Optimizing Resource Efficiencies for Scalable Full-Stack Quantum Computers

Marco Fellous-Asiani,[1,2,*] Jing Hao Chai,[2,3,4] Yvain Thonnart,[5] Hui Khoon Ng,[6,3,7,†] Robert S. Whitney,[8,‡] and Alexia Auffèves[2,3,7,§]

[1] *Centre for Quantum Optical Technologies, Centre of New Technologies, University of Warsaw, Banacha 2c, Warsaw 02-097, Poland*

[2] *Université Grenoble Alpes, CNRS, Grenoble INP, Institut Néel, Grenoble 38000, France*

[3] *Centre for Quantum Technologies, National University of Singapore, Singapore 117543, Singapore*

[4] *Entropica Labs, 186b Telok Ayer Street, 068632 Singapore*

[5] *Université Grenoble Alpes, French Alternative Energies and Atomic Energy Commission (CEA)–Laboratory for Integration of Systems and Technology (LIST), Grenoble F-38000, France*

[6] *Yale–National University of Singapore (NUS) College, Singapore*

[7] *MajuLab, CNRS-UCA-SU-NUS-NTU International Joint Research Laboratory, Singapore*

[8] *Université Grenoble Alpes, CNRS, Laboratoire de Physique et Modélisation des Milieux Condensés (LPMMC), Grenoble 38000, France*

(Received 29 November 2022; accepted 31 July 2023; published 30 October 2023)

In the race to build scalable quantum computers, minimizing the resource consumption of their full stack to achieve a target performance becomes crucial. It mandates a synergy of fundamental physics and engineering: the former for the microscopic aspects of computing performance and the latter for the macroscopic resource consumption. For this, we propose a holistic methodology dubbed metric noise resource (MNR) that is able to quantify and optimize all aspects of the full-stack quantum computer, bringing together concepts from quantum physics (e.g., noise on the qubits), quantum information (e.g., computing architecture and type of error correction), and enabling technologies (e.g., cryogenics, control electronics, and wiring). This holistic approach allows us to define and study resource efficiencies as ratios between performance and resource cost. As a proof of concept, we use MNR to minimize the power consumption of a full-stack quantum computer, performing noisy or fault-tolerant computing with a target performance for the task of interest. Comparing this with a classical processor performing the same task, we identify a quantum energy advantage in regimes of parameters distinct from the commonly considered quantum computational advantage. This provides a previously overlooked practical argument for building quantum computers. While our illustration uses highly idealized parameters inspired by superconducting qubits with concatenated error correction, the methodology is universal—it applies to other qubits and error-correcting codes—and it provides experimenters with guidelines to build energy-efficient quantum computers. In some regimes of high energy consumption, it can reduce this consumption by orders of magnitude. Overall, our methodology lays the theoretical foundation for resource-efficient quantum technologies.

DOI: 10.1103/PRXQuantum.4.040319

I. INTRODUCTION

There is a lot of excitement and hope that quantum information processing will help us solve problems of importance for society. Potential applications are numerous, ranging from optimization [1,2] and cryptography [3,4] to finance [5,6]. The simulation of quantum systems [7–9] for quantum chemistry and material science holds the promise of understanding fundamental phenomena and designing new materials and new drugs [10,11]. Different experimental platforms are currently investigated, including photonics [12], ion traps [13], spin qubits [14], and superconducting qubits [15], among others [16–18]. Owing to impressive experimental efforts, qubit fidelities are starting to approach the fault-tolerance thresholds for scalable quantum computers. Quantum computational advantages have been claimed [19,20], even as the concept is still being discussed [21,22].

*fellous.asiani.marco@gmail.com
†huikhoon.ng@yale-nus.edu.sg
‡robert.whitney@grenoble.cnrs.fr
§alexia.auffeves@cnrs.fr

Making quantum computers a concrete reality has a physical resource cost, especially when large-scale processors are targeted. In this spirit, the number of physical qubits required to implement large-scale computations has started to be investigated in various fault-tolerant schemes, including those that employ concatenated codes [23–25], surface codes [24–26], and bosonic qubits [27–29]. The total number of logical gates and qubits required by many algorithms have also been estimated, e.g., for decryption tasks [26,30–32] and material [33–37] or electromagnetic [38] simulations. These studies play an important role in identifying strategies for scaling up quantum processors.

The question of energy consumption was mentioned in the seminal experimental demonstration reported in Ref. [19]: a 5-orders-of-magnitude difference was announced between the power consumption of the quantum processor and that of the classical supercomputer performing the same task. However, the study was on a 50-qubit quantum processor and at the present time it remains unclear how the energy consumption of future quantum processors will scale. On the one hand, some studies anticipate energy savings, due to the complexity gains provided by quantum logic, see, e.g., Ref. [39]. They rely on subtle algorithmic details but then assume very simple models for the hardware. On the other hand, studies based on precise hardware details (but agnostic on algorithmic details) [40,41] usually conclude that the overheads needed to control the physical qubits could be so large that they will be to be an issue for scalability [40,42–48], especially in achieving large-scale fault-tolerant quantum computers. This lack of consensus reveals the need for a holistic methodology coupling data from the hardware and the algorithmic frameworks.

Setting up a holistic methodology is highly challenging, as it requires modeling the full stack of the quantum computer [42,44,45,48–52], and coordinating inputs from currently separated areas of expertise, as sketched in Fig. 1. Improving computational performance requires programming the processor to implement a given algorithm and to reach a satisfactory level of control over noise while the algorithm is being executed. These optimizations are performed at the quantum level and rely on detailed knowledge of quantum hardware and software, e.g., quantum control, noise modeling, environmental engineering, quantum error correction, quantum algorithms, qubit fabrication, etc. However, achieving that in a physical device requires the use of macroscopic resources provided by enabling technologies at the classical level (e.g., cryogenic systems [40,41], classical computers for control [43,53–57], lasers, detectors [58,59], amplifiers [60–62], etc.). Hence, understanding and managing the resource bill of future quantum processors cannot be restricted to the quantum level, as it provides no access to the macroscopic costs. Reciprocally, a solely macroscopic approach is blind to the computing performance—we do not know what we are paying for. Resource-cost assessments and optimizations must be jointly conducted by an interdisciplinary combination of expertise. In the energetic context, this has been dubbed the Quantum Energy Initiative [63].

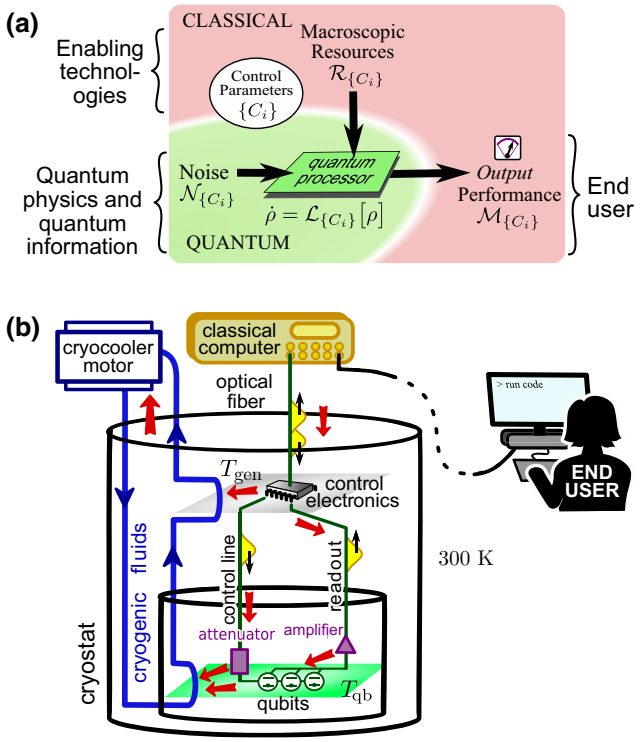


FIG. 1. (a) A schematic of our metric noise resource (MNR) methodology, which models the battle of control against noise. The control parameters affect the performance metric, the noise, and the resource consumption. Such parameters include the qubit temperature, the amount of error correction, etc. The performance metric can be improved by using resources to reduce the noise (e.g., cooling the qubits) or spending resources to make the metric less sensitive to existing noise (e.g., better error correction). (b) A simplified sketch of the physical elements in a typical full-stack superconducting quantum computer, with qubits at temperature $T_{\rm qb}$ and classical control electronics at temperature $T_{\rm gen}$. The classical computer, at room temperature, compiles the user-specified algorithm and code into a sequence of physical gate operations, interprets detected errors in real time, and can modify the gate sequence to correct them. The black arrows indicate information flows. The red arrows indicate heat flows that bring noise that can cause gate errors (heat conducted by wiring, heat generated by attenuators and amplifiers, etc.); more details are shown in Fig. 9. We model the full stack by considering each physical element in (b) in terms of its effect on the metric, noise, and resource in (a).

In this paper, we present a methodology that allows us to optimize the resource cost for the full stack of a quantum computer, under the constraint of a certain performance. Such an optimization under constraint is only possible if the methodology can treat all elements of the quantum computer in a single framework, including both

the hardware (such as attenuators and cryogenics) and the quantum software (such as different quantum gates that implement the same quantum algorithm). We provide such a holistic methodology and show how it relates the microscopic description of the quantum computation to the macroscopic resources consumed by the cryogeny, wiring, classical control electronics, etc. We apply it to examples of quantum tasks with increasing complexity, from the operation of a single-qubit gate or a noisy circuit to a fault-tolerant algorithm. Each requires the specification of a relevant performance metric and a detailed description of the physical processes at play. In each example, we use our approach to show how resource costs scale with the size of the computational task. This reveals concrete instances of extreme sensitivity to both hardware characteristics (e.g., the qubit quality or the efficiency of the electronics) and software characteristics (the architecture of the circuits or the type of error-correcting code) and the necessity of treating both aspects in a coordinated manner.

As an important outcome, we analyze quantum resource efficiencies as ratios between the computing performance metric and the resource cost. Different efficiencies can be defined, depending on the metric and the resource(s) of interest. Some permit the benchmarking of different qubit technologies or computing architectures. Others allow a comparison between quantum and classical processors and to define a quantum advantage from the resource perspective, which can further boost practical interest in quantum computers. Focusing on the example of Shor's algorithm to break Rivest-Shamir-Adleman (RSA) encryption, our calculations single out regimes of quantum energy advantage over classical supercomputers for problem sizes still accessible by such supercomputers, including in cases where the quantum computer takes more time to do the job than the supercomputer. These results show that the energy advantage is reached in different regimes than the computational advantage, providing a new and so far overlooked potential practical reason to build quantum computers.

Throughout this paper, we take parameters inspired by the superconducting platform and existing technologies for the control electronics, wiring, and cryogeny. However, our approach is generic and versatile, capable of providing general forms of behavior and typical orders of magnitude for a wide variety of settings. In particular, it shows how diverse the parameters that can affect power consumption are, with a crucial one being qubit quality. It also singles out surprising effects: while it is often optimal to make the qubits cold enough to minimize error correction, our approach shows that there are regimes where the opposite is true, regimes where it is more energy efficient to have warmer qubits with more error correction.

For illustrative simplicity, we base our examples on the concatenated seven-qubit code, which is well documented and allows for straightforward analytical expressions but can be demanding in terms of physical quantum resources. This choice leads us to use parameter values that are sometimes beyond the current state of the art. Nevertheless, we invite the reader to appreciate our results as proofs of concept of our methodology. It can provide on-demand practical guidelines to experimentalists and engineers looking to build resource-efficient quantum processors, allowing them to clearly identify the sequence of challenges to be met. Ultimately, systematic applications of the methodology can help avoid ecologically unacceptable outcomes, such as the current rapid increase in energy consumption of servers for consumer electronics [64] and artificial intelligence [65]. Thus, throughout the paper, we keep the methodology as apparent as possible, so as it can be applied to different qubit platforms and enabling technologies, as well as other error-correcting codes.

Our paper is organized as follows. We present our general optimization methodology in Sec. II, for any kind of resource and any kind of quantum computing platform. In Secs. III– V, we apply it to the special case of a superconducting quantum computer, focusing on energy and power, to illustrate the use of our methodology and of the kinds of conclusions one can draw from such an analysis. Section III describes a simple example for a noisy single-qubit gate, establishing the basic connection between microscopic qubit parameters and macroscopic power consumption. Section IV focuses on a noisy circuit, revealing the close interplay between inputs from the software and the physics of the hardware. Sections III and IV are largely pedagogical in their aims, to shed light on how the performance defined at the quantum level can impact the resource consumption at the macroscopic level. Section V considers a full fault-tolerant quantum computer, using concatenated codes for error correction and performing a calculation of difficulty similar to breaking the well-known RSA encryption. Estimates for fault-tolerant quantum computing based on the currently popular surface codes are given at the end of Sec. V. We summarize our findings in Sec. VI.

II. METRIC NOISE RESOURCE METHODOLOGY

A quantum computer is a programmable machine; its job is to perform a well-defined sequence of operations on an ensemble of qubits. After the circuit is programmed, the qubits are prepared in a reference state, unitary operations implementing a computational task are then applied, and the qubits are finally measured to extract the result of the calculation. Quantum noise perturbs this sequence, giving rise to errors that decrease the computing performance. This has to be countered by an increase in noise-mitigation measures, in an attempt to reduce the occurrence of errors and to remove their effects on the computation. Such increased noise mitigation is usually associated with increased resource costs. In some cases,

the increased resource cost can itself result in more noise. For instance, more error correction requires more physical qubits and that can result in additional sources of crosstalk, increasing the noise affecting the quantum processor [66]. Hence, finding the minimal resource cost to reach a target performance requires one to explore nontrivial sweet spots. Such an investigation involves coordinated inputs from the software and hardware, at the quantum and classical levels of description. Here, we present a holistic methodology to treat the whole range of inputs. For reasons that become clear below, we have dubbed it the metric noise resource (MNR) methodology, or more simply MNR [67].

The basics of MNR are sketched in Fig. 1(a). The first step consists of identifying the set of parameters—dubbed "control parameters," $\mathcal{C}_i$s—that allows us to execute a quantum algorithm with a given target performance. It is with respect to these parameters that the resource cost will be minimized. The control parameters can be of various kinds. Some characterize the quantum processor or the hardware controlling it. Typical examples include the temperature of the qubits or the strength of the attenuators on the control lines. Some are of software nature, reflecting the fact that the same algorithm can be executed by different circuits. Examples include the degree of compression of the circuit or any other quantity capturing the circuit architecture. A crucial control parameter is the size of the quantum error-correcting code, i.e., the number of physical qubits per logical qubit in a fault-tolerant quantum computation.

Once this identification has been done, we can turn to the first element of MNR: the performance metric $\mathcal{M}$ (later dubbed "metric," for brevity). It is a number measuring the quality of the computation, for which a larger number means a better computation. Naturally, there is some flexibility in the choice of the metric. Some are defined at a low level, focusing on the precision with which states can be generated by executing the programmed sequence of gates. A natural example is the fidelity, which quantifies the distance between the ideal and the real processor states before the extraction of the result. Other metrics are user oriented, such as the Q score [68] or the quantum volume [69], which estimates the maximal size of the problems that can be solved on a quantum computing platform. Some user-oriented metrics aim to benchmark classical and quantum processors and to identify quantum computational advantages. Whatever the chosen metric, it depends directly on the level of control over physical processes in the quantum computer.

This brings us naturally to the second element of MNR: the noise in the physical processes. It is taken into account by modeling the dynamics of the noisy quantum processor executing the algorithm. This involves a given time-dependent Hamiltonian, together with a noise model, in the form of a master equation the expression of which depends on the control parameters. Many parameters can enter this noise model, such as the temperature of the qubits and the temperature of the external control electronics. The time-dependent Hamiltonian corresponds to the sequence of gates applied to the qubits, which is set by the circuit architecture. Hence, such a model allows us to derive a quantitative expression for the metric, as a function of the C_is.

The third ingredient of MNR is the resource $\mathcal{R}$ of interest that we wish to minimize. Formally, a resource can be any cost function that depends on the set of control parameters. While MNR is general and could tackle economic costs, here we shall focus on physical resource costs. They can be extremely diverse in nature, e.g., the physical volume occupied by the quantum processor, the total frequency bandwidth allocated to the qubits, the duration of the algorithm, the amount of classical information processed to perform error correction, or the energy consumption. In this paper, we will address the last of these, by considering the electrical power consumed while a quantum computation is being performed.

Once these steps are completed, MNR basically reduces to a constrained optimization. Fixing a target metric $\mathcal{M}_0$ boils down to setting a tolerable level of control over noise for a properly programmed processor: it provides a first constraint that the control parameters have to satisfy. The resource cost $\mathcal{R}$ is then minimized as a function of the control parameters under this implicit constraint. An optimal set of control parameters gives the minimal resource consumption $\mathcal{R}^{\min}(\mathcal{M}_0)$ needed to reach the metric $\mathcal{M}_0$ [70]. For instance, if the qubit temperature is a control parameter, MNR provides an optimal working temperature for the qubits to reach a target metric $\mathcal{M}_0$ with a minimal resource cost and can lead to nontrivial values as shown below. It thus provides practical inputs to designing resource-efficient quantum computations.

MNR relates a metric to its macroscopic resource cost. This makes it drastically different from the common point of view to date, which has been to target the largest metrics, whatever the resource cost. It has been inspired by our earlier work [66], which pointed out that a constraint on resources has a profound effect on fault-tolerant quantum computing. However unlike here, that work only considered quantum-level resources.

A. Resource efficiencies

In general, efficiencies characterize the balance between a performance and the resource consumed in achieving it. MNR provides systematic relations between performance and resource costs. Hence, it naturally leads one to define and optimize resource efficiencies for quantum computing. For classical supercomputers, the target performance is computing power, expressed in floating-point operations per second (FLOPS). The energy efficiency is built as the ratio of the computing power to the power consumption

(the resource) of the processor. It is measured in FLOPS/W, has the dimension of the inverse of an energy, and gives rise to the Green500 ranking of the most energy-efficient supercomputers [71]. In this paper, we shall explore quantum equivalents of this energy efficiency. Within the MNR methodology, the resource efficiency is generically defined as $\eta = \mathcal{M}/\mathcal{R}$, where $\mathcal{M}$ is the metric and $\mathcal{R}$ the resource cost. As mentioned above, two kinds of metrics can be considered: low-level metrics and user-oriented metrics. The resource cost can be defined at the quantum level or at the macroscopic level, giving rise to bare and dressed efficiencies, respectively. The quantum level is crucial for understanding the physics of qubits, while the macroscopic level will be what matters for large-scale applications.

Sections III and IV, respectively, involve noisy gates and circuits. A low-level metric, the fidelity, is natural in both cases. Modeling the processor at the quantum level provides an implicit relation between the noise, the control, and the chosen metric. Applying the MNR methodology to minimize the power consumption $\mathcal{R}^{\min}(\mathcal{M}_0)$ for the target metric $\mathcal{M}_0$ unambiguously sets the maximal efficiency of the task at the target $\mathcal{M}_0$. Such an efficiency is well suited for benchmarking different technologies of qubits or different computing architectures implementing the same algorithm, i.e., to compare different quantum computing platforms.

Sections IV and V involve algorithms. We thus introduce user-oriented metrics there to explore another kind of resource efficiency. As a typical example, in Sec. V we consider the breaking of RSA encryption. There the relevant metric is the maximal key size that can be broken with a well-defined probability of success. We estimate the energy consumed by full-stack quantum and classical processors as a function of this size. Beyond a typical size, we show that quantum processors are more energy-efficient than classical ones, highlighting a new and essential practical advantage of quantum computing.

III. NOISY SINGLE-QUBIT GATE

We start by applying the MNR methodology to the simplest component of a quantum computer: a resonant noisy single-qubit gate. This allows us to introduce the generic type of qubits we will be working with throughout the paper, which take values inspired by the superconducting platform [15,62,72]. We only consider errors due to spontaneous emission and thermal noise, both of which are unavoidable as soon as the qubit is driven by control lines bringing pulses from the signal-generation stage to the processor (see below). All other sources of noise are neglected.

A. Quantum level

Let us first focus on the characteristics of the gate at the quantum level. The ground and first excited states of the superconducting qubit are denoted by $|0\rangle$ and $|1\rangle$, respectively, with a transition frequency set to $\omega_0 = 2\pi \times 6\,\mathrm{GHz}$. The gate is implemented by driving the qubit with resonant microwave pulses at a frequency ω_0 and an amplitude that induces a classical Rabi frequency Ω. We consider square pulses of duration $\tau_{1\mathrm{qb}}$, giving rise to the qubit Hamiltonian $H = -\frac{1}{2}\hbar\omega_0\sigma_z + \frac{1}{2}\hbar\Omega(e^{i\omega t/2}\sigma_- + e^{-i\omega t/2}\sigma_+)$, with $\sigma_\pm \equiv \frac{1}{2}(\sigma_x \mp i\sigma_y)$. For model simplicity, all single-qubit gates are taken as X gates, i.e., each gate is a π pulse of duration $\tau_{1\mathrm{qb}} = \pi/\Omega$. For driving pulses propagating in control lines, Ω and the spontaneous emission rate γ are not independent, with $\Omega = \sqrt{(4\gamma)/(\hbar\omega_0)}\sqrt{P}$, where P is the average power of the microwave pulse [66,73]. In the present section, dedicated to study at the quantum level, the gate duration $\tau_{1\mathrm{qb}}$ is taken as the control parameter. The resource cost is defined as the power P_π consumed to bring the qubit from $|0\rangle$ to $|1\rangle$:

$$P_\pi = \frac{\hbar\omega_0\pi^2}{4\gamma\tau_{1\mathrm{qb}}^2}. \tag{1}$$

The spontaneous emission rate γ is set by the specific qubit technology; we use $\gamma^{-1} = 1$ ms in Sec. III and $\gamma^{-1} = 10$ ms in Sec. IV. The action of the noise alone is described by a map $\mathcal{N}$, obtained by integrating the Lindblad equation over a time interval $\tau_{1\mathrm{qb}}$,

$$\frac{d\rho}{dt} = \gamma n_{\mathrm{noise}}\mathcal{L}[\sigma_-^\dagger](\rho) + \gamma(n_{\mathrm{noise}}+1)\mathcal{L}[\sigma_-](\rho), \tag{2}$$

where $\mathcal{L}[A](\cdot) \equiv A\cdot A^\dagger - \frac{1}{2}\{A^\dagger A,\cdot\}$, $\{\cdot,\cdot\}$ is the anticommutator, and n_{noise} denotes the number of thermal photons. We assume this same noise map $\mathcal{N}$ for every single-qubit gate and write $\mathcal{G} = G\mathcal{N}$, where $\mathcal{G}$ and G are the maps for the noisy and ideal gates, respectively.

We define our "low-level" metric to quantify the performance of the noisy single-qubit gate as the worst-case gate fidelity,

$$\mathcal{M}_{1\mathrm{qb}} \equiv \min_\rho \mathcal{F}_\mathcal{G}(\rho), \tag{3}$$

where $\mathcal{F}_\mathcal{G}(\rho)$ is the (square of the) fidelity between the output of the ideal gate and the output of the noisy gate. Then, $\mathcal{F}_\mathcal{G}(\rho) \equiv \left[\mathrm{Tr}\sqrt{(G\rho G^\dagger)^{1/2}\mathcal{G}(\rho)(G\rho G^\dagger)^{1/2}}\right]^2$. The concavity of the fidelity ensures that the worst-case fidelity is attained on a pure state. The minimization in Eq. (3) can thus be restricted to over pure states only and $\mathcal{F}_\mathcal{G}(\rho)$ simplifies to $\langle\psi|G^\dagger\mathcal{G}(\psi)G|\psi\rangle = \langle\psi|\mathcal{N}(\psi)|\psi\rangle$ for $\rho \equiv |\psi\rangle\langle\psi| \equiv \psi$. It is useful to rewrite the metric as $\mathcal{M}_{1\mathrm{qb}} \equiv 1 - \mathrm{IF}_{1\mathrm{qb}}$, where $\mathrm{IF}_{1\mathrm{qb}}$ is now the worst-case (i.e., the maximum) gate *in*fidelity. Straightforward algebra yields an expression for $\mathrm{IF}_{1\mathrm{qb}}$ in terms of the gate and noise

parameters:

$$\mathrm{IF}_{1\mathrm{qb}} = \gamma\tau_{1\mathrm{qb}}\,(1 + n_{\mathrm{noise}})\,. \tag{4}$$

$\mathrm{IF}_{1\mathrm{qb}}$ scales like $\gamma\tau_{1\mathrm{qb}}$, the number of spontaneous events during the gate, and increases with the number of thermal photons n_{noise}. Equation (4) provides us with an implicit relation between the noise (γ and n_{noise}), the control ($\tau_{1\mathrm{qb}}$), and the metric ($\mathcal{M}_{1\mathrm{qb}} = 1 - \mathrm{IF}_{1\mathrm{qb}}$). The metric can be increased by reducing the time to perform the gate operation, $\tau_{1\mathrm{qb}}$. However, Eq. (1) tells us that this comes at the cost of increased power consumption.

1. Bare efficiency

We now define the bare efficiency η_0, with "bare" meaning that the resource cost P_π is defined at the quantum level:

$$\eta_0 = \frac{\mathcal{M}_{1_{\mathrm{qb}}}}{P_\pi}. \tag{5}$$

In the MNR methodology, we impose the metric to be equal to a given target value, $\mathcal{M}_{1_{\mathrm{qb}}} = \mathcal{M}_0$. Let us consider the case where the thermal noise is negligible compared to spontaneous decay, i.e., $n_{\mathrm{noise}} \ll 1$, yielding $\mathcal{M}_0 = \mathcal{M}_{1_{\mathrm{qb}}} = 1 - \gamma\tau_{1\mathrm{qb}}$. Now, we wish to minimize the resource cost P_π for the desired $\mathcal{M}_0$. Such minimization is performed as a function of the control parameter $\tau_{1\mathrm{qb}}$ and gives rise to the maximal efficiency $\eta_0^{\max}$. From Eqs. (1) and (4), we can see that $\tau_{1\mathrm{qb}}$ affects both the resource and the metric. This allows us to write $\eta_0^{\max}$ solely as a function of the target metric,

$$\eta_0^{\max}(\mathcal{M}_0) = \frac{\mathcal{M}_0}{P_\pi^{\min}(\mathcal{M}_0)} = \frac{4}{\pi^2}\frac{\mathcal{M}_0(1-\mathcal{M}_0)^2}{\gamma\hbar\omega_0}. \tag{6}$$

Equation (6) tells us that the bigger the target performance metric, the smaller is the efficiency. In other words, increasing the target by one digit (e.g., to take $\mathcal{M}_0$ from 0.9 to 0.99) costs more and more power—we will see this general trend in all our examples below. It also reveals the natural unit of power to be $\gamma\hbar\omega_0$, which is the power dissipated into the environment through spontaneous decay events. The larger the noise rate γ, the larger is the power dissipated, as the gate has to be performed more quickly to maintain the equality $\mathcal{M}_{1_{\mathrm{qb}}} = \mathcal{M}_0$. Hence, $P_\pi^{\min}(\mathcal{M}_0)$ increases [74], decreasing the efficiency. Hence, at the level of single gates, good qubits characterized by small γ are typically more energy efficient than bad ones. This observation will carry through to the macroscopic level in all examples in this work.

