


**Hybrid quantum-classical algorithms in the noisy intermediate-scale quantum era and beyond**Adam Callison <sup>1,\*</sup> and Nicholas Chancellor <sup>2,†</sup><sup>1</sup>*Department of Physics and Astronomy, University College London, London WC1E 6BT, United Kingdom*<sup>2</sup>*Department of Physics; Joint Quantum Centre (JQC) Durham-Newcastle, Durham University, South Road, Durham DH1 3LE, United Kingdom*

(Received 25 March 2022; published 8 July 2022)

Hybrid quantum-classical algorithms are central to much of the current research in quantum computing, particularly when considering the noisy intermediate-scale quantum (NISQ) era, with a number of experimental demonstrations having already been performed. In this perspective, we discuss in a very broad sense what it means for an algorithm to be hybrid quantum-classical. We first explore this concept very directly, by building a definition based on previous work in abstraction-representation theory, arguing that what makes an algorithm hybrid is not directly how it is run (or how many classical resources it consumes), but whether classical components are crucial to an underlying model of the computation. We then take a broader view of this question, reviewing a number of hybrid algorithms and discussing what makes them hybrid, as well as the history of how they emerged and considerations related to hardware. This leads into a natural discussion of what the future holds for these algorithms. To answer this question, we turn to the use of specialized processors in classical computing. The classical trend is not for new technology to completely replace the old, but to augment it. We argue that the evolution of quantum computing is unlikely to be different: Hybrid algorithms are likely here to stay well past the NISQ era and even into full fault tolerance, with the quantum processors augmenting the already powerful classical processors which exist by performing specialized tasks.

DOI: [10.1103/PhysRevA.106.010101](https://doi.org/10.1103/PhysRevA.106.010101)**I. INTRODUCTION**

The first proposals for the creation of computing resources based on the principles of quantum mechanics emerged in the 1980s [1–3]. This led quickly to the discovery, throughout the 1990s, of a number of quantum algorithms for toy problems designed to show a complexity theoretic separation between quantum and classical computation [4–6]. In the mid- to late 1990s, however, Peter Shor developed the polynomial-time quantum algorithm for factoring products of large prime numbers [7,8], now widely regarded as the first *useful* quantum algorithm that achieves a speedup over purely classical counterparts.

The explosion of interest in quantum computing that resulted from Shor’s work is partly responsible for the rapid development thereafter of early experimental quantum hardware that could run small, noisy instances of quantum algorithms (see Refs. [9–22], among others). However, as these experimental platforms with severe limits on computation size and quality emerged, it quickly became apparent that for quantum computers to be useful in the near and mid-term, the community needed to find algorithms tailored to this era of quantum computing (later termed the “noisy intermediate-scale quantum” (NISQ) era [23]). This includes not only algorithms run on gate model quantum devices, but

also *analog* optimization, simulation, and machine learning through quantum annealing [24].

Perhaps the most influential development in the field of gate-model NISQ-suitable quantum algorithms is the framework of variational quantum algorithms (VQAs) that was introduced with the variational quantum eigensolver (VQE) [25]. These algorithms make use of short-depth parametrized quantum circuits, particularly suited to NISQ hardware, embedded in an otherwise classical variational loop, a structure that is manifestly *hybrid quantum-classical*. While some of the other quantum algorithms discovered previously are also hybrid, VQAs offered a clear demonstration on the potential computing power available by coupling quantum and classical resources together. Thus, due to VQAs, hybrid quantum-classical algorithms have been an integral part of quantum algorithms research ever since, including through experimental demonstrations of applying NISQ devices to problems in quantum chemistry [25–27], machine learning [28,29], and combinatorial optimization [30].

Separately from VQAs, the key advance which has allowed for hybrid techniques in *analog* quantum computing is the notion of biased search [31–34], a family of techniques that allow the system to be called as a subroutine within a broader algorithm, aided by previously results [33,35]. These can be used in simple ways such as biasing toward a previously found good solution from a classical greedy search [36], or in more complicated ways such as in genetic algorithms [37].

In this perspective, we take a broad view of hybrid quantum-classical algorithms, going beyond variational algorithms and hybrid quantum annealing. We discuss what

\*a.callison@ucl.ac.uk

†nicholas.chancellor@gmail.com

it means to be a hybrid algorithm, in the broadest sense of being necessarily a combination of quantum and classical computation. We explore this question both in the literal sense of developing a definition and in the more abstract sense of where these algorithms sit within the space of quantum computing and more broadly computing in general.

First, we propose that defining hybrid algorithms should be done in terms not of the computation itself but of the abstract model of the computation, and whether that model requires significant classical computation (computation which does not need a quantum hardware to be performed). We further examine how hybrid algorithms are developed and find that there are many paths, and that this development can be driven by any number of factors, including the capabilities and limitations of the hardware, the discovery of issues with “pure” quantum algorithms, and the identification of bottlenecks in current classical methods.

Furthermore, we consider the long and successful history of specialized processors being used within classical computing and note that while quantum computers can be very powerful and greatly enhance computational capabilities, it is likely that they will play a role similar to many of these other accelerators. Finally, looking to the future, we see no clear advantage in algorithms being “purely” quantum, and therefore argue that hybrid algorithms are likely here to stay, rather than just an artifact of the NISQ era.

We start in Sec. II by discussing how a useful definition can be constructed and how to make the distinction between pure and hybrid quantum algorithms meaningful. Within this discussion, we reference two of the most iconic quantum algorithms: Shor’s algorithm, which we argue is in fact hybrid, and Grover’s algorithm, which we argue is not. In Sec. III, we discuss examples of hybrid quantum-classical algorithms and how they fit into the definition. Next, in Sec. IV, we examine the considerations that must be made when thinking of how hybrid algorithms will be run on real hardware. We then, in Sec. V, examine the bigger picture, in particular, how classical computation makes use of heterogeneous hardware arrangements as specialized accelerators and argue that quantum processors are likely to follow as similar path. Finally, in Sec. VI, we conclude by discussing the future of hybrid algorithms.

