Helios: A 98-qubit trapped-ion quantum computer

Quantinuum[1, 2, 3, 4, 5, 6] and Quantum Performance Laboratory, Sandia National Laboratories[7, 8]

[1] *Quantinuum, Broomfield, CO 80021, USA*
[2] *Quantinuum, Brooklyn Park, MN 55422, USA*
[3] *Quantinuum, Plymouth, MN 55422, USA*
[4] *Quantinuum K.K., Tokyo, Japan*
[5] *Quantinuum, London, UK*
[6] *Quantinuum, Cambridge, UK*
[7] *Quantum Performance Laboratory, Sandia National Laboratories, Livermore, CA 94550*
[8] *Quantum Performance Laboratory, Sandia National Laboratories, Albuquerque, NM 87185*

We report on Quantinuum Helios, a 98-qubit trapped-ion quantum processor based on the quantum charge-coupled device (QCCD) architecture. Helios features $^{137}Ba^{+}$ hyperfine qubits, all-to-all connectivity enabled by a rotatable ion storage ring connecting two quantum operation regions by a junction, speed improvements from parallelized operations, and a new software stack with real-time compilation of dynamic programs. Averaged over all operational zones in the system, we achieve average infidelities of $2.5(1) \times 10^{-5}$ for single-qubit gates, $7.9(2) \times 10^{-4}$ for two-qubit gates, and $4.8(6) \times 10^{-4}$ for state preparation and measurement, none of which are fundamentally limited and likely able to be improved. These component infidelities are predictive of system-level performance in both random Clifford circuits and random circuit sampling, the latter demonstrating that Helios operates well beyond the reach of classical simulation and establishes a new frontier of fidelity and complexity for quantum computers.

I. INTRODUCTION

Quantum computing hardware has progressed significantly in the last decade, providing strong experimental evidence of quantum supremacy [1–3] and the feasibility of fault-tolerance [4, 5]. As an increasing number of different modalities check off the basic requirements for quantum computing, the focus of progress is shifting toward scaling these systems to much larger sizes without sacrificing performance.

Like all modalities, the trapped-ion QCCD architecture [6–11] has a unique set of engineering challenges in scaling. For example, trapped-ions can require laser systems for loading, cooling, state-preparation, measurement and coherent control (or a subset of these), introducing somewhat non-standard integration constraints between sub-systems. However, atomic-qubit architectures that use qubit transport, including QCCD and optical tweezers [12, 13], can distribute these computational resources more efficiently than stationary qubits. Mobile qubit architectures allow qubits to flow through the QPU like bits in classical processing architectures, with separated memory structures, data buses, and logic processing units, each optimized for their function. Conversely, stationary-qubit architectures, like superconducting qubits [1, 14] or even atomic-qubits without transport [15, 16], deliver quantum operations to each individual qubit (or connected qubits), which can pose significant engineering and calibration issues. Since transport-based qubits can share expensive hardware resources, the relative complexity of optical control systems (for example) is largely offset by reducing the multiplicative complexity associated with the number of processing zones [17]. Of course, the effectiveness of mobile qubit design principles depends on how sensitive the quantum information is to the required transport operations and how easy the controls are to build and operate. Using hyperfine clock-states and standard scalable micro-fabricated traps for transport control, the QCCD architecture can readily take advantage of these strategies. Indeed, as we show in this work, trapped-ion QPUs are roughly scaling in qubit number as fast or faster than solid state technologies, with the first QCCD computer demonstrated five years ago with 6 qubits [9], to now using 98 qubits.

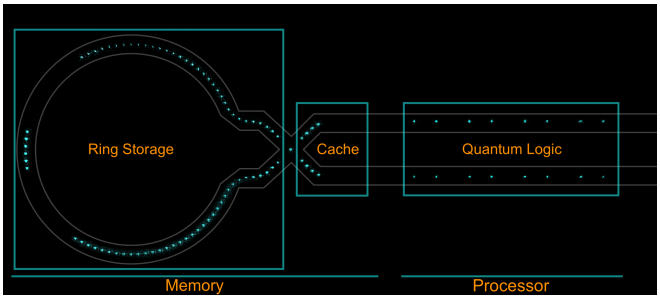

FIG. 1. An image of 98 atomic ions illuminated by resonant laser light in the Helios 2D surface trap illustrated in Fig. 2. The overlaid lines indicate different regions of the device with the quincunx of ions showing the location of the ion trap junction.

In this paper, we present Helios, the next generation system from Quantinuum, which introduces three advances to transport-based, trapped-ion quantum computers. First, Helios uses barium ions as the qubits [18], achieving improved quantum operation error rates with a more scalable laser architecture compared to ytterbium ions used in earlier Quantinuum QPUs [9, 10]. Second, we use a four-way "X" junction [19–27] to efficiently connect memory regions to quantum logic regions without

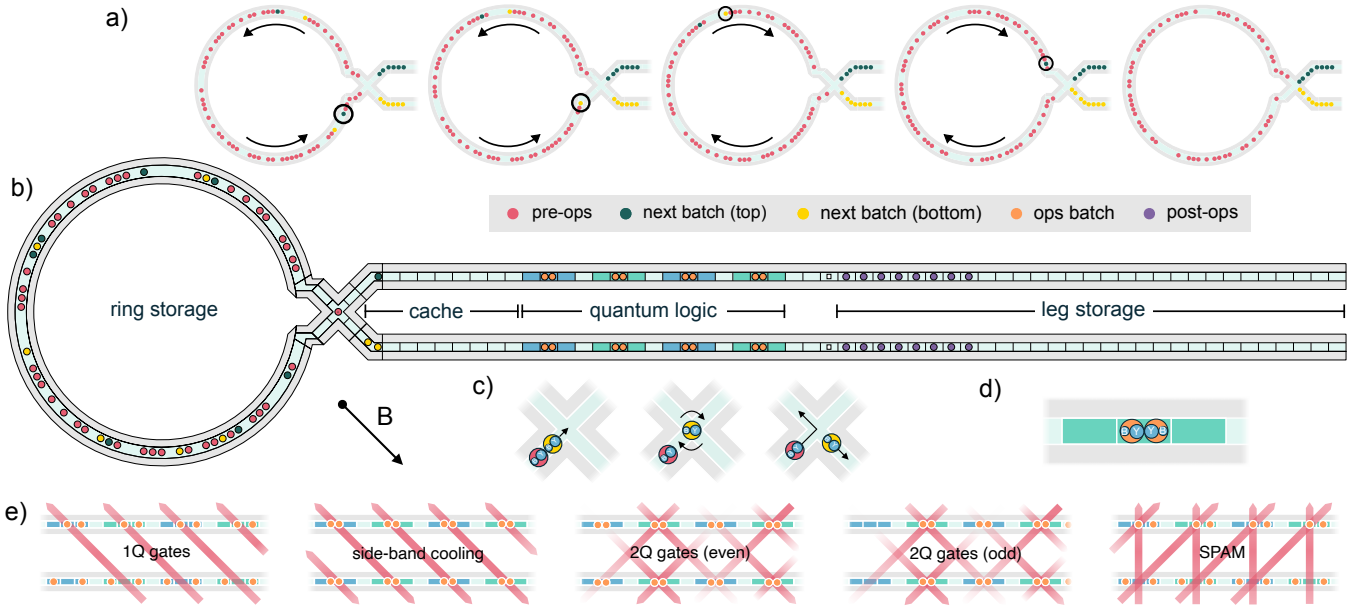

FIG. 2. An illustration of the Helios design and conception of operations. (a) The final five stages of loading the cache region with qubits from ring storage. The ring rotates ions in both directions to move the circled qubit into the cache. (b) Diagram of trap (not to scale) part-way through a program, ring storage qubits are being loaded into the cache and qubits in the quantum logic region are undergoing ground-state cooling. The actual horizontal length is 15.3 mm, the ring diameter is 2.8 mm, and the operational zones are 750 μm apart. (c) Junction operations showing the retrieval and alignment of an ion crystal, and an ion crystal moving through the junction to stay in the ring storage. (d) The proper alignment of a 4-ion crystal in the quantum logic zones. (e) Laser beam and crystal configurations during example quantum operations as labeled. Beams are focused to operate on top/bottom legs as shown by color gradients. The 2Q gate beams are tilted both vertically and horizontally away from the 45 degree line that intersects the ion crystals in both legs by approximately 1 degree so to only interact with a single ion crystal at a time.

increasing electrical control or device fabrication complexity compared to the Quantinuum H2 [10]. Third, Helios is orchestrated by a new classical control implementation capable of making real-time decisions about all transport and quantum operations, enabling execution of truly arbitrary quantum programs with all-to-all connectivity. We show that these generational advancements set a new state-of-the-art in digital quantum computers according to several figures of merit—average two-qubit, single-qubit, and state preparation and measurement (SPAM) fidelities—confirmed by component and system-level benchmarks.

We organize this paper by first providing an overview of the notable advances in the Helios system—the architecture and trap design II A, the ion species II B, the concept of operations of the QCCD II C and the real time compilation of programs II D. We then establish the performance of the QPU with component-level and system-level benchmarking data in Sec. III. Finally, we discuss the outlook in Sec. IV and provide additional experimental and theoretical details in the appendix.

II. HARDWARE AND SOFTWARE ARCHITECTURE

A. QPU architecture and ion trap design

Helios is a transport-based quantum processor with spatially separated qubit memory regions and quantum logic regions. These elements are realized on a 2D surface electrode QCCD [9, 28], which confines ions with electric fields generated by a pattern of electrodes (see Fig. 1 and Fig. 2), and the QPU uses individual ions for qubits. To apply gates to qubits or pairs of qubits, the ions are physically transported to isolated trapping zones to facilitate low-crosstalk addressing and maintain high fidelity.

Figure 2 illustrates how Helios operates. The quantum logic region processes batches of up to 16 qubits at a time, using 8 high-fidelity operation zones, each with the capability to perform state preparation, measurement, ground-state laser cooling, and quantum logic gates. Each operation is implemented via focused laser beams propagating parallel to the chip surface as shown in Fig. 2e. High-fidelity operation necessitates low noise, independent electrode voltages and multiple laser beams for each zone, so they consume most of the control resources in the processor. By using shared lasers across

multiple operation zones (Fig. 2e), the quantum logic region design scales these essential components more efficiently than previous systems.

Qubits outside the operation zones are stored in functionally distinct memory regions: ring storage, leg storage, and cache, see Fig. 2b. Memory regions require less control resources as the only operations available are sympathetic laser cooling [29] and qubit transport. To minimize the number of transport control signals, segmented DC electrodes in the memory regions use voltages that are broadcast in a repeating triplet pattern similar to Ref. [10]. The cache is a small memory region that holds the next batch of pre-sorted qubits before going to the quantum logic region. The leg storage operates as a first in, last out memory, while the ring storage acts as a random access memory, because it connects to the operational region via an X-junction.

The junction is a key structure enabling this architecture. As qubits move through the junction, they can be routed to remain in memory or be added to the cache in either the upper or lower legs. Furthermore, by implementing qubit routing in a separate structure from the quantum logic region, qubit sorting can proceed in parallel with the ground state cooling of ions in the logic region, reducing the effective clock-speed of the QPU. Comparisons to the Quantinuum H1 [9] and Quantinuum H2 [10] QPUs summarize the cumulative impact of these design choices in the electrical control subsystems (Table I).

System	Num. Electrodes	Num. Signals	Signals/Qubit
H1	198	198	9.9
H2	376	268	4.8
Helios	1228	273	2.8

TABLE I. The number of electrodes and independent voltage signals per qubit for three different generations of Quantinuum QPUs.

B. Ion Species - qubit and coolant

Helios is the first quantum computer to utilize $^{137}Ba^{+}$. We define $|F=1, m_f=0\rangle$ and $|F=2, m_f=0\rangle$ hyperfine levels in the $^{137}Ba^{+}$ electronic ground state as $|0\rangle$ and $|1\rangle$ respectively. The optical transitions used to implement quantum operations are in the visible part of the wavelength spectrum, allowing for laser and optical components that are more mature, reliable, and cost-effective and enables fundamentally better performance. Using more available laser power with better phase performance, we can suppress the leading sources of errors in logic gates, including spontaneous emission errors, laser phase fluctuations, and higher-order Lamb-Dicke errors [30].

Specifically, the single-qubit (1Q) and two-qubit (2Q) gates are implemented with pairs of 515 nm laser beams separated by the ∼8.04 GHz qubit frequency splitting. The 1Q gates, $U_{1Q}(\theta,\phi) = e^{(-i\theta/2)(\cos\phi X+\sin\phi Y)}$, are implemented with co-propagating laser beams for improved phase stability of the Raman interaction and minimal sensitivity to the ions' thermal motion. 1Q Z-rotations, $R_Z(\theta) = e^{-iZ\theta/2}$, are implemented by phase changes in software. The 2Q gates are implemented with beams intersecting the quantum logic zones at 90 degrees to each other such that the difference k-vector is parallel to the crystal axis (Fig. 2e). The 2Q gate protocol is based on the Mølmer-Sørensen interaction using wrapper pulses to remove optical phase sensitivity [9, 31], yielding a native 2Q gate $R_{ZZ}(\theta) = e^{-iZZ\theta/2}$. The gate angle θ is specified by the user and is varied by adjusting the detuning and duration of the gate. Gate infidelities have been shown to improve for smaller angles [10], but here we only benchmark the perfect entangler $R_{ZZ}(\pi/2)$.

State preparation and measurement (SPAM) are achieved in $^{137}Ba^{+}$ with a combination of lasers at 493 nm, 614 nm, 650 nm and 1762 nm via narrow-band optical pumping Ref. [32, 33]. The 1762 nm laser is locked to a narrow linewidth cavity to facilitate high-fidelity mapping pulses between the $S_{1/2}$ ground state and $D_{5/2}$ state (Fig. 3). The standard measurement protocol first maps the $|F=1, m_f=0\rangle$ qubit state to the $D_{5/2}$ manifold with multiple π pulses to different levels in $D_{5/2}$. Then the 493 nm and 650 nm lasers are turned on to induce fluorescence from all $S_{1/2}$ states. Additionally, the 1762 nm laser is used to protect neighboring qubits from measurement crosstalk errors (Fig. 3b) and enables a ternary (three outcome) measurement to detect leakage population (Fig. 3c) without the use of ancillas or 2Q gates [34–36].

The QCCD architecture relies on mid-circuit recooling of ions, achieved here with sympathetic cooling applied to $^{171}Yb^{+}$ ions co-trapped with the $^{137}Ba^{+}$ qubit ions. The $^{171}Yb^{+}$ ion is chosen because of similar mass to $^{137}Ba^{+}$ and for the established and straightforward methods for qubit control and state measurement [37]. The cooling is performed with lasers tuned near the $S_{1/2}$ to $P_{1/2}$ transition of $^{171}Yb^{+}$ at 369 nm.

To load ions into the QCCD, we photoionize both species from cold atomic beams produced by an atomic source similar to Ref. [10], based on a neutral atom magneto-optical trap (MOT) [38, 39]. Other hardware details, including implementation of all quantum operations are described in the Appendix.