B. Macroscopic level

We now model the macroscopic chain of control to take its resource cost into account. Note that from now on, only such macroscopic resource costs will be considered, for which "dressed" efficiencies are appropriate. Here, we present the basic approach within a simplified model, which will be made more realistic in Sec. V. This simple example is limited to a control line funneling driving pulses from outside the cryostat onto the qubit through a single attenuator, with the cryogenics evacuating the heat dissipated by that attenuator. The implementation of the gate is depicted in Fig. 2(a). The qubit is put in a cryostat and cooled down to the temperature T_{qb} (typically below a kelvin). The driving signal is generated at room temperature T_{ext} and sent into the cryostat through a control line. Alongside the signal, the line also brings in unwanted room-temperature thermal noise, which is unavoidable whenever we require external control.

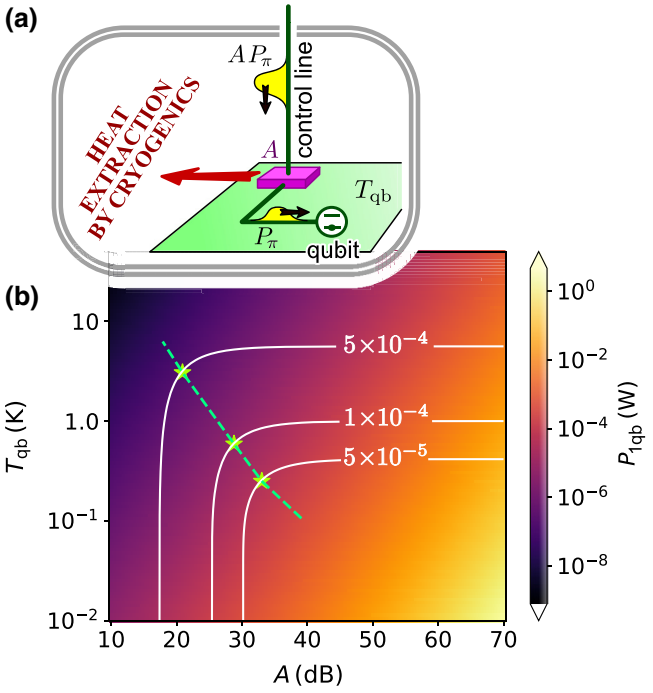


FIG. 2. (a) A sketch of our simplified model of a single qubit in a cryostat. (b) The color scale is the power consumption (in watts) of a single-qubit gate as a function of the qubit temperature T_{qb} and attenuation A. The parameter values are $\gamma^{-1} = 1$ ms and $\tau_{1\mathrm{qb}} = 25$ ns. Each contour is associated with the target metric (worst-case infidelity) indicated in white. The green stars mark the optimal parameters (those that minimize the power consumption) for each value of the target metric.

To mitigate the noise, the signal is first generated with a high amplitude for a strong signal-to-noise ratio. An attenuator is then placed on the line [75] (at the qubit level at temperature T_{qb}), which lowers the input pulse power by an amount A. Thus A and T_{qb} are the two control parameters optimized in the present section. For simplicity, we fix the gate duration to be $\tau_{1\mathrm{qb}} = 25$ ns [15,62,72], chosen to avoid leakage errors [76] that are not modeled here. The two control parameters, A and T_{qb}, impact the gate noise in

the following manner (see, e.g., Eq. (10.13) in Ref. [77]):

$$n_{\text{noise}} = \frac{A-1}{A} n_{\text{BE}}(T_{\text{qb}}) + \frac{1}{A} n_{\text{BE}}(T_{\text{ext}}), \tag{7}$$

where $n_{\text{BE}}(T) = 1/[e^{\hbar\omega_0/(k_B T)} - 1]$ is the Bose-Einstein photon distribution at temperature T. Here, A is expressed in natural units: $A = 10^{A_{\text{dB}}/10}$, where A_{dB} is the attenuation expressed in decibels [78]. The noise model is now entirely defined by Eqs. (2) and (7). Keeping the fidelity in Eq. (3) as the metric, increasing it boils down to increasing the level of attenuation A or decreasing the qubit temperature T_{qb}.

We finally define the macroscopic resource of interest. To get a signal of power P_π on the qubit, a power AP_π is injected into the cryostat, giving $\dot{Q} \approx AP_\pi$ as the rate of heat generation from the attenuator at T_{qb} [79]. We assume Carnot-efficient heat extraction, as it already gives the right order of magnitude for large-scale cooling to cryogenic temperatures that can be done at 10–30% of Carnot efficiency, such as the cooling capabilities at CERN [80]. Then, the cryogenic electrical power consumption (dubbed the "cryo-power" below) needed to run the gate is

$$P_{1\text{qb}}(T_{\text{qb}}, A) = \frac{T_{\text{ext}} - T_{\text{qb}}}{T_{\text{qb}}} AP_\pi. \tag{8}$$

This is the resource that we consider in the present section. Putting together Eqs. (4)–(8), we can see that increasing the metric by reducing T_{qb} or increasing A (taking $A \gg 1$ as in typical experiments) increases the resource cost $P_{1\text{qb}}(T_{\text{qb}}, A)$. This behavior is apparent in Fig. 2(b), where the cryo-power is plotted as a function of A and T_{qb}. If we target a specific value $\mathcal{M}_0$ for the metric, i.e., we require $\mathcal{M}_{1\text{qb}} = \mathcal{M}_0$, this sets an implicit relation between A and T_{qb}, giving rise to the contours marked in the figure.

In the MNR methodology, $\mathcal{M}_{1\text{qb}} = \mathcal{M}_0$ is the constraint under which $P_{1\text{qb}}(T_{\text{qb}}, A)$ is minimized. Using Eqs. (4) and (7), this constraint can be explicitly written as

$$1 - \gamma\tau_{1\text{qb}} \left(1 + \frac{A-1}{A} n_{\text{BE}}(T_{\text{qb}}) + \frac{1}{A} n_{\text{BE}}(T_{\text{ext}})\right) = \mathcal{M}_0. \tag{9}$$

In Fig. 2(b), this is indicated by the white contours, while the points with minimum power consumption are marked with green stars. This provides our first example of a nontrivial sweet spot, where the metric defined at the quantum level impacts the macroscopic resource cost, and is an explicit illustration of the necessity of coordinating inputs from both levels of description.

1. Dressed efficiency

We now minimize the cryogenic power consumption as a function of the two control parameters A and T_{qb}, under the constraint $\mathcal{M}_{1\text{qb}} = \mathcal{M}_0$. We denote this minimum by $P^{\min}_{1\text{qb}}(\mathcal{M}_0)$. This defines the maximal dressed efficiency of the single-qubit gate:

$$\eta^{\max}_{1\text{qb}}(\mathcal{M}_0) = \frac{\mathcal{M}_0}{P^{\min}_{1\text{qb}}(\mathcal{M}_0)}. \tag{10}$$

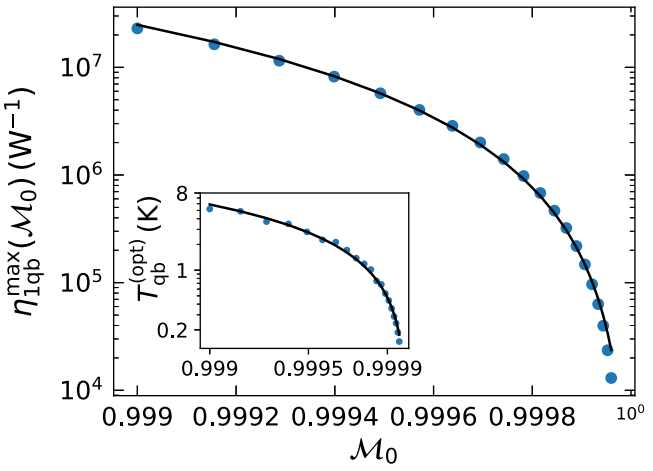


FIG. 3. The maximal dressed efficiency $\eta^{\max}_{1\text{qb}}(\mathcal{M}_0)$ (in W^{-1}) for a single-qubit gate, as defined in Eq. (10). The inset shows the optimal qubit temperature as a function of the target metric $\mathcal{M}_0$. Here, $\gamma^{-1} = 1$ ms.

$\eta^{\max}_{1\text{qb}}$ is plotted in Fig. 3 as a function of $\mathcal{M}_0$, revealing the same behavior as $\eta^{\max}_0$: the larger the target metric, the smaller is the efficiency. The inset gives the qubit temperature that achieves the minimal power consumption, as a function of $\mathcal{M}_0$. The maximal dressed efficiency is much lower than the maximal bare efficiency $\eta^{\max}_0$, with a typical reduction by 3 orders of magnitude. For example, $\eta^{\max}_{1\text{qb}}(0.99965) = 3 \times 10^6$ W^{-1}, while $\eta^{\max}_0(0.99965) \sim 10^{10}\ W^{-1}$. While these two examples are not strictly comparable (the gate duration was optimized for the microscopic efficiency but fixed at $\tau_{1\text{qb}} = 25$ ns for the macroscopic case), the main difference is that the cryogenic power consumption is larger than the microscopic power P_π by a magnification factor of $AT_{\text{ext}}/T_{\text{qb}}$, which can be very large (approximately 2×10^4 for $\mathcal{M}_{1\text{qb}} = 0.99965$). This illustrates the reduction of efficiency when going from the microscopic to the macroscopic level.

IV. EXAMPLE OF NOISY COMPUTATION

Noisy computations are currently considered in the search for use cases with a quantum computational advantage in the noisy intermediate-scale quantum (NISQ) [81] setting, as opposed to fault-tolerant quantum computing (FTQC), which we discuss in Sec. V.

Here, we consider a simplified model of noisy computation (chosen for pedagogy rather than realism), performed

with the simplified qubit model from Sec. III. We use this simplified model to introduce how details of the algorithmic implementation enter MNR as control parameters that can be adjusted to minimize the resource consumption.

Readers who want a more realistic indication of the minimal power consumption of a noise computation should look at the corner of Fig. 10(b) marked $k = 0$ (recalling that $k = 0$ means that there is no error correction). It is based on our complete full-stack model in Sec. V, rather than the simplified model presented here. Although its assumptions are for large-scale fault-tolerant calculations not NISQ ones (it assumes large-scale cryogenics and certain approximations mentioned in Ref. [82]), we expect its conclusion of a few milliwatts per physical qubit to be reasonable for an optimistic estimate of the NISQ regime.

To understand how the algorithm is taken into account by MNR, it is important to note that the same algorithm can be implemented using different circuits. Here, the *algorithm* refers to the overall operation that we want to perform on the qubits, while a *circuit* is an instruction set specifying the sequence of gates on the qubits to carry out the algorithm. In the MNR methodology, the architecture of the circuit can be viewed as a control parameter for the algorithmic task. Our simple example here is the algorithm implemented by the circuit in Fig. 4, which bears structural similarities with a quantum Fourier-transform circuit [83]. It comprises sequences of two-qubit (2qb) gates grouped into subcircuits, marked with different colors in Fig. 4). For Q qubits, this circuit has $Q - 1$ subcircuits and $\frac{1}{2}Q(Q-1)$ 2qb gates. It can be "compressed" by having the subcircuits overlap, noting that there are idling qubits within each subcircuit [84]. We define a compression parameter ϵ, set to be zero for the scenario in which all subcircuits are performed in sequence with no overlap (the top circuit of Fig. 4). We can then make a succession of compression where some subcircuits are partially performed in parallel with their preceding subcircuits. The maximum compression occurs when $(Q-3)/(Q-1)$ subcircuits are partially parallelized in this manner (the bottom circuit of Fig. 4).

In this section, the compression ϵ plays the role of a control parameter of software nature. This comes in addition to the hardware parameters used for the single-qubit gate, namely, the processor temperature T_{qb} and the control-line attenuation A (here taken to be identical for all lines). We will thus minimize the resource cost as a function of the triplet $(T_{\text{qb}}, A, \epsilon)$.

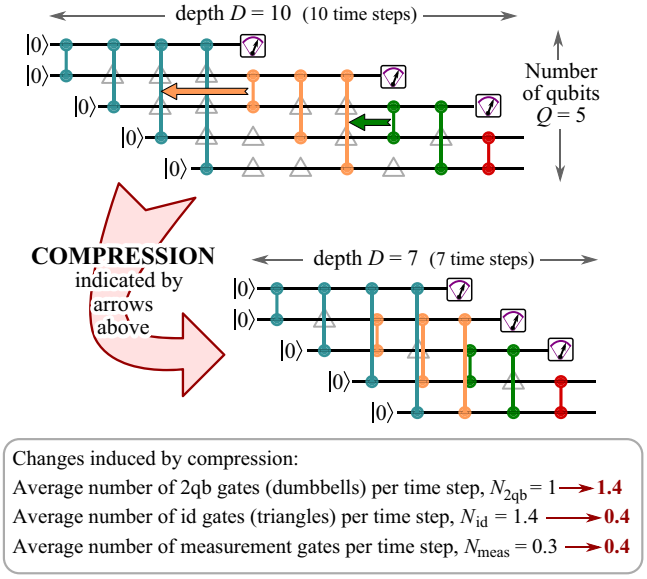


FIG. 4. A hypothetical circuit made of two-qubit (2qb) gates, each indicated by a colored dumbbell. Each qubit has a 2qb gate with each of the qubits below it. Thus, if the circuit has Q qubits, then it contains $\frac{1}{2}Q(Q-1)$ 2qb gates. In the upper circuit, no two 2qb gates are performed in parallel, so its depth is $D = \frac{1}{2}Q(Q-1)$ and there are many time steps in which qubits are simply storing information for a later time step. These are indicated by gray triangles, corresponding to noisy identity (id) gates. One can "compress" the circuit step by step, by moving 2qb gates in the direction of the arrows shown on the upper circuit. This increases the average number of 2qb gates per time step, reducing the number of id gates and thereby reducing D. The lower circuit is the fully compressed version of the upper circuit.

A. Noise model and low-level metric

In this section, we consider circuits built from a typical minimal gate set consisting of identity (id), single-qubit (1qb), and two-qubit (2qb) gates acting on Q qubits. We assume that the 2qb gates are implemented with a cross-resonance scheme [85,86] in which the two qubits interact by sending a microwave signal to one qubit at the frequency of the other qubit. Such 2qb gates rely on resonant excitations similar to those employed in the 1qb gates. We thus assume the 1qb and 2qb gates to have similar costs and take that cost to be P_π of Eq. (1). The 2qb gates are, however, slower than the 1qb gates and we set $\tau_{\text{2qb}} = 100$ ns [85,86]. Finally, the quantum computer runs at a clock frequency determined by its slowest gate. We thus set the clock period, or the time step for gate applications, to be $\tau_{\text{step}} = \tau_{\text{2qb}} = 100$ ns.

We first establish the relation between the local noise afflicting individual gate operations and the global metric characterizing the overall circuit performance.

We follow Sec. III in assuming that the only noise felt by the qubits is the unavoidable noise coming from the control lines. This is modeled as simple probabilistic noise in which each qubit has a probability of having an error during one time step equal to the worst-case infidelity IF_{1qb} of the process at each time step, determined from Eq. (2). Here, IF_{1qb} is defined as in Eq. (4) but τ_{1qb} is replaced by $\tau_{\text{step}} = 100ns$. Then, a two-qubit gate has twice the infidelity of a single-qubit gate, because two qubits participate

in a two-qubit gate. So each id gate and each 2qb gate, respectively, have probabilities equal to $\mathrm{IF}_{1\mathrm{qb}}$ and $2\,\mathrm{IF}_{1\mathrm{qb}}$ of generating an error in the computation.

To quantify the algorithmic performance, we choose a low-level metric, $\mathcal{M}_{\mathrm{algo}} = 1 - P_{\epsilon}^{\mathrm{error}}$, where $P_{\epsilon}^{\mathrm{error}}$ is the probability that at least one error has occurred within the circuit with compression ϵ. For the algorithm to have a reasonable chance of success, $\mathrm{IF}_{1\mathrm{qb}}$ should be small. Because of that, the error probability of the circuit can be approximated by $P_{\epsilon}^{\mathrm{error}} = \mathcal{N}_g(\epsilon)\mathrm{IF}_{1\mathrm{qb}}$, where $\mathcal{N}_g(\epsilon) \equiv \big(\mathcal{N}_{\mathrm{id}}(\epsilon) + \mathcal{N}_{1\mathrm{qb}}(\epsilon) + 2\mathcal{N}_{2\mathrm{qb}}(\epsilon)\big)$. Here, $\mathcal{N}_i(\epsilon)$ is the total number of gates of type i in the circuit with compression ϵ. From this, we deduce that

$$\mathcal{M}_{\mathrm{algo}}(\epsilon, A, T_{\mathrm{qb}}) = 1 - \mathcal{N}_g(\epsilon)\,\mathrm{IF}_{1\mathrm{qb}}(A, T_{\mathrm{qb}}). \tag{11}$$

Equation (11) makes explicit the influence of control parameters of software (ϵ) and hardware (A, T_{qb}) natures on the global performance of the algorithm.

B. Resource cost

Whenever the calculation time is a parameter (as it is when we introduce the circuit compression shown in Fig. 4), minimizing the average power consumption during the calculation is *not* the same as minimizing the energy cost of the calculation (since that energy cost is the average power times the calculation time). So should we minimize the power or the energy?

We argue that the power consumption should be minimized whenever that power consumption is large enough to be the principal engineering challenge. For example, there are many engineering reasons why it is much harder to consume 1 GW for 1 min, rather than 250 kW for 3 days, even though the two have similar total energy costs. Our main full-stack calculation, in Sec. V, is in the regime in which the power consumption is so high that it will be a huge engineering challenge. Thus it is critical to minimize this power. Hence, for simplicity, we also minimize the power consumption for the pedagogical examples in the work, including for the simplified model of a NISQ calculation considered in this section.

In many cases, we believe that both minimizations will give similar results. Minimizing the energy cost will tend to promote shorter calculation times than minimizing the power consumption alone (since it corresponds to minimizing the power times the calculation time). However, we observe that minimizing the power consumption already tends to favor solutions with fairly short calculation times (see, e.g., Sec. V E 4). So the parameters that minimize the power consumption may not be far from those that minimize the energy cost.

We take the resource cost to be the total cryo-power averaged over a specified circuit that implements the algorithm. This is given by

$$P_{\epsilon}(A, T_{\mathrm{qb}}) = P_{1\mathrm{qb}}(A, T_{\mathrm{qb}})N_{1\mathrm{qb},\epsilon} + P_{2\mathrm{qb}}(A, T_{\mathrm{qb}})N_{2\mathrm{qb},\epsilon}. \tag{12}$$

Here, the cryo-powers supplied to perform a 1qb and 2qb gate are $P_{1\mathrm{qb}}(A, T_{\mathrm{qb}})$ and $P_{2\mathrm{qb}}(A, T_{\mathrm{qb}})$, respectively, and it is assumed that id gates require no power. $N_{1\mathrm{qb},\epsilon}$ and $N_{2\mathrm{qb},\epsilon}$ are the average number of 1qb and 2qb gates, respectively, run in parallel per time step of the circuit with compression ϵ. Since we consider the power consumption during the execution of the algorithm, we shall only consider one run. The energy considerations of specific NISQ algorithms such as the variational quantum eigensolver (VQE) or the quantum approximate optimization algorithm (QAOA) may require one to take into account the number of runs needed to reach a result with a certain accuracy.

In our plots—see Fig. 5 and onward—we have to choose certain parameters. There, we assume that each time step is 100 ns, where this is the time for a 2qb gate, whereas 1qb gates take only 25 ns. We assume that it takes about the same power to drive 1qb and 2qb gates, given by Eqs. (1) and (8). However, as a 1qb gate is completed in a quarter of a time step, its power consumption averaged over the time step is a quarter that in Eq. (8), so $P_{1\mathrm{qb}}(A, T_{\mathrm{qb}}) = P_{2\mathrm{qb}}(A, T_{\mathrm{qb}})/4$.

Equations (11) and (12) allow us to optimize the total cryo-power as a function of the three control parameters, $(A, T_{\mathrm{qb}}, \epsilon)$, under the constraint of a target metric $\mathcal{M}_{\mathrm{algo}} = \mathcal{M}_0$. To make the impact of the circuit compression obvious, we first minimize the cryo-power with respect to A and T_{qb}, for various values of the compression ϵ. We have performed this optimization for a circuit with $Q = 25$ qubits (see Fig. 5). The plot shows that the sweet spot of minimal power consumption occurs when the circuit is partially compressed, revealing a competition

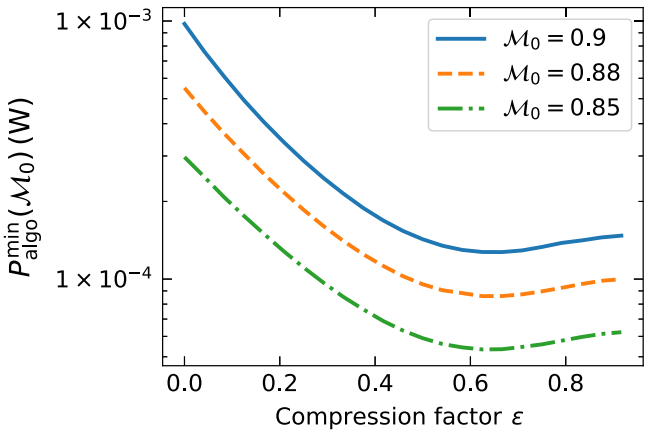


FIG. 5. The minimum power as a function of the compression ϵ of the circuit (see text). The circuit is a 25-qubit version of the one shown in Fig. 4. Here, $\gamma^{-1} = 10$ ms.

between two mechanisms. On the one hand, compressed circuits correspond to a reduced total number of gates $\mathcal{N}$ (including id gates), with a reduced risk of error according to Eq. (11), but to a larger number of gates run in parallel, leading to a larger power consumption. On the other hand, uncompressed circuits, with more idling qubits, increase the total error probability, which has to be compensated for by lowering the error rate per gate, which is achieved by lowering $T_{\rm qb}$, hence also increasing the cryo-power per gate (see Fig. 2). This demonstrates that resource optimizations require coordinated inputs from the hardware and the software.

Figure 5 illustrates that minimizing the average power consumption is not equivalent to minimizing the energy cost of the calculation. Multiplying this power by the calculation time (with its relation to the compression factor explained above), we see that one can obtain a lower total energy cost for a higher compression factor (shorter calculation time) than that which minimizes the power consumption. However, one also sees that the difference is not huge (less than a factor-of-2 difference in energy consumption for the simple model in Fig. 5), so a circuit optimized for minimum power consumption will not be far from one optimized for minimum energy consumption.

C. Resource efficiencies

Resource efficiencies for quantum algorithms executed on noisy circuits can be defined in two ways, depending on the target performance. First, one may adopt a low-level metric, such as the fidelity used above. This invites us to define an efficiency $\eta_{\rm algo} \equiv \mathcal{M}_{\rm algo}/P_\epsilon(A, T_{\rm qb})$ for a specified circuit implementation of the algorithm. For a given target metric $\mathcal{M}_0$, minimizing $P_\epsilon(A, T_{\rm qb})$ as a function of A, $T_{\rm qb}$ and the circuit compression ϵ defines the maximal algorithmic efficiency $\eta^{\rm max}_{\rm algo}(\mathcal{M}_0)$. This is plotted in Fig. 6 as a function of $\mathcal{M}_{\rm algo} \equiv \mathcal{M}_0$, for circuits with $Q = 25$ qubits. While it follows the same behavior as the single-qubit gate efficiency, a direct comparison is difficult, because of the complexity of the relationship between the fidelity of a single gate and the fidelity of a whole circuit.

Second, the performance of an algorithm can be quantified by user-oriented metrics. A typical such metric is the size of the problem, which can be solved with a given probability of success. In this example, it can be measured by the size of the data register Q that carries out the algorithm, assuming that the algorithm is executed with a $2/3$ success probability. The maximal user-oriented efficiency is given by $\mathrm{H}^{\rm max}_{\rm algo} = Q/P^{\rm min}_{Q,\epsilon}(A, T_{\rm qb})$. Here, we have introduced $P^{\rm min}_{Q,\epsilon}(A, T_{\rm qb})$, the minimal cryo-power for a success probability of $2/3$ for a circuit of size Q, optimized with respect to the compression ϵ, attenuation A, and processor temperature $T_{\rm qb}$. The maximal efficiency is plotted in Fig. 7 as a function of the target metric Q.

