, and $|\overline{\text{bad}}(c)\rangle$ contains the *orthogonal* state $K_{c\notin i}(\alpha_i |R_i\rangle + \alpha_{i+1}|R_{i+1}\rangle)^{\perp}$. It follows that the $j = i+1$ terms vanish in Eq. (D53) for even i. The $j = i-1$ terms vanish for odd i by the same argument. To keep the notation simple, however, we will continue to simply write $j \neq i$ and reference this Remark whenever this distinction is relevant.

The first term in Eq. (D53) can be simplified using the relation

$$b_{R_j} K_{c\notin j} |R_j^{\perp}\rangle = K_c |R_j\rangle - a_{R_j} K_{c\notin j} |R_j\rangle , \tag{D56}$$

which is equivalent to Eq. (D21). Inserting this expression into Eq. (D53) yields

$$\langle\overline{\text{good}}(c)|\overline{\text{bad}}(c)^{\parallel}\rangle = \sum_{i\in g(c)} \sum_{j\neq i} (\alpha_i a_{R_i})^* \alpha_j \langle R_i|K_{c\notin i}^{\dagger} K_c|R_j\rangle , \tag{D57}$$

which we have simplified using the fact that $\langle R_i|K_{c\notin i}^{\dagger} K_{c\notin j}|R_j\rangle = \delta_{ij}$. Eq. (D57) may be equivalently expressed as an inner product between two N-dimensional complex vectors, $\boldsymbol{\alpha_g}$ and $M\boldsymbol{\alpha}$,

$$\langle\overline{\text{good}}(c)|\overline{\text{bad}}(c)^{\parallel}\rangle = \langle \boldsymbol{\alpha_g}, M\boldsymbol{\alpha}\rangle \tag{D58}$$

where the i-th entry of the vector $\boldsymbol{\alpha}$ is α_i, the i-th entry of $\boldsymbol{\alpha_g}$ is

$$\alpha_{g,i} = \begin{cases} a_{R_i}\alpha_i & \text{for } i \in g(c) \\ 0 & \text{for } i \notin g(c), \end{cases} \tag{D59}$$

and M is an $N \times N$ complex matrix with entries $M_{ij} = \langle R_i|K_{c\notin i}^{\dagger} K_c|R_j\rangle (1-\delta_{ij})$.

Consider the quantity $\langle R_i|K_{c\notin i}^{\dagger} K_c|R_j\rangle$, with $i \in g(c)$ and $j \neq i$. Since all states and operators involved are tensor products, this term is a product of $N-1$ matrix elements, one for each of the $N-1$ routers. We now enumerate these matrix elements, first assuming $j \in g(c)$. Of the $N-1$ matrix elements, $(N-1) - 2\log N - |c|$ elements are $\langle W|K_0^{\dagger}K_0|W\rangle = 1-\varepsilon_W$, corresponding to error-free routers outside both branches i and j. (There may be additional such elements, depending on how many routers i and j share, see Eq. (D60) below). With $j \in g(c)$, an additional $|c|$ matrix elements are $\langle W|K_1^{\dagger}K_1|W\rangle = \varepsilon_W$, corresponding to routers that suffer errors. The remaining $2\log N$ matrix elements can be enumerated as follows. For a given state $|R_i\rangle$, let $|R_i^{(\ell)}\rangle$ denote the corresponding state of the router at level ℓ (1-indexed) in branch i of the tree. Additionally, let ℓ_{ij} denote the smallest value ℓ for which $\langle R_i^{(\ell)}|R_j^{(\ell)}\rangle = 0$. At each level ℓ of the tree, there are two remaining matrix elements—associated with routers at level ℓ—that we have not yet enumerated. These matrix elements are

$$\langle R_i^{(\ell)}|K_0|R_j^{(\ell)}\rangle \langle W|K_0^{\dagger}K_0|W\rangle , \text{ for } \ell \leq \ell_{ij}, \tag{D60}$$
$$\langle R_i^{(\ell)}|K_0|W\rangle \langle W|K_0^{\dagger}K_0|R_j^{(\ell)}\rangle , \text{ for } \ell > \ell_{ij}. \tag{D61}$$