C. QCCD operation

In this section, we describe how Helios executes quantum programs using the operations depicted in Fig. 2. An arbitrary quantum program is decomposed into ion transport and quantum operations. These operations are not pre-planned but instead executed with a new real-

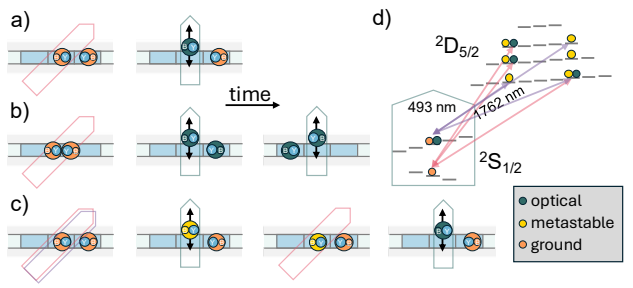

FIG. 3. Three types of measurements are available in all 8 quantum operation zones. All measurements are made with the target ion displaced from the RF null to reduce stray light interacting with non-measured ions [40] as shown with double arrows. (a) Standard measurement occurs when the user specifies a measurement but not for all the qubits in the batch. (b) Protected measurement occurs when the compiler detects an entire batch of qubits will be measured, such as at the end-of-program measurement. Protected measurement performs the $D_{5/2}$ mapping operations on both qubits prior to state detection such that crosstalk from 493 nm detection light does not affect the measurement outcome. (c) User specified ternary measurement allows the user to obtain a result of 0, 1, or L. In this case, each qubit state amplitude is mapped to different parts of the $D_{5/2}$ manifold [41] and any remaining population in the $S_{1/2}$ population (representing leakage errors) is measured via induced fluorescence with the 493 nm and 650 nm lasers. Afterwards, a series of pulses independently maps each state amplitude back into the $S_{1/2}$ and $D_{3/2}$ manifolds allowing measurement of the qubit state (0 or 1). Ternary and protected measure can be combined when an entire batch is measured. (d) Energy level diagram for $^{137}Ba^+$ with $S_{1/2}$ ground state manifold used for storage and quantum operations and the $D_{5/2}$ used during measurement.

time and dynamic classical control software called "Helios runtime", which is described in detail in Sec. II D.

Ions move through the trap using transport operations from four categories: shift, split/combine, junction transport, and rotate. Shift operations translate ions along linear sections in the cache, quantum logic, and leg storage regions. These operations can move both two-ion Ba−Yb (BY) and four-ion Ba−Yb−Yb−Ba (BYYB) crystals. Split (combine) operations separate (merge) BYYB (BY and YB) crystals in the eight operation zones. Junction exit (enter) operations move crystals from (into) the junction into (from) the desired leg in the cache with the desired order, BY or YB. Rotate operations collectively move crystals in the ring clockwise or counterclockwise.

Programs use these transport operations to move qubits between the memory and processor regions of the trap. This cycle occurs during a single layer in a program, in which qubits are removed from ring storage, processed in batches within the quantum logic region, and then returned to ring storage. Every program begins with qubits in a default configuration: 8 BYYB crystals in the quantum logic region and 82 BY crystals in ring storage. Each layer contains up to 7 batches, with a maximum of 16 qubits per batch.

Using appropriate ion-to-qubit assignments, quantum operations immediately begin on the qubits already in the eight operation zones with individual addressing operations occurring first: state preparation (or reset), 1Q gates, and measure operations. Next, if 2Q gates are required, the BY and YB pairs associated with each zone are combined to BYYB crystals and ground-state cooling begins. In parallel with cooling, qubits for the next batch of gating are moved from the ring storage to the cache. This parallel sorting with ground state cooling allows cooling and gating cycles to run nearly continuously, as the next batch of qubits is ready to shift in as the current batch finishes operations.

Unlike 1Q, reset, and measure operations, 2Q operations are executed in only four of the eight quantum logic zones (second and fourth zones on top and bottom legs as shown in green in Fig. 2b,e). To perform 2Q gates on all 8 qubit pairs, the qubits are first merged and cooled as 8 four-ion crystals in all operation zones and then 2Q beams are applied in the four 2Q operation zones. Immediately after executing the 2Q gates, the four-ion shift operation moves all crystals over by one zone (the crystals in the right edge operational zones are split to BY and YB pairs and then shifted into the storage legs). We then apply a small ($\sim$300 μs) additional amount of cooling to remove any energy gained from the shift operation and then gate the remaining four crystals. The 2Q gate operation itself requires approximately $\sim$70 μs to execute.

After executing quantum operations, a batch is complete: its qubits move to leg storage, while qubits in the cache shift to the quantum logic region. This process repeats until all qubits requiring operations have been processed. Lastly, all qubits move from leg storage to the ring, and the cycle begins for the next layer.

Fig. 4a shows timing estimates and a breakdown of operations per layer for a representative program on Helios. The program is constructed as a sequence of 10 layers, in which qubits are randomly paired and receive 1Q and 2Q gates each layer. We define the "depth-1 time" as the time required to perform the random pairing and 1Q and 2Q gates in a single layer, and use this time as our characteristic figure of merit for processor speed. We estimate the average depth-1 time by measuring the duration of the depth-10 program and dividing it by the number of layers to average any fortunate sort cases, resulting in an average of 55 ms per layer. To illustrate how program details such as 2Q gate density and qubit connectivity impact depth-1 time, we present timing results in Fig. 4b for three example programs as a function of the number of active qubits (for more details, see App. A2).

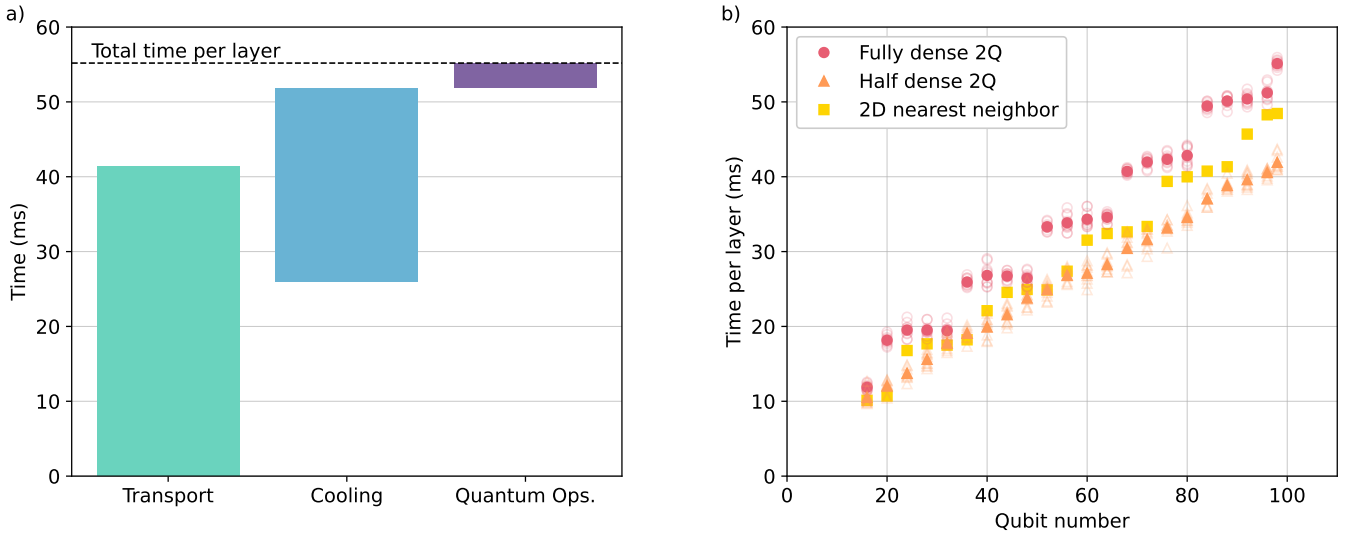

FIG. 4. (a) Time budget per layer for an example depth-10 random program that executes 1Q and 2Q gates on all 98 qubits after an arbitrary permutation each layer, broken down into three categories: ion transport; ground-state cooling; and quantum operations (1Q and 2Q gates). (b) Total time per layer versus number of active qubits for three programs: a random program with fully dense 2Q gates, the same random program with approximately half the 2Q gate density, and a program with 2D nearest-neighbor 2Q gate pairing. For the two random programs, solid points represent the mean of 10 program instances; hollow points show the individual values.

D. Real time compilation of sorting and gates

To realize the full capability of the Helios QPU, the system must be capable of executing arbitrary quantum programs efficiently, including dynamic quantum programs. Optimal decision making for dynamic quantum programs requires a new classical control hardware unit and software compilation stack. This new stack both allows for real-time qubit routing decisions and increases the level of abstraction of quantum programs—mirroring the way classical computers advanced from writing assembly code to writing high-level programs.

In particular, Helios is the first trapped-ion QPU to translate operations on a program's "virtual qubits" [42] into operations on corresponding physical qubits on the device in real time—that is, while the program is executing and quantum state is live. This is enabled by the Helios runtime, whose responsibility is to efficiently map virtual qubits to physical qubits on the device and turn declarative gates on virtual qubits into operations on physical qubits. This runtime enables state-of-the-art user programming constructs for use on a quantum computer (functions that can allocate and de-allocate qubits depending on the control flow of the program), early termination of programs based on mid-circuit measurement or arbitrary classical logic, and classical control flow such as `if-then-else` statements, `for` loops, and `while` loops. This is in stark contrast to the way most gate-level quantum programs, commonly referred to as "dynamic circuits" [43], are written right now—as a flat series of gates with conditional gates conditioned on measurements. Many of the Guppy [44] programs for the applications discussed in Sec. III use some of these features. Additionally, any programming language compiling to QIR such as Q# [45], qiskit [46], Open QASM 2.0/3.0 [47, 48], cirq [49], and CUDA-Q [50] can use QIR adaptive profile features to implement these control flow constructs for programs executing on Helios.

An example of high-level operations enabled by the Helios runtime is the "gate streaming" used in [51]. In the Guppy program executed on Helios for this work, a section of the program performs a remote-procedure-call out to a classical server that is separate from the control system but which is allowed to communicate to the control system via a networking interface [52]. The information transmitted to the control system by the classical server is the measurement basis for each qubit. If a qubit needs no change in measurement basis then the runtime receives no 1Q gate to apply before measurement. In the case that a whole row of BY or YB crystals on the top or bottom legs needs no basis change, the Helios runtime will not perform any extraneous transport to address these qubits. Importantly, this reduces the overall shot time, improving the critical latency times in that application. Efficient gate streaming would be impossible without the real-time identification of qubits provided by the runtime.

The core responsibilities of the Helios runtime are the following: (1) receive qubit allocation requests on virtual qubits and resolve them to physical qubits; (2) receive gating requests on allocated virtual qubits; (3) transform requested gates on sets of virtual qubits into parallel operations on as many physical qubits as can fit in the

quantum operation zones; and (4) transport batches of physical qubits from the ring into these zones, referred to as a "sort".

Responsibility (1) is performed using a model of the physical QPU state as the program runs and determining efficient mappings from virtual qubits to physical qubits. Responsibilities (2) and (3) are performed by identifying which quantum logic operations can be done in parallel by storing them in sets contained in a data-structure we refer to as a "slice". Sequences of slices are accumulated into another data-structure that drives the sorting of each slice to execute the quantum logic operations within. Responsibility (4) is performed by doing an $O(n)$ traversal over the ring storage to determine which two pairs in a slice have qubits closest to the cache. The runtime then assigns one pair to move to the top leg and the other to the bottom. Subsequently, the algorithm determines the smallest number of rotations needed to move the two pairs into BYYB crystals in both legs. This process is visualized in Fig. 2. This process repeats until either enough pairs are moved into the cache to fill a batch, or until no more pairs need to be sorted. Finally, the runtime dispatches the calculated sort by generating these operations as a queue of commands to lower-level control system software for performing transport operations and parallelized cooling as outlined in II C. After all of the quantum logic operations have been executed in a given slice via repetitions of this sort, transport is generated to return the qubits back into the ring storage–and the sorting algorithm repeats for subsequent slices.

III. BENCHMARKING

A. Overview

To see how Helios performs in practice and understand current limitations, we characterize individual operations with component-level benchmarks and full-device operation with system-level benchmarks [10]. Operations include SPAM, 1Q and 2Q gates, mid-circuit measurements and resets (MCMRs), and qubit idle during ion transport. We perform two separate system-level benchmarking experiments [53–58], both of which are examples of volumetric benchmarks [54]. The first involves random Clifford circuits with MCMR, which can be simulated classically. We include MCMRs, unlike most prior work, because they are necessary for quantum error correction. The second experiment is mirror benchmarking of random circuit sampling (RCS), which is an appealing benchmark because the quantum computational power can be measured by the classical simulation cost. The use of mirroring allows for estimating the circuit fidelity where classical simulation is unfeasible.

In the following, we first describe the component-level benchmarks in Sec. III B, with a summary of the results given in Tab. II. We then present our system-level benchmark results with a detailed comparison to the prediction from the component-level benchmarks in Sec. III C.

Component	Metric	Value ($\times 10^{-4}$)
SPAM (standard)	Average error	4.8(6)
SPAM (ternary)	Average error	17(1)
1Q gates	Clifford avg. infidelity	0.25(1)
2Q gates	Avg. infidelity (2QRB)	7.9(2)
2Q gates	Avg. infidelity (CB)	8.1(2)
Transport idle	Linear memory error rate	5(1)
Transport idle	Quadratic memory error parameter	0.7(2)
MCMR crosstalk	Avg. infidelity (global)	0.48(1)

TABLE II. Component-level benchmark values, averaged over all operation zones.

B. Component-level benchmarks

1. *State-preparation and measurement*

It is difficult to differentiate state preparation errors from measurement errors [59], although from detailed modeling of $^{137}Ba^{+}$ qubits we expect state preparation errors to be the largest contributor [32].

We measure SPAM errors by preparing 16 qubits in the 8 operation zones in the $|0\rangle$ or $|1\rangle$ states, and measuring each qubit. For any given shot, the state preparations are randomized among the different qubits, but we ensure that each qubit is prepared in each state for the same total number of shots. We run two experiments: standard measurement that ideally differentiates $|0\rangle$ from $|1\rangle$ but falsely returns $|1\rangle$ in the event that the qubit has leaked, and a ternary measurement, shown in Fig. 3c, that ideally differentiates $|0\rangle$, $|1\rangle$, and leaked states. For both experiments, we take 4000 shots per state preparation.

For the standard measurement, we measure errors of $8(1)\times10^{-4}$ and $1.6(5)\times10^{-4}$ when preparing $|0\rangle$ and $|1\rangle$, respectively. Because this measurement protocol mistakenly detects leaked states as $|1\rangle$, the reported error for preparing and measuring $|1\rangle$ will not catch all errors [32]. For the ternary measurement, we find an average leakage probability of $4.2(7)\times10^{-3}$, and in the event of non-leakage we measure SPAM errors of $7(1)\times10^{-4}$ and $2.8(2)\times10^{-3}$, for $|0\rangle$ and $|1\rangle$, respectively. Although the ternary measurement reveals more information as it can detect leakage, it also has a larger SPAM error due to a larger number of shelving pulses involved. The SPAM errors reported in Tab. II are averaged between the two state preparations.