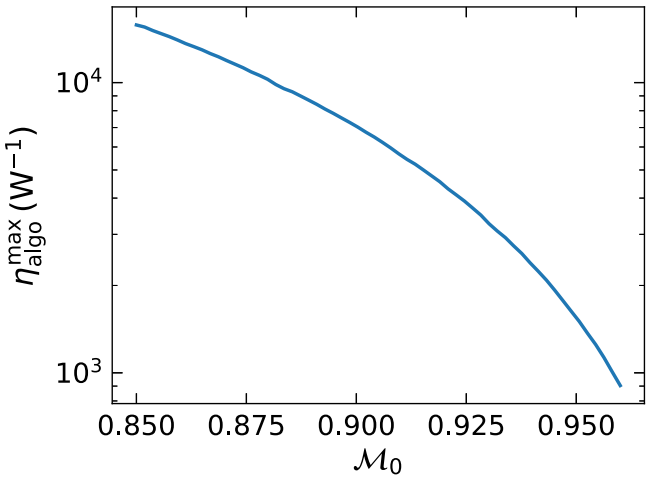


FIG. 6. The optimal efficiency $\eta^{\rm max}_{\rm algo}(\mathcal{M}_0)$ as a function of the target algorithmic fidelity $\mathcal{M}_0$. The circuit is a 25-qubit version of the one shown in Fig. 4. Here, $\gamma^{-1} = 10$ ms.

This section provides a pedagogical example to understand the impacts of the hardware and software choices on the energy consumption of a quantum computation. It also allows us to play with two different kinds of metrics and efficiencies, either low level or user oriented. To be truly informative, the user-oriented efficiency should be compared to a classical value quantifying the efficiency of a classical processor performing the same algorithm. A larger efficiency reached by the quantum processor provides a signature of a quantum energy advantage. This regime and the potential to reach it are studied in Sec. V, in the context of fault-tolerant computation.

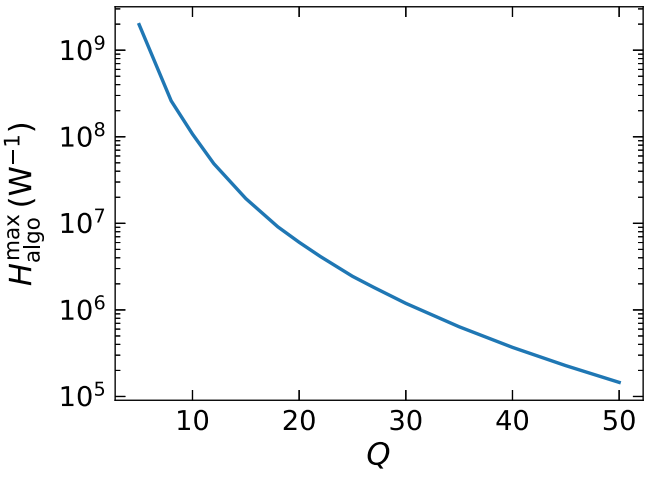


FIG. 7. The optimal efficiency $H^{\rm max}_{\rm algo}(Q)$ as a function of the target metric Q, the number of qubits needed for an algorithmic success probability of $2/3$ (see text). Here, $\gamma^{-1} = 10$ ms.

V. FAULT-TOLERANT COMPUTATION

We now turn to the macroscopic power consumption and energy efficiency of fault-tolerant quantum computation, currently the only known route to useful large-scale quantum computers. Fault-tolerant quantum computation is built upon the technique of quantum error correction. The basic idea of quantum error correction is to distribute each qubit of information over many physical qubits, to form what is known as a logical qubit. This gives the logical qubit some resilience against noise that usually affects physical qubits individually. It requires, however, the use of several physical qubits to carry one logical qubit and the addition of regular error-correction operations, namely, syndrome measurements to diagnose what errors have occurred, and recovery gates to remove the effects of those errors on the logical qubit. This means a significant increase in the number of qubits, measurements, and gates, each of which has imperfections and can potentially add noise to the computer. Computing with such logical qubits is said to be done in a fault-tolerant manner if the error-correction operations, as well as the computational operations on the logical qubits (i.e., the logical gates), are designed so that the addition of so many more noisy physical components for the error correction still has the net effect of removing more errors than it introduces. This turns out to be possible only if the physical error rate is below some threshold level, often referred to as the fault-tolerance threshold.

From the user perspective, the fault-tolerant nature of the quantum computer is invisible. The user states the problem to be solved, and the algorithm to solve it, in terms of an ideal (noise-free) operation performed on a given input and specifies a target metric (e.g., the probability of success). This is then converted by the compiler to physical noisy qubits, gates, and measurements, using a prescribed fault-tolerant quantum computing scheme. The user-given input is represented by the logical qubits, encoded into the physical qubits carrying the information. The logical gates between those logical qubits that carry out the steps of the user-specified algorithm are converted into a sequence of physical gates between the physical qubits that make up each logical qubit.

For our simulation, we shall consider fault-tolerant quantum computing built from concatenating a seven-qubit code [83,87–90]. This is a very well-studied scheme and it has the advantage over more recent proposals (e.g., those based on topological codes) in that it has fairly complete and well-documented analyses, allowing us to be sure that we do not overlook any resource requirements. However, it is widely believed that fault-tolerant proposals based on surface codes require vastly less resources than the seven-qubit code. In Sec. V G, we extend our results to such surface codes, confirming this but pointing out open questions there that possibly make our estimates unreliable. The advantage of our complete analysis of the seven-qubit code allows the reader to clearly see what questions they need to answer before doing similar estimates with their favorite fault-tolerant scheme.

A. MNR on a fault-tolerant algorithm

Before coming to the specifics of our model, we provide the reader with a general view on the approach that is valid for any quantum error-correcting code. We let p_{err} denote the error probability of a physical qubit [91], which is provided by a microscopic model of the noise. If the error correction is successful, the error probability of a logical qubit is reduced to $p_{\mathrm{err};L} = f\,(k, p_{\mathrm{err}})$, where f is a function and k quantifies the amount of error correction—the concatenation level in the case of our concatenated seven-qubit-code example. The price to pay for this reduction of errors is that the number of physical qubits per logical qubit grows with k; we denote this number by $g(k)$.

Throughout this section, we consider a simple "rectangular" circuit, with the goal of preserving Q_L (logical) qubits of quantum information, for a total of D_L (logical) time steps. Such a rectangular circuit approximates well many fault-tolerant quantum algorithms based on Q_L qubits and having a circuit with depth D_L and still yields similar orders of magnitude for the power consumption and the metric (see Sec. V D). As (Q_L, D_L) is set by the choice of algorithm and circuit, k is the only software parameter that we are left with to perform our optimizations.

The metric $\mathcal{M}_{\mathrm{FT}}$ that we will consider is the probability of success of the rectangular circuit. Denoting the number of locations where logical errors can happen as $\mathcal{N}_L = Q_L \times D_L$, we find that

$$\mathcal{M}_{\mathrm{FT}} = (1 - p_{\mathrm{err};L})^{\mathcal{N}_L}. \tag{13}$$

Targeting a total success probability of $\mathcal{M}_{\mathrm{FT}} = 2/3$, it translates into a maximal allowed value for $p_{\mathrm{err};L}$. This maximal allowed value shrinks as the size $\mathcal{N}_L$ of the circuit grows. Hence, performing bigger computations while maintaining the same target metric mandates more error correction and hence the consumption of more physical resources.

Estimating the physical resource cost requires the use of a full-stack model. Elaborations of the simple cases studied earlier to give a full-stack model that incorporates more experimental details are presented in Sec. V C, leading to a larger set of hardware control parameters. We also need to specify the physical circuit that carries out the quantum algorithm, which depends on the parameter k. Altogether, we can establish the generic expression of the full-stack power consumption P_{FT}:

$$P_{\mathrm{FT}} = P_{1\mathrm{qb}} N_{1\mathrm{qb}} + P_{2\mathrm{qb}} N_{2\mathrm{qb}} + P_{\mathrm{meas}} N_{\mathrm{meas}} + P_Q Q. \tag{14}$$

The first three terms capture the dynamical power consumption: they are nonzero only when a computation is running and involve active gates and measurements. Measurements must be modeled since syndrome measurements for error correction take place all along the fault-tolerant quantum computation. As in Eq. (12), N_{1qb} and N_{2qb} are the average numbers of physical 1qb gates and 2qb gates, respectively, performed in parallel, while N_{meas} is the average number of physical qubit measurements performed in parallel. These three quantities are determined solely by the software, i.e., the algorithm, the choice of error-correcting code, and the parameter k. Conversely, P_{1qb}, P_{2qb} and P_{meas} are, respectively, the full-stack power consumption of 1qb gates, 2qb gates, and measurements, including all cryogenic and electronic costs; these depend solely on the hardware parameters. Finally, the fourth term in Eq. (14) captures the static power consumption, which we will take to be proportional to the number of physical qubits, $Q = g(k) \times Q_L$. Its expression depends both on software and hardware parameters.

The MNR methodology then simply consists of the following steps. (i) Consider an algorithm characterized by (Q_L, D_L) and a target probability of success equal to $2/3$. As in Sec. IV, this $2/3$ is a common choice for the success probability for a single run of an algorithm, with an exponential chance of yielding the correct answer with a constant number of reruns. Owing to Eq. (13), this sets an implicit relation between the hardware control parameters and k. (ii) Minimize the power consumption P_{FT} as a function of the control parameters under the constraint of reaching a probability of success $\mathcal{M}_{FT} = 2/3$.

B. Noise and metric for the seven-qubit code

Let us first consider the noise at the level of a single physical qubit. Instead of the infidelity, we consider the error probability p_{err} of the qubit. We employ the same noise model as that of Sec. III. We write $p_{err} = \frac{1}{2}\gamma\tau_{step}\left(\frac{1}{2} + n_{noise}\right)$ [82]. Here, τ_{step} is the time step of the quantum computer, taken to be equal to the time taken to perform the slowest qubit gate, i.e., the 2qb gate in our model.

From now on, we focus on the seven-qubit code. The basic components of the fault-tolerance scheme are illustrated in Fig. 8, starting at the top with the logical circuit to be implemented (drawn here, for simplicity, for just single-logical-qubit gates). Each logical gate is broken down into the physical qubits and gate operations that are needed to implement it in a fault-tolerant manner, with qubits that carry the actual logical information, as well as ancillary qubits (or just "ancillas") that permit the syndrome measurements for error correction. Also shown in Fig. 8 are the details of the preparation of the state of the ancilla for the error correction to work in a fault-tolerant manner. These details are critical inputs to our power-consumption calculations below (see Appendix A).

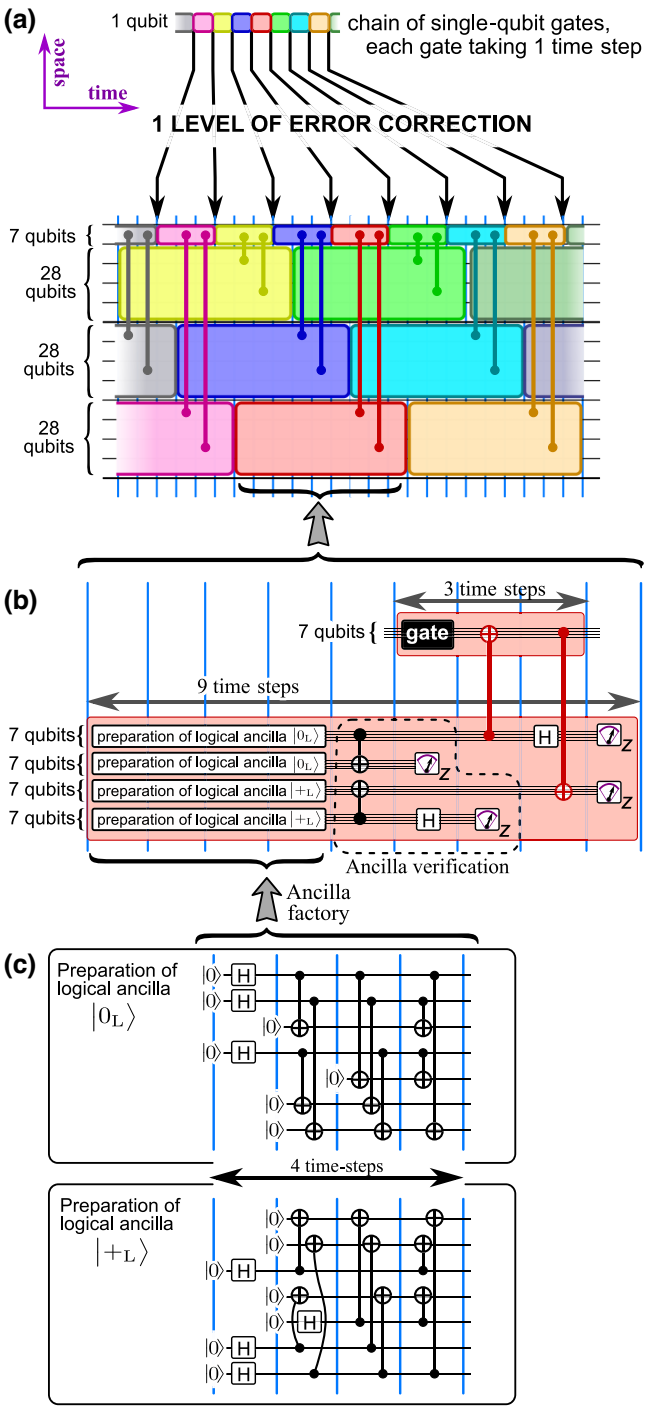


FIG. 8. The circuit for one level of the seven-qubit error-correction code on a series of single-qubit gates [89,90]. Gates acting on groups of seven qubits are transversal. The *ancilla factory* prepares the state of the logical ancillas [92]. The vertical blue lines indicate time steps in the algorithm. See the main text for a brief description of this circuit, with additional technical details given in Appendix A.

The power of the code can be increased, thereby acquiring the ability to remove more errors, by concatenating the basic seven-qubit code. At the first level of concatenation, the logical qubit is encoded into seven physical qubits; at the next level of concatenation, the logical qubits of the previous level are treated like physical qubits, and the logical qubit at this level is encoded into seven logical qubits of the previous level, thus employing 7^2 physical qubits in all; and so on in a recursive manner. Error correction is done at every level of the concatenation. After k levels of concatenation, the error probability per logical qubit per (logical) time step can be shown to be [90]

$$p_{\mathrm{err};L} = p_{\mathrm{thr}} \left(p_{\mathrm{err}}/p_{\mathrm{thr}}\right)^{2^k}, \tag{15}$$

where $p_{\mathrm{thr}} \approx 2 \times 10^{-5}$. Here, $1/p_{\mathrm{thr}}$ is an integer that counts the number of ways in which the extra physical elements (qubits, gates, and measurements) added to correct errors can have faults (for a fuller explanation, see, e.g., Ref. [83]). p_{thr} is the aforementioned fault-tolerance threshold: The error per logical qubit decreases as k increases only if the qubit error probability p_{err} is less than p_{thr}. This is an important constraint on the physical qubits that we will consider for our simulations, requiring fidelities that can be significantly beyond the state of the art.

Increasing k increases the ability to remove errors and hence compute more accurately. The price to pay, however, is a large increase in the number of physical qubits. For a computation with Q_L logical qubits, the fault-tolerant scheme requires Q physical qubits, where

$$Q \equiv g(k) \times Q_L = (91)^k Q_L. \tag{16}$$

This formula can be understood from Fig. 8, which illustrates how fault-tolerant quantum computation with the seven-qubit code works. We focus on the first level of concatenation, where one logical qubit is encoded into seven physical qubits. In addition to the seven physical qubits, one needs 28 physical qubits as ancillas to facilitate syndrome measurements for the code. The 28 ancillas are explained in Refs. [89,90]. In short, they should be understood as two groups of 14 ancillas each, with one group for each of the two kinds (X or Z) of syndrome measurements needed for the seven-qubit code. Each group of 14 should again be thought of as two groups of seven ancillas; one of these groups is used to verify the quality of the ancillas in the other group, which is necessary to guarantee fault tolerance. Furthermore, the ancillas have to be prepared in specific states for the syndrome measurement. As the ancilla preparation takes a certain number of time steps to complete (four time steps, as shown in Fig. 8), in order for all 28 ancillas to be ready at the time at which they are needed in the syndrome measurement, we find that, at any one time step, there must be three groups of 28 ancillas each in various stages of preparation (see Fig. 8). This then gives the $91 = 7 + 3 \times 28$ in Eq. (16), for $k = 1$. Then, the recursive structure of the concatenation, treating each logical qubit as if it were a physical qubit at the next level, gives the $(91)^k$ factor for k levels of concatenation in Eq. (16).

This systematic analysis allows us to derive the number of 1qb gates, 2qb gates, and measurements running in parallel as needed in Eq. (14) to estimate the power consumption; the details are given in Appendix A. Finally, the overall metric introduced in Sec. V A for our generic algorithm is given by

$$\mathcal{M}_{\mathrm{FT}} = \left[1 - p_{\mathrm{thr}} \left(p_{\mathrm{err}}/p_{\mathrm{thr}}\right)^{2^k}\right]^{\mathcal{N}_L}. \tag{17}$$

For simplicity, we use the linear approximation,

$$\mathcal{M}_{\mathrm{FT}} = 1 - \mathcal{N}_L p_{\mathrm{thr}} \left(p_{\mathrm{err}}/p_{\mathrm{thr}}\right)^{2^k}, \tag{18}$$

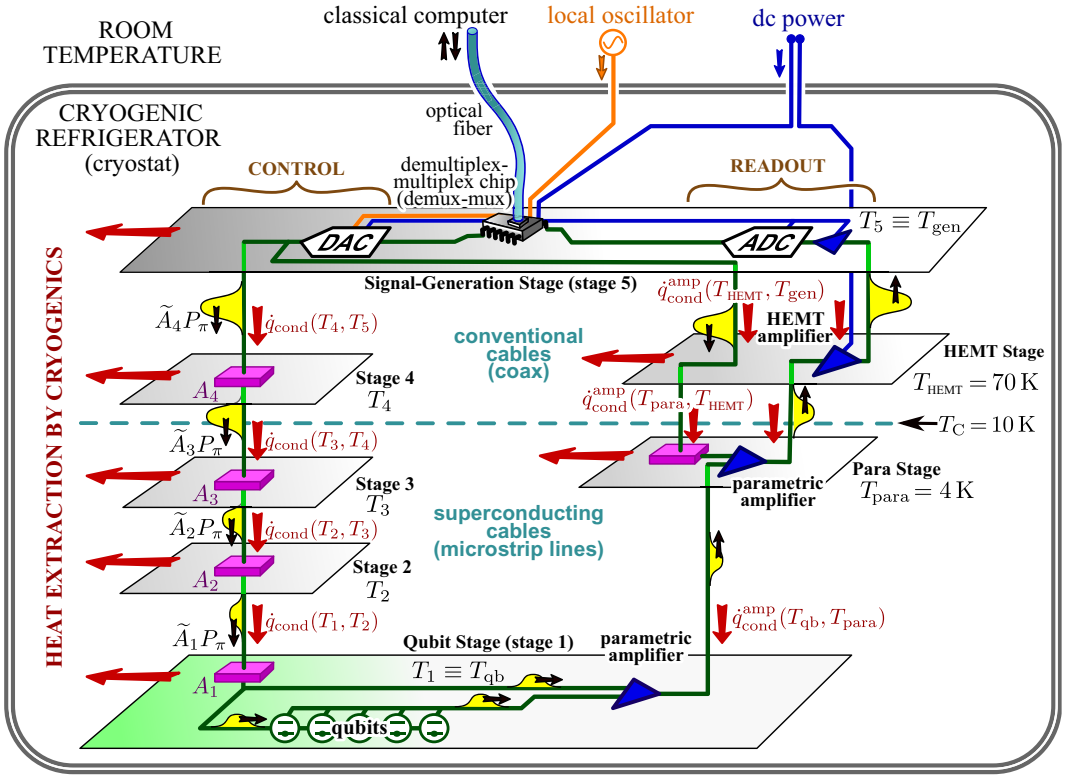
which slightly overestimates the effect of the errors (i.e., slightly underestimates the metric).

C. Full-stack hardware model

We now present our full-stack model, which goes significantly beyond the pedagogical model used in earlier sections. In short, we replace the simplified setup of Fig. 2(a) by the full setup of Fig. 9. This involves key improvements over the simplified setup of Fig. 2(a) that bring us closer to experimental reality. These improvements dealing with the control electronics and the cryogeny are presented below, with more details in the appendixes. We take inspiration from current technologies for the improved model. Nevertheless, our interest is in understanding general trends that will provide guidelines for ongoing and future research and this leads us to consider values that are beyond the current state of the art.

1. Cryogenic model

The first improvement to bring us closer to experimental reality is that we spread the attenuation on the microwave control lines over multiple temperature stages (see the left-hand side of Fig. 9). This is known to be much more energy efficient than placing all the attenuation at T_{qb}, as we had done in Fig. 2. Much of the heat generated by attenuators is thus dissipated at higher temperatures, where it costs much less power to extract it. Adding more temperature stages always reduces power consumption but it is often technically challenging. We observe that the benefits of adding another stage becomes small once there are about five stages, so we take five temperature stages here. Appendix B 1 gives the detailed specifications of these five stages of attenuation. The heat conducted by the control lines turns out to be significant and to minimize this



Component	Ratio to qubits
optical fiber + demux-mux	1 for 270 qubits
local oscillator	1 wire (mm ⌀) for 10^6 qubits
dc (demux-mux+DAC+ADC)	1 wire (mm ⌀) for 10^6 qubits
dc (HEMT amplifiers)	1 wire (mm ⌀) for 10^8 qubits
Qubit control (1 & 2qb gates)	1 DAC+cable (with attenuators) for 25 qubits
Qubit readout (measurement)	1 cable+paramps+HEMT amp+ADC for 100 qubits

FIG. 9. A sketch of our model of the multistage cryogenics with all components. It is important to maximize the number of physical qubits per other component (using multiplexing etc.), and the table gives reasonable values for the ratio of the number of components to the number of physical qubits. The qubit control lines are particularly crucial to the energy consumption; we model them with four stages of attenuation (attenuators in purple), with conventional coaxial cables down 10 K and superconducting microstrip lines below that. The readout is less crucial to the energy consumption if one uses superconducting parametric amplifiers (paramps) at $T_{\rm qb}$ and $T_{\rm amp} = 4$ K, with a third amplification stage using high-electron-mobility transistors (HEMTs) at $T_{\rm HEMT} = 70$ K. The black arrows indicate the flow of information and/or signals, while the red arrows indicate heat conduction. The demux-mux, digital-to-analog converters (DACs), analog-to-digital converters (ADCs), attenuators, and amplifiers all also generate heat.

heat conduction, we assume all wiring to be superconducting below 10 K and thus to conduct vastly less heat than normal-metal wires. The heat-conduction properties of these control lines are given in Appendix B 2. As above, we assume that the cryogenics have Carnot efficiency and thus use the minimal possible power to evacuate heat as allowed by the laws of thermodynamics. We take this for simplicity as it already gives the right order of magnitude for large-scale cryogenics, where the state of the art is 10–30% of Carnot efficiency [80]. Evidently, the results change if one considers small-scale cryostats that operate far below Carnot efficiency, as shown in the example in Appendix E.

2. Control electronics

A second improvement to bring us closer to experimental reality is that we now add the circuitry to read out the qubits (see the right-hand side of Fig. 9). The signal from the qubits has to be amplified significantly above the thermal noise level at the temperature stage to which the signal is being sent. As amplifiers generate heat, it is again much more energy efficient to have a chain of amplifiers at different temperature stages than to have all the amplification occur at the qubit temperature. Superconducting parametric amplifiers (paramps) generate much less heat than conventional amplifiers but they can only operate at temperatures below 10 K. At

higher temperatures, the best option is amplifiers based on high-electron mobility transistors (HEMTs). Here, we take the amplification chain inspired by recent experiments [60]: We assume one superconducting paramp at the qubit temperature, which sends the signal to another superconducting paramp at 4 K. This then sends the signal to a HEMT amplifier at 70 K, which finally sends it to the chip that reads out the signal. Appendix B 3 gives the detailed specifications for this chain of amplifiers. The readout lines are the same materials as the control lines, so their heat-conduction properties are those described in Appendix B 2.

The third and final improvement is that we assume there is a signal-generation stage at temperature T_{gen}, with chips that carry out the signal generation and readout. Below, we want to find the optimal value of T_{gen} that minimizes the power consumption. For this, we need to know the heat dissipated by the signal-generation stage, which requires a specification of what it contains. Our model assumes that the signal-generation stage receives digitized instructions of the wave form to generate for each gate operation down an optical fiber from a conventional (classical) room-temperature computer. The signal-generation stage contains a demultiplex (demux) chip that demultiplexes the photonic signal in the optical fiber and turns it into digital electrical signals. These digital signals are turned into analog signals in the digital-to-analog converters (DACs) and are then superimposed on the local-oscillator signal (at 6 GHz) to make the microwave signal that performs the desired gates on the target qubits. At the same time, the signal-generation stage takes the microwave wave form coming from the measurement of the qubit through the amplifiers and digitizes it in the analog-to-digital converter (ADC). This is then turned into a multiplexed photonic signal, which is sent through the optical fiber to the conventional room-temperature computer. This (classical) computer demultiplexes it and digitally demodulates the wave form, allowing it to deduce the state of the qubit in question [62]. It also decodes (i.e., interprets) the syndromes coming from the error-correction procedure and manages the algorithm at the logical level. Further details of this are provided in Appendix B 5, which argues that this classical computer will not be a significant contribution to the power consumption, and so can be neglected at our level of approximation.

3. Control parameters

We can now summarize the four control parameters that we will use for our optimizations, namely, T_{qb}, T_{gen}, A (the *total* attenuation on the lines), and k, the concatenation level. The temperature of each stage and the amount of attenuation put on these stages are taken to be functions of T_{qb}, T_{gen} and A [see Eq. (B11) in Appendix B]. As explained around Eq. (B11), we consider such constraints to lead to a relatively optimal distribution of attenuation and temperatures.

D. Full-stack power cost for the seven-qubit code

1. Software assumptions

We first write down Eq. (14), describing the power consumption for the specific case of fault-tolerant quantum computing based on the seven-qubit code. For this, we need to look at the circuit for one level of error correction for any Clifford logic gate (see Fig. 8). The circuit looks the same for such any logic gate, except for the contents of the black box marked "gate" [this "gate" in Fig. 8(b) is transversal, containing seven physical gates corresponding to the logic gate]. However, this black box makes a very small contribution to the total number of gate operations in the circuit, so once the error correction is included (i.e., $k \geq 1$), all logical gates require about the same number of physical gate operations. Thus one expects that any logical gate will have a power consumption very similar to that of a (logical) identity gate that does nothing except preserve the quantum state of the logical qubit. This intuition is confirmed and carefully quantified in Appendix A. As a result, the power consumption of any given algorithm is almost independent of what the algorithm is actually doing at the logical level; it only depends on the number of logical qubits Q_L and logical depth D_L of that algorithm. We can thus take the power consumption of any algorithm to be close to that of a logical memory the only job of which is to preserve the state of Q_L logical qubits for D_L logical time steps.