The absolute values of these products can be bounded using Eqs. (D4) and (D5),

$$\left|\langle R_i^{(\ell)}|K_0|R_j^{(\ell)}\rangle \langle W|K_0^{\dagger}K_0|W\rangle\right| \leq \begin{cases} (1-\varepsilon_W) & \text{for } \ell \neq \ell_{ij} \\ (1-\varepsilon_W)\sqrt{\varepsilon} & \text{for } \ell = \ell_{ij} \end{cases} \tag{D62}$$
$$\left|\langle R_i^{(\ell)}|K_0|W\rangle \langle W|K_0^{\dagger}K_0|R_j^{(\ell)}\rangle\right| \leq \varepsilon_W\sqrt{\varepsilon}, \tag{D63}$$

where the bound on the first line is tighter for $\ell = \ell_{ij}$ because $\langle R_i^{(\ell_{ij})}|R_j^{(\ell_{ij})}\rangle = 0$ by definition. We can thus bound the product

$$|\langle R_i|K_{c\notin i}^{\dagger} K_c|R_j\rangle| \leq (1-\varepsilon_W)^{[(N-1)-2\log N-|c|+\ell_{ij}]}\, \varepsilon_W^{(|c|+\log N-\ell_{ij})}\, \varepsilon^{(1+\log N-\ell_{ij})/2}, \text{ for } i \in g(c) \text{ and } j \neq i. \tag{D64}$$

We have only shown that the bound (D64) holds for $j \in g(c)$. In fact, though, it also holds for $j \notin g(c)$. The calculation of the bound in this case proceeds almost identically, with the only difference being that we also utilize the bound

$$\left|\langle R_i^{(\ell)}|K_0|W\rangle \langle W|K_1^\dagger K_1|R_j^{(\ell)}\rangle\right| \leq \varepsilon_W \sqrt{\varepsilon}. \tag{D65}$$

We now proceed to bound the norm of the vector $M\boldsymbol{\alpha}$. We do so by first bounding the maximum absolute value column and row sum norms, $||M||_1$ and $||M||_\infty$, respectively. To bound $||M||_\infty$, observe that

$$||M||_\infty = \max_i \sum_j |M_{ij}| = \max_i \sum_{j\neq i} \left|\langle R_i|K_{c\notin i}^\dagger K_c|R_j\rangle\right| \tag{D66}$$

$$\leq \sum_{j\neq i} (1-\varepsilon_W)^{[(N-1)-2\log N-|c|+\ell_{ij}]}\, \varepsilon_W^{(|c|+\log N-\ell_{ij})}\, \varepsilon^{(1+\log N-\ell_{ij})/2}, \tag{D67}$$

where the second line follows from Eq. (D64). To perform the sum, note that there at most $2^{(\log N-\ell)}$ branches j for which $\ell_{ij} = \ell$. Grouping such branches together, we have

$$||M||_\infty \leq \sum_{\ell=1}^{\log N-1} 2^{(\log N-\ell)}(1-\varepsilon_W)^{[(N-1)-2\log N-|c|+\ell]}\, \varepsilon_W^{(|c|+\log N-\ell)}\, \varepsilon^{(1+\log N-\ell)/2} \tag{D68}$$

$$= q(c) \sum_{\ell=1}^{\log N-1} (1-\varepsilon_W)^{[-2\log N+\ell]}\, (2\varepsilon_W)^{(\log N-\ell)}\, \varepsilon^{(1+\log N-\ell)/2} \tag{D69}$$