2. *Single-qubit gates*

Single-qubit gate errors are primarily caused by spontaneous emission during the Raman gate, laser phase and intensity noise, and finite qubit coherence. Importantly,

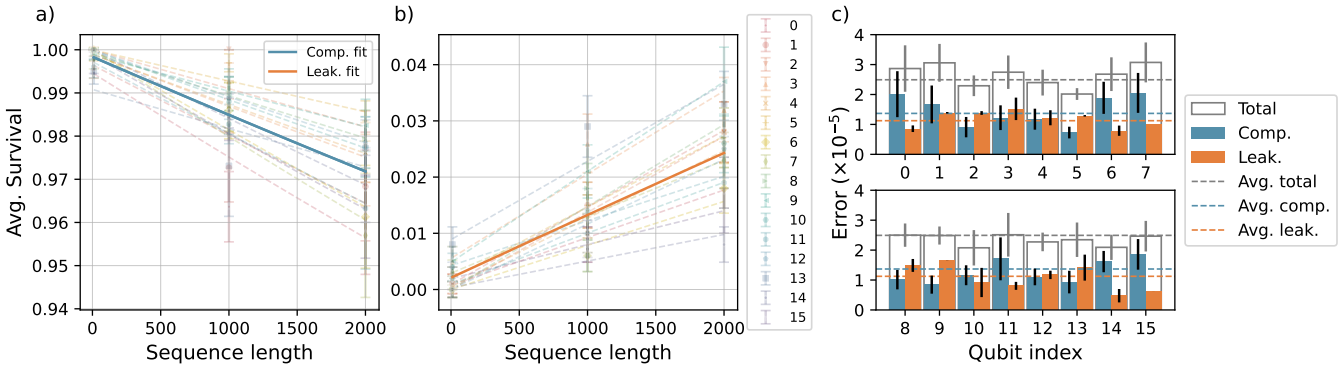

FIG. 5. 1QRB data: (a) 1QRB survival probability as a function of sequence length, for the 16 qubits occupying the 8 operation zones. (b) 1QRB measured leakage population as a function of sequence length. The leakage rate is combined with the survival decay rate to compute the 1Q Clifford infidelity. (c) Breakdown of 1Q Clifford error rates into computational (comp.) errors and leakage (leak.) errors, for the 16 individual qubits. Label locations correspond to qubit locations in Fig. 2b with qubits 0-7 in the top operation zones and 8-15 in the bottom (two per zone ordered left to right).

spontaneous emission causes leakage outside of the computational subspace. We quantify 1Q gate errors by Clifford randomized benchmarking (RB) [60], with details provided in App. A3 B.

We follow the methods in Ref. [61] to account for leakage in the 1Q infidelity estimate. The ternary measurement allows us to measure the leakage population at the end of every circuit without the use of ancilla qubits (as was done in Ref. [10]). We estimate the rate of leakage per 1Q Clifford r_L by the rate at which the measured leakage population increases with sequence length. The probability of observing the expected computational state decays exponentially due to non-leakage errors as $p(l) = A(1-r)^l + 1/2$ for sequence length l. The reported 1Q error is the Clifford average infidelity $\epsilon_{avg,1Q} = r/2 + r_L$ [61].

Figure 5 shows the survival probability and the leaked population as a function of l, for all 16 qubits in the 8 operation zones. We obtain a zone-averaged 1Q error of $2.5(1) \times 10^{-5}$, which includes a leakage rate of $1.12(6) \times 10^{-5}$. The error bars represent a 1-sigma confidence interval obtained from bootstrapping [62]. The leakage rates and infidelities for each individual qubit are given in Tab. A1. The measured errors can be compared with our predictions from physical error models of $2.6(6) \times 10^{-5}$ that account for measured laser intensity noise, calculated spontaneous emission, and measured memory error.

Finally, we ran a statistical hypothesis test for correlated errors in the simultaneous 1QRB data. An error channel on multiple subsystems is correlated if it cannot be factored into a tensor product of individual error channels on each subsystem, and such correlated errors are a signature of crosstalk. We found no evidence of correlated errors at the 95% confidence level (see App. A3 A for analysis details).

3. *Two-qubit gates*

Errors in the $R_{ZZ}(\theta)$ gates are caused by spontaneous emission from the Raman lasers and experimental imperfections including laser phase and intensity noise at the ion's position, thermal motion of the ions, voltage noise on the electrodes, and imprecise calibrations of the gate parameters. We validate the performance of the maximally entangling $R_{ZZ}(\pi/2)$ gate (referred to as the 2Q gate) using both Clifford 2QRB and cycle benchmarking (CB). Additional details of each implementation is in App. A3 B.

We again follow the methods in Ref. [61] to account for leakage in the 2QRB infidelity estimate. The leaked population versus sequence length is used to extract a leakage rate per Clifford, which is rescaled into a leakage rate per 2Q gate $r_{L,2Q}$, using the fact that there are 1.5 2Q gates per 2Q Clifford on average. We fit the survival probability of the remaining population to the decay model $p(l) = A(1-r)^l + 1/4$, and the average infidelity of the non-leakage error component per Clifford is given by $3r/4$, which is rescaled into an average infidelity per 2Q gate of $r/2$. The average infidelity per 2Q gate (including leakage) is then computed as $\epsilon_{avg,2Q} = r/2 + r_{L,2Q}$. We note that our rescaling of the error per Clifford into an error per 2Q neglects the errors from 1Q gates and memory errors during the 2QRB sequence, which we estimate to contribute $1.2(2) \times 10^{-4}$ per 2Q gate.

The experimental 2QRB data is shown in Fig. 6. We obtain a zone-averaged 2Q infidelity of $\epsilon_{avg,2Q} = 7.9(2) \times 10^{-4}$, which includes a leakage rate of $r_{L,2Q} = 2.4(1) \times 10^{-4}$. The leakage rates and infidelities for each individual qubit pair are given in Tab. A2. The leakage errors arise both from spontaneous emission error, which we measure to be $1.0(2) \times 10^{-4}$ in agreement with the model of [63], and from the leakage memory error (discussed in Sec. III B 4). In total, we expect leakage to contribute $1.7(2) \times 10^{-4}$ of the error.

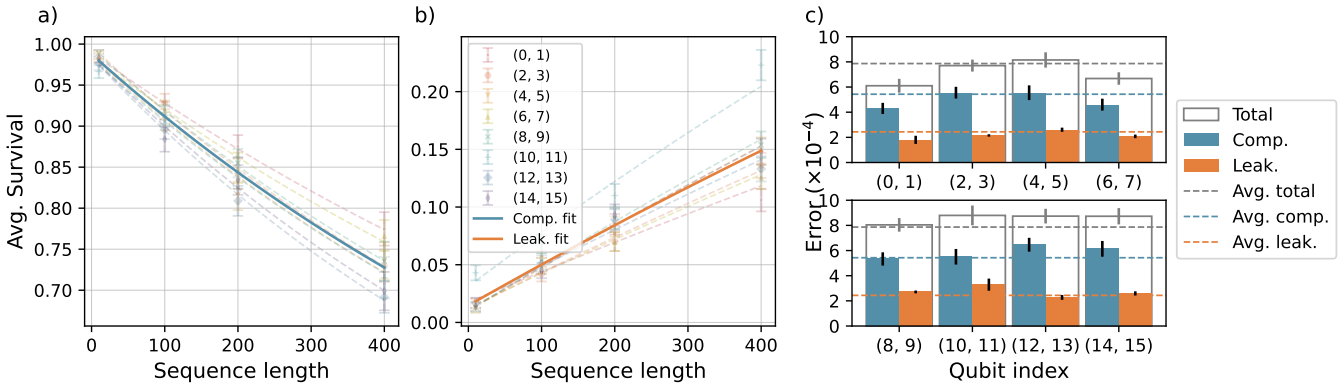

FIG. 6. 2QRB data: (a) 2QRB survival probability as a function of sequence length for 8 qubit pairs in the 8 operation zones. Sequence length here refers to the number of Clifford group elements. (b) 2QRB measured leakage rate as a function of sequence length. The leakage rate is combined with the survival decay rate to compute the 2Q infidelity. (c) Breakdown of $R_{ZZ}(\pi/2)$ errors into computational and leakage errors, for the 8 qubit pairs. Label locations correspond to qubit locations in Fig. 2b with qubits 0-7 in the top operation zones and 8-15 in the bottom (two per zone order left to right)

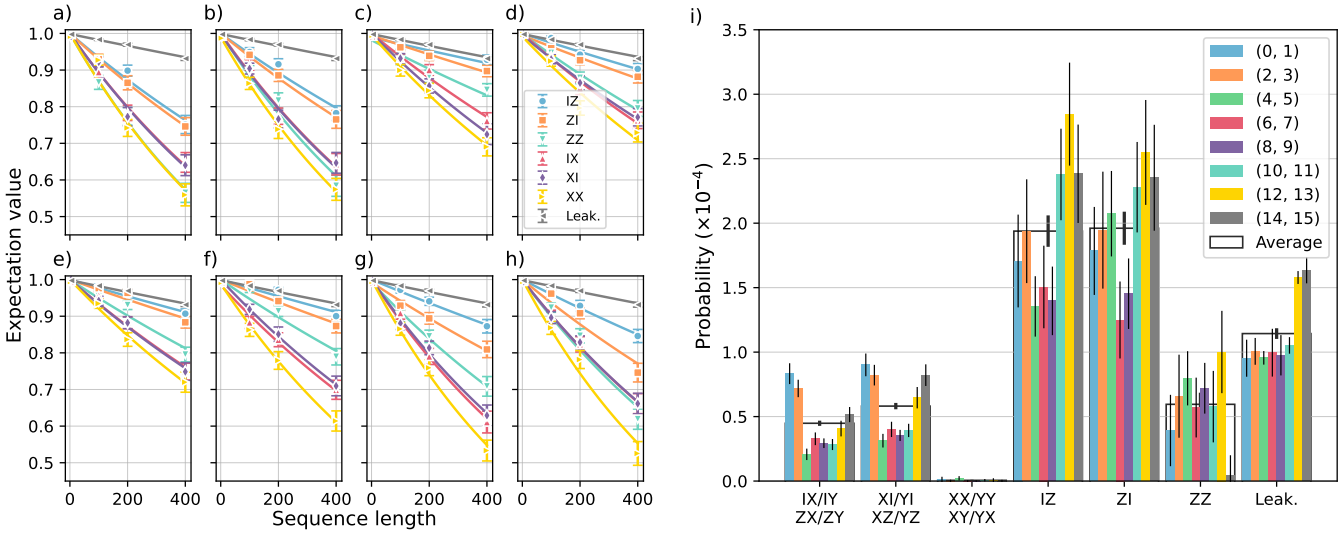

FIG. 7. 2QCB data: (a-h) Pauli expectation values and measured leakage rate as function of sequence length, for the 8 operation zones in order of Fig. 2 (i) Pauli error probabilities and leakage survival probability for the native $R_{ZZ}(\pi/2)$ gate, up to unlearnable degrees of freedom, for the 8 operation zones.

Our measured value of $7.9(2) \times 10^{-4}$ can be compared to a total expected error per 2Q gate of $3.5(4) \times 10^{-4}$, which we predict from an error budget consisting of spontaneous emission errors, memory error, and 1Q pulse errors plus other characterized experimental sources of noise such as laser phase and intensity noise, thermal motion of the ions, and imprecise calibrations. The discrepancy of the measured 2Q error with predicted value could be explained by a number of factors including higher leakage error in the operational zones due to finite extinction of the resonant detection beams present, non-thermal motional distributions, crosstalk, or other unaccounted for effects.

Just as with the 1QRB data, we performed a statistical test for the presence of correlated errors in the 2QRB data and found no significant evidence of correlated errors across the qubit pairs (see App. A3 A for details).

We also perform two-qubit cycle benchmarking (2QCB) [64] to estimate a partial Pauli error model for the 2Q gate in each operation zone, with the experimental and theoretical details supplied in App. A3 B. Fig. 7 shows the expectation value decays and estimated Pauli error channels, for each qubit pair. We find the zone-averaged infidelity is $8.1(2) \times 10^{-4}$, which includes a leakage rate of $1.14(4) \times 10^{-4}$, and is dominated by IZ and ZI errors. We note that our estimate of leakage rate per 2Q gate from 2QCB is about a factor of two smaller than the estimate from 2QRB.

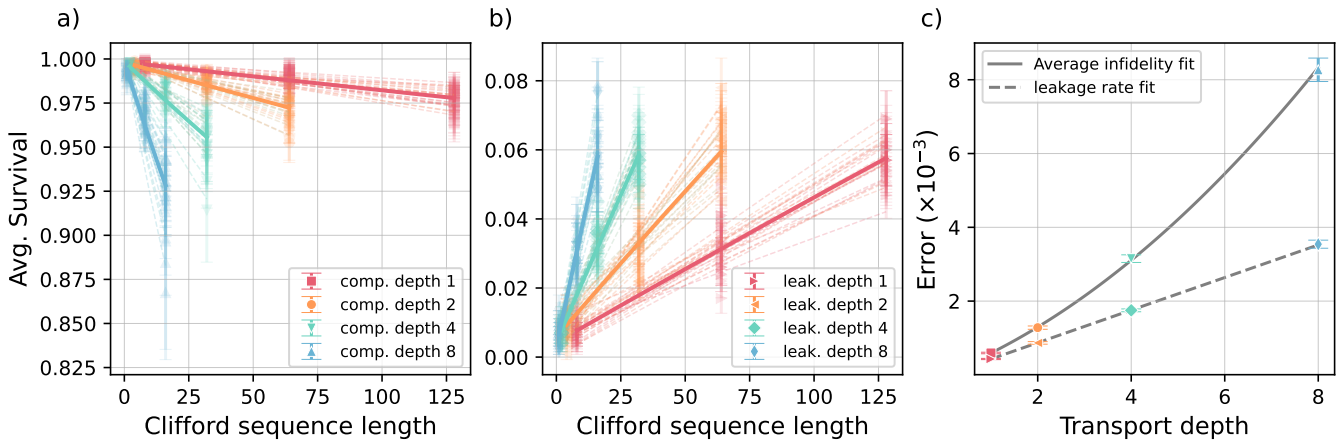

FIG. 8. Transport-1QRB data: (a) Survival probability as a function of Clifford sequence length for 98 qubits grouped into 4 groups. (b) Measured leakage population as a function of Clifford sequence length. (c) Qubit-averaged leakage rate (dashed curve) and total memory error (solid curve) as a function of the number of depth-1 transport operations.

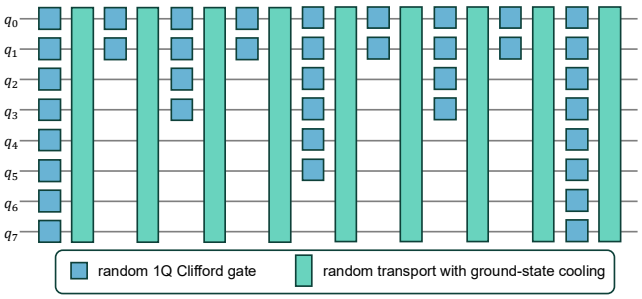

FIG. 9. Example circuit diagram for transport-1QRB where different qubits receive different number of transport rounds between gates: q_0 and q_1 have depth-1, q_2 and q_3 have depth-2, q_4, q_5 have depth-4, and q_6 and q_7 have depth-8.

4. *Transport idle memory errors*

Qubits idle during ion transport and cooling and incur memory errors due to spatiotemporal magnetic field inhomogeneities, with their impact being heavily dependent on the circuit structure and its specific transport schedule. As a figure of merit we define the depth-n memory error to be the average infidelity per qubit after randomly pairing all qubits, performing the transport and cooling operations that would be required to apply 2Q gates on all pairs (but no actual gate operations), and repeating this process n times.