The power consumption of such a circuit can be taken as proportional to Q_L (see Appendix A), with

$$P_{\text{FT}} \simeq Q_L \left(\frac{4(64)^k}{185} \left[16P_{\text{2qb}} + 7P_{\text{1qb}} + 7P_{\text{meas}} \right] + (91)^k P_{\text{Q}} \right), \tag{19}$$

using an approximation that gets better at higher k. Appendix A shows that this approximation gets the order of magnitude right for any circuit of Clifford gates at $k = 1$ and is within a few percent of the correct result for $k \geq 2$.

To keep the modeling here as compact as possible, we neglect the power consumption associated with fault-tolerant non-Clifford gates (such as T gates). While a quantum computer without at least one type of non-Clifford gate is not universal (and can be efficiently simulated on a classical computer), the modeling of non-Clifford gates is very different than that of Clifford gates. Appendix A 2 discusses this modeling and points out how rare non-Clifford gates are in the algorithms that we consider. It then argues that accounting for them would complicate the modeling without significantly changing the resulting power consumption.

2. Hardware assumptions

What remains to be calculated is the contribution of each hardware component to P_{1qb}, P_{2qb}, P_{meas}, and P_Q. We compute P_{1qb} and P_{2qb} in the same manner as in the noisy quantum circuit in Sec. IV, except that we now also account for the chain of attenuators at different temperatures. The expressions for P_{1qb} and P_{2qb} are given in Appendix B 6. For P_{meas}, we use a similar approach to that for P_{1qb}, as the measurement in our model involves sending a microwave signal similar to that for a gate operation. Our estimations, however, show that in most cases P_{meas} is negligible compared to P_{1qb}, so we drop it. P_{1qb}, P_{2qb}, and P_{meas} grow whenever the qubit temperature is reduced to raise the physical qubit fidelity. A larger physical qubit fidelity hence requires a larger power consumption for gates and measurements.

For our simulations, we make the qubit lifetime vary between 3.5 ms and 1 s. However, our main discussion will be based on a lifetime of 50 ms, which is about 100 times better than the state of the art in transmon qubits [93]. This is necessary since, as mentioned above, a successful calculation using a fault-tolerant scheme built from the seven-qubit code requires an error probability smaller than the threshold of $p_{thr} = 2 \times 10^{-5}$, which is achievable only for qubits with a long enough lifetime.

Next, P_Q is the part of the power consumption that scales as the number of physical qubits, independent of whether gates or measurements are being performed. It has two different contributions. The first is the power consumption of the cryogenics to remove the heat conducted down the microwave lines that control and read out the qubits. Their thermal properties are given in Appendix B 2. The second is the heat generated by all electronics that are always on. This includes the amplifiers at 4 K and 70 K, the electronics for control and readout at T_{gen}, and the classical computer at ambient temperature. Their detailed specifications are given in Appendix B 3, with the full list of parameters summarized in Tables II and III.

We consider three generic scenarios, labeled A, B, and C, for the control electronics. Scenario A can be taken as a futuristic scenario for conventional CMOS technology, where the control electronics typically dissipates 1 mW of heat per qubit at the temperature T_{gen}. Current best estimates are closer to 15–30 mW per qubit [46,94,95] but these numbers are dropping as research progresses. Taking this optimistic value compared to current CMOS also reinforces the observation that we make in Sec. V E 2 [96].

Scenarios B and C, respectively, correspond to improvements by 2 and 4 orders of magnitude compared with scenario A in terms of heat dissipation per qubit. Scenario C can be taken as a futuristic projection for classical logic performance based on superconducting circuits known as single-flux quantum (SFQ) [43], which may potentially generate about 10 000 times less heat than CMOS. However, our results should mainly be taken as an indication of the importance of research in this direction.

We conclude this summary of our full-stack model by noting that our simulations use generic numbers and orders of magnitude. Our results should thus not be considered as precise estimates for a specific technology or platform. Instead, they enable us to observe general trends and thereby provide understanding that can guide future experiments.

E. Minimization of power consumption

We now use our model to minimize the macroscopic power consumption P_{FT}, under the constraint of a fixed target metric, $\mathcal{M}_{FT} = 2/3$, following the MNR methodology presented in Sec. V A. Our results for P_{FT} do not vary significantly for slight variations of the target metric from the specified 2/3 value [97]. This target constrains the control parameters, since the metric depends on p_{err}, which in turn depends on the control parameters [see Eqs. (18) and (B1)]. Under this constraint, we optimize the power consumption with respect to four control parameters: the temperature T_{gen} of the signal generation (top stage in Fig. 9), the temperature T_{qb} of the qubits, the total attenuation A between T_{gen} and T_{qb}, and the level of concatenation k for the fault-tolerant scheme. Figure 10(b) presents a two-dimensional map of our optimizations, i.e., the minimal power P_{FT} consumed by our generic circuit of size (Q_L, D_L). P_{FT} increases with the number of logical qubits Q_L and depth D_L, with the discontinuities corresponding to the change of concatenation level k. We have considered scenario A, with high-quality qubits characterized by $1/\gamma = 50$ ms, corresponding to $p_{err} \sim p_{thr}/40$ (i.e., $p_{err} = p_{thr}/40$ when T_{qb} is small enough and A large enough so that $p_{err} = \frac{1}{4}\gamma\tau_{step}$).

As mentioned at the beginning of this section, our chosen circuit provides a good approximation of the power consumption of any circuit involving the same values of Q_L and D_L. We use this property to estimate the minimum power required to implement the set of quantum gates given in Ref. [26] for the Shor's algorithm that breaks the RSA encryption of an n-bit key. Figure 10(a)(i) shows P_{FT} for as a function of the qubit quality γ^{-1}. The different curves are for different values of power dissipation for the electronics. Figures 10(a)(ii) and 10(a)(iii) show the values of temperatures and attenuation that give the minimal power consumption. Finally, Fig. 10(c) shows the heat evacuated (and the corresponding power consumption) at each temperature stage in the cryogenics. Our results allow us to make a number of observations that are likely to hold for a range of fault-tolerant quantum computing schemes, including those based on surface codes. These observations are detailed below and deal with

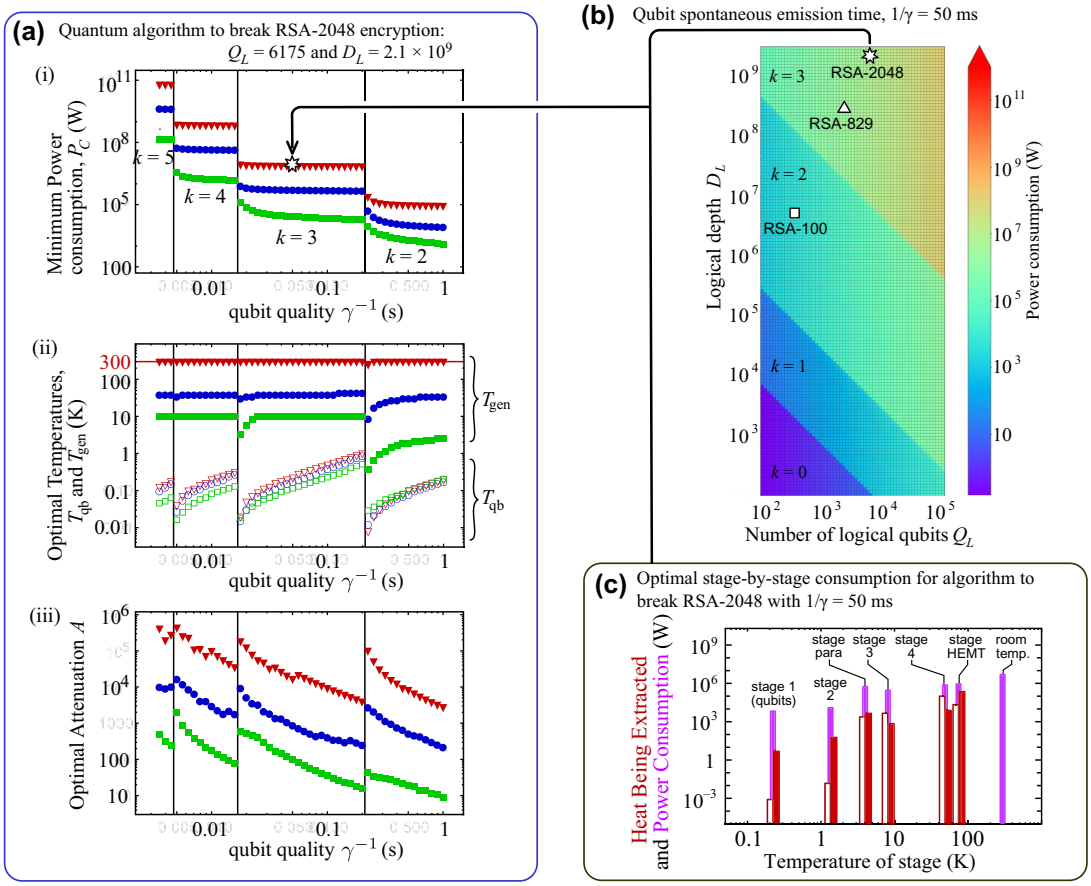


FIG. 10. (a) The upper plot is the minimum power consumption, P_{FT}, as a function of the qubit quality (qubit lifetime $1/\gamma$) under the constraint that the computation successfully cracks the RSA-2048 encryption, a calculation that requires $Q_L = 6175$ logical qubits, and a logical depth of $D_L = 2.1 \times 10^9$. The different symbols are for the different types of control electronics at T_{gen}. The red triangles, blue circles, and green squares, respectively, correspond to our scenarios A, B, and C (see text). The lower two plots show the optimal temperatures and attenuations for which this minimum P_{FT} is achieved. Transitions between values of k are first-order-like and we observe that their positions are fairly well estimated by a simple formula (see Appendix C 3). (b) The minimum power consumption, P_{FT}, as a function of the size of the calculation being carried out, for scenario A with $1/\gamma = 50$ *ms*, which corresponds to $p_{err} \simeq p_{thr}/40$ (see text). The star indicates values corresponding to an algorithm that breaks the RSA-2048 encryption, with a total power consumption of 7 MW for 4.6×10^9 physical qubits. (c) For every set of parameters, we have the heat extracted at each refrigeration stage after optimization. We show this here for the parameters corresponding to the point marked by the star in (a) and (b). The red and white bars are the heat extracted at each refrigeration stage, with white corresponding to the heat conduction down the cables and red being the heat generated at each stage by attenuators or amplifiers. Each purple bar is the power consumption to extract this heat (assuming Carnot-efficient refrigeration). The purple bar at $T_{gen} = 300$ K (room temperature) is the power consumption of the control electronics.

the respective impacts of the qubit fidelity, the control electronics, the cryogenics, and the logical depth.

1. Impact of qubit fidelity

If the error probability is only slightly below the fault-tolerance threshold, we observe that the power consumption is unreasonably large. However, the power consumption drops very rapidly as the quality of the qubits increases. In the present model, increasing qubit quality means having the physical qubits couple more weakly to the microwave control line and hence having more microwave power to drive the qubits [see Eq. (1) for how the power to flip a qubit from $|0\rangle$ to $|1\rangle$ is proportional to the qubit quality, γ^{-1}]. Despite this, the gains from reducing the error rate (and hence reducing the necessary amount of error correction) greatly outweigh the costs of increasing the microwave power per physical qubit. Figure 10 shows that a factor-of-10 increase in the qubit quality (i.e., dividing γ by 10) leads to a factor-of-100 reduction in the overall power consumption for a given computational accuracy. We believe that a large reduction in power consumption from improved qubit quality is likely to be a general trend in all parameter regimes

and, indeed, in all qubit technologies, placing a significant emphasis on developing qubits of the highest possible quality.

It is worth noting that additional sources of noise (beyond the unavoidable noise in the lines that control the qubits) will always add to the resource cost. They will always increase the power consumption, as we are required to cool the qubits further, or provide additional error correction to achieve a given performance metric. Errors due to long-range crosstalk between qubits are particularly dangerous, since error correction can be of limited use against them [66].

2. Impact of control electronics

Once the cryogenics are optimized, we observe that the control electronics are a dominant contribution for scenario A. This is clear from Fig. 10(c), where we plot the heat to be extracted per stage in the cryogenics and the corresponding power consumption per stage. The absolute magnitude of the heat and power varies dramatically with the quality of the gates and with Q_L and D_L but we observe that the ratios between different stages do not vary very much. In all cases, we find that the total power consumption per physical qubit (given by P_{FT}/Q) is 1.3–2 mW [it is 1.5 mW for the star in Fig. 10(a)]. Relatively little of that comes from the cryogenics below 4 K; the dominant part (approximately 1 mW) comes from the control electronics at T_{gen}.

For this reason, our optimization in scenario A puts the control electronics at room temperature (i.e., the optimal T_{gen} is ambient temperature), with the consequence that there are many millions of room-temperature cables (a few cables for every 25 physical qubits, in our model of multiplexing) going down into the cryostat. Placing the electronics at 4 K will reduce the heat conduction into the cryostat, as there will then be almost no cables between 300 K and 4 K. However, the heat generated by these control electronics is vastly more than that brought in by the wires and the resulting increased demand for cooling will increase the power consumption by a factor of about 75 (see Fig. 12 in Appendix A). It is only when the dissipation of the control electronics is orders of magnitude lower (such as 10 μW in scenario B) that we observe a significant energetic advantage in placing these electronics at lower temperatures, as shown in the middle plot in Fig. 10(a).

We recall that we have assumed Carnot-efficient cryogenics. If the cryogenics are only at 10–30% of Carnot efficiency [see Sec. B 10], then the optimal T_{gen} will remain at room temperature in scenario A (or for any CMOS technology consuming more power than scenario A). The power consumption of the cryogenics will be higher (e.g., 10 times larger for cryogenics at 10% Carnot efficiency) but the power consumption of the control electronics will remain a major cost. Hence, research to minimize this cost is crucial.

At the same time, we do not yet have a technology that can reliably install many millions of wires (with attenuators) between the room-temperature stage and the qubits. It may thus be necessary to put the control electronics at low temperatures, despite the cryogenic cooling costs. This makes it crucial to pursue ongoing research to improve the efficiency of cryo-CMOS, hand in hand with designing cryogenics that can efficiently evacuate large amounts of heat at the temperature chosen for the cryo-CMOS control electronics.

3. Impact of cryogenics

When we assume that the cryogenics are close to achieving Carnot efficiency, we observe that the cryo-power comes mainly from evacuating heat generated at temperatures above 4 K. An example of this is shown in Fig. 10(c). This means that its power consumption is almost independent of the qubit temperature, which is always significantly less than 4 K. More precisely, the total power consumption has a large T_{qb}-independent contribution, with the contribution for T_{qb} (which diverges at $T_{\mathrm{qb}} \to 0$) dominating only for very small T_{qb}. This causes the abrupt change in P_{FT} as $k \to (k-1)$ visible in Fig. 10, although, if one were to magnify the curves sufficiently, one would see that the curves are continuous with a discontinuity in their derivative at the transition from k to $k-1$ (see Appendix C 3).

Notably, this observation comes with a caveat: it relies on the cryogenics being reasonably close to Carnot efficiency at low temperatures. If this is not so, one can find cases where the power consumption depends largely on T_{qb}. For example, many small-scale laboratory cryogenic systems have a heat extraction at ultralow temperature far from the Carnot efficiency and some experimental qubits have significant additional sources of heat at T_{qb}. In such situations, the overall power consumption may be dominated by the evacuation of heat at T_{qb}. Appendix E gives an example of this in which the power consumption per physical qubit can vary by 3 orders of magnitude as T_{qb} changes. The minimal power consumption for a given performance metric then depends more strongly on the qubit quality and, in a more subtle manner, on many other hardware and software parameters. Without the systematic optimization proposed here, one simply cannot know the optimal values of all the control parameters. Appendix E has examples where a poor choice of parameters can induce a power consumption of gigawatts, compared with megawatts when the optimal parameters are used. In contrast to conventional wisdom (quantified in Appendix E), the power consumption is sometimes reduced by a strategy of raising the qubit temperature (and hence increasing the errors per physical qubit) and compensating for it by

having more error correction. Unexpectedly, whether or not this strategy is optimal depends on parameters such as the power consumption of the control electronics.

The above caveat means that, for any given hardware, one has to carefully optimize the full-stack model to know whether the power consumption is dominated by effects above 4 K or by effects at T_{qb}. Our analysis suggests that the most promising cases fall into the former category (dominated by effects above 4 K) and so this observation will apply to them. Nevertheless, this has to be checked on a case-by-case basis.

4. Impact of logical depth

We observe that the power consumption increases with the algorithmic depth. By reasoning in terms of power per qubit, it is easy to guess that the power consumption at each time step in the calculation goes up with the number of qubits. However, the power consumption at each time step in the calculation also grows with the depth of the algorithm—it grows with the number of logical time steps in the calculation, D_L. This is because a longer computation requires a lower error probability per gate operation, in order to keep the quantum information error free until the end of the computation. A longer calculation thus requires more noise mitigation by, e.g., lowering the qubit temperature or performing more error correction. This means more power consumption at every time step in the calculation. This behavior is quite different from deterministic computation in classical computing, where the power consumption of the computer depends only on the number of operations performed in parallel and usually not on the duration of the computation.

This behavior is, however, not unique to quantum computers. An example of an analogous situation in classical computing is floating-point calculus for the simulation of chaotic systems (e.g., a three-body problem [98]). To achieve a given precision for a simulation of the chaotic system for D time steps, one is required to take a floating-point precision at each time step that grows with D, requiring more power consumption at each time step. However, while power consumption increasing with the algorithmic depth, D, occurs in certain specific cases in classical computing, it is unavoidable in quantum computing.

F. Quantum energy advantage

Finally, we make use of our model to explore the potential of fault-tolerant quantum computers to achieve a quantum energy advantage. Usually, the concept of quantum advantage refers to a comparison between the computing powers of classical and quantum processors, with a quantum advantage being present if the quantum processor solves a problem in less time or space than the best-in-class classical processor. Here, we bring the discussion to the energetic level and ask when a fault-tolerant quantum computer can solve problems using less energy than a classical supercomputer, a feature we dub the quantum energy advantage.

Intuitively, one expects such quantum energy advantage whenever the quantum computer solves a problem, which is intractable on a classical computer, in a reasonable time. Even if the quantum computer requires more power than the classical computer, the considerably shorter time taken for the quantum computer will give a lower energy cost. But the quantum energy advantage involves much more surprising regimes. In particular, it can appear for problems that are solvable on a classical computer and even in cases where the quantum computation takes more time than the classical one. We see examples of these various cases below.

The manner in which a quantum computer solves a problem is so different from a classical computer that there is little sense in comparing the power consumption per gate operation or per floating-point operation; the energy-efficiency measure of FLOPS/W used in classical computing is not applicable to quantum computers. Instead, one must compare the power consumption of quantum and classical computers performing the same useful task. As an example of such a useful task, we consider the cracking of an n-bit RSA key discussed earlier. There are well-documented quantum [26] and classical [99] algorithms for this task, allowing us to perform a fair energetic comparison. This time, we choose a user-oriented metric to quantify the performance of the quantum processor, namely, the size of the key n that can be broken with a success probability of $2/3$. In a similar spirit as the quantity defined in Sec. IV, a convenient user-oriented energy efficiency is $H_{FT} = n/E_{\mathrm{FT}}(n)$, where $E_{\mathrm{FT}}(n) = P_{\mathrm{FT}}(n) \times t_{\mathrm{FT}}(n)$ is the energy consumed by the fault-tolerant quantum computation and $t_{\mathrm{FT}}(n)$ is the duration of the computation. Such an efficiency corresponds to the inverse of the energy cost per bit of the key and has the dimensions of a bit/J.

We first consider a situation of quantum computational advantage for a calculation that can be completed on a classical computer in a week or two. This is the case for the RSA-830 encryption (i.e., public-key encryption with a 830-bit key), which has been cracked on a classical supercomputer [99], using the equivalent of 2700 core years on Intel Xeon Gold 6130 CPUs. These consume about 12 W per core, so cracking the encryption classically requires about a terajoule of energy. This yields a typical efficiency of 8×10^{-10} bit/J. If it were done using all the cores on a top-100 supercomputer, such as the JUWELS Module 1 [100], then we estimate that its power consumption would be about 1.3 MW (including cooling) and it would crack the encryption in 8–9 days (see Appendix D).

In contrast, a quantum computer with high-quality qubits ($1/\gamma = 50$ ms) and control electronics corresponding to scenario A, corresponding to the triangle in Fig. 10(b), can complete the calculation in about 16 min,

with a power consumption of about 2.9 MW after optimization. We again recall that we are optimistic on hardware parameters compared to the state of the art and that we have made some simplifying assumptions (see Appendixes A 2 and C 2) but we also base our optimizations on an error-correcting code that is highly demanding in resource. This corresponds to a total energy consumption of 2.7 GJ and an efficiency of 3×10^{-7} bit/J. This is more than 2 orders of magnitude less than the classical supercomputer, clearly pointing to a quantum energy advantage.

For RSA-2048 encryption, considered uncrackable by classical supercomputers (even with an arbitrarily large power supply), Fig. 10(b) gives a quantum computer power consumption of about 7 MW after our optimization. If this could be achieved, it would be similar to the power consumption of some of the largest existing supercomputer clusters. The quantum computer would solve this RSA-2048 problem in 1.5 h, for a total energy cost of about 38 GJ, equivalent to the energy in about 20 car tanks of gasoline [101] and an efficiency of 5×10^{-8} bit/J.

In these two examples, the smaller energy cost of the quantum calculation arises because the quantum computation is much faster, while both the classical supercomputer and the quantum computer require megawatts of power. However, this is not the only route to a quantum energy advantage. To investigate this point more thoroughly, we have computed the time $t_{\mathrm{FT}}(n)$ needed for the quantum computer to break keys of increasing size n and the corresponding energy efficiency $H_{\mathrm{FT}}(n)$. The comparison with classical computers allows us to define the respective regimes of computational and energy advantages. Both quantities, together with the estimations for their classical counterparts, are plotted in Fig. 11 for various qubit and control-electronics technologies. As it appears on the figure, the quantum computational advantage solely depends on the qubit fidelity and remains insensitive to the dissipation by the control electronics—the better the qubits, the sooner the advantage is gained. Conversely, the quantum energy advantage is sensitive to the qubit fidelity and other parameters such as the efficiency of the control electronics. Hence, both advantages are reached for sufficiently large keys, at the price of huge experimental and technological challenges.

Interestingly, our study singles out cases where the quantum computer consumes less energy than a classical computer, even when the quantum calculation takes longer than the classical one. An example of this can be seen on the green curve in Fig. 11, obtained for high-quality qubits with extremely efficient control electronics (scenario C). Figure 11 illustrates that the computational advantage in the upper plot is of a different nature than the energy advantage in the lower plot, with the former being related to time and the latter to energy. Hence, it should not come as a surprise that they occur in different parameter regimes.

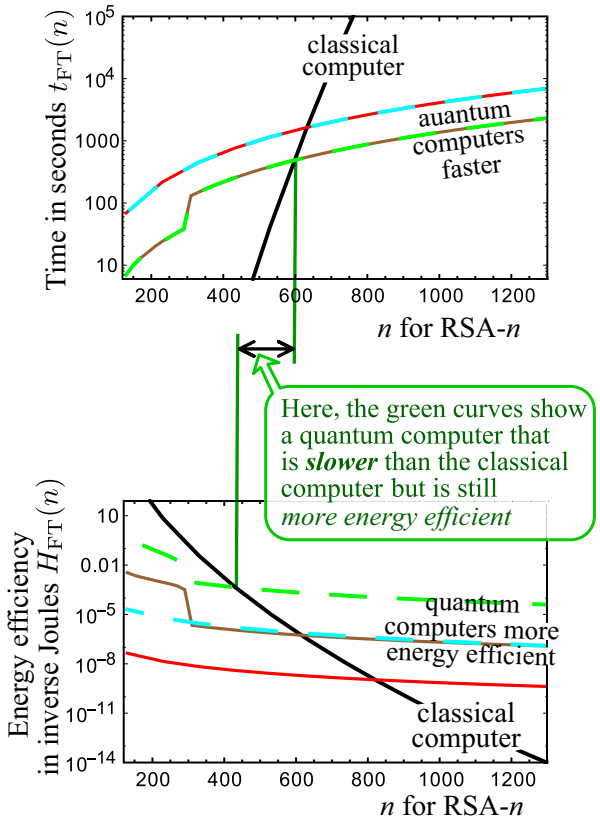


FIG. 11. A comparison between classical and quantum computers performing the calculation to crack RSA-n encryption as a function of the key size n. The plots show (a) the time t_{FT} taken to perform the calculation and (b) the energy efficiency of the calculation. The time of the quantum computation is the number of time steps in the algorithm multiplied by the time per step, $\tau_{\mathrm{step}} = 100$ ns. The efficiency is defined as $H_{\mathrm{FT}}(n) = E_{\mathrm{FT}}(n)$, where $E_{\mathrm{FT}}(n) = P_{\mathrm{FT}} \times t_{\mathrm{FT}}(n)$ for the quantum computer. We plot this efficiency for the value of P_{FT} found from our minimization. The black curve is a classical supercomputer based on Ref. [99] (see Appendix D). The colored curves are for quantum computers with various parameters: red for qubits with $1/\gamma = 5$ ms and the control electronics in scenario A, brown for $1/\gamma = 50$ ms and scenario A, cyan for $1/\gamma = 5$ ms and scenario C, and green for $1/\gamma = 50$ ms and scenario C. In all three cases, the quantum computers become faster and more efficient than the classical computer as n grows. The green curves show a quantum computer that becomes more energy efficient than the classical computer *before* it becomes faster than the classical computer.