## II. WHAT IT MEANS TO BE HYBRID

To begin a meaningful discussion of hybrid quantum-classical algorithms, it is first necessary to establish an understanding of what it means for a quantum algorithm to be considered a hybrid algorithm and what makes hybrid algorithms different from other quantum algorithms. For this task, it is essential to develop a working definition of the term “hybrid” in this context. An agreed upon, meaningful, and useful definition can help facilitate communication and allow a useful grouping of similar algorithms. It is tempting to argue that simple techniques such as “repeat until success” play an important role in almost all quantum algorithms, and therefore almost all quantum algorithms should be considered hybrid. While potentially a valid viewpoint, this approach does not lead to a useful definition: If everything is hybrid, then the term “hybrid quantum algorithm” has little meaning.

Rather than considering to be hybrid any quantum algorithm for which the implementation is supported by classical computing, it seems that the salient feature connecting hybrid quantum-classical algorithms is that the use of classical processing is fundamentally inseparable from the computational model. While we will discuss later in this section what exactly this term means, for now it suffices to think of this as the basic theoretical description of how a system solves problems. As a starting point, therefore, we propose the following definition of a hybrid quantum-classical algorithm:

*Definition.* An algorithm that requires nontrivial amounts of both quantum and classical computational resources to run, and which cannot be sensibly described, even abstractly, without reference to the classical computation.

While the term “nontrivial” in this context is open to interpretation, a sensible interpretation is that the classical computation should go beyond simple repetition and conditioning on measurement outcomes.

Quantum error correction [38] (QEC) provides a simple example of classical computers supporting quantum algorithms, but in which the classical computation is not fundamentally tied into the computational model. In devices with QEC, the quantum computation occurs only on a small part of the device’s Hilbert space, known as the *logical subspace*. The logical subspace is chosen such that almost all physical errors will take the system out of this subspace; that is, errors within the subspace are unlikely. Well chosen measurements then can be performed to detect whether the state has left the logical subspace without destroying the quantum information (more correctly, the error will lead to a superposition over the logical subspace and the error subspaces, but the measurement will force the state to collapse onto one of the subspace, a mechanism referred to as *error digitization*). Classical analysis on these measurement outcomes can then be used to diagnose the error and to determine the quantum operations required to rotate the state back into the logical subspace, in a mechanism known as *decoding*. The high-level structure of typical QEC strategies is shown in Fig. 1, and a thorough review can be found in Ref. [38].

Decoding of quantum error correction can be very intensive in terms of classical computing resources [39]; however, the fundamental model of (noise-free) gate-model quantum computing does not depend on *how* the errors are corrected. Similarly, the classical strategy for decoding does not depend on exactly what quantum algorithm is running, only on the device architecture. We would therefore argue that the fact that classical computing resources are required to protect a fault-tolerant quantum algorithm does not in itself make the algorithm hybrid. A fault-tolerant algorithm can be hybrid in the sense that the quantum component of the algorithm can be run in a fault-tolerant way, but the fact that the algorithm relies on QEC does not in itself make it hybrid.

To approach the question of when a quantum computation is hybrid, it is useful to first look at work which has already been done in understanding when a physical system computes. The abstraction-representation theory [40] of computation was motivated by exactly this question [41]. The key distinction which abstraction-representation theory draws between a physical system which is computing and one which is

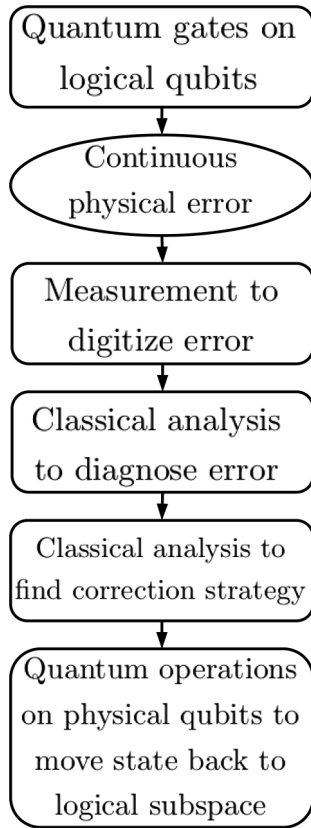


FIG. 1. High-level schematic of typical forms of QEC in which some classical processing is used to protect a quantum algorithm from errors.

not is that a system undergoing computation has information encoded in it and a corresponding abstract model for the computation it is doing. In other words, computation is not about what a system is doing so much as what it is *representing*.

By abstraction-representation theory, a physical system needs to be paired with a (possibly imperfect) abstract model of the computation it is doing. For example, a gate-model quantum computer can be abstractly represented by a quantum circuit (possibly including random elements from noise), a coherent quantum annealer could be represented by the Schrödinger equation, and a dissipative quantum annealer by a master equation.

To understand whether a computation is hybrid or not, it is necessary to think in terms of what the role classical computation plays within this model. Returning to the example of QEC, the model for fault-tolerant quantum computing is already a (noise-free) quantum circuit, and the classical computation is just performing a supporting role in ensuring that the model is accurate enough. For contrast, consider a variational algorithm: In this case, the classical computation is directly involved within the computational model; it gives the updates for the gate parameters and fundamentally cannot be removed from the computational model.