We measure memory error with a variant of 1QRB that interleaves random transport between 1Q Clifford gates, referred to as transport-1QRB [10, 65]. Our method here differs from Ref. [10] in that we partition the 98 qubits into groups where the qubits in each group have a random 1Q Clifford operation applied after every k rounds of depth-1 transport operations as shown in Fig. 9. The qubits in the different groups will have a different amount of transport and idle time between Clifford operations, which allows us to extract how memory errors scale with the number of depth-1 transport operations for random circuits.

We run transport-1QRB circuits on the 98 qubits with one Clifford between every $k \in \{1, 2, 4, 8\}$ transport operations Additionally, we use the ternary measurement to extract any leakage errors during transport. Fig. 8a and b show the measured decay in transport-1QRB for computational and ternary measurements respectively. The decay curves are clustered into 4 groups determined by k. By fitting the decay curves and accounting for the leakage rate using the same procedure as in Sec. III B 2, we obtain the Clifford infidelity for each qubit.

Fig. 8c shows a plot of the Clifford infidelity as a function of the number of depth-1 transport operations, averaged over all qubits in the corresponding group. The expected scaling of memory error with delay time varies depending on the time scale of the noise sources [66]. For this reason we fit the memory error versus l to a quadratic equation $a + bl + cl^2$ where b and c capture the linear memory error rate (from fast noise) and quadratic memory error parameter (from slow noise), respectively [65].

From the fit to the data, we infer a linear memory error rate of $5(1) \times 10^{-4}$ and a quadratic memory error parameter of $7(2) \times 10^{-5}$. We find that the leakage error scales linearly with the number of transport operations, with a rate of $4.0(2) \times 10^{-4}$ and accounts for nearly all of the linear memory error. The expected coherent error from typical drift in magnetic fields between calibrations (every $\sim$ 5 s) of approximately 10 μG is 3×10^{-5} in a depth-1 circuit. The remaining coherent error may be explained by imperfections in the phase tracking routine or other unaccounted sources of noise.

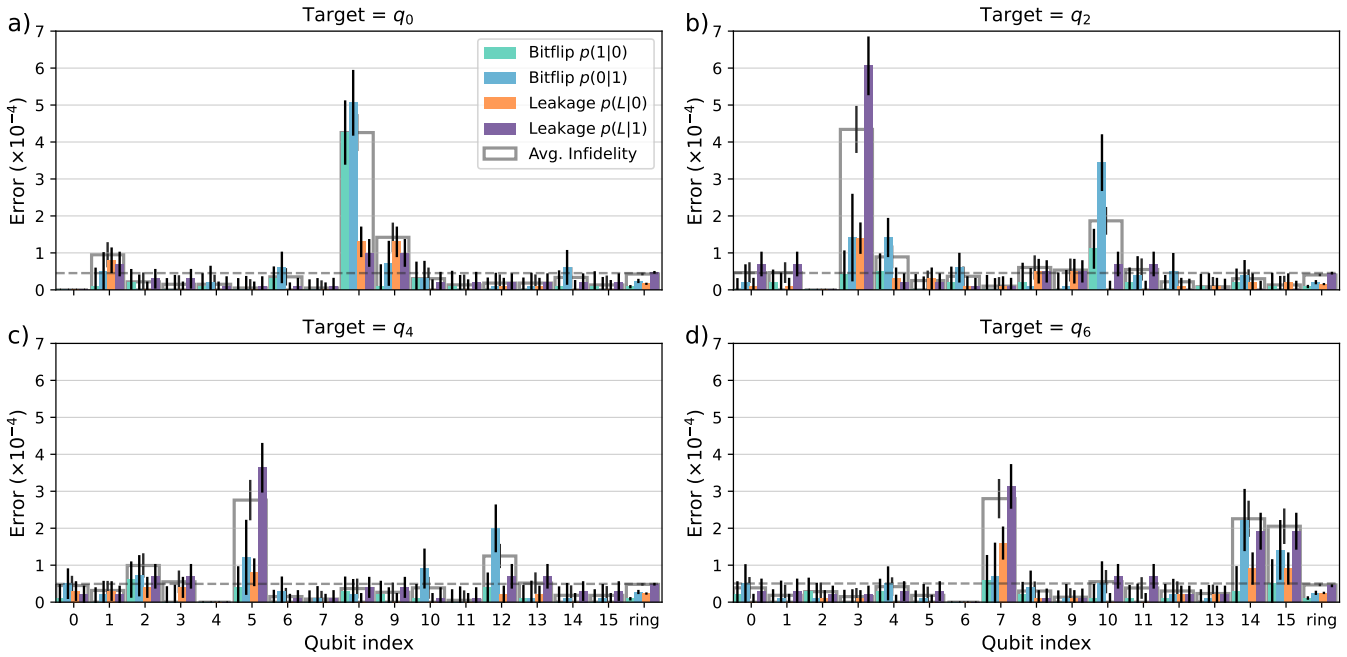

FIG. 10. MCMR crosstalk data with estimated rates of different error channels (smaller bars) and average infidelity (wider bars) with different individual target qubits (repeatedly MCMR'ed) and spectators (witnesses to crosstalk). We report conditional probabilities $p(i|j)$ for transitioning to state i given state j where $i, j = \{0, 1, L\}$ and L represents the sum of all leaked state populations. The x-axis labels the qubit locations in the quantum operation zones with 0-7 in the top and 8-15 in the bottom (left to right). The "ring" bin contains an average over the remaining 82 qubits that sat in the ring during the test. The dashed horizontal lines indicate the global average infidelity which is the mean crosstalk error for all 97 spectator qubits.

5. *Mid-circuit measurement and reset crosstalk*

MCMR causes crosstalk errors on un-measured or un-reset qubits that absorb stray measurement or reset light. The resulting spontaneous emission can lead to bit-flip, leakage, or dephasing errors.

We measure MCMR crosstalk errors by partitioning the 98 qubits into target qubits that are measured and reset repeatedly. Spectator qubits are prepared in the $|0\rangle$ or $|1\rangle$ and we use the ternary measurement at the end. The combination allows us to differentiate bit-flip rates from leakage rates to get a more detailed picture of the crosstalk error channel. The test was repeated for individual target qubits in the operation zones to illustrate the structure of MCMR crosstalk errors as shown in Fig. 10. Further details are provided in App. A 3 B.

It is clear that ions sitting adjacent to the 493 nm lasers applied to the target ion (in the same zone or neighboring zone above/below as shown in Fig. 2e) have much larger crosstalk errors. We distinguish between local (three ions that are laser-adjacent) and global (all 97 spectators) crosstalk, reporting per MCMR average crosstalk infidelities $2.1(1) \times 10^{-4}$ and $4.8(1) \times 10^{-5}$, respectively. The linear memory error rate (see Table II) contributes background leakage at a per MCMR rate of roughly $9(2) \times 10^{-6}$ to the measured average infidelities.

C. System-level benchmarks

1. *Random Clifford circuits with mid-circuit measurements*

To test the ability of Helios to execute arbitrary 98-qubit circuits using all primitive components, we run circuits with layers consisting of random Clifford 1Q and 2Q gates and MCMRs. Ref. [67] introduced circuits with random Clifford layers as a scalable system-level benchmark called binary randomized benchmarking (BiRB). An extension allowing for MCMRs was given in [68], called quantum instrument randomized benchmarking (QIRB). Our circuits are constructed similarly to Ref. [68] with a few small modifications. An example circuit diagram is shown in Fig. 11.

In our implementation, a length l circuit on N qubits with n_m MCMRs per layer consists of the following for each layer:

- A distinct uniformly random 1Q Clifford is applied to each qubit.
- The N qubits are uniformly randomly paired into $\lfloor \frac{N}{2} \rfloor$ qubit pairs, and the 2Q gate $R_{ZZ}(\pi/2)$ is applied to each pair, with Pauli-twirling applied to the 2Q gates.
- A uniformly random subset of n_m qubits are sampled, and for each qubit a 1Q Clifford is applied to

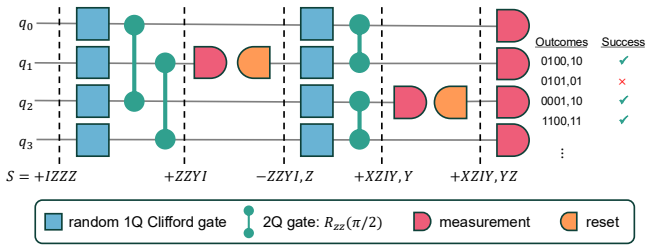

FIG. 11. Example of a random Clifford circuit with MCMRs, with parameters $N = 4$, $n_m = 1$, and $l = 2$. In each layer the qubits are randomly paired and gated and then a random subset of qubits is measured and reset. An initial random stabilizer S is chosen and evolved through the circuit to determine a binary outcome (success or failure) for each shot.

prepare a measurement in a particular Pauli basis, followed by an MCMR operation.

To classically verify correct circuit outputs, we track a random initial stabilizer through the circuit, as explained in App. A3 C. The parity of the evolved stabilizer defines a success/failure trial. For the purpose of fidelity estimation, the average success probability is rescaled into a quantity called the polarization [67], defined as $y_{pol} = 2p_{succ} - 1$. A polarization of 1 corresponds to perfect success, whereas a polarization of 0 corresponds to 50% success, or random guessing. A plot of $y_{pol}(l, n_m)$ versus l for different values of n_m is shown in Fig. 12a. Let $F(n_m)$ be the process fidelity per circuit layer as a function of n_m. We estimate $F(n_m)$ by fitting the polarization to an exponential decay model.

Figure 12b shows a plot of $F(n_m)$ versus n_m. We note that the layer fidelity actually increases slightly (with overlapping error bars) as n_m increases from 8 to 16. This is explained by the fact that a batch of 16 measurements in the operation zones utilizes the protected measure scheme (explained in Fig. 3b), which protects against MCMR crosstalk in the operation zones.

To see whether the results are consistent with our component benchmarks, we first compute an effective 2Q gate error ϵ_{2Q} from the $n_m = 0$ data, using

$$F(n_m = 0) = (1 - 5\epsilon_{2Q}/4)^{\lfloor \frac{N}{2} \rfloor}, \qquad (1)$$

where the factor of 5/4 comes from the conversion between process and average fidelity [69]. The effective 2Q gate error includes errors from 2Q gates, 1Q gates, and memory errors, and it can be thought of as the infidelity of a 2Q depolarizing channel that would best fit the data in the absence of all other errors. We find $\epsilon_{2Q} = 2.0(3) \times 10^{-3}$, whereas an accounting of 2Q and memory errors from Tab. II predicts $2.2(1) \times 10^{-3}$ (see Sec. A3 C for details).

We next compute effective MCMR errors ϵ_M for the $n_m = 8$ and $n_m = 16$ data, using the heuristic formula

$$F(n_m) = (1 - 5\epsilon_{2Q}/4)^{\lfloor \frac{N}{2} \rfloor}(1 - 3\epsilon_M/2)^{n_m} \qquad (2)$$

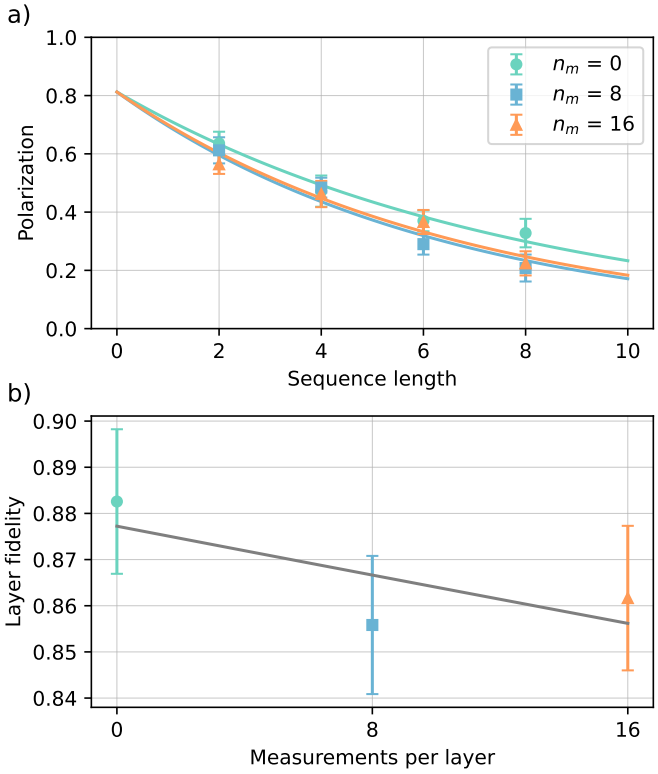

FIG. 12. Random Clifford circuits with MCMR data: (a) Polarization versus sequence length for different numbers of MCMRs per layer. (b) Layer fidelity versus number of MCMRs per layer.

together with our computed value of ϵ_{2Q}. We find $\epsilon_M(n_m = 8) = (2.6 \pm 1.3) \times 10^{-3}$ and $\epsilon_M(n_m = 16) = 1.0(7) \times 10^{-3}$. By comparison, adding the component-level SPAM error and the MCMR crosstalk error, we predict effective MCMR errors of $2.2(1) \times 10^{-3}$ and $1.7(1) \times 10^{-3}$ for $n_m = 8$ and $n_m = 16$ (see Sec. A3 C for details). We conclude that the data from our random Clifford with MCMR circuits is consistent with our measured component-level 2Q error, but tighter error bars are needed to assess the consistency of the effective MCMR errors. We remark that our method of comparison is heuristic and a rigorous methodology for comparing component-level to system-level benchmarking performance is an open problem.

2. *RCS mirror benchmarking*

Random circuit sampling (RCS) is a system-level benchmark assessing how effectively a quantum computer can generate computationally complex quantum states [1]. Like BiRB, RCS probes the extent to which quantum circuits obtain the performance expected from component-level benchmarks. At the same time, because the classical difficulty of sampling from the out-

puts of random quantum circuits has been extremely well-studied over the last decade [70], RCS provides a well-vetted benchmark for the computational power of a quantum computer.

Leveraging the arbitrary connectivity of the Helios quantum computer, we consider RCS with circuit geometries constructed from colorings of random-regular graphs [3]: A layer depth-l random circuit is constructed by interleaving l layers of 2Q $R_{ZZ}(\pi/2)$ gates (each layer containing $N/2$ 2Q gates) with $l+1$ layers of Haar-random 1Q gates (each layer containing N 1Q gates). While the fidelity of such circuits can in principle be inferred by running them and performing cross-entropy benchmarking [72], evaluating the cross-entropy requires exact simulation of the circuits in question and is infeasible except for small depth or qubit number. To estimate the expected state fidelity in RCS (and therefore the anticipated performance in cross-entropy benchmarking), we follow the strategy of Refs. [3, 73–76] and infer the fidelity of a layer depth-l circuit by computing the return-probability F_{MB} of a "mirrored" layer depth-$l/2$ circuit, with the second (mirrored) half of the circuit employing randomized compiling to prevent unintended cancellation of coherent errors. The randomness for randomized compilation is sampled in real-time at the start of each shot, and the corresponding random 1Q gates are compiled on the fly (with the existing Haar-random 1Q gates), resulting in only one physical 1Q gate per qubit per layer. Following Ref. [3], we also initialize each mirrored circuit into a random computational basis state to prevent unequal SPAM errors between the two basis states from biasing the fidelity estimate. At each depth, we execute between 1000 and 2500 shots spread evenly across 100 random circuit connectivities.