These observations suggest that an important future motivation for using quantum computers could be that they can solve problems in a more energetically efficient manner than classical computers, even when those problems are solvable on classical computers in an acceptable time (or even solvable faster on a classical computer). We believe that this may become a crucial source of applications for quantum computing, especially because an energy advantage without a computational one might ask for less

demanding qubit quality and a quantum computer of much more moderate size. The search for quantum energy advantage should thus be an active goal of research, alongside the search for quantum computational advantage.

G. The case of surface code

It is likely that we can significantly improve the efficiency of quantum computers by taking more resource-efficient error-correcting codes than the concatenated seven-qubit code discussed in the previous sections. An example could be surface code, the current choice for fault-tolerant quantum computing being pursued in many experiments. Unlike the concatenated seven-qubit code, the literature lacks certain information about surface codes, which limits our capacity to make concrete claims. In particular, the error diagnosis from the measured syndromes is trivial for concatenated codes (see Appendix B 5), while surface codes require complicated algorithms run on conventional computers to "decode" the syndrome information to deduce what errors may have occurred and to do this fast enough to correct them. These decoding algorithms remain a subject of investigation today and may carry a significant energy cost. We currently know the general scaling properties of some of the most well-known decoders (e.g., minimum-weight perfect matching [102–106], and union-find [107,108] decoders. However, we do not have access to the detailed prefactors necessary for our resource estimates for a realistic classical computer that is powerful enough to decode the errors fast enough to correct them.

Nevertheless, we can give estimates for some aspects of the full-stack fault-tolerant quantum computer with surface codes. We start from the proposal of Ref. [26] to crack the RSA-2048 encryption in about 8 hours, using 20 million superconducting qubits, with the error probability per gate set to be 0.1%. For scenario A, a suitably optimized concatenated code has a total power consumption of 1–2 mW per physical qubit (including cryogenics, control and readout electronics, etc.), whatever the qubit quality and whichever algorithm is being performed (see Eqs. (16)–(19)). We recall our caveat that scenario A calls for parameters beyond the current state of the art. Assuming that similar numbers apply in the surface-code situation, the 20 million qubits needed to break RSA-2048 will then only require a power of 20–40 kW [109]. Then, this part of the quantum computer will be truly green: it will take about 8 hours to do something no classical supercomputer could ever do, while using about the same amount of power as an electric car driving on an interstate highway. Of course, this estimate excludes the power needed for running the decoding algorithm, which, as mentioned above, may be significant. However, as decoding algorithms get better, there is a reasonable hope that the quantum computer will require no more power than a classical supercomputer to do useful calculations that no supercomputer can ever do.

These considerations make it likely that such a quantum computer would have a much better quantum energy advantage than that discussed in Sec. V F, including for tasks that could be done on a classical supercomputer in a reasonable time.

VI. DISCUSSION AND OUTLOOK

This work presents and applies a new methodology, dubbed the metric noise resource (MNR), that provides a holistic and quantitative model of the full stack of a quantum computer. It identifies and quantifies the links between the quantum and macroscopic levels of such a quantum computer. This provides the theoretical foundation for the minimization of the resource costs of quantum technologies. We use it to arrive at a quantitative relation between the computing performance of a quantum processor and its macroscopic resource cost and thus minimize the latter under a given performance constraint. The MNR methodology provides a common language to connect fundamental research and enabling technologies, fostering new synergies such as those recently called for in the Quantum Energy Initiative [63]. It allows us to define and optimize various quantum computing efficiencies, i.e., new versatile resource-based figures of merit that we expect to serve as a tool to benchmark a large range of hardware and software features, including qubit technologies, processor architectures, quantum error-correcting codes, etc.

To illustrate our methodology, we have applied it to the full-stack model of a superconducting quantum computer. There, we have established the first estimates of energy costs in relation to the quality of the qubits, the type of quantum error-correcting codes employed for fault-tolerant computing, as well as the architecture and energetic performance of control electronics and cryogeny. We have considered the case of RSA key breaking, a task with well-defined classical and quantum algorithms, and the complexity of which is indicative of useful applications in physics research, chemical simulations, and various optimization tasks.

Our model is based on highly optimistic assumptions for the qubit lifetimes and the noise model (as needed for the seven-qubit code) and somewhat optimistic characteristics for other elements in the model. We have sought to model all the main sources of energy consumption, while maintaining a reasonable simplicity in our model. Our results should hence not be interpreted as design blueprints for an energy-efficient quantum computer. Instead, they are a guide to what must be understood and improved in the physics and engineering of such a quantum computer.

We have provided examples where the minimization of resources costs can reduce power consumption from gigawatts to megawatts. While this may arguably be an

extreme case, our results suggest that such resource optimizations are likely to be critical in the success of the current state of the art and also in futuristic scenarios. Existing prototypes of quantum computers (for which energy optimization was not a priority) often consume hundreds of watts per physical qubit [19]; our proposed optimization suggests that future generations of quantum computers could consume only a few milliwatts per physical qubit. In our main example, this includes the power cost of classical computing to decode the error-correction syndrome (this cost is negligible there). In some other cases, this syndrome-decoding cost is not yet known (see Sec. V G), so it may be in addition to our prediction of a few milliwatts per physical qubit.

In some of our examples, our minimization has led to optimal parameters that could not have arisen from simpler estimates and that are contrary to conventional wisdom. For example, it is sometimes energetically favorable to put the qubits at higher temperature and compensate with more error correction, an unexpected strategy given how many more physical components are needed for the extra error correction. Only a systematic full-stack analysis using our MNR methodology can reliably determine these optimal parameters for a given quantum computing platform.

In addition, we have observed evidence of a potential quantum energy advantage for RSA key breaking. While our optimization there has involved somewhat futuristic scenarios with qubit quality and control electronics beyond the existing state of the art, it nevertheless suggests that quantum processors can consume less energy than classical ones even in the regime where the RSA keys are small enough for classical computers to break in a reasonable time. This is an extremely encouraging result, particularly as other error-correcting codes (such as surface code) could demand fewer resources than the simple one used in our model, while also working with much noisier qubits.

We have also shown that a quantum energy advantage and a quantum computational advantage are different concepts that can correspond to different parameter regimes. Energy savings provided by the quantum logic thus emerge as a crucial practical interest of quantum computing, distinct from computational advantage. This is a particularly interesting and open problem for the NISQ regime. From this perspective, it should be investigated whether the current generation (or the next generation) of noisy quantum computers could be better than classical computers for useful algorithms, not necessarily by showing a computational advantage but by being more energy efficient instead.

The MNR methodology presented in this work will help in the design of scalable quantum processors using various qubit technologies and different energetic regimes (cat qubits, photons, electron spins, atoms, etc.). It can provide a consistent set of design specifications that keeps resource costs reasonable and helps to identify potential technological bottlenecks to achieving these specifications. It can help define road maps for the various research and industry teams working on these technologies, as it relies heavily on the development of efficient and integrated control electronics, with paramps, signal-multiplexing techniques, software engineering techniques, and the like. The MNR methodology will be useful in prioritizing the coordinated development of all components in the full-stack quantum computer. Developing this mindset will be vital in ensuring that quantum computing avoids dead ends that await environmentally unsustainable technologies. Beyond the case of quantum computing, we also suggest the application of the MNR approach to optimization of the resource cost of other quantum technologies, such as quantum communications and quantum sensing.

ACKNOWLEDGMENTS

This work benefits from the support of the European Union (EU) Horizon 2020 research and innovation program under the collaborative project "Quantum Large Scale Integration in Silicon" (QLSI) (Grant No. 951852), the French National Research Agency (ANR) Research Collaborative Project "QuRes" (Grant No. ANR-PRC-CES47-0019), the Merlion Project (Grant No. 7.06.17), the ANR program "Investissements d'avenir" (Grant No. ANR-15-IDEX-02), the Committee of Laboratories of Excellence (Labex) Laboratoire d'Alliances Nanosciences-Energies du futur (LANEF), and the "Quantum Optical Technologies" project within the International Research Agendas program of the Foundation for Polish Science cofinanced by the EU European Regional Development Fund. We warmly thank O. Ezratty for his constant feedback and support, as well as O. Buisson, C. Gidney, O. Guia, B. Huard, T. Meunier, V. Milchakov, L. Planat, M. Urdampilleta, and P. Zimmermann for useful discussions that greatly contributed to this work. H.K.N. acknowledges the support of a Centre for Quantum Technologies (CQT) Fellowship. CQT is a Research Centre of Excellence funded by the Singapore Ministry of Education, and the National Research Foundation of Singapore.

APPENDIX A: COUNTING QUBITS AND GATES FOR *k* LEVELS OF CONCATENATED ERROR CORRECTION

1. Fault-tolerant Clifford gates

Figure 8 is the complete circuit diagram for any Clifford gate with one level of seven-qubit code for error correction, including the ancilla factory [83,87–90]. It shows that one level of error correction (i.e., one concatenation level) replaces one qubit by seven data qubits and uses 28 ancilla qubits to detect errors [89,90]. Gates acting on groups of seven qubits are *transversal*, which means the relevant gate is applied to each of the seven qubits individually. For example, a transversal controlled-NOT (CNOT) between one

group of seven qubits and another group of seven qubits involves a CNOT between qubit i in the first group and qubit i in the second group for all i from 1 to 7 in parallel.

Figure 8 shows that the preparation and use of the ancillas takes a significantly longer time than data-qubit operations. The full evolution of the ancillas (including their preparation, verification, interaction with data qubits, and final measurement) takes nine time steps, while the data-qubit operations take only three time steps. Hence, while 28 ancillas are being used in the current gate, an additional 2×28 ancillas undergo the preparation and verification steps, to be ready in time for the next two gates. Thus, each additional level of concatenation in the error correction involves replacing one qubit by seven data qubits and 3×28 ancillas, giving a total of 91 qubits.

The logical ancillas must be verified to be sufficiently error free for use, before they interact with the data qubits, and this is done by the verification part of the circuit in Fig. 8. Each logical ancilla has a small chance of failing the verification, so the code must always prepare and verify a small percentage of extra logical ancillas at each clock cycle (not shown in Fig. 8), to immediately replace any ancillas that fail verification. We will show elsewhere [110] that this increases the resource consumption associated with ancillas by less than 2%. This is small enough to neglect here and so we simplify the analysis by assuming that all ancillas pass verification.

The concatenated nature of the seven-qubit code scheme means that this counting is repeated k times for k levels of error correction. For k levels of error correction, the number of physical qubits is

$$Q = (91)^k \, Q_L \tag{A1}$$

for Q_L logical qubits. This number is independent of the type of logical Clifford gates that are being implemented on the logical qubits.

A counting of gates in Fig. 8 gives the numbers of physical gates per logical gate in Table I. Then, with k concatenation levels, the number of physical gates done in parallel is related to the number of logical gates done in parallel, according to

$$\begin{pmatrix} N_{2\mathrm{qb}} \\ N_{1\mathrm{qb}} \\ N_{\mathrm{Id}} \\ N_{\mathrm{meas}} \end{pmatrix} = A^k \begin{pmatrix} N_{2\mathrm{qb};L} \\ N_{1\mathrm{qb};L} \\ N_{\mathrm{Id};L} \\ N_{\mathrm{meas};L} \end{pmatrix}, \tag{A2a}$$

with

$$A = \frac{1}{3} \begin{pmatrix} 135 & 64 & 64 & 0 \\ 56 & 35 & 28 & 0 \\ 58 & 29 & 36 & 0 \\ 56 & 28 & 28 & 7 \end{pmatrix}, \tag{A2b}$$

where the elements of A come from transposing Table I. The prefactor of $1/3$ in A is because it takes three time steps to perform each logical gate. Thus, the number of physical gates acting in parallel (averaged over those three time steps) is $1/3$ the number of physical gates in a single level of concatenation. This $1/3$ appears at each level of concatenation, giving the prefactor in A.

TABLE I. The number of physical gates in a given logical Clifford gate for one concatenation level of the seven-qubit code, as shown in Fig. 8. Here, "1qb" refers to any single-qubit gate in the set $\{X, Y, Z, H, S\}$ and "2qb" refers to the two-qubit CNOT gate. The numbers include the gates required to prepare and verify the logical ancillas.

	Number of physical gates for given logical gate			
Logical gate	2qb gates	1qb gates	id gates	Measure
Logical 2qb	135	56	58	56
Logical 1qb	64	35	29	28
Logical id	64	28	36	28
Logical measure	0	0	0	7

The matrix A has two eigenvalues, $192/3 = 64$ and $7/3$. The larger one dominates Eq. (A2). For example, Eq. (A2) gives the number of physical single-qubit gates as

$$N_{1\mathrm{qb}} = \frac{28(64)^k}{185} \left(2N_{2\mathrm{qb};L} + N_{1\mathrm{qb};L} + N_{\mathrm{id};L}\right) - \frac{(7/3)^k}{185} \left(56N_{2\mathrm{qb};L} - 157N_{1\mathrm{qb};L} + 28N_{\mathrm{id};L}\right). \tag{A3}$$

This is well approximated by its first term, which comes solely from the eigenvalue 64. The approximation is best for large k but it works reasonably for order-of-magnitude calculations of $N_{1\mathrm{qb}}$, $N_{2\mathrm{qb}}$, N_{id}, and N_{meas} for all nonzero k; the differences between approximate and exact results are less than 25% for $k = 1$ and less than 1% for $k \geq 2$. This means that the number of physical components in parallel is well approximated by a function of $(2N_{2\mathrm{qb};L} + N_{1\mathrm{qb};L} + N_{\mathrm{id};L})$, independent of the individual values of $N_{2\mathrm{qb};L}$, $N_{1\mathrm{qb};L}$ and $N_{\mathrm{id};L}$. This is nice because, for any time step where no logical measurements are occurring, $(2N_{2\mathrm{qb};L} + N_{1\mathrm{qb};L} + N_{\mathrm{id};L})$ is equal to the number of logical qubits storing information at that time step, irrespective of what (if any) gates are being performed on those logical qubits.

This is particularly simple in the context of quantum algorithms that can be approximated by *rectangular* circuits. A rectangular circuit is one that uses the same number of qubits in each time step and does the logical measurements on these qubits at the end. This is a reasonable approximation of the algorithm to break the RSA encryption in Ref. [26] (see Sec C 2). In such a case, the number of logical qubits storing information in any given time step is equal to the total number of logical qubits available in the computer, Q_L. We can then write

$$2N_{2\mathrm{qb};L} + N_{1\mathrm{qb};L} + N_{\mathrm{id};L} = Q_L, \tag{A4}$$

and the number of physical elements in parallel in any time step is

$$N_{2\mathrm{qb}} \simeq \frac{64}{185}(64)^k Q_L, \quad N_{1\mathrm{qb}} \simeq \frac{28}{185}(64)^k Q_L,$$
$$N_{\mathrm{id}} \simeq \frac{29}{185}(64)^k Q_L, \quad N_{\mathrm{meas}} \simeq \frac{28}{185}(64)^k Q_L, \tag{A5}$$

where $\simeq$ indicates the approximations and assumptions made in the previous paragraphs.

This gives us a useful intuitive rule for a quantum computer with error correction: the number of physical gates in parallel after $k \geq 1$ levels of concatenations is about the same for any algorithm, so long as the algorithm is using nearly all the logical qubits in the quantum computer in nearly all time steps, with measurements of logical qubits remaining rare during the calculation [111]. The number of each type of physical gates is then approximately given by Eq. (A5).

2. Fault-tolerant *T* gates

The gate set in Table I allows us to perform any Clifford operation. However, to perform an arbitrary quantum calculation (i.e., to perform an arbitrary unitary operation on the qubits), we also require at least one non-Clifford gate in the gate set. Without a non-Clifford gate in the gate set, the calculations that we can perform on the quantum computer are limited to ones that can be efficiently simulated with classical computers.

We take the common choice to add a non-Clifford gate called the T gate to the gate set. The T gate is a single-qubit rotation of $\pi/4$ around the z axis of the Bloch sphere, i.e., $T \equiv \exp(-i\pi/8\sigma_z)$. Then, any non-Clifford operation (such as a Toffoli gate) is included in the circuit as a suitable combination of T gates and Clifford gates. The fault-tolerant implementation of the T gate in the seven-qubit code scheme is done in a very different way from that of Clifford gates. It is done by making the logical qubit interact with ancillas in a so-called *magic state*. These magic states have to be fault-tolerantly prepared beforehand in the manner shown in Fig. 13 of Ref. [90]); this circuit is more complicated than for Clifford gates. Each magic state preparation also has a chance of failure and must be verified before being inserted into the circuit. This means one must also prepare extra magic states to compensate for those that are discarded when they fail verification. These considerations imply that a T gate will be more costly than a Clifford gate.

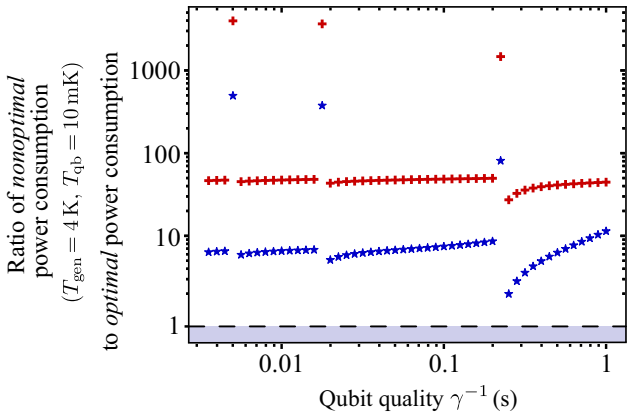


FIG. 12. The power saved by optimization, compared to forcing the signal-generation temperature $T_{\mathrm{gen}} = 4$ K and qubit temperature $T_{\mathrm{qb}} = 10$ mK and only optimizing the attenuation, A, and the error-correction concatenation level, k. All parameters are as in Fig. 10. The different symbols are for different heat generation at the signal-generation stage; the red crosses are for the current state of the art (scenario A in Table III) and the blue stars are 100 times better (scenario B in Table III). The discontinuities are when the fully optimized full stack has a lower optimal k than that given by the optimization in which we force $T_{\mathrm{gen}} = 4$ K and $T_{\mathrm{qb}} = 10$ mK.

However, T gates are often rare in circuits of interest. The algorithm we consider here (see Appendix C 2), and other algorithms in the literature [4,26,112–114], have less than about one logical T gate for every 70 logical Clifford gates (including identity gates in the count of Clifford gates). It is reasonable to guess that a T gate would not require 70 times more resources than a Clifford gate, so T gates could be neglected when calculating power consumption, making our results in Sec. V applicable to the type of algorithms that we consider here.

However, to confirm that this guess is correct, a detailed calculation was necessary, to be presented in Ref. [110]. There, we compute the number of physical qubits and gates required by logical T gates compared to the number required by logical Clifford gates. As the number of

TABLE II. A summary of the variables we optimize in our full-stack analysis of large-scale fault-tolerant quantum computing.

Variables in optimization	Symbol	Value
Power consumption	P_C	To be minimized under constraint of given metric, $\mathcal{M}$
Qubit temperature	T_{qb}	To be found from minimization of P_C
Signal-generation temperature	T_{gen}	To be found from minimization of P_C
Attenuation between T_{gen} and T_{qb}	A	To be found from minimization of P_C
Error-correction concatenation level	k	Integer to be found from minimization of P_C

TABLE III. A summary of the parameters used in our full-stack analysis of large-scale fault-tolerant quantum computing. These are the parameters used in Figs. 10, 11, and 12. Figure 14 also uses most of these parameters but it takes a lower cryogenic efficiency and adds a heat source at T_{qb} (see Sec. E). In all plots, we take the same scenario (A, B, or C) for $\dot{q}_{gen}$, $\dot{q}_{para}$, and $\dot{q}_{HEMT}$; we do not mix scenarios.

Parameters	Symbol	Value
Typical qubit frequency	$\omega_0/(2\pi)$	6 GHz (similar to Google Sycamore [15,62,72])
1qb gate time	τ_{1qb}	25 ns (similar to Google Sycamore [15,62,72])
2qb gate time	τ_{2qb}	100 ns (similar to cross-resonance scheme involving interaction via a bus [85,86], which allows implementation of the long-range 2qb gates necessary for concatenated error correction)
Measurement time	τ_{meas}	100 ns (similar to scheme in Ref. [115])
Time step of quantum computer	τ_{step}	100 ns (time of slowest gate)
Type of error correction	⋯	Seven-qubit error-correction code (concatenated error correction) [83,87–90]
Threshold for error correction	p_{thr}	2×10^{-5}
Error time scale ≡ spontaneous emission rate into microwave line ∼ decoherence time at $T_{qb} = 0, A \to \infty$	γ^{-1}	From 3 ms and 1 s [see Fig. 10(a)]—the lowest γ^{-1} here is *a few orders of magnitude larger than Google Sycamore* [15,62,72]; however, smaller γ^{-1} is not possible here, because it would put us above p_{thr}, meaning that the seven-qubit error-correction code would fail completely
Errors per physical gate	p_{err}	Given in terms of γ^{-1}, T_{qb} and A by Eq. (B1); at the smallest γ^{-1} in Fig. 10 ($\gamma^{-1} = 3$ ms) this corresponds to $p_{err} \simeq 0.4 p_{thr}$ in the limit of $T \to 0$ and $A \to \infty$
Cryostat efficiency	⋯	Carnot efficiency at all temperatures (the state of the art is 10–30% of Carnot efficiency [80])
Number of refrigeration stages for control lines	K	Five, as sketched in Fig. 9, with temperature and attenuation in Eq. (B11)
Thermal conductivity of control lines	$\dot{q}_{cond}$	Coax above 10 K and superconducting microstrip below 10 K; details in Appendix B 2
Heat produced at T_{gen} by signal generation and readout (demux-mux, DAC, amplifier, and ADC)	$\dot{q}_{gen}$	Scenario A: 1 mW per physical qubit; futuristic CMOS logic (see text) Scenario B: 10 μW per physical qubit Scenario C: 0.1 μW per physical qubit; perhaps future SFQ logic (see text)
Heat produced at T_{qb} by paramps	⋯	Smaller than other heat sources at T_{qb} (see text), so neglected
Heat produced at 4 K by paramps	$\dot{q}_{para}$	Scenario A: 1 μW per physical qubit, value from Ref. [60] Scenario B: 10 nW per physical qubit Scenario C: 0.1 nW per physical qubit
Heat produced at 70 K stage by HEMT amplifiers	$\dot{q}_{HEMT}$	Scenario A: 50 μW per physical qubit, value from Ref. [60] Scenario B: absent since $T_{gen} < 70$ K (see text) Scenario C: absent since $T_{gen} < 70$ K (see text)
Heat conduction from T_{ext} to T_{gen} (optical fiber, dc, and local oscillation lines)	⋯	Absorbed into $\dot{q}_{gen}$ (see text); heat generated at T_{gen}, so it does not modify the values of $\dot{q}_{gen}$
Joule heating in all lines	⋯	The number and cross section of lines in Fig. 9 is chosen to ensure that Joule heating is less than other heat sources, and so can be neglected

levels of concatenation k increases, the physical requirements for logical T gates could grow differently than for Clifford gates, because the preparation of a logical magic state contains physical T gates and physical Clifford gates, while logical Clifford gates only need physical Clifford gates. However, Ref. [110] will show that this is not the case. This can be seen by carefully considering the circuit in Fig. 13 of Ref. [90], including all the Clifford gates necessary to prepare the logical states $|\text{cat}\rangle$, $|0\rangle$, and $|+\rangle$ that are required inputs into that circuit. This shows that the magic state preparation has a low number of physical T gates compared to physical Clifford gates. A straightforward calculation then shows that this means that the physical resources required by a T gate have the same scaling with k as we have given above for a Clifford gate but with a different prefactor. The prefactor is about 5 times greater for a T gate than for a Clifford gate for two reasons: (i) a logical T gate implemented on a logical qubit needs

the additional logical magic state as an ancilla and (ii) some extra magic states are needed to replace those that fail verification. Thus at any value of k, a logical T gate requires about 5 times more physical resources than a logical Clifford gate. Thus for a circuit in which logical T gates represent only $1/70$ of the total number of logical gates (including identity gates), it is a good approximation to neglect the T gates when calculating the power consumption.

APPENDIX B: PARAMETERS FOR THE FULL-STACK SUPERCONDUCTING QUANTUM COMPUTER

As stated in Sec. V E, we find the optimal values of four parameters: the temperature of the signal generation, T_{gen} (the top stage in Fig. 9), the temperature of the qubits, T_{qb}; the total attenuation A between T_{gen} and T_{qb}; and the concatenation level, k, for the error correction. However, there are many more parameters in our model that are not optimized. For these, our philosophy is to take optimistic but realistic numbers of the current state of the art. When we are forced to make simplifications, we aim for a simplification that gets that contribution to power consumption within an order of magnitude of the correct result. There is, however, one critical parameter for which we are vastly more optimistic than the state of the art: We assume that the qubits and gates are at least *a few orders of magnitude better* than those in Google's current Sycamore chips. The reason for this vastly optimistic assumption is that the concatenated error-correction scheme based on the seven-qubit code fails unless the error is below the threshold $p_{\text{thr}} = 2 \times 10^{-5}$. While Google's current Sycamore chips typically have a two-qubit gate error probability of about 0.01, other recent works [93] suggest that an error probability per gate of 2×10^{-4} might soon be achievable (taking their T_2 with a gate time of 100 ns). Thus, we hope that error probabilities significantly below the threshold p_{thr} should be achieved within a few years.

We take the gate times for physical gates from Google's Sycamore chip [72], except that we take a longer two-qubit gate time of $\tau_{2\text{qb}} = 100$ ns, because we have in mind the long-range two-qubit gates necessary for the seven-qubit code scheme. This can be done by, e.g., making the qubits interact with each other through a bus via a cross-resonance technique [85,86]. This makes the two-qubit gate slower than in the Sycamore chip and makes this the longest gate time in our modeling. The time step of the computer (its clock cycle) is fixed by this gate time and hence $\tau_{\text{step}} \sim 100$ ns in our model.