To demonstrate the differences between pure quantum algorithms and hybrid quantum-classical algorithms, we now turn to two widely known algorithms: Grover’s algorithm for searching unsorted databases and Shor’s algorithm for integer

factorization. In what follows, we argue that Grover’s algorithm should not be considered meaningfully hybrid, while Shor’s algorithm should. We discuss other example of hybrid algorithms in more detail in Sec. III.

### A. Grover’s algorithm

Grover’s algorithm [42] is a well-known quantum algorithm for solving the “unstructured search” problem. For the purpose of this section, we limit ourselves to “pure” implementations of Grover’s algorithm, where the entire problem is phrased as an unstructured search; we discuss more sophisticated algorithms based on the same amplitude amplification principle as Grover’s algorithm in Sec. III C. Very simply, this is the problem of finding any of  $M$  marked labels among a set of  $N \geq M$  of labels, given knowledge of the numbers  $M$  and  $N$  access to an oracle that is able to check whether a particular label  $j$  is marked. Many computational problems can be phrased as unstructured search, such as database search and some forms of constraint satisfaction. Without access to additional structure among the labels, the only option for solving unstructured search classically is to repeatedly select labels, either randomly or in a particular order, which has a time complexity that scales as  $O(\frac{N}{M})$ .

The labels  $j$  are mapped to computational basis states  $|j\rangle$  of a Hilbert space of size  $N$ , and we assume *coherent* access to the oracle; that is, it is possible to apply a unitary operator  $O_{\text{check}}$  defined by

$$O_{\text{check}}|j\rangle = \begin{cases} -|j\rangle & \text{for marked } j \\ |j\rangle & \text{for unmarked } j. \end{cases} \quad (1)$$

It can be shown that by initializing the quantum computer in the uniform superposition state  $|s\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^N |j\rangle$  and applying  $r$  iterations of the Grover operator

$$Q_{\text{Grover}} = (2|s\rangle\langle s| - \mathbb{1})O_{\text{check}}, \quad (2)$$

a state is produced that can be measured in the computational basis to return a marked state with probability

$$P_{\text{marked}}(r) = \sin^2((2r+1)\theta), \quad (3)$$

where  $\theta = \sqrt{\frac{M}{N}}$ . Thus, by choosing

$$r = \text{Round}\left(\frac{\pi}{4}\sqrt{\frac{N}{M}} - \frac{1}{2}\right) \in O\left(\sqrt{\frac{N}{M}}\right), \quad (4)$$

a marked state can be found with a probability that quickly approaches unity for  $M \ll N$ . This represents a quadratic speedup over what is possible classically.

While single run of the algorithm does not guarantee a marked state with unit probability, the output can be checked via the oracle and the algorithm repeated a small number of times until a marked state is confirmed, without affecting the quadratic speedup. It would not be useful to consider a quantum algorithm with this trivial level of classical postprocessing to be hybrid, as it is difficult to imagine any useful quantum algorithms which did not contain at least these elements of classical control.

## Algorithm 1. Structure of Shor's algorithm

- 
- 1: Randomly select  $a$  from  $1 < a < N$
  - 2: Compute  $K = \gcd(a, N)$  **classically**, via Euclidean algorithm
  - 3: If  $K \neq 1$ ,  $K$  is a factor of  $N$ , so return  $a$  and exit. Otherwise, continue to step **I**.
  - 4: Implement the  $a$ -dependent unitary  $U_a$  that acts as  $U_a|x\rangle_n = |a^x \bmod N\rangle$ .
  - 5: Perform **quantum** phase estimation (QPE), via a circuit like the one shown in Fig. 2, with a  $q$ -qubit control register where  $N^2 < 2^q < 2N^2$ , on  $U_a$  with the initial state of the target register set to  $|1\rangle$  (the binary representation of 1), to produce an integer  $y$ .
  - 6: Use **classical** continued-fraction expansion to produce an irreducible fraction approximation  $\frac{d}{r}$  of  $\frac{y}{2^q}$  for which  $r < N$  and the approximation error is small ( $< \frac{1}{2^{q+1}}$ ).
  - 7: If  $r$  is odd or it is true that  $a^{\frac{r}{2}} = -1 \bmod N$ , go back to step **I**. Otherwise  $r$  is highly likely to be the period of the modular exponentiation with base  $a$ ,  $f(x) = a^x \bmod N$ , so continue to step **I**.
  - 8: If  $r$  is the period of the modular exponentiation (which can be checked **classically**), continue to step **I**. Otherwise, go back to step **I**, and find a different irreducible fraction with a different  $r$ . If out of candidates, go back to step **I**.
  - 9: Compute  $\gcd(a^{\frac{r}{2}} - 1, N)$  and  $\gcd(a^{\frac{r}{2}} + 1, N)$  **classically**, via the Euclidean algorithm. These are now guaranteed to be nontrivial factors of  $N$ , so return these and exit.
- 

## B. Shor's algorithm

Shor's polynomial-time algorithm [8] for factoring products  $N = p_1 p_2$  of two large prime numbers  $p_1$  and  $p_2$ , a feat thought to be impossible classically, is regarded as the first useful quantum algorithm to have been invented. Given the risk a practical implementation of Shor's algorithm would have on much of modern computer security, it is perhaps the most high-profile quantum algorithm and is seen as a quintessential example of the power of quantum computing.

However, Shor's algorithm is far from being a *purely* quantum algorithm and should certainly be considered a hybrid algorithm. The algorithm relies on a polynomial-time reduction of the factoring problem to the problem of finding the order  $r$  of a periodic function and is described in Algorithm **I**. It can be seen that almost all of the steps are entirely classical, including the calculation of greatest common divisors via the classical Euclidean algorithm in steps **I** and **I** and the classical continued-fraction expansion in step **I**. The only part of the algorithm that is quantum is the call to the quantum phase estimation circuit (see Fig. 2) in step **I**, which forms the core of the order-finding subroutine.