The fidelity of RCS as a function of depth inferred in this manner is reported in Fig. 13a. We perform a least-squares best fit to the gate-counting model from [3],

$$F_{GC}(l) = (1-p_{\text{spam}})^N \left(1-\frac{5}{4}\epsilon_{2Q}\right)^{\frac{N}{2}(l-\delta)}. \qquad (3)$$

Here, $N=98$, $\delta=1.12$ is a correction to effective circuit layer depth from boundary effects in mirror circuits [3], p_{spam} is the effective SPAM error, and ϵ_{2Q} is the effective average 2Q error rate, which includes effects from 1Q, 2Q, and memory errors as in the previous section. From the fit, we estimate $p_{\text{spam}} = 5.3(51)\times 10^{-4}$ and $\epsilon_{2Q} = 2.00(6)\times 10^{-3}$. This effective 2Q error is also consistent with the estimate obtained from random Clifford circuits as well as component benchmarks reported in Table II.

Heuristic estimates of the classical cost of drawing samples from forward circuits at the same depths is shown in Fig. 13b. The reported costs are for optimized tensor-network contraction assuming so-called "embarrassing parallelization" (via slicing) into independent computations involving various amounts of available memory, and were obtained using (sliced) simulated annealing built into `cotengra` [77]. We note that the contraction-cost optimization performed here is only approximate, and the costs could certainly be mildly improved by providing the optimization heuristics with more computational power. However, we do not expect such improvements

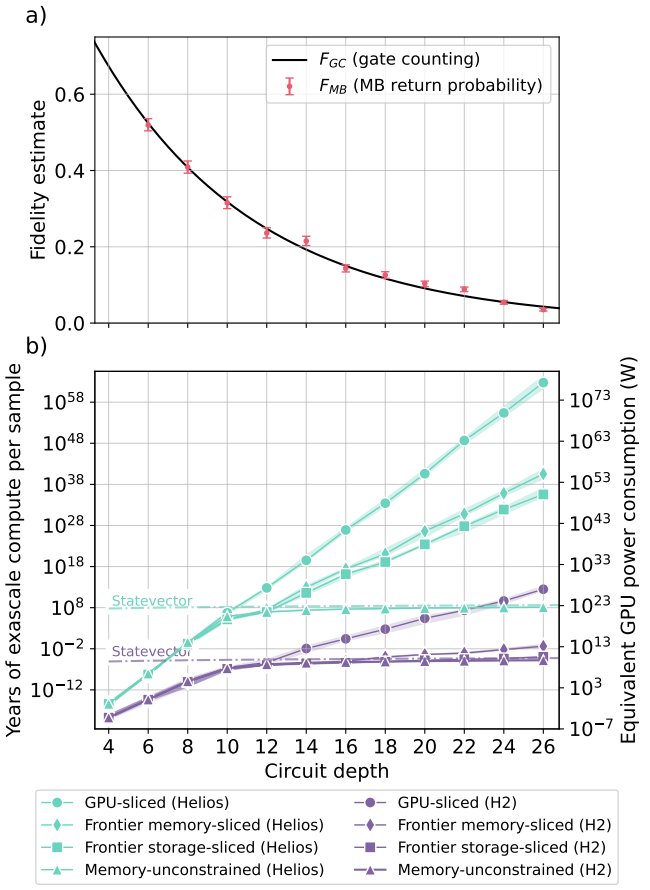

FIG. 13. (a) Fidelity of $N=98$ mirrored RCS circuits as a function of circuit depth (red). The best-fit gate-counting curve is overlaid (black), demonstrating consistency with an exponential decay with depth. (b) Estimated cost of classical sampling via tensor-network contraction from RCS circuits of varying depth on both H2 (purple) and Helios (teal). The left axis reports time in years required to draw a single sample by tensor-network contraction on a state-of-the-art supercomputer (achieving about 10^{18} FLOPs per second). The right axis shows the required power (assuming state-of-the-art GPU power efficiency of roughly 10^{11} FLOPs/W) to perform contraction-based sampling at the same rate that Helios can draw samples. Costs are quoted across different assumptions on the total memory footprint of the contraction (in a similar fashion to [71]), corresponding to the `cotengra` contraction width $\mathcal{W}$. Triangles show costs assuming access to unlimited memory ($\mathcal{W}=\infty$), which saturates at large depths to the $\sim 2^N$ scaling of statevector simulation; squares ($\mathcal{W}=54$) allow use of all external storage of the Frontier supercomputer, while diamonds ($\mathcal{W}=49$) restrict to the available memory on Frontier, and circles ($\mathcal{W}=30$) correspond to spreading the slices independently among state-of-the-art GPUs. Shaded bands indicate the range of costs obtained over 5 random circuit instances at each depth, with the markers indicating the median cost.

to change the overall conclusion that Helios can produce states at high global fidelity for which the (classical) sampling cost is vastly beyond the capabilities of existing supercomputers.

IV. OUTLOOK

In this manuscript, we reported on how Helios operates and its current performance. Even at this early stage in its lifecycle, Helios exhibits state-of-the-art capabilities at the scale of ~ 100 qubits. Like its predecessors Quantinuum H1 and H2, we expect Helios's performance to improve over time. Examples of relatively straight-forward performance improvements include: (1) fewer gate errors as our two-qubit gate error model suggests the 2Q gate error could be cut in half, (2) smaller memory errors using dynamic decoupling strategies [78] and (3) reduced circuit times from both faster transport operations [79–81] and better compilation methods.

Beyond these performance improvements, increasing clock speed is one scaling challenge for the QCCD platform. In this work we begin to address this issue through a fundamental architectural shift by parallelizing operations [17]. Previous generations, H1 and H2, used the same space for ground-state cooling and gating operations, with cooling operations being up to two orders of magnitude slower [9, 10]. Helios, on the other hand, spreads the cooling operation over space to allow ions to spend less time in the zones used for 2Q gates. By increasing the ratio of cooling zones to gate zones, future QCCD-based QPUs can optimize the processor zone complexity while simultaneously increasing the clock speed.

While we do not yet fully understand the power or limitations of Helios, the combination of a new qubit choice, device architecture, and control software runtime already represents significant progress in the push for more powerful devices, scalable architectures, and capabilities for fault-tolerant computation. Helios is far beyond the simulation abilities of classical computers, as evidenced by the RCS demonstration described above, and well poised to expand upon the set of tasks best suited for contemporary quantum computers. Indeed, as reported in Refs. [51, 82, 83], Helios is already enabling advancements in quantum simulations of superconductivity and in cryptographic protocols to generate certified randomness.

Looking further ahead, the successful integration of the four-way junction paves the way for much larger QCCD processors. Junction-based architectures should allow QCCD machines to maintain all-to-all connectivity for large numbers of qubits, opening the design space for fault-tolerance to high-efficiency encodings [84], transversal logic [85, 86], low-overhead magic state factories [87], and single-shot error correction [88–90].

ACKNOWLEDGMENTS

We thank the entire Quantinuum team for numerous contributions that enabled this work. The contributions of the SNL authors were funded in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Testbed Pathfinder Program and in part by an Office of Advanced Scientific Computing Research Early Career Award. Sandia National Laboratories is a multi-program laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. All statements of fact, opinion, or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of the U.S. Department of Energy or the U.S. Government.

DATA AVAILABILITY

Most of the component benchmarking data are available on the Quantinuum website and will be updated as Helios improvements are introduced.

APPENDICES

A1. HARDWARE DETAILS

A. Quantum logic

For 2Q gates, we create a Mølmer-Sørensen[30] interaction by using pairs of Raman beams aligned at 90 degrees to each other with the $\delta\vec{k}$ aligned along the axes of BYYB crystals, and we use the axial stretch mode at 1.86 MHz to couple the internal states of the ions. The uncontrolled optical phase of the gate is removed using wrapper pulses to generate a ZZ interaction [9, 31]. 2Q gates can be performed in four of the operation zones.

As in Ref. [32], state-preparation uses narrow-band optical pumping by driving $S_{1/2}$ leakage states first to $D_{5/2}$ with a narrow linewidth 1762 nm laser, and then to $P_{3/2}$ $F = 0$ using 614 nm light where it will decay back to $S_{1/2}$ leading to population accumulation in the qubit subspace. The measurement protocol begins by transferring the $|F, m_F\rangle = |1, 0\rangle$ qubit state to the $D_{5/2}$ manifold (shelving) with the 1762 nm laser, and population remaining in the $S_{1/2}$ manifold is measured with resonant fluorescence. We reduce measurement crosstalk by shelving all qubits located in the quantum logic region before measurement if an entire batch of 16 qubits is to be measured, called "protected measure." Furthermore, the end user can also choose to shelve both $|1, 0\rangle$ and $|2, 0\rangle$ qubit states and check for leakage out of the qubit subspace, called "ternary measure", and then measure the qubit state by de-shelving one of the qubit states and applying resonant light to check for fluorescence. The ternary measurement doubles the measure time as shelving-and-detect needs to be performed twice. All shelving operations use multiple pulses (cabinet shelving) with different final states to exponentially reduce population transfer errors.

B. Ground state cooling

For the $^{171}Yb^+$ coolant ion, the nuclear spin $I = 1/2$ allows for a fast frequency selective state-preparation scheme not reliant on a particular polarization [37]. For ground state cooling, we use counter-propagating linperp-lin Raman beams aligned at 45 degrees to the crystal axis to get a $\delta\vec{k}$ projection on all three principal axes (the radial modes are rotated so as to not be orthogonal or parallel to the trap surface). The parallel cooling is achieved using 5 pairs of Raman beams detuned from the $S_{1/2}$ to $P_{1/2}$ transition near 369.4 nm. The laser beam angles are aligned to 45 ± 0.2 degrees with respect to the storage legs such that the beams can simultaneously intersect an operational zone in each leg. The beam waist focii are positioned between the two zones so each zone has the same beam waist and intensity. Carrier Rabi rates of up to 1 MHz are achieved in all zones for the cooling operations. In this configuration, we perform sideband cooling sequences to achieve ground state cooling times of approximately 3 ms [9].

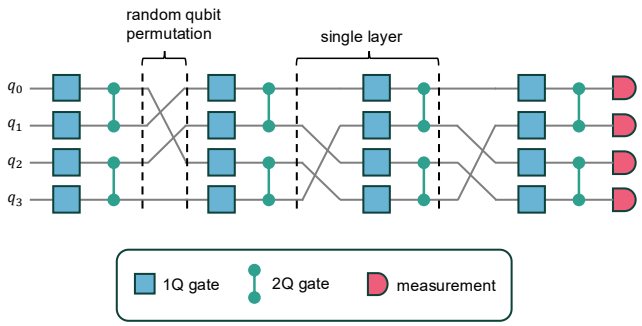

FIG. A1. Example four-qubit program where all qubits are arbitrarily permuted and receive 1Q and 2Q gates each layer.

C. Qubit frequency calibrations

Helios operates with an externally imposed bias field of 3.95 G, making the qubit states approximate clock states, meaning they are naturally robust to magnetic field fluctuations with a second-order field coefficient of 488.8 Hz/G^2 (at zero field they become true clock-states that are insensitive to magnetic fields up to second order).

Variations in the qubit frequency arise primarily from the slow drift of magnetic fields at the level of ~ 200 μG (over 24 hrs) and their gradient, as well as varying AC Zeeman shifts of the clock transition in $^{137}Ba^+$ from the trap RF current. To mitigate these effects, we employ a real-time spatial phase tracking routine [85]. The routine gets corrections to the reference qubit frequency from measurements of the average magnetic field in the quantum operation zones and the spatially-varying magnetic field in all 277 wells. After these calibrations, the routine applies the appropriate corrections to 1Q operations.

A2. PROGRAM PROFILING

To profile programs written in Guppy (see Sec. II D), the compiled code is executed on a real-time control system simulator. Although this simulator is separate from Helios, it accurately captures timing information by using the same compiler, real-time software, and system settings used on Helios.

In Fig. 4b of the main text, we present timing results for three example programs run on the control system simulator. The first two programs consist of 1Q and 2Q gates executed on arbitrary qubit pairs, which are randomized each layer. Fig. A1 illustrates their structure using an example four-qubit program with four layers. To get an accurate estimate of the time per layer, we

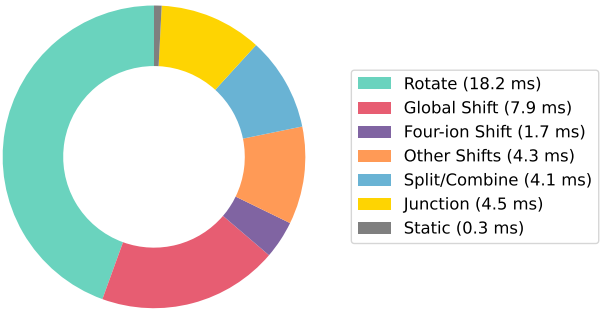

FIG. A2. Timing breakdown of transport operations for a single layer of the 98-qubit program profiled in Fig. 4a of the main text. Operations are broken down into the five categories described in Sec. II C, plus four additional ones: global shift operations that collectively move ions in the cache, quantum operation, and storage leg regions; four-ion shift operations; all other shift operations; and static operations that do not move ions.

exclude reset and measure operations that occur during the first and last layers of the program. We reduce the density of 2Q gates to roughly half by randomly selecting qubit pairs and gate them such that the 2Q gates occur $\lfloor d\frac{N}{2} \rfloor$ times per layer where N is the number of active qubits in the program and $d = 1$ ($d = \frac{1}{2}$) corresponds to fully (half) dense.

For the fully dense random program, Fig. A2 shows a breakdown of the transport operation times per layer. Ring rotations dominate, while global shifts are the second-largest contributor. Future work will focus on reducing the total time spent on transport operations, thereby improving the depth-1 time. For example, compiler optimizations can reduce the number of transport operations in a program, while improvements in transport operation speed can lower their execution time.

The third program we profile uses 2D nearest-neighbor qubit pairing, which reflects common use cases such as quantum error correction and quantum simulation. While similar to the previously discussed example, this program restricts qubit pairings to one of four possible configurations on a square grid of qubits described as follows. Configurations (1) and (2) define horizontal pairings that alternate by row such that in (1), even rows pair adjacent qubits starting at the first column, while odd rows start at the second column; configuration (2) reverses this pattern. Configurations (3) and (4) follow the same alternating pattern but for vertical pairings, alternating the starting row by column parity: (3) starts at the first row for even columns and the second row for odd columns, while (4) reverses this pattern. Each layer in the program applies one of these pairing configurations, cycling through all four configurations every four layers.

A3. BENCHMARKING DETAILS

A. Correlated Error Analysis for randomized benchmarking

In this section, we detail our method for identifying correlated errors in simultaneous 1QRB and 2QRB experiments. Our analysis for correlations uses *subsystem RB polarizations*. Consider an experiment of k simultaneous RB experiments (e.g. qubits in operation zones of Helios). The result of running a simultaneous RB circuit consisting of parallel RB circuits on k disjoint qubit subsets, labeled $1, 2, \ldots, k$, is described by a k-bit string $s_1 s_2 \ldots s_k$, where $s_i = 0$ if the bit string outcome of the RB subexperiment on qubit subset i matches its target outcome, and $s_i = 1$ otherwise. A *subsystem parity* z_S for $S \subseteq \{1, 2, \ldots, k\}$ is defined as

$$z_S = \prod_{j \in S} (-1)^{s_j}, \tag{A1}$$

and the expected value $\langle z_S \rangle$ is called a *subsystem polarization*.