The principal components of the full-stack model of a fault-tolerant quantum computer are sketched in Fig. 9. They are explained in detail in the following subsections but can be briefly summarized as follows:

(a) A stage for the qubits at temperature T_{qb}, which also houses attenuation to remove thermal noise on the drive signals, and superconducting paramps to boost the readout signal.

(b) A stage containing superconducting paramps [60] at $T_{\text{Amp}} = 4\ K$, to boost the readout signal to a level above the noise at 70 K.

(c) A stage containing amplifiers made from HEMTs [60] at $T_{\text{Amp}} = 70$ K, to boost the readout signal to a level above the noise at T_{gen}.

(d) A number of stages between T_{qb} and T_{gen}, with attenuators on each stage to attenuate the thermal noise in the qubit driving signal. Their role is to evacuate heat at intermediate temperatures, to reduce the amount of heat to be evacuated at the lowest temperature stage.

(e) A stage at temperature T_{gen} which we call the signal-generation stage. It contains the classical electronics consisting of demultiplex-multiplex chips (demux-mux), digital-to-analog converters (DACs), and analog-to-digital converters (ADCs). The demultiplex part of the chip takes as inputs the list of gates to be performed at a given instant from the optical fibers (information that has been multiplexed in the room-temperature computer). This is then sent to the appropriate DAC that generates the appropriate wave form from one it has stored in memory. The wave forms are then multiplied by the local oscillator and sent toward the qubits. The ADCs receive measurements on the qubits, coming through the chain of amplifiers (including one at T_{gen}). The multiplex part of the chip takes the digital version of the readout signal from the ADC, multiplexes the data, and sends it back up the optical fiber, for it to be analyzed by the classical computer at room temperature.

(f) Electronics at room temperature, including the generation of the local oscillator, and a classical computer. The role of the latter is to manage the algorithm on the logical level, to decode the syndromes from error correction and to digitally demodulate the readout signals. In Appendix B 5, we argue that all the power consumption at room temperature can be neglected in comparison to the power dissipated by the stage at T_{gen}, at least for our scenario A, which is inspired by a futuristic view of CMOS electronics. To help understand the parameter dependencies, we also neglect it in our scenarios B and C but it can easily included using the information in Appendix B 5.

At each stage, the cryogenics must evacuate the heat generated at that stage and the heat conducted down cables from higher temperatures. We now explain the details of our full-stack model and our motivations

for neglecting the power consumption for some of the components.

1. Attenuation on microwave lines

As is standard, we assume that the thermal-photon contribution to the noise is reduced to an acceptable level by a chain of attenuators on the incoming microwave line (see Fig. 9). These attenuators are kept cold by the cryogenics, so they thermalize the signal coming down the line from hotter temperatures, reducing the population of thermal photons. For a chain of K cooling stages, with $K-1$ attenuators (Fig. 9 shows a case with $K=5$), the error probability of a physical qubit is

$$p_{\rm err} = \frac{\gamma\tau_{\rm step}}{2}\left(\frac{1}{2} + n(T_1) + \sum_{i=1}^{K-1}\frac{n(T_{i+1}) - n(T_i)}{\widetilde{A}_i}\right), \tag{B1}$$

where $T_1 = T_{\rm qb}$ and $n(T) = (\exp[\hbar\omega/(k_B T)] - 1)^{-1}$ is the Bose-Einstein function at the qubit frequency. Here, A_i is the attenuation on stage i at temperature T_i (as sketched in Fig. 9) and we define $\widetilde{A}_i$ to be the total attenuation between T_i and the qubits, so that $\widetilde{A}_i = A_i \cdots A_2 A_1$. The sum in Eq. (B1) is due to thermal photons that leak through the attenuators from higher temperatures.

We see that the noise can always be reduced by increasing the attenuation but this comes at the cost of greater power consumption. This makes the attenuation a crucial parameter in our optimization, as explained in Appendix B 6.

2. Heat conducted by microwave lines

We consider that the microwave signals are carried by coaxial cables above 10 K and superconducting microstrip line below 10 K, as sketched in Fig. 13. Unfortunately, these cables also conduct heat from higher to lower temperatures and this heat must be evacuated by the cryogenics. Thus, we must quantify the heat flow $\dot{q}_{\rm cond}$ from a higher temperature T_2 to a lower temperature T_1. For a cable of length L, made of both metal and Kapton (polyimide) dielectric, Fourier's law gives

$$\dot{q}_{\rm cond} = \frac{1}{L}\int_{T_1}^{T_2} dT\,[\mathcal{A}_{\rm kd}(T)\lambda_{\rm kd}(T) + \mathcal{A}_m\lambda_m(T)], \tag{B2}$$

where the Kapton dielectric has cross-section area $\mathcal{A}_{\rm kd}(T)$ and thermal conductivity $\lambda_{\rm kd}(T)$ at temperature T, while the metal has cross-section area $\mathcal{A}_m(T)$ and thermal conductivity $\lambda_m(T)$ at temperature T. Note that $\mathcal{A}_m(T)$ is the total cross section of the stainless steel in the coaxial cable including the ground or the total cross section of the Nb-Ti in the microstrip line including the ground.

As the heat conduction is inversely proportional to the length L of the cables between stages in the cryostat, it is good to have long microwave cables (coiled if necessary); we take $L = 1$ m. In the coaxial cable, the heat conduction is dominated by the stainless steel, so we can neglect the Kapton dielectric. In contrast, the heat conduction in the microstrip lines is dominated by the Kapton dielectric (because the Nb-Ti is superconducting and so its thermal conductivity is similar to that of Kapton but its cross section is smaller). Then, Eq. (B2) is reasonably well approximated by

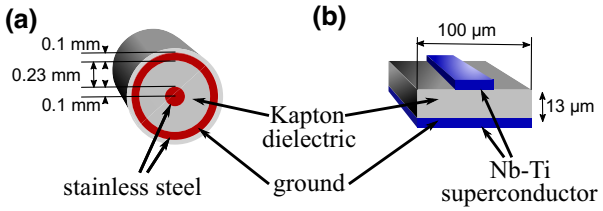


FIG. 13. The types of control and readout lines considered in our modeling. We need explicit models of these to calculate the heat that they will conduct between temperature stages in the full-stack quantum computer: (a) $T > 10$ K, coaxial cable; (b) $T < 10$ K, superconducting microstrip. The coaxial cables are called ULT-23 and have a radius of the order of 1 mm. The microstrip lines are those discussed in Ref. [43] and are much smaller than the coaxial cables (their largest dimension is 0.1 mm).

$$\dot{q}_{\rm cond} = \frac{1}{L}\int_{T_1}^{T_2} dT\,\mathcal{A}(T)\lambda(T), \tag{B3}$$

where the relevant cross sections are

$$\mathcal{A}(T) = \begin{cases} 2.7\times10^{-7}\ {\rm m}^2, & \text{for } T > 10\ {\rm K}, \\ 1.3\times10^{-9}\ {\rm m}^2, & \text{for } T < 10\ {\rm K}. \end{cases} \tag{B4}$$

The conductivity above 10 K is that of stainless steel, given by a fitting to experimental data in Ref. [116] as

$$\lambda(T > 10\ {\rm K}) = 10^{Z(T)} \quad \text{with } Z(T) \equiv \sum_{\alpha=0}^{8} a_\alpha \left(\log_{10} T\right)^\alpha, \tag{B5}$$

where the fitting parameters are $a_0 = -1.4087$, $a_1 = 1.3982$, $a_2 = 0.2543$, $a_3 = -0.6260$, $a_4 = 0.2334$, $a_5 = 0.4256$, $a_6 = -0.4658$, $a_7 = 0.1650$, and $a_8 = -0.0199$. The conductivity below 10 K is that of the Kapton (polyimide) dielectric, which we model phenomenologically as

$$\lambda(T < 10\ {\rm K}) = \begin{cases} 4.6T^{0.56}, & \text{for } T < 4\ {\rm K}, \\ 3.0T^{0.98}, & \text{for } 4\ {\rm K} < T < 10\ {\rm K}. \end{cases} \tag{B6}$$

The form for $T < 4$ K is taken from experimental measurements on Kapton in the range 0.5 K $< T <$ 5 K [117], which we assume can be extrapolated down to 0 K, although we rarely need to extrapolate it below about 0.05 K. The form for $T > 4$ K is a simplification of the form

extracted from experiments in the range 4–300 K [116]. This form slightly overestimates the conductivity in the range 4–10 K compared to Ref. [116] and leads to a small discontinuity at 4 K, which would be absent in a more accurate model. However, a more realistic T dependence would not change $\dot{q}_{\rm cond}$ by more than a few percent in the range 0–10 K and we observe that $\dot{q}_{\rm cond}$ in this range makes only a small contribution to the total power consumption. Hence, for our purposes, Eqs. (B3)–(B6) give a sufficiently accurate model of the heat conduction in the microwave lines.

3. Control electronics for signal generation and readout

The electronics on the signal-generation stage (at $T_{\rm gen}$) have two jobs. Their first job is to generate the microwave signals driving the qubits. This is done by first demultiplexing (demux) the data sent down the optical fiber (which contains a few bits of information describing which gate has to be performed). Such information is then sent to the appropriate DAC, which (i) reads the description of the wave form to generate from memory, (ii) generates the wave form, and (iii) multiplies it with the local oscillator. The microwave signal is then sent toward the qubits through the coaxial cables.

The second job of the electronics at this stage is to take the output of the measurements, digitize them (ADC), and multiplex (mux) them to be sent up the optical fiber to the traditional computer at room temperature.

We consider three scenarios to do the jobs of these control electronics (demux-muxs, DACs, and ADCs), with increasing levels of energy efficiency. Scenarios A and C are inspired by futuristic views of technologies that could do the job, while scenario B is simply a point halfway between A and C, to better understand the parameter dependencies.

Scenario A [the least energy efficient; see the red curves in Fig. 10(a)] is a parameter regime that is a futuristic view for CMOS, for which we assume the power consumption $\dot{q}_{\rm gen} = 1$ mW per qubit (assuming that the time step of the quantum computer, $\tau_{\rm step} = 100$ ns). This is a bit more than an order of magnitude better than current state-of-the-art CMOS [46,94,95] but that state of the art is rapidly improving at present. We use Eq. (B13) for the heat dissipated to multiplex and demultiplex (mux-demux chip). Existing multiplexing consumes 0.8 pJ/bit [118], giving a heat dissipation $\lesssim 0.5$ mW. Thus, we take the optimistic view that the control electronics at $T_{\rm gen}$ will generate a total heat of $\dot{q}_{\rm gen} = 1$ mW per physical qubit.

Scenario B [intermediate energy efficiency; see the blue curves in Figs. 10(a)] is taken to be halfway between scenarios A and C, so it is 100 times more energy efficient than scenario A, with $\dot{q}_{\rm gen} = 10\ \mu$W.

Scenario C [the most energy efficient; see the green curves in Figs. 10(a)] is in a parameter regime inspired by SFQ logic. This is a type of classical logic that is under development, based on superconducting circuits. Initial estimates suggest that it could be 10^4 times more energy efficient than CMOS [43], so we take $\dot{q}_{\rm gen} = 100$ nW.

At various points in the following sections, we will compare certain heating mechanisms at or near $T_{\rm gen}$ to the heat generated by the control electronics in scenario A ($\dot{q}_{\rm gen} = 1$ mW), as a way of justifying neglecting those heating mechanisms. We choose to also neglect those heating mechanisms when plotting results for scenarios B and C (even when this may not be justified in a technology corresponding to these scenarios) to help us understand the parameter dependence of the mechanisms that we do *not* neglect. In practice, this means that one should not expect our results for scenario C to apply directly to SFQ logic. We prefer to say that scenario C is an indication of the potential interest of SFQ logic, while being modest about our lack of concrete knowledge of what an SFQ implementation would look like. For example, would its implementation involve an optical fiber from SFQ logic to a classical computer at room temperature or would all the classical computing be done with SFQ logic at low temperature? It is too early to tell and this would need to be clarified before going beyond the very naive estimates made with our scenario C.

4. Amplification stages

The qubit measurements require amplifiers. We first assume that we can measure about 100 physical qubits with a single readout line. We obtain the factor of 100 as follows. The readout time of 100 ns means that the readout on each physical qubit needs a bandwidth of about 0.01 GHz. If we assume that the qubit frequencies are spread over about 1 GHz, then we can have about 100 qubits operating at different frequencies on a single readout line. We assume that the signal in each readout line is amplified by three amplifiers, one at the qubit temperature, one at 4 K, and one at 70 K. Here, we take experimental numbers for the 4 K and 70 K amplifiers from Ref. [60], while the paramps at $T_{\rm qb}$ are treated separately in Sec. B 7 [119]. In Ref. [60], the authors propose HEMT amplifiers at $T_{\rm HEMT} = 70$ K, which cannot be turned off between measurements, so that they are continually generating 5 mW of heat generation per amplifier. With one amplifier for 100 physical qubits, this gives us $\dot{q}_{\rm HEMT} = 50\ \mu$W. The job of this amplifier is to amplify the signal well above the noise at 300 K. However if the signal readout temperature $T_{\rm gen}$ is below 70 K, it is then clear that this amplifier is unnecessary and we can set $\dot{q}_{\rm HEMT}$ to zero. It turns out that our optimization places $T_{\rm gen}$ above 70 K in our scenario A (so we keep $\dot{q}_{\rm HEMT} = 50\ \mu$W) but it places $T_{\rm gen}$ below 70 K in our scenarios B and C, for which $\dot{q}_{\rm HEMT} = 0$.

Reference [60] proposes using superconducting paramps at $T_{\rm para} = 4$ K, which are powered by microwave pump

signals sent from room temperature to 4 K through a 20 dB attenuator (to reduce the noise on the line). We estimate that these will need a driving power of order 10^{-6} W to amplify the readout of 100 physical qubits. Thus, the 20 dB attenuator will dissipate heat of order 10^{-4} W for the 100 physical qubits. This gives $\dot{q}_{\text{para}} = 1\ \mu\text{W}$. For simplicity, we take the worst-case scenario, where the paramp microwave pump signal is always on, and we take all the attenuation on this microwave driving to be at $T_{\text{para}} = 4\,\text{K}$, so that $\dot{q}_{\text{para}}$ is entirely dissipated at 4 K. Clearly, this can be improved significantly, by turning off the pump signal when no qubits are being measured and by having the 20 dB of attenuation being generated by a chain of attenuators at different temperatures. In practice, we take $\dot{q}_{\text{para}} = 1\ \mu\text{W}$ for our scenario A. In the case of scenario C, the readout electronics are already at a temperature close to 4 K, given the results of our optimizations, so it is likely that this amplifier at 4 K will not be necessary. In our model, it is modeled by considering $\dot{q}_{\text{para}} = 0.1$ nW (a negligible contribution compared to the heat dissipated by control electronics at T_{gen} in scenario C). Finally, scenario B, which is chosen to be a situation halfway between A and C, has $\dot{q}_{\text{para}} = 10$ nW.

5. Power consumption at room temperature

At room temperature, we have different sources of power consumption. First, there is the generation of the local oscillator. We believe it reasonable to neglect it compared to the power consumed by the DACs and ADCs for our scenario A, given the information available in the literature [120]. Then, there are electronics multiplexing and demultiplexing data coming from the optical fibers. In Sec. B 3, we will show that this power consumption can be neglected. Finally, there is a classical computer that has three main purposes: (i) to digitally demodulate the readout signals coming from the ADC (in order to interpret the state of the measured qubits from the readout signals); (ii) to decode the syndromes from error correction; and (iii) to manage the algorithm on the logical level. We argue here that the power consumption for all these can be neglected compared to the power consumption that we take for the signal-generation stage.

For (i), the digital demodulation, we consider $N_d = 100$ discrete points in time to provide a good accuracy for the digitization of the readout signals [121]. Two quadratures of the readout signal must be found to obtain the state of the qubit from the phase of this signal. This is done by multiplying the digitized signal once by cos and once by sin (for the two quadratures) [62], requiring approximately $2N_d$ operations. Then, we multiply this quantity by the number of measurements per unit time, $N_{\text{meas}}/\tau_{\text{step}}$. This is further multiplied by the energy required to perform each operation. We overestimate this by taking the energy cost of a floating-point operation to be $q_{\text{Float}} \approx 0.85$ pJ [122]. Then, the power consumption *per physical qubit* required for the demodulation is given by

$$\dot{q}_{\text{demodulation}} = 2N_d \frac{N_{\text{meas}}(k)}{Q_P(k)\tau_{\text{step}}} q_{\text{Float}}. \tag{B7}$$

This decreases with k and is around 200 μW for $k = 1$ (and hence negligible compared to the power required for DACs and ADCs in scenario A). To be more precise, in a heterodyne demodulation scheme, the readout signals are multiplied by a local oscillator before being filtered and digitized [62], with the cost of digitization given by Eq. (B7). We assume that this multiplication of signals with a local oscillator is being done by the ADCs at T_{gen} and that the energy needed for it is accounted for in the $\dot{q}_{\text{gen}} = 1$ mW of scenario A.

For (ii), decoding the syndrome, we need the number of operations per unit time required to infer which error has occurred. Multiplying that number by the energy cost of one operation will give the required power. The recursive nature of concatenated code makes an exact calculation of the number of operations unnecessarily complicated for our goal here. Instead, we assume that one operation has to be performed per physical qubit per time step. This gives a pessimistic cost per physical qubit to decode the syndrome:

$$\dot{q}_{\text{syndrome}} = \frac{1}{\tau_{\text{step}}} q_{\text{Float}} \approx 8\ \mu\text{W}. \tag{B8}$$

This cost is several orders of magnitude lower than the power consumption of the electronics in our scenario A (1 mW per physical qubit), so we neglect it in our calculations. A more exact calculation of the number of operations necessary to decode a syndrome is sufficiently complicated that we have not rigorously shown this to be an upper bound on $\dot{q}_{\text{syndrome}}$ but we believe that it is not far from such a bound.

For (iii), we have considered it reasonable to neglect the total cost because the logical algorithm and its decomposition into a physical circuit comprising gates from the chosen gate set can be precomputed and stored in memory. Because memory storage is usually cheap in electronics, we consider it reasonable to neglect the associated cost. The only processing necessary is to add the gates required to correct any errors to this precomputed circuit. As errors are rare (less than one every 10^5 time steps for each qubit), adding such error-correction gates will be equally rare and this will require a tiny fraction of the processing required to decode the syndrome, so it can be safely neglected.

6. Power consumption per physical gate

The power consumption per physical gate is averaged over the time step of the computer, τ_{step}, assuming that the 2qb gates take one time step but the 1qb gates are faster

(taking a time $\tau_{1qb} < \tau_{step}$). This gives

$$P_{2qb} = P_\pi \sum_{i=1}^{K} \frac{T_{ext} - T_i}{T_i} (\widetilde{A}_i - \widetilde{A}_{i-1}), \tag{B9}$$

$$P_{1qb} = \frac{\tau_{1qb}}{\tau_{step}} P_{2qb}, \tag{B10}$$

where $\widetilde{A}_i$ is given below Eq. (B1). Note that, as all signals arriving at stage 1 (the cryogenic stage containing the qubits) are eventually dissipated as heat in that stage, one must take $\widetilde{A}_0 = 0$ in the sum. The power supplied for 1qb and 2qb gates during the gate is the same but 1qb gates are faster, so the power averaged over the time step is smaller for 1qb gates than 2qb gates in Eqs. (B9) and (B10). In our examples, we take $\tau_{1qb} = \frac{1}{4}\tau_{step}$.

To keep the optimization tractable, we do not optimize the temperatures and attenuation at each refrigeration stage. Instead, we take the common rule of thumb that stages should have equal attenuation and be regularly spaced in orders of magnitude of temperature between T_{gen} and T_{qb}. In other words, if we want a total attenuation of A, we take

$$A_i = A^{1/(K-1)}, \quad T_i = T_{qb} \left(\frac{T_{gen}}{T_{qb}} \right)^{(i-1)/(K-1)} \tag{B11}$$

for K stages of cooling. While this is not optimized, we suspect that the power consumption is not far from the optimal, because we have observed in specific cases that increasing K from 4 to 5 does not greatly reduce the power consumption. All plots in this work are for $K = 5$, since this is typical of current cryostats.

7. Power consumption per physical measurement

To estimate the power consumption per measurement, P_{meas}, we note that it originates from the heat dissipated when performing the amplification of the signals coming from the qubits. The first amplification occurs using the superconducting paramps at temperature T_{qb} in Fig. 9. Being superconducting, these dissipate negligible heat but they require microwave driving signals to be sent down through the attenuators. The microwave driving of the paramp must be about 100 times its output signal; i.e., 100 times the input signal times the amount of amplification. For the measurement of a single-qubit state, the input signal is one photon during the measurement time, τ_{meas}, so the power being measured is $\hbar\omega_0/\tau_{meas} \sim 10^{-17}$. This needs to be amplified by a factor of 100, so the microwave driving signal for the paramp is about 10^{-13} W per physical measurement. This is at least 40 times smaller than the microwave driving necessary for a single-qubit gate for the values of γ that we consider, given by P_π in Eq. (1). Thus the heat generated in the attenuators due to the driving of the paramps can be neglected, compared to the heat generated in the attenuators due to the driving of the single-qubit and two-qubit gates.

The other stages of amplification (at 4 K, 70 K, and T_{gen}) are assumed to be always on and so they dissipate heat constantly (see Sec. B 4). This means that they contribute to P_Q (see Sec. B 8) rather than to P_{meas}. Thus, P_{meas} is at least 40 times smaller than P_{1qb} and P_{1qb} and we neglect it.

8. Power consumption per physical qubit

The heating proportional to the number of qubits (independent of the number of gates or measurements being performed) comes from thermal conduction in cables and the heat generated by electronics that cannot be switched off (amplifiers and signal generation and readout). We define $\dot{q}_{cond}(T_i, T_{i+1})$ as the heat conduction per physical qubit due to the cables between cryogenic stages at T_{i+1} and T_i. It has been given in Sec. B 2. We define $\dot{q}_{gen}$ as the power consumed (and turned into heat) per physical qubit by the control electronics at T_{gen}. We define $\dot{q}_{HEMT}$ as the power consumed (and turned into heat) per physical qubit by the conventional HEMT amplifiers at $T_{HEMT} = 70$ K and we define $\dot{q}_{para}$ as the power consumed (and turned into heat) per physical qubit by the superconducting paramps at $T_{para} = 4$ K. Values for $\dot{q}_{gen}$, $\dot{q}_{HEMT}$ and $\dot{q}_{para}$ are given in Appendixes B 3 and B 4. For K stages of cryogenics between signal generation and qubits (as sketched in Fig. 9 for $K = 5$), the power consumption per qubit is

$$P_Q = \frac{T_{ext}}{T_{gen}} \dot{q}_{gen} + \frac{T_{ext}}{T_{HEMT}} \dot{q}_{HEMT} + \frac{T_{ext}}{T_{para}} \dot{q}_{para} + \sum_{i=1}^{K} \frac{T_{ext} - T_i}{T_i} \left(\dot{q}_{cond}(T_i, T_{i+1}) - \dot{q}_{cond}(T_{i-1}, T_i) \right), \tag{B12}$$

where $T_1 \equiv T_{qb}$ and $T_K \equiv T_{gen}$. We take $K = 5$ in Eq. (B12) for the reason explained in Appendix B 6. Note that terms such as $(T_{HEMT}/T_{gen})\dot{q}_{HEMT}$ are the sum of the power supplied to the HEMT amplifier, which is dissipated as heat, $\dot{q}_{HEMT}$, and the cryogenic power cost to remove that heat with Carnot efficiency, $((T_{HEMT} - T_{gen})/T_{gen})\dot{q}_{HEMT}$. Note also that as there is no stage below stage 1, one must take $\dot{q}_{cond}(T_0, T_1) = 0$ in the sum.

We have neglected the heat conduction between the laboratory and the stage at $T_K = T_{gen}$. This is because relatively few cables are required for the local oscillator and dc cables. In principle, the wires from the local oscillator and the dc sources need filtering to ensure that thermal noise at T_{ext} does not perturb the signal generation at T_{gen}. As each such wire carries a single specific frequency (rather than complex wave forms), we assume that the filtering can be done by reflection rather than absorption, so we neglect heating due to such filtering. Hence, the main source of

heat conduction at T_{gen} is dominated by the optical fibers that allow the exchange of data with the laboratory. To show that such heat conduction can be neglected, we need the ratio of optical fibers to physical qubits. This is given by the ratio of the bit rate of an optical fiber, $N_{\text{bit rate}} = 400$ Gb/s, to the information exchanged per unit time (the bit rate) for a single physical qubit. Most of the information exchanged between T_{ext} and T_{gen} is the digitized version of the readout signals going up the fiber from ADCs to the room-temperature computer. We take [121] each readout to have its amplitude digitized with $N_{\text{encoded}} = 14$ bits at $N_{\text{samples}} = 100$ discrete points in time during a time step, $\tau_{\text{step}} = 100$ ns. There are $N_{\text{meas}}(k)/\tau_{\text{step}}$ measurements per unit time, so we arrive at the following bit rate per physical qubit:

$$\frac{N_{\text{encoded}} N_{\text{samples}}}{\tau_{\text{step}}} \frac{N_{\text{meas}}(k)}{Q(k)} \lesssim 1.5 \text{ Gb/s}. \tag{B13}$$

Equations (16) and (A5) show that the ratio of measurements to physical qubits $N_{\text{meas}}(k)/Q(k)$ is maximal for $k = 1$. Thus, we have taken $k = 1$ to obtain the above overestimate of the bit rate. This gives one optical fiber for every approximately 270 physical qubits.

The heat conducted from T_{ext} to T_{gen} per physical qubit is thus $1/270$ times the heat conducted of an optical fiber, which is much less than 1 mW, per physical qubit, giving a heat conducted down to T_{gen} of much less than 4 μW per physical qubit. This is a significant overestimate, because 1 mW is the approximate heat conducted between 300 K and 0 K by a coaxial cable in Appendix B 2, when the temperature drop here is less than 300 K to 0 K and optical fibers carry much less heat than coaxial cables (they are insulators rather than metallic). We thus neglect the heat conduction from T_{ext} to T_{gen}, because it is negligible compared to the 1 mW of heat dissipated by control electronics at T_{gen} in our scenario A.