While it is absolutely true that the quantum part of Shor's algorithm provides the speedup over the best known purely classical methods, a significant (but polynomial) amount of classical processing is done before and after the call to the quantum subroutine in step **I** in order to transform the problem into a specific form that the quantum processor can handle efficiently. This classical processing includes calls to identifiable classical algorithms such as the Euclidean algorithm, In other words, what is being calculated by the quantum computer is not directly useful without the surrounding classical steps.

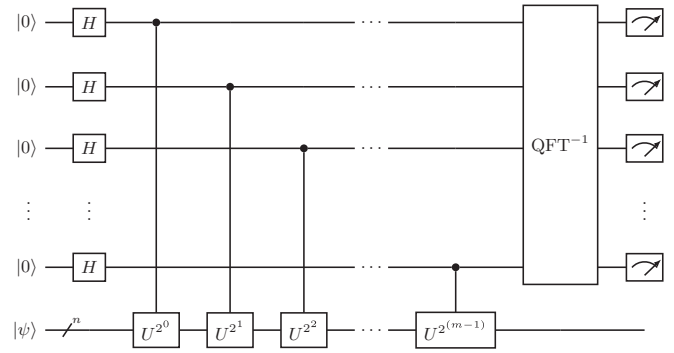


FIG. 2. Phase estimation circuit.

Having discussed two of the possibly most iconic quantum algorithms, in the next section we get into a more detailed discussion about other important examples of hybrid algorithms. This is not intended to be an exhaustive list, but to start a discussion on what it means to be hybrid under different contexts, and to provide examples that can help extend these to other algorithms.

## III. ALGORITHM EXAMPLES

## A. Quantum annealing and continuous-time quantum computing

Quantum annealing (QA) consists of using continuous-time dynamics of a quantum system to solve an optimization problem by searching the solution space through quantum fluctuations. QA as we know it today was proposed in the late 1990s by Kadowaki and Nishimori [24] (although a similar suggestion for chemistry-related problems had been made earlier [43]). After the development of the basic principles of QA, Farhi *et al.* introduced the specific case of adiabatic quantum computing (AQC) [44] in which the adiabatic theorem of quantum mechanics guarantees a solution if the algorithm is run long enough. Roland and Cerf then showed that AQC can attain the same speedup as the gate model Grover's algorithm on unstructured search [45], while around the same time Santoro *et al.* further developed the theory of QA and showed that it can in some cases outperform classical annealing algorithms [46]. For a complete review of AQC, see Ref. [47], and for a more application-focused review of QA, see Ref. [48]. For a forward-looking perspective on techniques which operate outside of the adiabatic limit, see Ref. [49].

Since its inception, QA has grown to encompass a variety of protocols, but the typical structure involves implementing a time-dependent transverse Ising model Hamiltonian

$$H(t) = -A\left(\frac{t}{T}\right) \sum_{j=1}^n X_j + B\left(\frac{t}{T}\right) H_P, \quad (5)$$

where  $T$  is the total annealing time,  $A$  and  $B$  are monotonic control functions such  $A(0) \gg B(0)$  and  $A(1) \ll B(1)$ , and the problem Hamiltonian  $H_P$  is of the form

$$H_P = - \sum_{j=1}^{n-1} \sum_{k=j+1}^n J_{jk} Z_j Z_k - \sum_{j=1}^n h_j Z_j, \quad (6)$$

where the couplings  $J_{jk}$  and fields  $h_j$  define the problem of interest such that lower energy eigenstates of  $H_P$  correspond to better quality solutions. In this way, QA is qualitatively similar to its classical analog *simulated* annealing, but with the physical quantum fluctuations and/or tunneling induced by the transverse-field driver term  $-\sum_{j=1}^n X_j$  playing the role of a simulated temperature. QA protocols typically start the system in the ground state of the transverse-field driver term and aim to set the controls functions  $A$  and  $B$  such that the system is driven toward low-energy states of the problem Hamiltonian.

### 1. Hybrid forms of quantum annealing

Much of the work on hybrid algorithms in continuous-time quantum computing has focused on protocols which allow an educated guess at the solution to be incorporated into the protocol.

The idea of incorporating a heuristic guess was first proposed by Perdomo-Ortiz *et al.* [31], in a scheme which has come to be known as (one variant of) *reverse* annealing. In this scheme, the standard QA Hamiltonian of Eq. (5) is replaced with

$$H_{\text{rev},1}(t) = \left[1 - s\left(\frac{t}{T}\right)\right]H_{\text{init}} + \text{hat}\left[s\left(\frac{t}{T}\right)\right]\left(-\sum_{j=1}^n X_j\right) + s\left(\frac{t}{T}\right)H_P, \quad (7)$$

where  $s$  is a *schedule* function that starts at  $s(0) = 0$ , ends at  $s(1) = 1$ , and monotonically increases in between. The hat function premultiplying the transverse field driver is a function that begins and ends at zero, but is positive in between, such as  $\text{hat}(s) = \sin(\pi s)$ ; that is, the transverse field driver is turned on and then off during the anneal. The completely new ingredient is the *initial* Hamiltonian

$$H_{\text{init}} \equiv -\sum_{j=1}^n g_j Z_j, \quad (8)$$

where  $g_j$  is a sign such that  $g_j = 1$  ( $g_j = -1$ ) if the  $j$ th bit of the heuristic guess is 0 (1). The ground state of the initial Hamiltonian  $H_{\text{init}}$  is then the guess state, and the annealing protocol aims to transfer a nontrivial amount of the amplitude into the true ground-state of the problem Hamiltonian  $H_P$ , with the transverse field turned up and then down in order to assist this transfer via quantum fluctuations. It is worth noting that since the aim of a reverse annealing run is to improve upon a previously known reasonable solution, it is possible to embed reverse annealing in a classical loop, whereby the output state measured one run is then incorporated as the initial state (with the appropriate initial Hamiltonian) for the next run. This hybrid structure was suggested in Ref. [31], and schemes of this sort have come to be known as *iterated* reverse annealing.