$$\leq 2q(c)\varepsilon^2/(1-2\varepsilon^{\frac{3}{2}}) \tag{D70}$$

where the summation does not include an $\ell = \log N$ term as a consequence of Remark 1. The inequality on the last line is obtained by applying the bounds $(1-\varepsilon_W) \leq 1$ and $\varepsilon_W \leq \varepsilon$, evaluating the sum, then further simplifying the result by assuming that $\varepsilon \leq 1/2$. By a nearly identical calculation, we find that $||M||_1$ obeys the same upper bound as $||M||_\infty$. Because $||O||_2^2 \leq ||O||_1||O||_\infty$ for arbitrary O, it follows that $||M||_2$ obeys this upper bound as well. Therefore,

$$||M\boldsymbol{\alpha}||_2 \leq ||M||_2||\boldsymbol{\alpha}||_2 \leq q(c)\varepsilon^2/(1-2\varepsilon^{\frac{3}{2}}), \tag{D71}$$

as $||\boldsymbol{\alpha}||_2 = 1$.

We can now finally bound $|\langle\overline{\text{good}}(c)|\overline{\text{bad}}(c)^{\|}\rangle|$. By the Cauchy-Schwarz inequality,

$$|\langle\overline{\text{good}}(c)|\overline{\text{bad}}(c)^{\|}\rangle| = |\langle\boldsymbol{\alpha}_{\boldsymbol{g}}, M\boldsymbol{\alpha}\rangle| \leq ||\boldsymbol{\alpha}_{\boldsymbol{g}}||_2\, ||M\boldsymbol{\alpha}||_2. \tag{D72}$$

Using Eqs. (D51) and (D71) together with the fact that $||\boldsymbol{\alpha}_{\boldsymbol{g}}||_2 = \Lambda(c)$, we thus obtain

$$|\langle\overline{\text{good}}(c)|\overline{\text{bad}}(c)^{\|}\rangle| \leq \varepsilon^2 q(c)\Lambda(c)/(1-2\varepsilon^{\frac{3}{2}}) \leq B\varepsilon^2 \langle\overline{\text{good}}(c)|\overline{\text{good}}(c)\rangle. \tag{D73}$$

where we have defined

$$B \equiv \frac{1}{1-2\varepsilon^{\frac{3}{2}}} \left(\frac{1-\varepsilon_W}{1-\varepsilon}\right)^{\log N}. \tag{D74}$$

This inequality is equivalent to Eq. (D48), so the proof is complete. □

Lemma 2, which incorporates the detrimental effects of $|\overline{\text{bad}}(c)^{\|}\rangle$, is analogous to Eq. (20). It remains to quantify the detrimental effects of $|\overline{\text{bad}}(c)^{\perp}\rangle$, and we do so in the following lemma.

Lemma 3. *The overlap of $|\overline{\text{bad}}(c)^{\perp}\rangle$ with the ideal state can be bounded as*

$$|\langle\psi_{\text{out}}, f(c)|\overline{\text{bad}}(c)^{\perp}\rangle| \leq \sqrt{p(c)} - (1-B\varepsilon^2)\sqrt{q(c)}\,\Lambda(c) \left(\frac{1-\varepsilon}{1-\varepsilon_W}\right)^{\frac{3}{2}\log N}, \tag{D75}$$

where $p(c)$ is the probability of error configuration c, defined in Eq. (D16).

Proof. The proof is a direct application of the identity

$$|\langle\psi|\overline{\phi}^{\perp}\rangle| \leq \left(\langle\overline{\phi}|\overline{\phi}\rangle + \langle\overline{\phi}^{\perp}|\overline{\phi}^{\perp}\rangle\right)^{\frac{1}{2}} - |\langle\psi|\overline{\phi}\rangle|, \tag{D76}$$

which holds for arbitrary $|\psi\rangle$ and unnormalized, orthogonal states $|\overline{\phi}\rangle$ and $|\overline{\phi}^{\perp}\rangle$. Taking $|\phi\rangle = |\overline{\text{good}}(c)\rangle + |\overline{\text{bad}}(c)^{\parallel}\rangle$ and $|\phi^{\perp}\rangle = |\overline{\text{bad}}(c)^{\perp}\rangle$, we have