9. Number of qubits per microwave line

Minimization of the number of cables going to the qubit stage minimizes the conduction of heat to the qubits from higher temperatures. This can be done by driving multiple qubits with a single microwave line and reading out multiple qubits with a single readout line. This is multiplexing at the level of qubits but it should not be confused with the multiplexing elsewhere in this paper (which is multiplexing of data in an optical fiber).

The basic idea is to place superconducting qubits at slightly different frequencies, i.e., ensure that each qubit has a slightly different ω. Then, one can have a single microwave line coupled to multiple qubits and one can send different signals on resonance with different qubits at the same time, thereby performing different gates on different qubits at the same time.

The fastest gate operations in our scheme take 25 ns and, thus, the signal that performs a gate operation will have a bandwidth of about 40 MHz. To ensure that the signal only affects the desired qubit, all qubits coupled to a given microwave line must have their frequencies spaced by 40 MHz. Assuming qubit frequencies that are spread over a range from 5.5 to 6.5 GHz, this means that a single microwave line can control about 25 qubits.

Similarly, we assume that each measurement of a qubit takes about 100 ns, following the scheme in Ref. [115]. Thus, the readout signal will have a bandwidth of about 10 MHz. If we consider 100 qubits and spread their frequencies in an intelligent way over the range from 5.5 to 6.5 GHz, we can have them being driven by four microwave lines but read out with only one line, i.e., one amplification chain per 100 qubits.

10. Efficiency of cryogenics

Here, we assume ideal (Carnot-efficient) cryogenics. As no cryogenics are ideal, real cryogenics will have a larger power consumption than those we consider here. The small-scale cryogenics used in most research laboratories put flexibility before efficiency and so often have very low efficiencies. However, once the quantum computing hardware is fixed and it is known how much heat will be evacuated at each temperature, then cryogenics engineers are good at optimizing efficiency. Some of the most efficient designs for cooling to cryogenic temperatures are used at CERN and they have efficiencies from 10% to 30% of Carnot efficiency [80].

Of course, high cryogenic efficiency requires that there is minimal heat leaking from one stage of the cryogenics to another. Here, we assume that this heat leakage is strictly zero. In other words, if the cryostat was empty (no cables conducting heat into it and no attenuators or amplifiers generating heat inside it), then it would require negligible power to keep it cold. This is clearly an idealization.

To be more realistic, we should take (i) the efficiency for heat extraction and (ii) the heat leakage into the cryostat from data sheets for the industrial state of the art. To treat point (i) in the full-stack model will require replacing the factor of $(T_{\text{ext}} - T_i)/T_i$ in Eqs. (8), (B9), and (B12) with a realistic efficiency for heat evacuation at each T_i, where each T_i is for a given temperature stage. To treat (ii) in the full-stack model, we could assume that the magnitude of the heat leakage between stages scales with the size of the cryostat and so scales linearly with the number of physical qubits. If this is the case, we just need to know the heat leakage between stages per physical qubit and then include it in the $\dot{q}_{\text{cond}}$ in Eq. (B12). If, in contrast, the heat leakage scales nonlinearly with the number of physical qubits, it must be included as a new term in Eq. (B12).

APPENDIX C: ALGORITHMIC PARAMETERS

1. Comment on Fig. 10

We explain here how the data in Fig. 10 are calculated. Rather than using Eq. (19), which is an approximation that fails for $k = 0$, Fig 10 is a calculation based on the exact formula given in Eq. (A3) (within the assumptions made in the model), in which we assume only identity gates at the logical level, so that $N_{1\mathrm{qb};L} = N_{2\mathrm{qb};L} = N_{\mathrm{meas};l} = 0$.

For $k = 0$, this is less power consuming than an arbitrary calculation that includes arbitrary gate operations, because identity gates add no dynamic costs (unlike other gates). The plot hence only includes the static costs of the calculation but not the dynamic costs [for the definition of static and dynamic, see below Eq. (14)]. However, as we have found that most of the power consumption is static rather than dynamic for the parameters in Fig. 10, it still gives a reasonable order-of-magnitude estimate of P_C for any calculation.

As soon as there is error correction, $k \geq 1$, so that many gates are necessary for the error correction, the power consumption of an arbitrary circuit is extremely similar to that of a circuit made only of logical identity gates. For $k = 1$, the dynamic part of the power consumption varies by much less than an order of magnitude between the case of a circuit solely composed of logical identity gates or solely composed of nonidentity gates [see Eq. (A3)]. As most of the power consumption is static rather than dynamic, the result in Fig. 10 for a circuit with only logical identity gates at $k = 1$ gives a fairly accurate estimate for an arbitrary circuit at $k = 1$.

For $k \geq 2$, the dynamic part of the power consumption will be the same, within a percent or so, for any computation. Thus, Fig. 10 gives a very accurate estimate for any calculation at $k \geq 2$.

Of course, when we say that Fig. 10 gives an accurate result for a given calculation, we mean accurate within the assumptions of the modeling. Specifically, we assume the absence of T gates. Appendix A 2 considers T gates and argues that they are rare enough in the circuit that we consider (a protocol to crack RSA encryption) that they will not contribute significantly to the power consumption of such a circuit. Thus we can apply the results in Fig. 10 to a circuit that is cracking RSA encryption.

2. Implementation of protocols to crack the RSA encryption

For our full-stack analysis of a fault-tolerant quantum computer in Sec. V, we need to know the number of logical qubits Q_L and logical depth D_L necessary for a typical calculation. The best-studied calculation is the cracking of RSA-n encryption, so we use that as a benchmark. The current record for cracking the RSA encryption with a classical supercomputer is for RSA-829 [99].

Here, we take the quantum protocol to crack the RSA-n encryption from Ref. [26], which proposes

$$\begin{aligned} Q_L &= 3n + 0.002n\mathrm{Log}_2[n], \\ D_L &= 500n^2 + n^2\mathrm{Log}_2[n], \end{aligned} \tag{C1}$$

for large n, where the algorithm has been decomposed using a gate set comprising Clifford, Toffoli, and T gates. Thus, to crack RSA-2048 ($n = 2048$), which is considered uncrackable on current classical supercomputers, one requires $Q_L = 6175$ and $D_L = 2.1 \times 10^9$, as in Fig. 10(a). The symbols (star, triangle, and square) in Fig. 10(b) indicate Q_L and D_L for different n. It might also be worth noting that the estimate from Ref. [26] is very similar to the early one from Zalka [112], where he found $Q_L = 5n$ and $D_L = 600n^2$.

If one wishes to minimize the power consumption, it is critical to chose the right implementation of the protocol. For example, another recent implementation of the protocol to crack RSA-n encryption [123] uses fewer logical qubits, at the cost of a larger logical depth. It has

$$\begin{aligned} Q_L &= 2n + 2, \\ D_L &= 52n^3 + \mathcal{O}[n^2]. \end{aligned} \tag{C2}$$

For RSA-2048, this gives $Q_L = 4098$ and $D_L = 4.4 \times 10^{11}$, so that Q_L is about two thirds of that in Eq. (C1) but D_L is 200 times that in Eq. (C1). Our full-stack analysis shows that even though this uses fewer qubits than in Eq. (C1), it uses much more energy. In many parameter regimes, the extra depth with respect to the implementation in Eq. (C1) means that more error correction is required, so there will be more physical qubits per logical qubit than for Eq. (C1). This means that the power consumption will be more than for Eq. (C1) and the calculation will take longer to run. However, for the parameters in Fig. 10(b), it happens that the extra depth required by the implementation in Eq. (C2) is not enough to require another concatenation level [see Eq. (C4)]. Hence the power needs of both implementations are similar. However, as the implementation in Eq. (C2) takes 200 times as long, the total energy cost of the computation is $200 \times 2/3 = 133$ times that of Eq. (C2). This takes its energy cost from the equivalent of about 20 car tanks of gasoline [101] to the equivalent of about 2666 car tanks of gasoline!

These results reveal the importance of research on optimizing the implementation of algorithms. Even modest reductions of the prefactors in equations such as Eq. (C1) can have significant effects. In particular, the effect can be huge if it happens that this reduction takes the system into

a regime in which the calculation can be done successfully with one fewer concatenation level.

This is different from common situations in classical computing algorithms, where implementing the algorithm in a way that runs faster does not make it consume less power. So, the total energy consumption is reduced linearly with the increase in speed. Here, the fact that a faster quantum algorithm also requires less power (because it requires less error correction) means that one will get a *better than linear* gain from any faster implementation of the algorithm.

There is also a *hardware* aspect of circuit implementation that is worth mentioning here; the results in Eq. (C1) depend on which gates can be directly implemented by the hardware of the quantum computer. Thus it is worth briefly examining an example of how this could affect the power consumption and energy cost of a calculation. Equation (C1) is based on the assumption that the hardware supports a fault-tolerant gate set consisting of Clifford, Toffoli, and T gates. However, it could well be that the fault-tolerant Toffoli gates in a concatenated seven-qubit code must be built out of T gates, something that takes about nine time steps, involving CNOT and T gates [124]. This is the point of view taken in Ref. [110]. In this case, the fact that the hardware cannot directly implement Toffoli gates would not change Q_L but it would increase D_L. A careful estimate of the amount by which D_L would increase would require delving into the details of the circuit to see how many Toffoli gates are done in parallel. We do not do that here. Instead, we look at the worst case and the best case. The worst case would be to start with Eq. (C1) and assume that there is no time step in the logical algorithm without a Toffoli gate and that when one replaces each Toffoli by a series of gates taking nine time steps, all other qubits just wait for this series of gates to finish before continuing with the algorithm. Then the logical depth, D_L, would be 9 times that in Eq. (C1). However, we believe this to be much too pessimistic. Reference [26] shows that only about one gate in every 5000 is a Toffoli gate; they say that their "Toffoli+T/2 count" is $0.3n^3 + 0.0005n^3\mathrm{Log}_2[n]$, which is about 5000 times smaller than $Q_L \times D_L$. Thus the best case would be that the depth is only increased by 0.02% (this would require all logical qubits to do all Toffoli gates in parallel). The true result will be somewhere between the two, although as Ref. [26] uses a lot of parallelization, we can hope that it will be much better than the worst case. In any event, we expect that such an increase of D_L combined with the fact that T gates are more demanding in physical resources (see Appendix A 2) could increase our energy estimates by an order of magnitude, which would not change the qualitative conclusions in our Sec. V F. Conversely, the field of circuit optimization is very active, so new protocols with smaller D_L than Eq. (C1) are likely to appear in the coming years. Thus, for simplicity, we use Eq. (C1) in our calculations here.

3. First-order phase transition between concatenation levels

The transition from the k to $k+1$ levels of concatenation in the error correction seen in Fig. 10(b) is analogous to a first-order phase transition. To understand why, suppose that one increases the depth of the calculation D_L for a given concatenation level k. To maintain the calculation metric of $\mathcal{M}_0$, one must reduce the error probability per gate operation, p_{err} in Eq. (B1), by reducing T_{qb} and increasing A. Doing this will cause P_C to diverge at a finite value of D_L, because p_{err} remains finite when $T_{\mathrm{qb}} \to 0$ and $A \to \infty$ but P_C diverges. Shortly before this divergence occurs, this power consumption becomes more than the power consumption if one allows larger p_{err} but adds a concatenation level. Thus, the transition occurs when the minimum P_C for k levels of error correction exceeds the minimum P_C for $(k+1)$ levels of error correction. This is analogous to a first-order phase transition, such as the liquid-gas transition, which occurs when the energies of two phases cross.

Equation (17) tells us that the maximum calculation size—i.e., the largest $\mathcal{N}_L$, where $\mathcal{N}_L$ is defined above Eq. (17)—for a given metric $\mathcal{M}$, concatenation level k, and error probability $p_{\mathrm{err}} < p_{\mathrm{thr}}$, is

$$\mathcal{N}_L = \frac{\ln[\mathcal{M}]}{\ln\left[\left(1 - p_{\mathrm{thr}}(p_{\mathrm{err}}/p_{\mathrm{thr}})^{2^k}\right)\right]} \simeq \frac{\ln[1/\mathcal{M}]}{p_{\mathrm{thr}}}\left(\frac{p_{\mathrm{thr}}}{p_{\mathrm{err}}}\right)^{2^k}, \tag{C3}$$

where the second line is a small p_{thr} approximation, using the fact that $p_{\mathrm{thr}} = 2 \times 10^{-5} \ll 1$.

This means that the maximum possible calculation size for a given k is when p_{err} in Eq. (B1) takes its minimum value, $\gamma\tau_{\mathrm{step}}/4$ (the value it takes when $n_{\mathrm{noise}} = 0$ because $T_{\mathrm{qb}} \to 0$ and $A \to \infty$). Thus, the maximum possible $\mathcal{N}_L$ for a given k is found by replacing p_{err} by $\gamma\tau_{\mathrm{step}}/4$ in Eq. (C3). This is the value of $\mathcal{N}_L$ at which the power consumption for k levels of error correction diverges. Following the above argument, it will be energetically favorable to switch from k to $k+1$ levels of error correction as $\mathcal{N}_L$ approaches this value from below.

Thus, as we assume $\mathcal{N}_L = Q_L \times D_L$, the transition from k to $k+1$ levels of error correction must occur for

$$Q_L \times D_L \lesssim \frac{\ln[1/\mathcal{M}]}{p_{\mathrm{thr}}}\left(\frac{4p_{\mathrm{thr}}}{\gamma\tau_{\mathrm{step}}}\right)^{2^k}. \tag{C4}$$

Now, in general, the power consumption per qubit has a term that depends on T_{qb} (and diverges as $T_{\mathrm{qb}} \to 0$) and another term that is T_{qb} independent. The latter term contains such things as the power consumption of the electronics at T_{gen} and the resources required to evacuate the

heat conducted down wires at temperatures above 10 K. We observe that when the $T_{\rm qb}$-independent term becomes larger, the transition then moves toward the line defined by the equality in Eq. (C4). Indeed, for the parameters considered in this work (summarized in Tables II and III), it is a reasonable approximation to say that the transitions occur at

$$Q_L \times D_L = \frac{\ln[1/\mathcal{M}]}{p_{\rm thr}} \left(\frac{4p_{\rm thr}}{\gamma \tau_{\rm step}} \right)^{2^k}. \tag{C5}$$

Indeed, deviations from this approximation are hardly noticeable, if one superimposes Eq. (C5) on the log-log plot in Fig. 10(b).

We note that Eq. (C5) can also be arrived at from the conventional wisdom that error correction is so expensive that, heuristically, one should always adjust other control parameters (the qubit temperature etc.) to minimize the amount of error correction (in this case, the concatenation level k). However, we emphasize that while this conventional wisdom works for the parameters given in Tables II and III, it can fail drastically for slightly different parameters (such as the less ideal parameters in Appendix E). Thus, while Eq. (C5) helps us give a simple interpretation of this aspect of the full-stack modeling, it would be dangerous to rely on it without *first* performing the full-stack modeling. The reason is that to know whether or not Eq. (C5) is a good approximation, one needs to know the relative strengths of the $T_{\rm qb}$-dependent and $T_{\rm qb}$-independent terms in the power consumption. This is typically information that is only accessible after one has optimized the full-stack quantum computer (and thereby already found the transitions). Thus, we see Eq. (C5) as a way to help to understand the result of the optimization *a posteriori*, rather than for making a prediction *a priori*.

Typically, Eq. (C5) fails when the $T_{\rm qb}$-dependent term in the power consumption is more significant than for the parameters given in Tables II and III. We give a plausible example of this below in Appendix E (see Fig. 14). There, we see that the position of the transition from $k = 3$ to $k = 2$ depends on unexpected control parameters, such as the power consumption of the control electronics. Hence, there is no simple way to guess the optimal value of k for a given algorithm size without performing the full-stack optimization.

APPENDIX D: CLASSICAL ENERGY EFFICIENCY

To compare quantum computers to classical computers, we need the energy and computational duration required by the classical supercomputer to crack an RSA-n private key, where n is the bit size of the private key. To proceed, we can consider the asymptotic estimation of the number of operations required by the classical algorithm called the general-number-field sieve (GNFS), which is the best-known classical algorithm to factorize a number into prime factors [99]. The number of operations required to crack the RSA-n encryption is

$$N_{\rm GNFS}(n) = \exp\left[\left(\frac{64n\ln(2)}{9} [\ln(n\ln(2))]^2 \right)^{1/3} [1 + o(1)] \right]. \tag{D1}$$

We consider $1 + o(1) \approx 1$; while it could hide a possibly large constant, to our knowledge, there is no better estimate available today. From this total number of operations, we extrapolate the total energy $E_{\rm GNFS}$ and time $t_{\rm GNFS}$ that this classical algorithm would take on the classical supercomputer JUWELS Module 1, which has been used in the state-of-the-art cracking of the RSA encryption [99]). This gives $E_{\rm GNFS}(n) = E_{\rm GNFS}(830)N_{\rm GNFS}(n)/N_{\rm GNFS}(830)$, where $E_{\rm GNFS}(830) \approx 1$ TJ. For $t_{\rm GNFS}$, we assume that the calculation can be fully parallelized on all cores of the JUWELS Module 1, so $t_{\rm GNFS}(830) \approx$ 8–9 days. This underestimates $t_{\rm GNFS}$, as some steps in the algorithm cannot be parallelized in this way (see Ref. [99]). Then, $t_{\rm GNFS}(n) = t_{\rm GNFS}(830)N_{\rm GNFS}(n)/N_{\rm GNFS}(830)$. We define the energy efficiency as $\eta = n/E(n)$ and this gives the black curve in Fig. 11.

APPENDIX E: A DIFFERENT SITUATION IN WHICH P_C IS DOMINATED BY HEAT PRODUCTION AT $T_{\rm qb}$

For the parameters taken throughout this work (summarized in Tables II and III), the power consumption of the cryogenic stage at temperature $T_{\rm qb}$ is a tiny contribution to the overall power consumption; see, e.g., Fig. 10(c). Many of our observations follow from this fact.

In this appendix, we look at a full-stack model in the *opposite* situation, where the power consumption is dominated by what happens at temperature $T_{\rm qb}$. This will be a common situation if the cryogenics are less ideal than we have assumed above and if extra heat must be dissipated at the qubit temperature for any reason. The model we take here differs from that elsewhere in this paper in two ways:

(i) We replace the Carnot efficiency with the efficiency for typical small-scale cryostats, given in Ref. [125] by a phenomenological formula constructed by fitting data. This formula says that the power required to remove heat at a rate $\dot{Q}$ from a cryogenic stage at T is $3.24 \times 10^5 \times \dot{Q}(1 - T/T_{\rm ext})/T^2$, for $T_{\rm ext} = 300\,{\rm K}$. This diverges much faster than Carnot-efficient refrigeration at small T and so the low-temperature stages will contribute a much bigger proportion of the cryogenic power consumption.

(ii) We add an additional heat load of 50 nW per physical qubit at the stage at T_{qb}. We have taken this from Ref. [41], which has estimated such an additional heat load in their technology due their flux biasing of each physical qubit to bring it to its operational sweet spot (the flux bias being necessary to compensate for intrinsic magnetic fields).

These two modifications greatly increase the power consumption of the part of the cryogenics that extracts heat at T_{qb} with respect to other contributions to the power consumption.

Figure 14 shows results for the same parameters as in Fig. 10(a). Of course, it is no surprise that the power consumption is much bigger now that the cryogenics are less efficient and there is extra heat to extract at T_{qb}. What is interesting is that now the power consumption varies more with γ^{-1}, within a given concatenation level of error correction, than at the jumps between levels of error correction. This means that it is so expensive to make the qubits colder that it is sometimes more competitive to perform an additional level of concatenation. As a consequence, there is no easy way to see what is the best level of concatenation for a given set of parameters. In Fig. 10, the places where the concatenation level had changed were

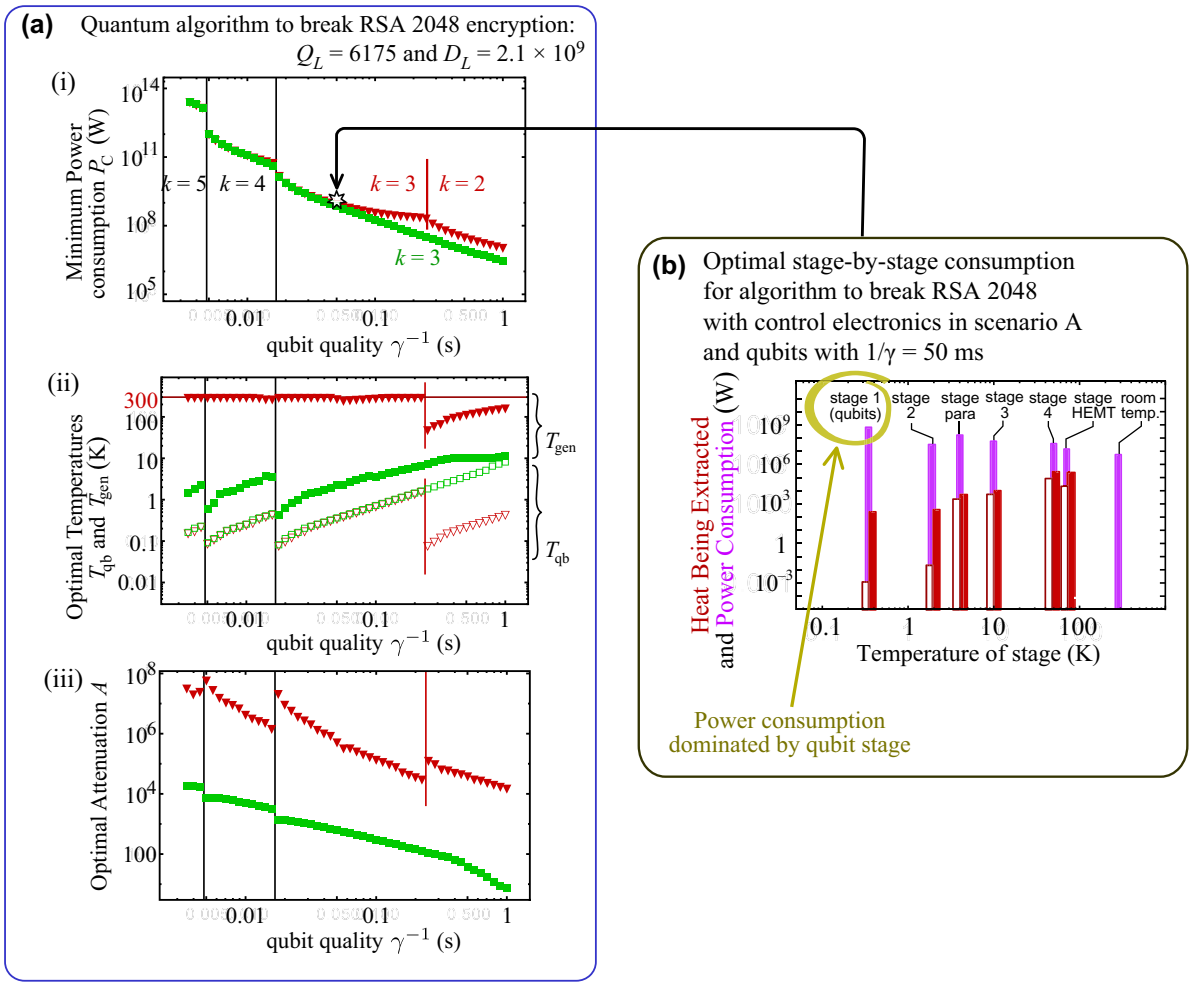


FIG. 14. (a) Minimization of the power consumption of Shor's algorithm breaking RSA-2048 with less efficient cryogenics and more heat sources at T_{qb} compared to Fig. 10 (see Sec. E). The plots show (i) the power consumption after minimization, with (ii) and (iii) showing the temperatures and attenuation. As in Fig. 10, the red curves are for control electronics in scenario A and the green curves are for scenario C. (b) The power consumption stage by stage at the point given by the star on the red curve in (a). Here, the power consumption is dominated by the cryogenic stage containing the qubits at T_{qb}, unlike in Fig. 10(c). This is typical for the parameter regime that we explore here (not just at the star). It is the reason that the power consumption depends strongly on the qubit temperature. This in turn means that the optimal amount of error correction can depend on the power consumption of the control electronics. We see that the transition from $k = 3$ to $k = 2$ is absent for the green curve (occurring outside the plot at $\gamma^{-1} > 1$ s). This is very different from Fig. 10(a), where all transitions are almost independent of the power consumption of the electronics.

the same for all curves and it was fairly well approximated by Eq. (C5). Here, the optimal concatenation level changes completely between the red and green curves in the region $\gamma^{-1} > 0.25$ s, simply because the control electronics have different power consumption for the red and green curves. This neatly shows the interdependence of technologies in the quantum computer. When the power consumption per physical qubit is low enough (green curve), adding more error correction (more physical qubits per logical qubit) costs less power than cooling the physical qubits further. Thus, the region of the green curve with $\gamma^{-1} > 0.25$ s is a regime in which it is better to have hotter physical qubits (more errors per physical qubit) and compensate with more error correction (i.e., $k = 3$ rather than $k = 2$). This is counter to the conventional wisdom outlined in Appendix C 3. It shows that, in general, only a full-stack optimization will find the optimal working conditions for a quantum computer.

This also gives us a clear example in which our MNR methodology allows us to reduce the power consumption by more than 3 orders of magnitude. Consider the power consumption for the SFQ electronics (green curve) in Fig. 14(a). Suppose that we had qubits with $\gamma^{-1} = 1$ s but we applied plausible but nonoptimal values of the attenuation, qubit temperature, and signal generation, such as those that would be optimal if $\gamma^{-1} = 0.02$ s. Then, the power consumption would be about 10 GW (the same as if $\gamma^{-1} = 0.02$ s), when the optimization tells us that a power consumption of only about 2 MW is necessary for $\gamma^{-1} = 1$ s. This would be a saving of more than 3 orders of magnitude, in a regime of high power consumption. Such an energy saving is possible because our MNR methodology links the fundamental noise model of the qubit to the classical hardware, electronics, cryogeny, etc.

[1] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, ArXiv:1411.4028 (2014).