Other protocols have been proposed that have also come to be known as reverse annealing. For example, Chancellor [33] and Yamashiro *et al.* [50] propose using the standard QA Hamiltonian of Eq. (5), but simply initializing to the guess state and using a schedule that begins entirely in the problem Hamiltonian  $H_P$  with  $A(0) = 0, B(0) = 1$ , anneals to some

intermediate values  $0 < A(t_{\text{intermediate}}/T), B(t_{\text{intermediate}}/T) < 1$ , and then back to  $A(1) = 0, B(1) = 1$ . These protocols often include pausing the schedule for some time at the intermediate values [51]. In Ref. [50], this two-term form of reverse annealing is embedded in a classical loop and is also called *iterated* reverse annealing (in fact, this is where the term was coined).

A third form of reverse annealing, proposed by Ohkuwa *et al.* [52], uses a Hamiltonian of the form

$$H_{\text{rev},2}(t) = \left[1 - s\left(\frac{t}{T}\right)\right]\left[1 - \lambda\left(\frac{t}{T}\right)\right]H_{\text{init}} + \left[1 - s\left(\frac{t}{T}\right)\right]\lambda\left(\frac{t}{T}\right)\left(-\sum_{j=1}^n X_j\right) + s\left(\frac{t}{T}\right)H_P, \quad (9)$$

where  $s(0), \lambda(0) = 0, s(1), \lambda(1) = 1$ , and both are monotonically increasing between. Recognizing that the system begins in the instantaneous ground state at  $t = 0$  and is targeting the instantaneous ground state at  $t = T$ , this protocol can in principle be run adiabatically and has come to be known as *adiabatic* reverse annealing. This Hamiltonian in Eq. (9) is quite similar to that described by Eq. (7) [in principle, the Eq. (7) protocol could also be run adiabatically], but the additional freedom that arises from having two schedule functions,  $s$  and  $\lambda$ , offers additional flexibility for the annealing path.

An independent proposal that incorporates a heuristic guess by using a biased driver Hamiltonian, instead of by reverse annealing, was given by Duan *et al.* [32] (later studies of this protocol can be found in Ref. [34] and to a lesser extent in Ref. [53]). An example of such a protocol is to modify the transverse-field driver term to include a bias toward a particular guess state; that is,

$$-\sum_{j=1}^n X_j \rightarrow -\sum_{j=1}^n (X_j + g_j b_j Z_j), \quad (10)$$

where  $b_j$  is a positive number representing the strength of the bias and  $g_j = 1$  ( $g_j = -1$ ) if the  $j$ th bit of the guess is 0 (1). The initial state should also be biased toward the guess state, such that it remains the ground state of the biased driver.

Graß [34] showed numerically that biasing both toward and away from solution candidates can improve performance of annealing protocols, and has also experimentally demonstrated that biased-driver QA can work in practice [54] (although technical limitations meant the bias had to be included as part of the problem Hamiltonian instead). Callison *et al.* [53] demonstrated that traditionally formulated QA is subject to an energy redistribution mechanism if certain conditions are met, and the conditions are met in the biased-driver protocols proposed of Refs. [32,34] (but not in the reverse annealing protocols).

Since both reverse QA and biased-driver QA fundamentally incorporate classical information, in the form of previous knowledge of the solution, into the protocol itself, any algorithm which makes use of them is necessarily hybrid. Many of the earlier papers focused on how closed system quantum annealing could incorporate prior knowledge, but later work explored how these ideas can be used in more sophisticated algorithms that use QA as a subroutine. An influential example is the work by Chancellor [33], in which it was proposed how

a known experimental protocol (see Ref. [55]) on a dissipative quantum annealer could be used within analogues of classical population annealing [56,57] and parallel tempering [58,59] algorithms. Furthermore, Chancellor [35] proposed a general formalism to understand annealing protocols that accept initial conditions, such as reverse anneals. Within a week of the first appearance of this paper on the arXiv preprint repository, Neven independently proposed a QA-based parallel tempering algorithm in the AQC 2016 opening remarks [60]. After this, D-Wave Systems Inc. made the necessary controls to perform a reverse annealing protocol (and, in fact, coined the term “reverse annealing” [61]) using a method similar to the proposal in Ref. [33]. The D-Wave implementation of reverse annealing most closely resembles the two-term proposals of Ref. [33,50], except that it operates in the dissipative regime.

The capability to perform reverse annealing has led to numerous experimental studies. Venturelli and Kondratyev [36] demonstrated that incorporating the result of a greedy search can improve the performance of a quantum annealer on a portfolio optimization problem. Ottaviani and Amendola [62] as well as Golden and O’Malley [63] demonstrated that reverse annealing can aid in matrix factorization. Chancellor [64] demonstrated how these techniques can be combined with other advanced controls to trade off between optimality and flexibility within solutions. Chancellor and Kendon [65] demonstrated that the underlying principle behind using reverse annealing for local search was correct and examined the effect of noise levels.

Data from an experimental demonstration of using reverse annealing for local searching, first presented in Ref. [65], are shown in Fig. 3. These experiments were performed by constructing an Ising system with known pathological behavior for QA. This was achieved by creating a “broad” false minimum where strong quantum fluctuations were allowed. The system was then initialized near (in the sense of number of bit flips) a much “narrower” true minimum (see the cartoon inset of Fig. 3). The results demonstrate that searching the solution locally by starting from a good guess and accessing only moderate fluctuations through reverse annealing allowed the broad false minimum to be avoided, thereby validating reverse annealing as a tool for search solution spaces locally, a key ingredient for hybrid QA algorithms.