$$|\langle\psi_{\text{out}}, f(c)|\overline{\text{bad}}(c)^{\perp}\rangle| \leq \langle\overline{\Omega}(c)|\overline{\Omega}(c)\rangle^{\frac{1}{2}} - \left|\langle\psi_{\text{out}}, f(c)|\left(|\overline{\text{good}}(c)\rangle + |\overline{\text{bad}}(c)^{\parallel}\rangle\right)\right| \tag{D77}$$

where we have used $|\overline{\Omega}(c)\rangle = |\overline{\text{good}}(c)\rangle + |\overline{\text{bad}}(c)^{\parallel}\rangle + |\overline{\text{bad}}(c)^{\perp}\rangle$. The proof is completed by leveraging the result of Lemma 2 and recognizing the first term on the right hand side as $\sqrt{p(c)}$.

□

Lemmas 2 and 3 are the analogies of Eqs. (20) and (21) and respectively. Following the main text, we can then write down the analogy of Eq. (22) using the reverse triangle inequality

$$\overline{F}(c) \geq \begin{cases} \left(2(1-C)\Lambda(c)\sqrt{q(c)} - \sqrt{p(c)}\right)^2, & \text{for } 2(1-C)\Lambda(c)\sqrt{q(c)} \geq \sqrt{p(c)}, \\ 0, & \text{otherwise.} \end{cases} \tag{D78}$$

where, to simplify notation, we have defined

$$1 - C \equiv (1 - B\varepsilon^2)\left(\frac{1-\varepsilon}{1-\varepsilon_W}\right)^{\frac{3}{2}\log N}. \tag{D79}$$

Now, recall that the query fidelity is given by $F = \sum_c \overline{F}(c)$. We can equivalently express the query fidelity as an expectation value with respect to the distribution $q(c)$ as

$$F = \mathbb{E}\left(\overline{F}/q\right) \equiv \sum_c q(c)\left(\frac{\overline{F}(c)}{q(c)}\right). \tag{D80}$$

The utility of expressing F this way will be made apparent shortly. Analogously to Eqs. (25) and (26) in the main text, we can then bound the query fidelity as

$$F \geq \mathbb{E}\left(\sqrt{\overline{F}/q}\right)^2 = \left[\sum_c \sqrt{q(c)\overline{F}(c)}\right]^2 \tag{D81}$$

$$\geq \left[2(1-C)\mathbb{E}(\Lambda) - \mathbb{E}(\sqrt{p/q})\right]^2 \tag{D82}$$

where the second line follows from Eq. (D78) under the assumption that $2(1-C)\mathbb{E}(\Lambda) \geq \mathbb{E}(\sqrt{p/q})$. Let us now consider the two expectation values. We have

$$\mathbb{E}(\sqrt{p/q}) = \sum_c \sqrt{q(c)p(c)} \leq 1, \tag{D83}$$

where the inequality follows from recognizing the sum as the Bhattacharyya distance between the distributions p and q. We also have

$$\mathbb{E}(\Lambda) = \sum_c q(c)\Lambda(c) = (1-\varepsilon_W)^{\log N}, \tag{D84}$$

which follows from a recursive calculation nearly identical to the one described in the main text. Thus, we find

$$F \geq \left[2(1-C)(1-\varepsilon_W)^{\log N} - 1\right]^2 \tag{D85}$$

in analogy to Eq. (27). Employing Bernoulli's inequality, we obtain

$$F \geq \left[2(1-C)(1-\varepsilon_W \log N) - 1\right]^2 \tag{D86}$$

$$\geq 1 - 4\varepsilon_W \log N - 4C(1-\varepsilon_W \log N), \tag{D87}$$

assuming $\varepsilon_W \log N + 4C(1 - \varepsilon_W \log N) \leq 1/4$. Equivalently,