[2] D. Amaro, C. Modica, M. Rosenkranz, M. Fiorentini, M. Benedetti, and M. Lubasch, Filtering variational quantum algorithms for combinatorial optimization, Quantum Sci. Technol. **7**, 015021 (2022).

[3] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. (2006).

[4] T. Häner, S. Jaques, M. Naehrig, M. Roetteler, and M. Soeken, in *Post-Quantum Cryptography* (Springer, Cham, Switzerland, 2020), p. 425.

[5] S. Chakrabarti, R. Krishnakumar, G. Mazzola, N. Stamatopoulos, S. Woerner, and W. J. Zeng, A threshold for quantum advantage in derivative pricing, Quantum **5**, 463 (2021).

[6] P. Rebentrost and S. Lloyd, Quantum computational finance: Quantum algorithm for portfolio optimization, ArXiv:1811.03975 (2018).

[7] B. Bauer, S. Bravyi, M. Motta, and G. Kin-Lic Chan, Quantum algorithms for quantum chemistry and quantum materials science, Chem. Rev. **120**, 12685 (2020).

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

[9] H. Ma, M. Govoni, and G. Galli, Quantum simulations of materials on near-term quantum computers, npj Comput. Mater. **6**, 1 (2020).

[10] Y. Cao, J. Romero, and A. Aspuru-Guzik, Potential of quantum computing for drug discovery, IBM J. Res. Dev. **62**, 6:1 (2018).

[11] M. Zinner, F. Dahlhausen, P. Boehme, J. Ehlers, L. Bieske, and L. Fehring, Quantum computing's potential for drug discovery: Early stage industry dynamics, Drug Discov. Today **26**, 1680 (2021).

[12] S. Slussarenko,and G. J. Pryde, Photonic quantum information processing: A concise review, Appl. Phys. Rev. **6**, 041303 (2019).

[13] C. D. Bruzewicz, J. Chiaverini, R. McConnell,and J. M. Sage, Trapped-ion quantum computing: Progress and challenges, Appl. Phys. Rev. **6**, 021314 (2019).

[14] G. Burkard, T. D. Ladd, J. M. Nichol, A. Pan, and J. R. Petta, Semiconductor spin qubits, ArXiv:2112.08863 (2021).

[15] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson,and W. D. Oliver, Superconducting qubits: Current state of play, Annu. Rev. Condens. Matter Phys. **11**, 369 (2020).

[16] L. Childress and R. Hanson, Diamond NV centers for quantum computing and quantum networks, MRS Bull. **38**, 134 (2013).

[17] E. A. Laird, F. Pei, and L. P. Kouwenhoven, A valley-spin qubit in a carbon nanotube, Nat. Nanotechnol. **8**, 565 (2013).

[18] V. Lahtinen and J. Pachos, A short introduction to topological quantum computation, SciPost Phys. **3**, 021 (2017).

[19] F. Arute, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[20] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, *et al.*, Quantum computational advantage using photons, Science **370**, 1460 (2020).

[21] E. Pednault, J. A. Gunnels, G. Nannicini, L. Horesh, and R. Wisnieff, Leveraging secondary storage to simulate deep 54-qubit sycamore circuits, ArXiv:1910.09534 (2019).

[22] Y. Zhou, E. M. Stoudenmire, and X. Waintal, What limits the simulation of quantum computers?, Phys. Rev. X **10**, 041038 (2020).

[23] C. Chamberland, T. Jochym-O'Connor, and R. Laflamme, Overhead analysis of universal concatenated quantum codes, Phys. Rev. A **95**, 022313 (2017).

[24] M. Suchara, J. Kubiatowicz, A. Faruque, F. T. Chong, C.-Y. Lai, and G. Paz, in *2013 IEEE 31st International Conference on Computer Design (ICCD)* (IEEE, Asheville, NC, 2013), p. 419.

[25] M. Suchara, A. Faruque, C.-Y. Lai, G. Paz, F. T. Chong, and J. Kubiatowicz, Comparing the overhead of topological and concatenated quantum error correction, ArXiv:1312.2316 (2013).

[26] C. Gidney and M. Ekerå, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits, Quantum **5**, 433 (2021).

[27] J. Guillaud and M. Mirrahimi, Error rates and resource overheads of repetition cat qubits, Phys. Rev. A **103**, 042413 (2021).

[28] C. Chamberland, K. Noh, P. Arrangoiz-Arriola, E. T. Campbell, C. T. Hann, J. Iverson, H. Putterman, T. C. Bohdanowicz, S. T. Flammia, A. Keller, G. Refael, J. Preskill, L. Jiang, A. H. Safavi-Naeini, O. Painter, and F. G. S. L. Brandão, Building a fault-tolerant quantum computer using concatenated cat codes, PRX Quantum **3**, 010329 (2022).

[29] K. Noh, C. Chamberland, and F. G. S. L. Brandão, Low-overhead fault-tolerant quantum error correction with the surface-GKP code, PRX Quantum **3**, 010315 (2022).

[30] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, in *Post-Quantum Cryptography* (Springer, Cham, Switzerland, 2016), p. 29.

[31] A. Pavlidis and D. Gizopoulos, Fast quantum modular exponentiation architecture for Shor's factorization algorithm, ArXiv:1207.0511 (2012).

[32] T. Häner, S. Jaques, M. Naehrig, M. Roetteler, and M. Soeken, in *International Conference on Post-Quantum Cryptography* (Springer, Daejeon, South Korea, 2020), p. 425.

[33] E. T. Campbell, Early fault-tolerant simulations of the Hubbard model, Quantum Sci. Technol. **7**, 015007 (2021).

[34] I. D. Kivlichan, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, W. Sun, Z. Jiang, N. Rubin, A. Fowler, A. Aspuru-Guzik, *et al.*, Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via Trotterization, Quantum **4**, 296 (2020).

[35] I. H. Kim, Y.-H. Liu, S. Pallister, W. Pol, S. Roberts, and E. Lee, Fault-tolerant re-source estimate for quantum chemical simulations: Case study on li-ion battery electrolyte molecules, Phys. Rev. Res. **4**, 023019 (2022).

[36] A. Delgado, P. A. M. Casares, R. dos Reis, M. Shokrian Zini, R. Campos, N. Cruz-Hernández, A.-C. Voigt, A. Lowe, S. Jahangiri, M. A. Martin-Delgado, J. E. Mueller, and J. M. Arrazola, Simulating key properties of lithium-ion batteries with a fault-tolerant quantum computer, Phys. Rev. A **106**, 032428 (2022).

[37] J. Lemieux, G. Duclos-Cianci, D. Sénéchal, and D. Poulin, Resource estimate for quantum many-body ground-state preparation on a quantum computer, Phys. Rev. A **103**, 052408 (2021).

[38] A. Scherer, B.ît Valiron, S.-C. Mau, S. Alexander, E. Van den Berg, and T. E. Chapuran, Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2D target, Quantum Inf. Process. **26**, 1484 (2017).

[39] D. Jaschke and S. Montangero, Is quantum computing green? An estimate for an energy-efficiency quantum advantage, ArXiv:2205.12092 (2022).

[40] M. James Martin, C. Hughes, G. Moreno, E. B. Jones, D. Sickinger, S. Narumanchi, and R. Grout, Energy use in quantum data centers: Scaling the impact of computer architecture, qubit performance, size, and thermal parameters, IEEE Trans. Sustainable Comput. **7**, 864 (2022).

[41] S. Krinner, S. Storz, P. Kurpiers, P. Magnard, J. Heinsoo, R. Keller, J. Luetolf, C. Eichler, and A. Wallraff, Engineering cryogenic setups for 100-qubit scale superconducting circuit systems, EPJ Quantum Technol. **6**, 2 (2019).

[42] C. G. Almudever, L. Lao, X. Fu, N. Khammassi, I. Ashraf, D. Iorga, S. Varsamopoulos, C. Eichler, A. Wallraff, L. Geck, A. Kruth, J. Knoch, H. Bluhm, and K. Bertels, in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017* (IEEE, Lausanne, Switzerland, 2017), p. 836.

[43] R. McDermott, M. G. Vavilov, B. L. T. Plourde, F. K. Wilhelm, P. J. Liebermann, O. A. Mukhanov, and T. A. Ohki, Quantum-classical interface based on single flux quantum digital logic, Quantum Sci. Technol. **3**, 024004 (2018).

[44] K. O. E. N. Bertels, A. Sarkar, T. Hubregtsen, M. Serrao, A. A. Mouedenne, A. Yadav, A. Krol, and I. Ashraf, in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (IEEE, Grenoble, France, 2020), p. 1.

[45] G. Donati, A look at the full stack, Nat. Rev. Phys. **3**, 226 (2021).

[46] D. J. Frank, S. Chakraborty, K. Tien, P. Rosno, T. Fox, M. Yeck, J. A. Glick, R. Robertazzi, R. Richetta, J. F. Bulzacchelli, *et al.*, in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 65 (IEEE, San Francisco, US, 2022), p. 360.

[47] M. R. Jokar, R. Rines, G. Pasandi, H. Cong, A. Holmes, Y. Shi, M. Pedram, and F. T. Chong, in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)* (IEEE, Seoul, South Korea, 2022), p. 400.

[48] M. Bandic, S. Feld, and C. G. Almudever, in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (IEEE, Antwerp, Belgium, 2022), p. 1.

[49] C. G. Almudever and E. Alarcon, in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (IEEE, Antwerp, Belgium, 2021), p. 762.

[50] P. Murali, N. M. Linke, M. Martonosi, A. J. Abhari, N. H. Nguyen, and C. H. Alderete, in *2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA)* (IEEE, Phoenix, USA, 2019), p. 527.

[51] S. Rodrigo, S. Abadal, E. Alarcón, M. Bandic, H. Van Someren, and C. G. Almudéver, On double full-stack communication-enabled architectures for multicore quantum computers, IEEE Micro **41**, 48 (2021).

[52] P. Gokhale, Ph.D. thesis, Department of computer science, Faculty of the division of the physical sciences, The University of Chicago, 2020.

[53] K. Ishida, M. Tanaka, T. Ono, and K. Inoue, Towards ultra-high-speed cryogenic single-flux-quantum computing, IEICE Trans. Electron. **101**, 359 (2018).

[54] E. P. DeBenedictis, in *2020 International Conference on Rebooting Computing (ICRC)* (IEEE, Atlanta, USA, 2020), p. 42.

[55] K. Kang, D. Minn, S. Bae, J. Lee, S. Bae, G. Jung, S. Kang, M. Lee, H.-J. Song, and J.-Y. Sim, in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 65 (IEEE, San Francisco, USA, 2022), p. 362.

[56] J.-S. Park, S. Subramanian, L. Lampert, T. Mladenov, I. Klotchkov, D. J. Kurian, E. Juarez-Hernandez, B. Perez-Esparza, S. Rani Kale, K. T. Asma Beevi, *et al.*, in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 64 (IEEE, San Francisco, USA, 2021), p. 208.

[57] J. C. Bardin, *et al.*, Design and characterization of a 28-nm bulk-CMOS cryogenic quantum controller dissipating less than 2 mW at 3 K, IEEE J. Solid-State Circuits **54**, 3043 (2019).

[58] L. You, Superconducting nanowire single-photon detectors for quantum information, Nanophotonics **9**, 2673 (2020).

[59] E. D. Walsh, W. Jung, G.-H. Lee, D. K. Efetov, B.-I. Wu, K.-F. Huang, T. A. Ohki, T. Taniguchi, K. Watanabe, P. Kim, D. Englund, and K. Chung Fong, Josephson junction infrared single-photon detector, Science **372**, 409 (2021).

[60] M. Malnou, J. Aumentado, M. R. Vissers, J. D. Wheeler, J. Hubmayr, J. N. Ullom, and J. Gao, Performance of a kinetic inductance traveling-wave parametric amplifier at 4 Kelvin: Toward an alternative to semiconductor amplifiers, Phys. Rev. Appl. **17**, 044009 (2022).

[61] L. Planat, A. Ranadive, R. Dassonneville, J. P. Martínez, S. Léger, C. Naud, O. Buisson, W. Hasch-Guichard, D. M. Basko, and N. Roch, Photonic-crystal Josephson traveling-wave parametric amplifier, Phys. Rev. X **10**, 021021 (2020).

[62] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, A quantum engineer's guide to superconducting qubits, Appl. Phys. Rev. **6**, 021318 (2019).

[63] A. Auffèves, Quantum technologies need a quantum energy initiative, PRX Quantum **3**, 020101 (2022).

[64] J. Puebla, J. Kim, K. Kondou, and Y. Otani, Spintronic devices for energy-efficient data storage and energy harvesting, Commun. Mater. **1**, 24 (2020).

[65] R. Desislavov, F. Martínez-Plumed, and J. Hernández-Orallo, Compute and energy consumption trends in deep learning inference, ArXiv:2109.05472 (2021).

[66] M. Fellous-Asiani, J. H. Chai, R. S. Whitney, A. Auffèves, and H. K. Ng, Limitations in quantum computing from resource constraints, PRX Quantum **2**, 040335 (2021).

[67] MNR has been briefly summarized for nonexperts in Auffèves' perspective article [63], citing this work as the place in which it would be presented as a complete scientific methodology, used to make quantitative predictions.

[68] S. Martiel, T. Ayral, and C. Allouche, Benchmarking quantum coprocessors in an application-centric, hardware-agnostic, and scalable way, IEEE Trans. Quantum Eng. **2**, 1 (2021).

[69] 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).

[70] In our models, to minimize the resource consumption, we should take the *smallest* metric that allows us to perform the task of interest. Put differently, the minimum resource required to achieve $\mathcal{M} \geq \mathcal{M}_0$ is reached when $\mathcal{M} = \mathcal{M}_0$. We have noted that this is true whenever the resources and metrics grow monotonically with at least one control parameter. This applies in our case, since the power consumption and metric always increase monotonically as the qubit temperature is reduced or as the attenuation is increased. In contrast, if one had a case where the resources or metrics were nonmonotonic functions of *all* the control parameters, then one might be able to achieve lower power consumption by going to a higher metric than the target necessary for the task of interest, taking $\mathcal{M} > \mathcal{M}_0$.

[71] The Green500 is at top500.org, the most recent list at the time of writing was Nov. 2021.

[72] See, e.g., *Google AI Quantum Computer Datasheet* (May 14, 2021) published online at https://quantumai.google/hardware/datasheet/weber.pdf.

[73] N. Cottet, S. Jezouin, L. Bretheau, P. Campagne-Ibarcq, Q. Ficheux, J. Anders, A. Auffèves, R. Azouit, P. Rouchon, and B. Huard, Observing a quantum Maxwell demon at work, Proc. Natl. Acad. Sci. **114**, 7561 (2017).

[74] More precisely, $\mathcal{M}_{1_{\mathrm{qb}}} = \mathcal{M}_0$ implies that $\tau_{1\mathrm{qb}} = (1 - \mathcal{M}_0)/\gamma$. Replacing it in the expression for P_π [see Eq. (1)] shows that the latter increases with γ.

[75] To keep this example pedagogical, we consider a single attenuator. A more realistic chain of attenuators is addressed in Sec. V.

[76] M. Werninghaus, D. J. Egger, F. Roy, S. Machnes, F. K. Wilhelm, and S. Filipp, Leakage reduction in fast superconducting qubit gates via optimal control, npj Quantum Inf. **7**, 1 (2021).

[77] D. M. Pozar, *Microwave Engineering* (John Wiley & Sons, 2011).

[78] Unlike some works, we take positive decibel values for attenuation. An attenuator with $A_{\mathrm{dB}} = 20$ dB has $A = 100$, corresponding to the output of the attenuator being 100 times smaller than the input.

[79] The signal and its reflection after interaction with the qubit are dissipated in the attenuator, which we take to have $A \gg 1$. Then, it is a reasonable approximation to take the heat dissipation of the attenuator as equal to the injected signal power.

[80] V. Parma, in *Superconductivity for Accelerators*, CERN Yellow Reports: School Proceedings, edited by R. Bailey (CERN, Geneva, 2014), p. 353.

[81] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[82] To obtain p_{err}, we first decompose the noise map $\mathcal{N}$ in the Pauli basis as $\mathcal{N}(\rho) = \sum_{ij} \chi_{ij} \sigma_i \rho \sigma_j$. We work under the Pauli-Twirling approximation and neglect the off-diagonal terms of the noise map; this approximation is well-justified in the context of error correction [126–128].The quantity $p_{\mathrm{err}} \equiv \max_{i>0} \chi_{ii}$ corresponds to the worst-case probability of Pauli error. This yields the expression for p_{err} given

in the main text once we put in the noise model from Sec. III.

[83] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, United Kingdom, 2011).

[84] We take the uncompressed circuit and label subcircuits from left to right (from 1 to $Q-1$). Then, at compression step i, we compress all subcircuits to the left of the subcircuit i and leave uncompressed all subcircuits to the right of subcircuit i. Here, "compress" means moving subcircuits into each other, as in Fig. 4, such that as many gates are performed in parallel as possible. The compression factor is then defined as $\epsilon = (i-1)/(Q-2)$, so $\epsilon = 0$ (meaning $i = 1$) is the uncompressed circuit and $\epsilon = 1$ (meaning $i = Q-1$) is the fully compressed circuit.

[85] J. M. Chow, A. D. Córcoles, J. M. Gambetta, C. Rigetti, B. R. Johnson, J. A. Smolin, J. R. Rozen, G. A. Keefe, M. B. Rothwell, M. B. Ketchen, *et al.*, Simple all-microwave entangling gate for fixed-frequency superconducting qubits, Phys. Rev. Lett. **107**, 080502 (2011).

[86] S. Sheldon, E. Magesan, J. M. Chow, and J. M. Gambetta, Procedure for systematically tuning up cross-talk in the cross-resonance gate, Phys. Rev. A **93**, 060302 (2016).

[87] A. M. Steane, Error correcting codes in quantum theory, Phys. Rev. Lett. **77**, 793 (1996).

[88] D. Gottesman, Ph.D. thesis, California Institute of Technology, 1997.

[89] A. M. Steane, Active Stabilization, Quantum computation, and quantum state synthesis, Phys. Rev. Lett. **78**, 2252 (1997).

[90] P. Aliferis, D. Gottesman, and J. Preskill, Quantum accuracy threshold for concatenated distance-3 codes, Quantum Info. Comput. **6**, 97 (2006).

[91] As in Sec. IV, we attribute all noise in the physical gates and measurements as arising from the noise in the individual qubits on which the gates and measurements are operating. Additional control noise occurs in a realistic device but this can be easily incorporated into our description by regarding p_{err} as the maximum over the physical qubit error probability and the error probability associated with the gate and/or measurement control.

[92] The logical ancillas must be verified as being error free, requiring the verification part of the circuit in Fig. 8. Each logical ancilla has a small chance of having an error, so the code must always prepare a small percentage of extra logical ancillas at each clock cycle, to replace those with errors. This increases the resource consumption associated with ancillas by less than 2% [110]. This is small enough to neglect here and so we simplify the analysis by assuming that all ancillas pass verification.

[93] C. Wang, *et al.*, Towards practical quantum computers transmon qubit with a lifetime approaching 0.5 ms, npj Quantum Inf. **8**, 1 (2022).

[94] J.-S. Park, S. Subramanian, L. Lampert, T. Mladenov, I. Klotchkov, D. J. Kurian, E. Juarez-Hernandez, B. Perez-Esparza, S. R. Kale, K. T.Asma Beevi, S. Premaratne, T. Watson, S. Suzuki, M. Rahman, J. B. Timbadiya, S. Soni, and S. Pellerano, in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 64 (IEEE, San Francisco, USA, 2021), p. 208.

[95] K. Kang, D. Minn, S. Bae, J. Lee, S. Kang, M. Lee, H.-J. Song, and J.-Y. Sim, A 40-nm cryo-CMOS quantum controller IC for superconducting qubit, IEEE J. Solid-State Circuits **57**, 3274 (2022).

[96] When we started this work, it was suggested that 1 mW per qubit was directly achievable [57] but very recent analyses [46,94,95] have accounted for more aspects of signal generation and argued that the current state-of-the-art is 15–30 mW. Nonetheless, we believe that reaching 1 mW per physical qubit is an optimistic but reasonable target given the rapid progress in the field.

[97] Note, however, that P_{FT} diverges for $\mathcal{M}_{\text{FT}} \to 1$ (if this is allowed by the algorithm) because that corresponds to an algorithm that never gives the wrong answer; this would require an infinite amount of error correction and hence would require infinite resources.

[98] A. M. Frolov and D. H. Bailey, Highly accurate evaluation of the few-body auxiliary functions and four-body integrals, J. Phys. B: At. Mol. Opt. Phys. **36**, 1857 (2003).

[99] F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé, and P. Zimmermann, in *Advances in Cryptology—CRYPTO 2020* (Springer, Cham, Switzerland, 2020), p. 62.

[100] JUWELS Module 1 has 114 000 cores with performance close to Xeon Gold 6130 CPUs and is also in the top 100 for energy efficiency [71].

[101] Gasoline has about 0.03 GJ of energy per liter and a typical car tank contains about 60 liters.

[102] J. Edmonds, Paths, trees, and flowers, Can. J. Math. **17**, 449 (1965).

[103] J. Edmonds, Maximum matching and a polyhedron with 0,1-vertices, J. Res. Natl. Bur. Stand. **69B**, 125 (1965).

[104] V. Kolmogorov, Blossom V: A new implementation of a minimum cost perfect matching algorithm, Math. Prog. Comp. **1**, 43 (2009).

[105] D. S. Wang, A. G. Fowler, A. M. Stephens, and L. C. L. Hollenberg, Threshold error rates for the toric and planar codes, Quantum Inf. Comput. **10**, 456 (2010).

[106] A. G. Fowler, Minimum weight perfect matching of fault-tolerant topological quantum error correction in average O(1) parallel time, Quantum Inf. Comput. **15**, 145 (2015).

[107] N. Delfosse and N. H. Nickerson, Almost-linear time decoding algorithm for topological codes, Quantum **5**, 595 (2021).

[108] P. Das, C. A. Pattison, S. Manne, D. Carmean, K. Svore, M. Qureshi, and N. Delfosse, A scalable decoder micro-architecture for fault-tolerant quantum computing, ArXiv:2001.06598 (2020).

[109] Considering a more conservative estimate for the consumption of the electronics generating the signals, i.e., 15–30 mW per physical qubit [46,94,95] (instead of 1 mW per physical qubit as used in our scenario A), the power consumption of the whole computer apart from the classical computer decoding the syndromes would be dominated by these electronics, which should be put at room temperature. The power consumption would then be 300–600 kW, which is less than, or comparable to, that of

a typical supercomputer (their consumption is usually in the range 1–10 MW).

[110] M. Fellous-Asiani, *et al.*, Fault-tolerant magic state preparation is not very costly in concatenated error-correction (provisional title). In preparation.

[111] Assuming that *logical* measurements are rare does not mean that *physical* measurements are rare. Physical measurements on the ancillas are a essential part of the error correction, so N_{meas} will be large even when $N_{\text{meas};L} = 0$.

[112] C. Zalka, Fast versions of Shor's quantum factoring algorithm, ArXiv:quant-ph/9806084 (1998).

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

[114] A. Scherer, B.ît Valiron, S.-C. Mau, S. Alexander, E. van den Berg, and T. E. Chapuran, Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2D target, Quantum Inf. Process. **16**, 60 (2017).

[115] E. Jeffrey, D. Sank, J. Y. Mutus, T. C. White, J. Kelly, R. Barends, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. Megrant, P. J. J. O'Malley, C. Neill, P. Roushan, A. Vainsencher, J. Wenner, A. N. Cleland, and J. M. Martinis, Fast accurate state measurement with superconducting qubits, Phys. Rev. Lett. **112**, 190504 (2014).

[116] E. D. Marquardt, J. P. Le, and R. Radebaugh, in *Cryocoolers 11* (Springer, Boston, Massachusetts, 2002), p. 681.

[117] J. Lawrence, A. B. Patel, and J. G. Brisson, The thermal conductivity of Kapton HN between 0.5 and 5 K, Cryogenics **40**, 203 (2000).

[118] M. Wade, M. Davenport, M. De Cea Falco, P. Bhargava, J. Fini, D. Van Orden, R. Meade, E. Yeung, R. Ram, M. Popović, *et al.*, in *2018 European Conference on Optical Communication (ECOC)* (IEEE, 2018), p. 1.

[119] Reference [60] has an amplifier at room temperature, which corresponds to an amplifier at T_{gen} for us. We assume that its power consumption is included within that of the control electronics at T_{gen}(see Appendix B 3).

[120] L. Le Guevel, G. Billiot, X. Jehl, S. De Franceschi, M. Zurita, Y. Thonnart, M. Vinet, M. Sanquer, R. Maurand, A. G. M. Jansen, and G. Pillonnet, in *2020 IEEE International Solid- State Circuits Conference—(ISSCC)* (IEEE, San Francisco, USA, 2020), p. 306.

[121] Y. Salathé, P. Kurpiers, T. Karg, C. Lang, C. Kraglund Andersen, A. Akin, S. Krinner, C. Eichler, and A. Wallraff, Low-latency digital signal processing for feedback and feedforward in quantum computing and communication, Phys. Rev. Appl. **9**, 034011 (2018).

[122] C.-W. Tung and S.-H. Huang, A high-performance multiply-accumulate unit by integrating additions and accumulations into partial product reduction process, IEEE Access **8**, 87367 (2020).

[123] T. Häner, M. Rötteler, and K. Svore, Factoring using $2n + 2$ qubits with Toffoli based modular multiplication, Quantum Inf. Comput. **17**, 673 (2017).

[124] P. Selinger, Quantum circuits of T-depth one, Phys. Rev. A **87**, 042302 (2013).

[125] Cryogenic electronics and quantum information processing. 2021 update. International roadmap for devices and systems (2021).

[126] M. R. Geller and Z. Zhou, Efficient error models for fault-tolerant architectures and the Pauli twirling approximation, Phys. Rev. A **88**, 012314 (2013).

[127] A. Katabarwa and M. R. Geller, Logical error rate in the Pauli twirling approximation, Sci. Rep. **5**, 1 (2015).

[128] M. Gutiérrez and K. R. Brown, Comparison of a quantum error-correction threshold for exact and approximate errors, Phys. Rev. A **91**, 022335 (2015).