The *subsystem decay factor* λ_S is found by fitting the empirical values of $\langle z_S \rangle$ to an exponential decay, i.e. $\langle z_S \rangle_l = \lambda_S^l$, where l is the circuit depth. In the absence of correlated errors, $\lambda_S = \prod_{k \in S} \lambda_k$. λ_S is a Pauli channel eigenvalue for the twirled error channel produced by Clifford twirling on the individual qubit subsystems on which RB is performed in parallel. Each $\lambda_{\{j\}}$ is simply one of the eigenvalues of a Clifford-twirled error channel, and the total error rate of the channel is $\frac{4^n-1}{4^n}(1 - \lambda_{\{j\}})$, where n is the number of qubits in the RB subexperiment.

To search for evidence of correlated errors across multiple RB subexperiments, and quantify any such errors, we start by estimating $\lambda_{\{i\}}$ for each subexperiment i. Then, we estimate $\lambda_{\{i,j\}}$ for each pair (i, j) of subexperiments and compute the test statistic

$$a = \log(\lambda_{\{i,j\}}) - \log(\lambda_{\{i\}}) - \log(\lambda_{\{j\}}). \tag{A2}$$

In the absence of correlated errors between subsystems i and j, $\lambda_{\{i,j\}} = \lambda_{\{i\}}\lambda_{\{j\}}$ and so $a = 0$, and if there are any correlated error between subsystems i and j, then $\lambda_{\{i,j\}} > \lambda_{\{i\}}\lambda_{\{j\}}$ and so $a > 0$. In our analysis, we only have access to estimates of each λ_S, which we denote by $\tilde{\lambda}_S$, and a corresponding estimate of a, denoted $\tilde{a}$. We therefore compute $\tilde{a}$ and implement a statistical test to ascertain whether $\tilde{a}$ is large enough to conclude that $a > 0$ with at least 95% confidence (an $\alpha = 0.05$ significance). We use a normal approximation for $\tilde{a}$'s distribution under the null hypothesis (no correlated errors), with a standard deviation for this distribution estimated using a non-parametric bootstrap. As we test for correlated errors between all k choose 2 pairs of subsystems, we implement our individual hypothesis tests (of whether each $a > 0$) at $\frac{\alpha}{\binom{k}{2}} = \frac{0.05}{\binom{k}{2}}$ significance (a Bonferroni correction). This means that if there are no correlated errors,

we will erroneously conclude there are correlated errors with at most 5% probability, known as the family-wise error rate of the hypothesis tests.

We find that none of the a are larger than zero, in our 95% confidence hypothesis test, indicating no statistically significant evidence for correlated errors. Sufficiently small correlated errors will probably not be detected by this analysis. Using simulations, we can estimate what fraction of the error would have to be correlated error in order for our analysis to detect it. For the simultaneous 2QRB experiments we find that, given the estimated RB error rates from the experimental data, a two-subsystem correlated error would need to constitute approximately 10% of the total error rate of the constituent subsystems (i.e., contributing 4.3×10^{-4} to the average 2QRB infidelity) to be identified as statistically significant in our analysis with at least 50% probability. For the 1QRB experiments, a correlated error would need to constitute approximately 50% (i.e., contributing 1.19×10^{-5} to the average 1QRB infidelity) of the total error rate of the constituent subsystems to be identified as statistically significant in our analysis with at least 50% probability.

B. Detailed component benchmarking data and experimental details

For component-level benchmarks including 1QRB, transport-1QRB, 2QRB, CB, and MCMR crosstalk, all error rates reported in the main text are the average infidelity, defined as follows. Let $\mathcal{E}$ be the error process (a completely-positive map) for a given operation $\mathcal{U}$, such that its noisy implementation is given by $\mathcal{E} \circ \mathcal{U}$. Then the average infidelity is

$$\epsilon_{avg}(\mathcal{E}) = 1 - \int d\psi \, \langle\psi| \, \mathcal{E}(|\psi\rangle \langle\psi|) \, |\psi\rangle \,, \qquad \text{(A3)}$$

where the integral is taken over all pure states in the computational Hilbert space with respect to the Haar measure.

1. SPAM

For $a, b \in \{0, 1\}$, let $p(a|b)$ denote the probability of measuring outcome a given state preparation b. For the standard measurement, we find $p(1|0) = 8.1(1) \times 10^{-4}$ and $p(0|1) = 1.6(5) \times 10^{-4}$. For the ternary measurement, we find leakage probabilities of $p(L|0) = 2.7(8) \times 10^{-3}$ and $p(L|1) = 5.7(1) \times 10^{-3}$, and SPAM errors of $p(1|0) = 7(1) \times 10^{-4}$ and $p(0|1) = 2.8(2) \times 10^{-3}$, conditioned on non-leakage.

2. Single-qubit RB

In 1Q Clifford RB, a sequence of l uniformly random Clifford group elements are applied to a qubit, followed by an inverse Clifford that randomly includes a bit-flip (X) gate. In the absence of error, this process prepares the qubit in a random computational basis state. In our decomposition of the 1Q Clifford group into native gates, the 24 group elements have 0.375 pi/2 pulses and 0.75 pi pulses on average.

We run 1QRB simultaneously on 16 qubits in the 8 operation zones with different random sequences applied to each qubit [91]. We use sequence lengths $l \in \{10, 1000, 2000\}$, generate 10 circuits per each sequence length, and run 100 shots of each circuit. Table A1 lists the measured leakage rates and average infidelities (including the contribution from leakage) for each individual qubit.

TABLE A1. 1QRB leakage rate and average infidelity.

Qubit	Leakage Rate ($\times 10^{-5}$)	Avg. Infidelity ($\times 10^{-5}$)
q_0	0.9(1)	2.9(8)
q_1	1.4(0)	3.1(7)
q_2	1.4(1)	2.3(4)
q_3	1.5(4)	2.8(4)
q_4	1.2(3)	2.4(4)
q_5	1.3(0)	2.0(2)
q_6	0.8(2)	2.7(4)
q_7	1.0(0)	3.1(6)
q_8	1.5(2)	2.5(3)
q_9	1.6(0)	2.5(3)
q_{10}	0.9(5)	2.1(3)
q_{11}	0.8(1)	2.5(7)
q_{12}	1.2(1)	2.3(3)
q_{13}	1.4(4)	2.4(4)
q_{14}	0.5(2)	2.1(4)
q_{15}	0.6(0)	2.5(4)
Mean	1.12(7)	2.5(1)

3. Two-qubit RB

Like 1QRB, 2QRB is performed by executing sequences of l uniformly random Clifford group elements (now drawn from the 2-qubit Clifford group). A final inverse Clifford then ideally prepares the qubit pair in a random computational basis state. The 2QRB circuits are performed on 8 pairs of qubits initialized in the 8 operation zones, each with a distinct random sequence. As described in Sec. II C, the 2Q gates are applied in parallel in only four out of eight zones. We select 8 pairs of qubits for benchmarking as this configuration corresponds to a typical batch of parallel operations during circuit execution.

TABLE A2. Two-qubit RB leakage rates and average infidelities. Following the protocol described in Sec. II C for performing 2Q gates on eight qubit pairs in the operation zones using 2Q gates applied in only four zones, pairs $(0,1)$ and $(2,3)$ utilize the same zone for the 2Q gate operation, similarly for pair sets $(4,5)$, $(6,7)$ and $(8,9)$, $(10,11)$ and $(12,13)$, $(14,15)$. Most sets of the qubit pairs utilizing the same zone have infidelities and leakage rates that agree within uncertainties, to the extent there are differences they may arise from differences in the 1Q gates, memory errors, and cooling, which occur in the eight separate zones.

Qubit Pair	Leakage Rate ($\times 10^{-4}$)	Avg. Infidelity ($\times 10^{-4}$)
$(0,1)$	1.8(3)	6.1(5)
$(2,3)$	2.2(1)	7.7(5)
$(4,5)$	2.6(2)	8.2(6)
$(6,7)$	2.1(1)	6.7(5)
$(8,9)$	2.7(1)	8.0(5)
$(10,11)$	3.3(5)	8.8(6)
$(12,13)$	2.3(2)	8.7(5)
$(14,15)$	2.6(2)	8.7(6)
Mean	2.4(1)	7.9(2)

4. *Two-qubit cycle benchmarking*

2QCB works by preparing eigenstates of a Pauli operator P, applying a Pauli-twirled 2Q gate l times, and measuring in the P basis. We Pauli-twirl [92] the 2Q gates so that the error channel $\mathcal{E}$ can be assumed to be a stochastic Pauli channel, which is defined as

$$\mathcal{E}(\rho) = \sum_i p_i P_i \rho P_i, \tag{A4}$$

where the sum is over all Pauli operators modulo an overall phase, and the p_i are probabilities that sum to one.

The eigenoperators of any stochastic Pauli channel are themselves Pauli operators, so $\mathcal{E}(P_i) = f_i P_i$, and their eigenvalues f_i are often called Pauli fidelities and are given by

$$f_i = \sum_j (-1)^{\langle i,j \rangle} p_j, \tag{A5}$$

where the symbol $\langle i,j \rangle$ equals 0 or 1, depending on whether P_i and P_j commute or anti-commute, respectively. The Pauli error probabilities can be computed from the Pauli fidelities via

$$p_i = \frac{1}{d^2} \sum_j (-1)^{\langle i,j \rangle} f_j, \tag{A6}$$

where d is the Hilbert space dimension. Denote the noisy 2QCB circuit of length l as $\mathcal{C}_l$. 2QCB estimates the Pauli fidelities by fitting the empirical expectation values $\mathbb{E}_l(P_i) = \mathrm{Tr}(P_i \mathcal{C}_l(P_i))$ to the model $\mathbb{E}_l(P_i) = A f_i^l$. The Pauli error probabilities are then computed via Eq. (A6).

In terms of the Pauli error probabilities, the average infidelity (not including leakage) is given by

$$\epsilon_{avg}(\mathcal{E}) = \frac{d}{d+1} \sum_{i>0} p_i, \tag{A7}$$

where the sum is over all non-identity Pauli probabilities.

Because gate sets have a gauge freedom, not all d^2 individual Pauli fidelities can be learned in a SPAM robust way, but rather, only the geometric means of subsets of Pauli fidelities that are related to each other by the action of the gate being benchmarked [93]. We therefore assume that pairs of Pauli operators within the same orbit of $R_{ZZ}(\pi/2)$ have the same fidelity (for example: $f_{IX} = f_{ZY}$). Furthermore, simulations of known error sources in $R_{ZZ}(\pi/2)$ shows symmetry between X and Y in the Pauli fidelities. We therefore only estimate f_i for $P_i \in \{IZ, ZI, ZZ, IX, XI, XX\}$, and we assume any two Paulis that are related by an X-Y symmetry to have the same fidelity (i.e., $f_{XY} = f_{XX}$).

In our 2QCB experiment, we prepare the 8 states in the tensor product bases $\{|0\rangle, |1\rangle\}^{\otimes 2}$ and $\{|+\rangle, |-\rangle\}^{\otimes 2}$, apply the $R_{ZZ}(\pi/2)$ gate l times with $l \in \{4, 100, 200, 400\}$, and measure each qubit in the same basis that it was prepared in. For each state preparation and sequence length we run 200 shots and employ runtime randomness in the software stack to implement single-shot Pauli-twirling on the 2Q gates. As in 1QRB and 2QRB, we use the ternary measurement and fit the probability of not leaking versus l to infer a leakage rate per gate. The non-leaked population is then used to compute expectation values that decay with l. We perform the experiment in parallel on 8 qubit pairs initialized in the 8 operation zones as in 2QRB and randomize the order of state preparations within each zone. The leakage rates and average infidelities (including leakage) are listed in Tab. A3. The zone-averaged Pauli error probabilities, up to unlearnable degrees of freedom and symmetry assumptions, are listed in Tab. A4.

TABLE A3. 2QCB estimated leakage rates and infidelities. Qubit pairs that share zones are the same as in A2

Qubit Pair	Leakage Rate ($\times 10^{-4}$)	Avg. Infidelity ($\times 10^{-4}$)
$(0,1)$	1.0(1)	9.6(4)
$(2,3)$	1.0(1)	9.6(5)
$(4,5)$	1.0(1)	6.1(4)
$(6,7)$	1.0(2)	6.0(4)
$(8,9)$	1.0(2)	5.9(3)
$(10,11)$	1.1(1)	7.4(4)
$(12,13)$	1.6(1)	10.1(4)
$(14,15)$	1.6(1)	9.7(5)
Mean	1.14(6)	8.1(2)

TABLE A4. 2QCB estimated Pauli error probabilities, averaged over all qubit pairs, up to unlearnable degrees of freedom and symmetry assumptions. For error classes with greater than one element, the right column is the probability of every Pauli error in the set.

Error Class	Probability ($\times 10^{-5}$)
$\{IX, IY, ZX, ZY\}$	4.5(2)
$\{XI, YI, XZ, YZ\}$	5.8(2)
$\{XX, XY, YX, YY\}$	0.06(4)
$\{IZ\}$	19(1)
$\{ZI\}$	19(1)
$\{ZZ\}$	5.9(9)

5. *Transport-1QRB*

We ran transport-1QRB with $k \in \{1, 2, 4, 8\}$ and sequence lengths $l \in \{8, 64, 128\}$, where sequence length here refers to the number of depth-1 transport operations. For each sequence length, we generate 10 circuits and run each circuit for 100 shots.

TABLE A5. Transport-1QRB leakage rates and average infidelities. Transport depth is the number of depth-1 transport operations between 1Q Cliffords. The reported numbers are averaged over qubits with a given transport depth.

Transport Depth	Avg. Leakage Rate ($\times 10^{-4}$)	Avg. Infidelity ($\times 10^{-4}$)
1	4.4(2)	6.0(3)
2	8.3(5)	12.8(7)
4	17.1(5)	34(2)
8	35(2)	84(5)

6. *MCMR crosstalk test*

We quantify MCMR crosstalk errors by fitting the spectator qubit survival probabilities to a linear decay model as a function of the number of applied MCMRs to the target qubits. We relate the fit parameters to error magnitude based on an effective quantum jump operator description of the error channel. This is an expansion of previous work on "bright-state depumping" for $^{171}Yb^+$ qubits [40] where the decay rate of spectator qubits prepared in the $|1\rangle$ state was used to determine the average infidelity. For $^{137}Ba^+$ qubits, however, the added complexity of the crosstalk decay channels requires that the spectator qubits be prepared in additional states. Furthermore, the ternary measurement is used to resolve bit-flip from leakage errors.

The MCMR crosstalk error channel is modeled as a set of effective quantum jump operators $\hat{L}_{ij} = \sqrt{\gamma_{ij}}|i\rangle\langle j|$ occurring at rates γ_{ij} between states i and j with $i, j \in \{0, 1, L\}$, leading to population transfer and decoherence [94, 95]. Evaluating Eq. (A3) results in an average infidelity

$$\epsilon_{avg}(\mathcal{E}) = \frac{1}{6}\left(p(0|1) + p(1|0) + 2p(L|0) + 2p(L|1) + 4p_Z\right) \tag{A8}$$

where $\mathcal{E}$ is the crosstalk error channel, $\epsilon_{avg}(\mathcal{E})$ is the average infidelity, $p(i|j)$ is the conditional probability for transitioning from state j to state i via a quantum jump, and p_Z is the phase-flip probability. Individual terms in Eq. (A8) can be resolved by preparing spectator qubits in eigenstates of the Pauli operators [96] and using the ternary measurement. In Sec. III B 5, circuits preparing the spectator qubits in each state of the computational basis were used to estimate bit-flip and leakage probabilities with results shown in Fig. 10.