$$1 - F \leq 4\varepsilon_W \log N + 4C(1 - \varepsilon_W \log N). \tag{D88}$$

Let us simplify this bound on the infidelity so that the scaling with N is evident. We use Bernoulli's inequality again to obtain the bound

$$C = 1 - (1 - B\varepsilon^2)\left(\frac{1-\varepsilon}{1-\varepsilon_W}\right)^{\frac{3}{2}\log N} \tag{D89}$$

$$\leq \frac{3}{2}\log N \frac{\varepsilon - \varepsilon_W}{1-\varepsilon_W}(1 + B\varepsilon^2). \tag{D90}$$

Inserting this bound, and using the definition of B, we find

$$1 - F \leq A\varepsilon \log N \tag{D91}$$

where we have defined

$$A \equiv 4\frac{\varepsilon_W}{\varepsilon} + 6\frac{1-\varepsilon_W/\varepsilon}{1-\varepsilon_W}\left(1 + \frac{\varepsilon^2}{1 - 2\varepsilon^{\frac{3}{2}}}\left(\frac{1-\varepsilon_W}{1-\varepsilon}\right)^{\log N}\right)(1 - \varepsilon_W \log N) \tag{D92}$$

In the relevant limit of $\varepsilon \log N \ll 1$, the coefficient A is well-approximated by keeping only the leading order terms,

$$A \approx 6 - 2\frac{\varepsilon_W}{\varepsilon}. \tag{D93}$$

Notice that, for mixed-unitary error channels, for which $\varepsilon_W = \varepsilon$, we have $A \approx 4$, in agreement with the proof in the main text.

This concludes the proof of the favorable error scaling, Eq. (D91). Recall, however, that we have made two simplifying assumptions. As described in Appendix D 1, we have assumed that errors are only applied at one time step and that the error channel has a Kraus rank of 2. It remains to relax these assumptions, and we do so in the following subsections.

Error channels with Kraus rank $\neq 2$

The proof can be straightforwardly adapted to handle the case of channels with Kraus rank $\neq 2$. First, let us consider the case of Kraus rank $= 1$, for which the error channel contains a single unitary Kraus operator K_0. This case can be understood as describing coherent errors in the routers and is actually already covered by the above proof. One simply sets $K_1 = 0$, and the bound Eq. (D91) holds.

Next we consider the Kraus rank > 2 case. We define two error configurations, c and c', to be *similar* if and only if $K_{c(r,t)} = K_0 \iff K_{c'(r,t)} = K_0$. That is, error configurations are similar if errors occur at the same locations, though which error $K_{m>0}$ occurs at a given location can differ. We further define c's similar set as the set of all configurations c' that are similar to c.

Let $|c_m|$ denote the number of errors $K_{m>0}$ in configuration c. We generalize the definition of $q(c)$ as

$$q(c) = (\langle W|K_0^\dagger K_0|W\rangle)^{(N-1)-|c|}\prod_{i>0}\left(\langle W|K_m^\dagger K_m|W\rangle\right)^{|c_m|}, \tag{D94}$$

where $|c| = \sum_m |c_m|$. Recall that $\varepsilon_W \equiv 1 - \langle W|K_0^\dagger K_0|W\rangle$. Then, because

$$\sum_{i>0}\langle W|K_i^\dagger K_i|W\rangle = 1 - \varepsilon_W, \tag{D95}$$

as a consequence of the Kraus operators' completeness relation, the total probability of obtaining any configuration from c's similarity set is

$$(1-\varepsilon_W)^{(N-1)-|c|}\varepsilon_W^{|c|}. \tag{D96}$$

This quantity matches the definition of $q(c)$ for the Kraus rank $= 2$ case. Therefore, by grouping all error configurations in the same similarity set together, the proof proceeds exactly as above. In other words, the proof is agnostic to the kind of errors that occur; so long as the total probability of *any* error occurring is fixed, the result is the same.