A clear demonstration of a hybrid quantum-classical algorithm employing reverse annealing as a quantum (although not fully coherent) subroutine was provided by King *et al.* [37]. In that work, reverse annealing is used as the mutation operator in an otherwise classical genetic algorithm. The algorithm is shown in Algorithm II.

Perhaps most surprisingly, reverse annealing has aided not just optimization but also quantum simulation using quantum annealers, in particular, demonstrations that quantum annealers can be used to simulate geometrically frustrated spin systems. The quantum simulations by King *et al.* [66,67] are only feasible through use of the reverse annealing feature to initialize the system in a previously found configuration. This is of particular interest since the work in Ref. [67] shows potential evidence of a scaling advantage over classical techniques based on path-integral quantum Monte Carlo.

There has also been significant numerical and theoretical work on understanding reverse annealing. Ohkuwa *et al.*

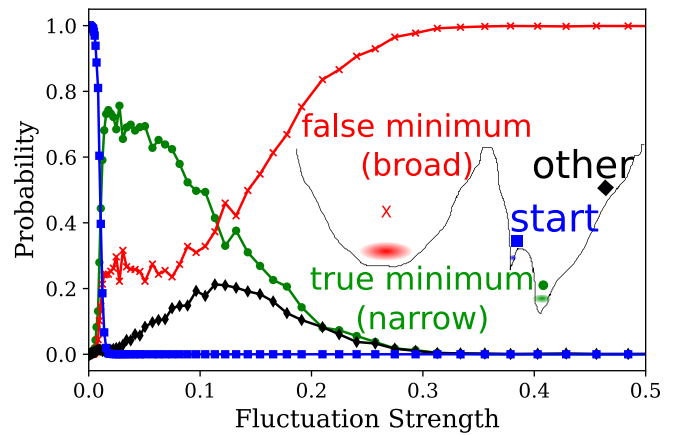


FIG. 3. Experimental demonstration of dissipative reverse annealing with a cartoon representation of an engineered energy landscape in the inset. Biasing toward a true solution leads to the problem being solved with a high probability, while traditional forward annealing (approximated in the limit of large fluctuation strength) does not solve the problem. Data plot reproduced from Ref. [65]; cartoon illustration is original. The colors (shades of gray in print) and symbols in the cartoon reproduce those in the figure. The cross symbol indicates probability to be found in the false minimum, while the circles represent the probability to be found in the true minimum. The squares indicate probability to be found in the original starting state, and diamonds indicate probability to be found in all states not previously mentioned.

[52] showed that reverse annealing techniques could mitigate or even remove a first-order phase transition in highly symmetric problems, though the analysis assumes a coherent rather than dissipative setting. In a similar vein, Yamashiro *et al.* [50] showed that a more coherent version of reverse annealing works better than dissipative reverse annealing for solving highly symmetric problems, and discussed how protocols similar to those proposed here could be experimentally implemented with an existing feature on D-Wave

Algorithm 2. Quantum-assisted genetic algorithm of Ref. [37]

---

```

1: classically initialize the population as  $N$  random states
2: FOR generation IN generations:
3:   FOR individual IN population:
4:     with probability  $\text{mutation\_rate}$ , use quantum reverse
       annealing, starting from individual to create
       mutated_individual
5:     add mutated_individual to population
6:   END FOR
7:   randomly match pairs of individuals to create list pairs (of
       size  $\text{recombination\_rate} \times |\text{population}|$ )
8:   FOR pair IN pairs:
9:     classically combine pair to create offspring
10:    add offspring to population
11:  END FOR
12:  exit loop if a stopping condition is met
13:  discard old individuals from population to maintain desired
       size  $N$ 
14: END FOR

```

---

devices. Furthermore, in a pair of papers considering the  $p$ -spin model as the problem Hamiltonian, Passarelli *et al.* show that open quantum system effects can enhance the performance of *iterated* reverse annealing [68] [using a Hamiltonian of the form of Eq. (5) with a nonmonotonic schedule], but could be detrimental in the case of *adiabatic* quantum annealing [69] [using a Hamiltonian of the form of Eq. (9)].

Overall, reverse annealing has been identified as a key technique to eventually realizing a quantum advantage through annealing techniques, especially in settings where coherence is low. In fact, in a recent forward-looking perspective, Crosson and Lidar [49] only rank QA in a moderately decoherent setting as promising if reverse annealing techniques are used.

Moving away from strategies that incorporate prior knowledge into the anneal, there are a number of other approaches to hybridizing QA, many of which involve optimizing QA schedules to increase success probability without requiring the runtime to grow toward the adiabatic limit, and fall under the umbrella of *shortcuts to adiabaticity* [70]. For example, in Ref. [71], a parameterized counterdiabatic term is introduced to the Hamiltonian to mitigate harmful aspects of the diabatic evolution, and its parameters are variationally optimized in a classical loop, similar to variational algorithms in the gate model (see Subsec. III B). Related work is presented in Refs. [72,73].

An alternative approach to hybridizing QA is to use classical machine-learning techniques to design annealing schedules for particular problems or problem types. For example, Chen *et al.* [74] show that a combination of tree-search algorithms and neural networks can be used to learn effective annealing schedules that appear to have the same effect as deliberately including counterdiabatic driving terms and which can be transferred between instances. A genetic algorithm is used for similar purposes by Hegde *et al.* in Ref. [75].