Measuring p_Z requires circuits preparing the spectator qubits in the X/Y eigenstates, which suffer from additional memory error that we separately quantify with transport-1QRB (see Sec. III B 4). Instead, to estimate p_Z (and consequently $\epsilon_{avg}(\mathcal{E})$), we expand $p_Z \approx [p(1|0) + p(0|1) + p(L|1) + p(L|0) + p_{el}]/4$, which reflects the scattering-induced random phase shifts leading to crosstalk-induced decoherence. This expansion makes an assumption that the intensity of crosstalk light is weak such that the duration of an MCMR on a target qubit is brief compared to the crosstalk transition rates γ_{ij}, which is well-satisfied in practice. The elastic (Rayleigh) contribution p_{el} was measured in Ref. [97] using a spin-echo sequence for $^9Be^+$. We estimate the contribution of p_e to the average infedlity to be roughly %8 of the total error budget, however measurement of this contribution on Helios remains the subject of future study.

In addition to the MCMR crosstalk experiments described in Sec. III B 5, we also run an MCMR crosstalk experiment on 8 target qubits simultaneously. This arrangement has one target qubit and one spectator qubit in each operation zone, with the remaining qubits in the storage ring as spectator qubits. We perform this experiment to estimate the contribution of MCMR crosstalk to the effective MCMR error in the system-level random Clifford circuits benchmark, for circuits that contain batches of multiple MCMRs per layer (see Sec. III C 1). The data is shown in Fig. A3 B 6. We find an average MCMR crosstalk error per qubit of $5.2(2) \times 10^{-5}$ in the operation zones and $1.21(4) \times 10^{-5}$ in the storage ring.

TABLE A6. Local and global MCMR crosstalk error channels estimated from Fig. 10.

Error Channel	Local ($\times 10^{-4}$)	Global ($\times 10^{-5}$)
$p(1|0)$	0.7(2)	1.2(2)
$p(0|1)$	1.6(2)	2.8(2)
$p(L|0)$	0.8(1)	2.1(2)
$p(L|1)$	1.8(2)	4.8(2)

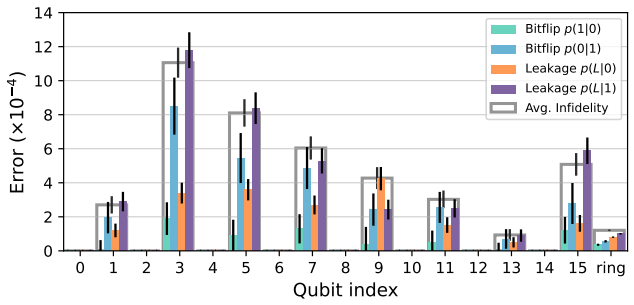

FIG. A3. MCMR crosstalk data with target qubits in {0, 2, 4, 6, 8, 10, 12, 14}. MCMR operations are applied simultaneously to the target qubits. The plot shows estimated rates of different error channels (smaller bars) and the average infidelity (wider bars) for spectator qubits in the operation zones (qubit index in {1, 3, 5, 7, 9, 11, 13, 15}) and in the storage ring.

C. Random Clifford circuits with mid-circuit measurements

Here we provide additional details on the system-level random Clifford circuits with MCMR benchmark, discussed in Sec. III C 1. We first describe the method of stabilizer tracking.

A stabilizer of a state $|\psi\rangle$ is a Pauli operator for which $|\psi\rangle$ is a $+1$ eigenstate. Initially, the state is $|0\ldots0\rangle$ and is stabilized by all Paulis tensor products of I and Z. We select a random initial stabilizer S with biased sampling of $\{I, Z\}$ with probabilities $\{1/4, 3/4\}$. We then propagate S through each layer of the circuit using an efficient binary matrix representation of the Clifford operations. When a qubit is chosen for an MCMR operation, a 1Q correction gate is applied to measure the qubit in the Pauli basis determined by S (or equivalently, since all measurements are in the Z basis, the correction maps the stabilizer from X or Y to Z). After an MCMR, a new qubit is appended to S with stabilizer randomly randomly chosen from $\{I, Z\}$ with the same biased sampling of probabilities in $\{1/4, 3/4\}$. Finally, at the end of the circuit, each qubit is measured in the Pauli basis according to S. The shot succeeds if the parity of the measured bitstring (including all the mid-circuit measurements) agrees with the sign of the evolved stabilizer S.

In our experiment we choose $n_m \in \{0, 8, 16\}$ and $l \in \{2, 4, 6, 8\}$. For each value of n_m and l, we generate 10 circuits, and run each circuit for 100 shots, with the order of all circuits randomized. The average success probability is rescaled into a quantity called the polarization [67], defined as $y_{pol} = 2p_{succ} - 1$. Let $F(n_m)$ be the process fidelity per circuit layer as a function of n_m. We estimate $F(n_m)$ by fitting the polarization to the model $y_{pol}(l, n_m) = AF(n_m)^l$, where A is a 98-qubit SPAM parameter that we fix to be equal for all values of n_m.

Below we list the process fidelities per circuit layer for $n_m \in \{0, 8, 16\}$, which are plotted in Fig. 12 in Sec. III C 1. Here n_m is the number of MCMRs per circuit layer.

TABLE A7. Process fidelity per layer versus number of mid-circuit measurements and resets per layer.

MCMRs per layer	Fidelity per layer
0	0.883(16)
8	0.856(15)
16	0.862(15)

As explained in Sec. III C 1, we compute an effective 2Q gate error ϵ_{2Q} from the $n_m = 0$ data, and separate effective MCMR errors ϵ_M from both the $n_m = 8$ and $n_m = 16$ data. The effective fidelities as well their predicted values from the component-level benchmarking data are listed in Tab. A8. Here, we explain our procedure for predicting the effective fidelities.

For ϵ_{2Q}, we take the 2Q gate error from 2QRB in Tab. II, and we convert the average infidelity into a process infidelity. We then take the depth-1 memory error per qubit from Tab. A5, again convert into a process infidelity and multiply by two (to get depth-1 error per two qubits). We add the process infidelities from the 2Q gate to the memory error and convert again into average infidelity:

$$\epsilon_{2Q} = \frac{4}{5}\left(\left(\frac{5}{4}\right)\epsilon_{RB} + 2\left(\frac{3}{2}\right)\epsilon_{\text{mem}}\right). \tag{A9}$$

Plugging in $\epsilon_{RB} = 7.9(2)\times10^{-4}$ and $\epsilon_{\text{mem}} = 6.0(3)\times10^{-4}$ and propagating uncertainties gives the value of ϵ_{2Q} in Tab. A8.

For $\epsilon_M(n_m = 8)$, we take the SPAM error of the standard measurement from Tab. II, and we add the measured crosstalk error from the MCMR crosstalk experiment with 8 simultaneous target qubits shown in Fig. A3 B 6, since that is the measurement configuration used for the MCMRs in the $n_m = 8$ QiRB circuits. We add the measured crosstalk error per MCMR in the storage ring times the number of spectator qubits in the ring, plus the crosstalk error per MCMR in the operation zones times the number of spectator qubits in the zones:

$$\epsilon_M(n_m = 8) = \epsilon_{SPAM} + 82 \times \epsilon_{MCMR,\,\text{ring}} + 8 \times \epsilon_{MCMR,\,\text{zones}}. \tag{A10}$$

For $\epsilon_M(n_m = 16)$, since the batch of 16 measurements is performed using the "protected measure" scheme, we omit the crosstalk error on qubits in the operation zones:

$$\epsilon_M(n_m = 16) = \epsilon_{SPAM} + 82 \times \epsilon_{MCMR,\,\text{ring}}. \tag{A11}$$

Plugging in with $\epsilon_{SPAM} = 4.8(6) \times 10^{-4}$, $\epsilon_{MCMR,\,\text{ring}} = 1.51(5) \times 10^{-5}$, and $\epsilon_{MCMR,\,\text{zones}} = 6.5(3) \times 10^{-5}$, and propagating uncertainties gives the values of ϵ_M listed in Tab. A8.

TABLE A8. Effective fidelities estimated from QiRB data (middle column), compared to predicted values from the component-level benchmarking data (right column).

Parameter	System-level value ($\times 10^{-3}$)	Component-level value ($\times 10^{-3}$)
ϵ_{2Q}	2.0(3)	2.2(1)
$\epsilon_M(n_m = 8)$	2.6(13)	2.2(1)
$\epsilon_M(n_m = 16)$	1.0(7)	1.7(1)

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[2] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, *et al.*, Strong quantum computational advantage using a superconducting quantum processor, Phys. Rev. Lett. **127**, 180501 (2021).

[3] M. DeCross, R. Haghshenas, M. Liu, E. Rinaldi, J. Gray, Y. Alexeev, C. H. Baldwin, J. P. Bartolotta, M. Bohn, E. Chertkov, *et al.*, Computational power of random quantum circuits in arbitrary geometries, Phys. Rev. X **15**, 021052 (2025).

[4] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. C. Brown, *et al.*, Realization of real-time fault-tolerant quantum error correction, Phys. Rev. X **11**, 041058 (2021).

[5] R. Acharya, D. A. Abanin, L. Aghababaie-Beni, I. Aleiner, T. I. Andersen, M. Ansmann, *et al.*, Quantum error correction below the surface code threshold, Nature **638**, 920–926 (2024).

[6] D. J. Wineland, C. Monroe, W. M. Itano, D. Leibfried, B. E. King, and D. M. Meekhof, Experimental issues in coherent quantum-state manipulation of trapped atomic ions, Journal of Research of the National Institute of Standards and Technology **103**, 259 (1998).

[7] D. Kielpinski, C. Monroe, and D. J. Wineland, Architecture for a large-scale ion-trap quantum computer, Nature **417**, 709 (2002).

[8] J. P. Home, D. Hanneke, J. D. Jost, J. M. Amini, D. Leibfried, and D. J. Wineland, Complete methods set for scalable ion trap quantum information processing, Science **325**, 1227 (2009).

[9] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, *et al.*, Demonstration of the trapped-ion quantum ccd computer architecture, Nature **592**, 209 (2021).

[10] S. A. Moses, C. H. Baldwin, M. S. Allman, R. Ancona, L. Ascarrunz, *et al.*, A race-track trapped-ion quantum processor, Phys. Rev. X **13**, 041052 (2023).

[11] C. Mordini, A. Ricci Vasquez, Y. Motohashi, M. Müller, M. Malinowski, C. Zhang, K. K. Mehta, D. Kienzler, and J. P. Home, Multizone trapped-ion qubit control in an integrated photonics qccd device, Phys. Rev. X **15**, 011040 (2025).

[12] D. Bluvstein, H. Levine, G. Semeghini, T. T. Wang, S. Ebadi, M. Kalinowski, A. Keesling, N. Maskara, H. Pichler, M. Greiner, *et al.*, A quantum processor based on coherent transport of entangled atom arrays, Nature **604**, 451 (2022).

[13] B. W. Reichardt, A. Paetznick, D. Aasen, I. Basov, J. M. Bello-Rivas, P. Bonderson, R. Chao, W. van Dam, M. B. Hastings, R. V. Mishmash, *et al.*, Fault-tolerant quantum computation with a neutral atom processor (2025), arXiv:2411.11822 [quant-ph].

[14] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, *et al.*, Evidence for the utility of quantum computing before fault tolerance, Nature **618**, 500 (2023).

[15] A. Radnaev, W. Chung, D. Cole, D. Mason, T. Ballance, M. Bedalov, D. Belknap, M. Berman, M. Blakely, I. Bloomfield, *et al.*, Universal neutral-atom quantum computer with individual optical addressing and nondestructive readout, PRX Quantum **6**, 030334 (2025).

[16] J.-S. Chen, E. Nielsen, M. Ebert, V. Inlek, K. Wright, V. Chaplin, A. Maksymov, E. Páez, A. Poudel, P. Maunz, *et al.*, Benchmarking a trapped-ion quantum computer with 30 qubits, Quantum **8**, 1516 (2024).

[17] M. F. Brandl, A quantum von neumann architecture for large-scale quantum computing (2017), arXiv:1702.02583 [quant-ph].

[18] M. Dietrich, N. Kurz, T. Noel, G. Shu, and B. Blinov, Hyperfine and optical barium ion qubits, Physical Review A—Atomic, Molecular, and Optical Physics **81**, 052328 (2010).

[19] W. C. Burton, B. Estey, I. M. Hoffman, A. R. Perry, C. Volin, and G. Price, Transport of multispecies ion crystals through a junction in a radio-frequency paul trap, Phys. Rev. Lett. **130**, 173202 (2023).

[20] W. K. Hensinger, S. Olmschenk, D. Stick, D. Hucul, M. Yeo, M. Acton, L. Deslauriers, C. Monroe, and J. Rabchuk, T-junction ion trap array for two-dimensional ion shuttling, storage, and manipulation, Applied Physics Letters **88**, 034101 (2006).

[21] R. B. Blakestad, C. Ospelkaus, A. P. VanDevender, J. M. Amini, J. Britton, D. Leibfried, and D. J. Wineland, High-fidelity transport of trapped-ion qubits through an **X**-junction trap array, Phys. Rev. Lett. **102**, 153002 (2009).

[22] C. Decaroli, R. Matt, R. Oswald, C. Axline, M. Ernzer, J. Flannery, S. Ragg, and J. P. Home, Design, fabrication and characterization of a micro-fabricated stacked-wafer segmented ion trap with two x-junctions, Quantum Science and Technology **6**, 044001 (2021).

[23] R. B. Blakestad, C. Ospelkaus, A. P. VanDevender, J. H. Wesenberg, M. J. Biercuk, D. Leibfried, and D. J. Wineland, Near-ground-state transport of trapped-ion qubits through a multidimensional array, Phys. Rev. A **84**, 032314 (2011).

[24] K. Wright, J. M. Amini, D. L. Faircloth, C. Volin, S. Charles Doret, H. Hayden, C.-S. Pai, D. W. Landgren, D. Denison, T. Killian, *et al.*, Reliable transport through a microfabricated x-junction surface-electrode ion trap, New Journal of Physics **15**, 033004 (2013).

[25] J. M. Amini, H. Uys, J. H. Wesenberg, S. Seidelin, J. Britton, J. J. Bollinger, D. Leibfried, C. Ospelkaus, A. P. VanDevender, and D. J. Wineland, Toward scalable ion traps for quantum information processing, New Journal of Physics **12**, 033031 (2010).

[26] D. L. Moehring, C. Highstrete, D. Stick, K. M. Fortier, R. Haltli, C. Tigges, and M. G. Blain, Design, fabrication and experimental demonstration of junction surface ion traps, New Journal of Physics **13**, 075018 (2011).

[27] G. Shu, G. Vittorini, A. Buikema, C. S. Nichols, C. Volin, D. Stick, and K. R. Brown, Heating rates and ion-motion control in a Y-junction surface-electrode trap, Phys. Rev. A **89**, 062308 (2014).