Errors at all time steps

The core idea of the proof still holds in the case when errors occur at all time steps. In particular, the arguments above can be iterated at each time step in order to isolate the component of the final state that lies along the ideal state. Accordingly, we define

$$|\overline{\text{good}}(c)\rangle = \sum_{i\in g(c)} \alpha_i \tilde{a}_i \, |i\rangle^A \, |x_i\rangle^B \, |\overline{f}_i(c)\rangle \tag{D97}$$

which differs from the definition of $|\overline{\text{good}}(c)\rangle$ given above only in the coefficients $\tilde{a}_i$. These coefficients are associated with the repeated application of Eq. (D21), which we now apply T times to each of the $\log N$ routers in branch i. We extend the definition Eq. (D5) to

$$\Re\left[\prod_{i=1}^{m} a_{\psi_i}\right] \geq (1-\varepsilon)^{m/2}, \text{ for any } m \leq T\log N, \tag{D98}$$

from which it follows that $\Re\tilde{a}_i \geq (1-\varepsilon)^{\frac{1}{2}T\log N}$. Lemma 1 is then generalized simply by replacing $\log N \to T\log N$. Conceptually, the idea is that each time the no-error operator K_0 is applied to an active router, we pay the price of a $\sim (1-\varepsilon)^{\frac{1}{2}}$ prefactor in order to isolate the ideal component resultant state. When we repeat this procedure T times for each of the $\log N$ routers, these prefactors multiply, giving a total prefactor of $\sim (1-\varepsilon)^{\frac{1}{2}T\log N}$.

Next consider the generalization of Lemma 2, which asserts that $|\langle\overline{\text{good}}(c)|\overline{\text{bad}}(c)\rangle| / \langle\overline{\text{good}}(c)|\overline{\text{good}}(c)\rangle = O(\varepsilon^2)$. Observe the proof of Lemma 2 does not actually depend on the time step when the errors are applied. For example, at time $t < t^*$, the QRAM has active routers in only the first $< \log N$ levels of the tree. One can equivalently view this state as the state of a QRAM with $< \log N$ levels at time step t^*, and so the proof of Lemma 2 directly applies. When errors occur at multiple time steps, it follows that

$$|\langle\overline{\text{good}}(c)|\overline{\text{bad}}(c)\rangle| \leq B'\varepsilon^2 \langle\overline{\text{good}}(c)|\overline{\text{good}}(c)\rangle, \tag{D99}$$

for some B'. In general $B' \neq B$, though $B' \approx 1$ in the limit $\varepsilon\log N \ll 1$. An equivalent statement of this result is that, regardless of when errors occur, the overlap of the good and bad states always contains sufficiently many off-diagonal matrix elements (like $\langle\psi|K_0^\dagger K_0|\psi^\perp\rangle \sim \varepsilon$) such that $\langle\overline{\text{good}}(c)|\overline{\text{bad}}(c)\rangle \sim \varepsilon^2$ to leading order.

Having generalized Lemmas 1 and 2, the generalization of Lemma 3 follows straightforwardly. One simply replaces $\log N \to T\log N$ and $B \to B'$ in the bound.

With these generalizations in hand, one can use the same argument as above to show that the fidelity can be expressed in terms of $\mathbb{E}(\Lambda)$, where the expectation is now taken with respect to error configurations with errors at multiple time steps. $\mathbb{E}(\Lambda)$ was already computed with respect to such configurations in the main text. Applying the same calculation, we obtain

$$\mathbb{E}(\Lambda) = (1-\varepsilon_W)^{T\log N}. \tag{D100}$$

Proceeding as above, we then find

$$1 - F \leq A'\varepsilon T\log N \tag{D101}$$

where, in the limit $\varepsilon\log N \ll 1$, the coefficient A' can be well-approximated by

$$A' \approx 6 - 2\frac{\varepsilon_W}{\varepsilon}. \tag{D102}$$

In general, $A' \neq A$ since $B \neq B'$.