## B. Variational algorithms

### 1. Variational quantum eigensolver

The NISQ era of quantum computing is characterized by devices with limited spatial (e.g., number of qubits) and temporal (e.g., achievable coherence time) resources. As such, near-term quantum algorithm design must focus on extracting as much performance as possible out of these limited resources. Hybrid algorithms offer a natural way to do this, by performing much of the computational effort on classical hardware and calling the quantum processor only for small subproblems to which it is particularly well suited. A well-known and illustrative example is the variational quantum eigensolver (VQE) [25], a class of algorithms intended for NISQ devices that combines small quantum circuits and classical optimization techniques to approximate eigenstates of a Hamiltonian  $\mathcal{H}$ , typically ground states, and associated eigenenergies. Comprehensive overviews of VQEs and related variational algorithms are available elsewhere [76,77], but we give a brief description here to illustrate a key example of a hybrid algorithm.

In VQE, a quantum processor is used to prepare a state

$$|\Psi(\theta)\rangle = U(\theta)|\Psi_{\text{init}}\rangle, \quad (11)$$

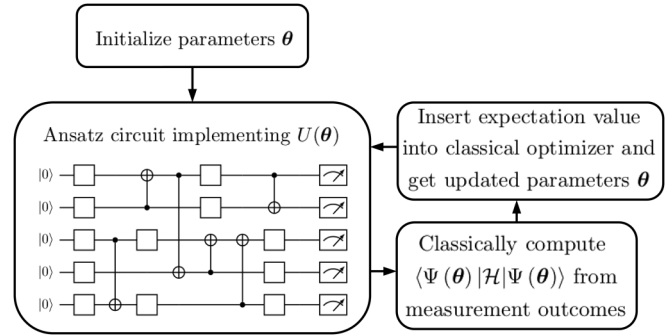


FIG. 4. Schematic showing a high-level overview of the structure of VQE algorithms.

where  $|\Psi_{\text{init}}\rangle$  is an easily prepared initial state. The unitary operator  $U(\theta)$  is the action of a parameterized quantum circuit (PQC); that is, a circuit with a fixed form (a carefully chosen ansatz) but a dependence on a set of parameters  $\theta$ . For Hamiltonians  $\mathcal{H}$  which can be expressed as sums over polynomially many products of Pauli operators, the state  $|\Psi(\theta)\rangle$  can be repeatedly prepared and measured  $O(1/\epsilon^2)$  times to efficiently estimate the expectation value  $\mathbb{E}_{\Psi, \mathcal{H}}(\theta) = \langle \Psi(\theta) | \mathcal{H} | \Psi(\theta) \rangle$  to some desired precision  $\epsilon$ . The variational principle gives the inequality

$$\langle \Psi(\theta) | \mathcal{H} | \Psi(\theta) \rangle \geq E_0^{(\mathcal{H})}, \quad (12)$$

where  $E_0^{(\mathcal{H})}$  is the ground-state energy of the Hamiltonian  $\mathcal{H}$ , and Eq. (12) becomes an equality only when  $|\Psi(\theta)\rangle$  is a ground state of the Hamiltonian  $\mathcal{H}$ . Thus, if the ansatz  $U(\theta)$  is sufficiently expressive, minimizing the expectation value  $\mathbb{E}_{\Psi, \mathcal{H}}(\theta)$  over the parameters  $\theta$  will yield a good approximation to the ground state  $|\Psi_0^{(\mathcal{H})}\rangle$ . The minimization can be performed by inserting the quantum processor into a feedback loop that incorporates a classical optimization algorithm; the original proposal [25] used the Nelder-Mead simplex method [78], but more recent work has used other optimization methods (see the reviews [79,80] for examples). Figure 4 shows a schematic of the VQE algorithm.

### 2. Quantum approximate optimization algorithm

The quantum approximate optimization algorithm (QAOA) is an algorithm for combinatorial optimization first proposed in Ref. [81] and which quickly inspired a number of others works developing the idea further [82–87]. While some forms of QAOA may be considered to be purely quantum algorithms [81], it is typically hybrid in form. In fact, QAOA in its most common form can be considered to be an example of a VQE algorithm applied to combinatorial optimization, with an ansatz inspired by QA (see Subsec. III A). The ansatz usually takes the form

$$|\Psi(\alpha, \beta)\rangle_n = \prod_{j=1}^p [\exp(-i\alpha_j H_D) \times \exp(-i\beta_j H_P)] |\text{gs}(H_D)\rangle_n \quad (13)$$

where  $p$  is the number of QAOA layers and  $\alpha = (\alpha_1, \dots, \alpha_p)$ ,  $\beta = (\beta_1, \dots, \beta_p)$  are  $2p$  variational parameters. The classical problem Hamiltonian  $H_P$  is a Hamiltonian into which,

as in QA, a combinatorial optimization cost function has been mapped such that its eigenstates correspond to candidate solutions and its eigenvalues correspond to their costs. The driver or mixing Hamiltonian  $H_D$  is a Hamiltonian which induces transitions between basis states. As in QA, this is typically a simple single-body transverse field Hamiltonian  $H_D = -\sum_{k=1}^n X_k$  applied to the qubits, although this has been generalized [87] to include a much broader class of mixing Hamiltonians, in a similar algorithm known as the quantum alternating operator ansatz, sharing an acronym with the original algorithm. In a typical QAOA implementation, the initial state  $|\text{gs}(H_D)\rangle$  is the ground state of the mixer Hamiltonian  $H_D$ .