[28] J. Chiaverini, R. B. Blakestad, J. Britton, J. D. Jost, C. Langer, D. Leibfried, R. Ozeri, and D. J. Wineland, Surface-electrode architecture for ion-trap

quantum information processing, Quantum Info. Comput. **5**, 419–439 (2005).
[29] D. Kielpinski, B. E. King, C. J. Myatt, C. A. Sackett, Q. A. Turchette, W. M. Itano, C. Monroe, D. J. Wineland, and W. H. Zurek, Sympathetic cooling of trapped ions for quantum logic, Phys. Rev. A **61**, 032310 (2000).
[30] A. Sørensen and K. Mølmer, Entanglement and quantum computation with ions in thermal motion, Phys. Rev. A **62**, 022311 (2000).
[31] P. J. Lee, K.-A. Brickman, L. Deslauriers, P. C. Haljan, L.-M. Duan, and C. Monroe, Phase control of trapped ion quantum gates, Journal of Optics B: Quantum and Semiclassical Optics **7**, S371 (2005).
[32] F. A. An, A. Ransford, A. Schaffer, L. R. Sletten, J. Gaebler, J. Hostetter, and G. Vittorini, High fidelity state preparation and measurement of ion hyperfine qubits with $I > \frac{1}{2}$, Physical Review Letters **129**, 130501 (2022).
[33] A. Ransford, C. Roman, T. Dellaert, P. McMillin, and W. C. Campbell, Weak dissipation for high-fidelity qubit-state preparation and measurement, Physical Review A **104**, L060402 (2021).
[34] J. Gaebler, A. Ransford, L. Sletten, F. An, J. Hostetter, A. Schaffer, and G. Vittorini, Detecting leakage errors in hyperfine qubits, U.S. patent 20,240,211,792 (2024), filed November 20, 2023.
[35] A. S. Sotirova, J. D. Leppard, A. Vazquez-Brennan, S. M. Decoppet, F. Pokorny, M. Malinowski, and C. J. Ballance, High-fidelity heralded quantum state preparation and measurement (2024), arXiv:2409.05805 [quant-ph].
[36] D. T. C. Allcock, W. C. Campbell, J. Chiaverini, I. L. Chuang, E. R. Hudson, I. D. Moore, A. Ransford, C. Roman, J. M. Sage, and D. J. Wineland, omg blueprint for trapped ion quantum computing with metastable states, Applied Physics Letters **119**, 214002 (2021).
[37] S. Olmschenk, K. C. Younge, D. L. Moehring, D. N. Matsukevich, P. Maunz, and C. Monroe, Manipulation and detection of a trapped yb^+ hyperfine qubit, Phys. Rev. A **76**, 052314 (2007).
[38] S. De, U. Dammalapati, K. Jungmann, and L. Willmann, Magneto-optical trapping of barium, Phys. Rev. A **79**, 041402 (2009).
[39] J. Johansen, B. Estey, M. Rowe, and A. Ransford, Fast loading of a trapped ion quantum computer using a 2d magneto-optical trap, in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (2022) pp. 299–303.
[40] J. P. Gaebler, C. H. Baldwin, S. A. Moses, J. M. Dreiling, C. Figgatt, M. Foss-Feig, D. Hayes, and J. M. Pino, Suppression of midcircuit measurement crosstalk errors with micromotion, Phys. Rev. A **104**, 062440 (2021).
[41] P. J. Low, N. C. F. Zutt, G. A. Tathed, and C. Senko, Quantum logic operations and algorithms in a single 25-level atomic qudit (2025), arXiv:2507.15799 [quant-ph].
[42] Openqasm live specification.
[43] N. C. Brown, J. P. C. III, C. Granade, B. Heim, S. Wernli, C. Ryan-Anderson, D. Lucchetti, A. Paetznick, M. Roetteler, K. Svore, *et al.*, Advances in compilation for quantum hardware – a demonstration of magic state distillation and repeat-until-success protocols, (2023), arXiv:2310.12106 [quant-ph].
[44] M. Koch, A. Lawrence, K. Singhal, S. Sivarajah, and R. Duncan, Guppy: Pythonic quantum-classical programming (2024), arXiv:2510.12582.
[45] K. Svore, A. Geller, M. Troyer, J. Azariah, C. Granade, B. Heim, V. Kliuchnikov, M. Mykhailova, A. Paz, and M. Roetteler, Q#: Enabling Scalable Quantum Computing and Development with a High-level DSL, in *Proceedings of the Real World Domain Specific Languages Workshop 2018*, RWDSL2018 (ACM, 2018).
[46] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, *et al.*, Quantum computing with Qiskit (2024), arXiv:2405.08810 [quant-ph].
[47] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta, Open quantum assembly language (2017), arXiv:1707.03429 [quant-ph].
[48] A. Cross, A. Javadi-Abhari, T. Alexander, N. De Beaudrap, L. S. Bishop, S. Heidel, C. A. Ryan, P. Sivarajah, J. Smolin, J. M. Gambetta, *et al.*, Openqasm 3: A broader and deeper quantum assembly language, ACM Transactions on Quantum Computing **3**, 1–50 (2022).
[49] Cirq Developers, *Cirq* (Zenodo, 2025).
[50] CUDA-Q Developers, Cuda-q (2025).
[51] M. Liu *et al.*, Certified randomness amplification by dynamically probing remote random quantum states, (To be made available simultaneously with this paper).
[52] Quantinuum, Gate streaming documentation (2025).
[53] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, Validating quantum computers using randomized model circuits, Phys. Rev. A **100**, 032328 (2019).
[54] R. Blume-Kohout and K. C. Young, A volumetric framework for quantum computer benchmarks, Quantum **4**, 362 (2020).
[55] A. Wack, H. Paik, A. Javadi-Abhari, P. Jurcevic, I. Faro, J. M. Gambetta, and B. R. Johnson, Quality, speed, and scale: three key attributes to measure the performance of near-term quantum computers (2021), arXiv:2110.14108 [quant-ph].
[56] T. Tomesh, P. Gokhale, V. Omole, G. S. Ravi, K. N. Smith, J. Viszlai, X.-C. Wu, N. Hardavellas, M. R. Martonosi, and F. T. Chong, Supermarq: A scalable quantum benchmark suite (2022), arXiv:2202.11045 [quant-ph].
[57] T. Lubinski, S. Johri, P. Varosy, J. Coleman, L. Zhao, J. Necaise, C. H. Baldwin, K. Mayer, and T. Proctor, Application-oriented performance benchmarks for quantum computing, IEEE Transactions on Quantum Engineering **4**, 1 (2023).
[58] T. Proctor, K. Young, A. D. Baczewski, and R. Blume-Kohout, Benchmarking quantum computers (2024), arXiv:2407.08828 [quant-ph].
[59] J. E. Christensen, D. Hucul, W. C. Campbell, and E. R. Hudson, High-fidelity manipulation of a qubit enabled by a manufactured nucleus, npj Quantum Information **6** (2020).
[60] E. Magesan, J. M. Gambetta, and J. Emerson, Characterizing quantum gates via randomized benchmarking, Physical Review A **85** (2012).
[61] Y.-H. Chen and C. H. Baldwin, Randomized benchmarking with leakage errors (2025), arXiv:2502.00154 [quant-ph].
[62] B. Efron and R. Tibshirani, An Introduction to the Bootstrap (1993).
[63] I. D. Moore, W. C. Campbell, E. R. Hudson, M. J. Boguslawski, D. J. Wineland, and D. T. C. Allcock, Photon scattering errors during stimulated raman transitions in

trapped-ion qubits, Phys. Rev. A **107**, 032413 (2023).
[64] A. Erhard, J. J. Wallman, L. Postler, M. Meth, R. Stricker, E. A. Martinez, P. Schindler, T. Monz, J. Emerson, and R. Blatt, Characterizing large-scale quantum computers via cycle benchmarking, Nature Communications **10**, 5347 (2019).
[65] S. Sheldon, L. S. Bishop, E. Magesan, S. Filipp, J. M. Chow, and J. M. Gambetta, Characterizing errors on qubit operations via iterative randomized benchmarking, Physical Review A **93**, 10.1103/physreva.93.012301 (2016).
[66] M. A. Sepiol, A. C. Hughes, J. E. Tarlton, D. P. Nadlinger, T. G. Ballance, C. J. Ballance, T. P. Harty, A. M. Steane, J. F. Goodwin, and D. M. Lucas, Probing qubit memory errors at the part-per-million level, Phys. Rev. Lett. **123**, 110503 (2019).
[67] J. Hines, D. Hothem, R. Blume-Kohout, B. Whaley, and T. Proctor, Fully scalable randomized benchmarking without motion reversal, PRX Quantum **5**, 030334 (2024).
[68] D. Hothem, J. Hines, C. Baldwin, D. Gresh, R. Blume-Kohout, and T. Proctor, Measuring error rates of mid-circuit measurements, Nature Communications **16** (2025).
[69] M. A. Nielsen, A simple formula for the average gate fidelity of a quantum dynamical operation, Physics Letters A **303**, 249 (2002).
[70] D. Hangleiter and J. Eisert, Computational advantage of quantum random sampling, Rev. Mod. Phys. **95**, 035001 (2023).
[71] D. A. Abanin, R. Acharya, L. Aghababaie-Beni, G. Aigeldinger, A. Ajoy, R. Alcaraz, I. Aleiner, T. I. Andersen, M. Ansmann, F. Arute, *et al.*, Observation of constructive interference at the edge of quantum ergodicity, Nature **646**, 825 (2025).
[72] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, Characterizing quantum supremacy in near-term devices, Nature Physics **14**, 595 (2018).
[73] K. Mayer, A. Hall, T. Gatterman, S. K. Halit, K. Lee, J. Bohnet, D. Gresh, A. Hankin, K. Gilmore, J. Gerber, *et al.*, Theory of mirror benchmarking and demonstration on a quantum computer (2023), arXiv:2108.10431 [quant-ph].
[74] T. Proctor, S. Seritan, E. Nielsen, K. Rudinger, K. Young, R. Blume-Kohout, and M. Sarovar, Establishing trust in quantum computations (2022), arXiv:2204.07568 [quant-ph].
[75] T. Proctor, K. Rudinger, K. Young, E. Nielsen, and R. Blume-Kohout, Measuring the capabilities of quantum computers, Nature Physics **18**, 75 (2022).
[76] A. Morvan, B. Villalonga, X. Mi, S. Mandrà, A. Bengtsson, P. V. Klimov, *et al.*, Phase transitions in random circuit sampling, Nature **634**, 328–333 (2024).
[77] J. Gray and S. Kourtis, Hyper-optimized tensor network contraction, Quantum **5**, 410 (2021).
[78] M. J. Biercuk, H. Uys, A. P. VanDevender, N. Shiga, W. M. Itano, and J. J. Bollinger, Optimized dynamical decoupling in a model quantum memory, Nature **458**, 10.1038/nature07951 (2009).
[79] R. Bowler, J. Gaebler, Y. Lin, T. R. Tan, D. Hanneke, J. D. Jost, J. P. Home, D. Leibfried, and D. J. Wineland, Coherent diabatic ion transport and separation in a multizone trap array, Phys. Rev. Lett. **109**, 080502 (2012).
[80] A. Walther, F. Ziesel, T. Ruster, S. T. Dawkins, K. Ott, M. Hettrich, K. Singer, F. Schmidt-Kaler, and U. Poschinger, Controlling fast transport of cold trapped ions, Phys. Rev. Lett. **109**, 080501 (2012).
[81] J. D. Sterk, H. Coakley, J. Goldberg, V. Hietala, J. Lechtenberg, H. McGuinness, D. McMurtrey, L. P. Parazzoli, J. Van Der Wall, and D. Stick, Closed-loop optimization of fast trapped-ion shuttling with sub-quanta excitation, npj Quanum Information **8**, 10.1038/s41534-022-00579-3 (2022).
[82] E. Granet *et al.*, Superconducting pairing correlations on a trapped-ion quantum computer, (To be made available simultaneously with this paper).
[83] P. Niroula *et al.*, Realization of a quantum streaming algorithm on long-lived trapped-ion qubits, (To be made available simultaneously with this paper).
[84] N. P. Breuckmann and J. N. Eberhardt, Quantum low-density parity-check codes, PRX Quantum **2**, 040101 (2021).
[85] C. Ryan-Anderson, N. C. Brown, M. S. Allman, B. Arkin, G. Asa-Attuah, C. Baldwin, J. Berg, J. G. Bohnet, S. Braxton, N. Burdick, *et al.*, Implementing fault-tolerant entangling gates on the five-qubit code and the color code (2022), arXiv:2208.01863 [quant-ph].
[86] C. Ryan-Anderson, N. C. Brown, C. H. Baldwin, J. M. Dreiling, C. Foltz, J. P. Gaebler, T. M. Gatterman, N. Hewitt, C. Holliman, C. V. Horst, *et al.*, High-fidelity teleportation of a logical qubit using transversal gates and lattice surgery, Science **385**, 1327 (2024).
[87] S. Dasu, S. Burton, K. Mayer, D. Amaro, J. A. Gerber, K. Gilmore, D. Gresh, D. DelVento, A. C. Potter, and D. Hayes, Breaking even with magic: demonstration of a high-fidelity logical non-clifford gate, arxiv:2506.14688 (2025).
[88] E. T. Campbell, A theory of single-shot error correction for adversarial noise, Quantum Science and Technology **4**, 025006 (2019).
[89] N. Berthusen, J. Dreiling, C. Foltz, J. P. Gaebler, T. M. Gatterman, D. Gresh, N. Hewitt, M. Mills, S. A. Moses, B. Neyenhuis, *et al.*, Experiments with the four-dimensional surface code on a quantum charge-coupled device quantum computer, Phys. Rev. A **110**, 062413 (2024).
[90] M. Cain, C. Zhao, H. Zhou, N. Meister, J. P. B. Ataides, A. Jaffe, D. Bluvstein, and M. D. Lukin, Correlated decoding of logical algorithms with transversal gates, Phys. Rev. Lett. **133**, 240602 (2024).
[91] J. M. Gambetta, A. D. Córcoles, S. T. Merkel, B. R. Johnson, J. A. Smolin, J. M. Chow, C. A. Ryan, C. Rigetti, S. Poletto, T. A. Ohki, *et al.*, Characterization of addressability by simultaneous randomized benchmarking, Phys. Rev. Lett. **109**, 240504 (2012).
[92] J. J. Wallman and J. Emerson, Noise tailoring for scalable quantum computation via randomized compiling, Phys. Rev. A **94**, 052325 (2016).
[93] S. Chen, Y. Liu, M. Otten, A. Seif, B. Fefferman, and L. Jiang, The learnability of pauli noise, Nature Communications **14**, 52 (2023).
[94] K. Mølmer, Y. Castin, and J. Dalibard, Monte carlo wave-function method in quantum optics, J. Opt. Soc. Am. B **10**, 524 (1993).
[95] F. Reiter and A. S. Sørensen, Effective operator formalism for open quantum systems, Phys. Rev. A **85**, 032111 (2012).

[96] M. D. Bowdrey, D. K. Oi, A. Short, K. Banaszek, and J. Jones, Fidelity of single qubit maps, Physics Letters A **294**, 258 (2002).
[97] H. Uys, M. J. Biercuk, A. P. VanDevender, C. Ospelkaus, D. Meiser, R. Ozeri, and J. J. Bollinger, Decoherence due to elastic rayleigh scattering, Phys. Rev. Lett. **105**, 200401 (2010).