The ansatz in Eq. (13) is inspired by the QA algorithm in the sense that the parameters  $\alpha, \beta$  can be chosen such that the unitary that prepares the ansatz can be expressed as a Trotterization of a QA protocol; that is,

$$\begin{aligned} & \lim_{p \rightarrow \infty} \left\{ \prod_{j=1}^p \exp \left( -i \left[ 1 - s \left( \frac{jT}{p} \right) \right] H_D \right) \right. \\ & \quad \times \left. \exp \left[ -is \left( \frac{jT}{p} \right) H_P \right] \right\} \\ & = \mathcal{T} \exp \left[ \int_{t=0}^{t=T} dt ((1 - s(t))H_D + s(t)H_P) \right]. \quad (14) \end{aligned}$$

In practice, running a QAOA algorithm with a large number  $p$  of layers in order to approach the QA limit requires a large circuit depth that quickly becomes impractical, particularly in the NISQ era. Instead, QAOA algorithms are typically considered at a small number of layers, even sometimes  $p = 1$  (e.g., in Ref. [88]). For a small number of layers, there are a manageable number of parameters  $\alpha, \beta$  which can be optimized in a classical variational loop as with other variational algorithms, as described in Sec. III B 1. Other strategies for optimizing the parameters have also been considered, such as posing the optimization as a learning and task and employing reinforcement learning [89] to find good parameters that can even be transferred between problem instances of different sizes.

### 3. Difficulties with variational algorithms

The performance of a variational algorithm, including QAOA, will depend strictly on the choice of ansatz for the PQC  $U(\theta)$ . Most critically, if the ansatz is too simple, the set of accessible states  $\{|\Psi(\theta)\rangle : \forall \theta\}$  may exclude a large portion of the physically relevant state space, meaning that it is not possible to achieve a good approximation to the target state  $|\Psi_0^{(T)}\rangle$ . This can be avoided by increasing the number of independent parameters  $\theta$  in the ansatz, but this comes at the cost of increased circuit depth, which for NISQ devices will lead to increased noise, again limiting the quality of achievable solutions.

In principle, matching the form of the ansatz to the problem under consideration can also improve VQE performance;

for example, the unitary coupled cluster ansatz is known to be particularly suitable for problems in quantum chemistry [25,90,91]. However, such domain-specific ansatz often require large circuit depths to implement in terms of gates native to quantum processors. An alternative approach is to employ hardware-efficient ansatz, which are designed to be easy to implement on real hardware [25,92]. Unfortunately, the hardware-efficient approach has been observed to lead to “barren plateaus,” regions of the optimization landscape with vanishing gradients, making the classical optimization much harder to perform. A useful discussion of ansatz design is available in Ref. [93].

An understanding is emerging that the origin of the barren plateaus relates to the expressivity of the ansatz [94]. In particular, if an ansatz is too expressive, then the broadness of the search of the space leads to barren plateaus since the majority of the solution space will consist of low-quality solutions which are relatively unaffected by small changes. This leads to tension between having an ansatz which is expressive enough to accurately describe a desired solution, but not so expressive that it is no longer trainable. It has also been found that there is a distinct mechanism by which noise in the circuit can lead to barren plateaus [95]. The fact that barren plateaus cause gradients to vanish begs the question whether they will be problematic if optimization of parameters is performed with a gradient-free method; unfortunately, recent work suggests that they are still an impediment [96]. While barren plateaus are a significant obstacle to variational algorithms, the situation is far from hopeless, and there are strategies which have shown some success. For example, it has been demonstrated in Ref. [97] that reducing the size of the search space by correlating circuit parameters can lead to large gradients (therefore destroying the plateaus). While many error-mitigation strategies do not improve the trainability of noisy variational algorithms, some can [98]. Furthermore, progress may be made by choosing ansatz or starting conditions which are inspired by the problem being solved. For example, it is known that QAOA can describe digitized QA, and indeed optimal QAOA protocols have been found to be very similar (but not identical) to digitized annealing [99,100] and sophisticated mathematical tools have been developed to understand when an ansatz will and will not be trainable due to barren plateaus [101].

A related but distinct question is how to encode information which is not naturally binary into a quantum computer. In particular, this raises the question of binary encodings, where bit strings are used directly to represent configurations, or encoding individual variables using unary (qubits required scale linearly with the number of configurations) encodings. In the case of annealing, the inability to efficiently engineer higher order terms means that it can be shown mathematically that a kind of unary encoding (domain wall) is the most efficient method in terms of qubit usage for completely general interactions [102]. However, for gate model approaches, the tradeoff is far more complicated and encoding strategies remain an area of active research [103–106] with many open questions. For example, to the best of our knowledge only Plewa *et al.* [106] have performed simulations to test the domain-wall encoding in a gate model setting.



### C. Quantum amplitude estimation

Quantum amplitude estimation (QAE) algorithms are collection of quantum methods to solve the following problem. Given black-box access to some unitary quantum algorithm  $A$  on  $n$  qubits a unitary oracle  $O$  that partitions the computational basis states into a set of good states  $G$  and bad states  $B$ ,

$$O|j\rangle = \begin{cases} |j\rangle, & \text{for } j \in G \\ -|j\rangle, & \text{for } j \notin G, \end{cases} \quad (15)$$

find the probability (referred to as the *amplitude* in this context)  $a$  that a computational basis measurement on the output of the algorithm  $|\psi\rangle = A|\psi_0\rangle$ , where  $|\psi_0\rangle$  is an easily prepared reference state (typically the all-zero state  $|0\rangle_n$ ), produces a good state.

Given no other information about the structure of the algorithm  $A$  or the good-bad partition, the only option to solve this problem classically (aside from the unavoidable quantum run of the algorithm  $A$  and the call to the oracle  $O$  to check the state) is to simply run the algorithm  $A$ , measure the output state, check if it is good, and repeat to build up statistics. To achieve an estimate  $\tilde{a}$  of the amplitude  $a$  to within an error  $\varepsilon$ , this approach requires a number of calls  $N_{\text{calls}}^{(\text{sampling})}$  to the algorithm  $A$  that scales as

$$N_{\text{calls}}^{(\text{sampling})} \in O\left(\frac{1}{\varepsilon^2}\right). \quad (16)$$

The first QAE algorithm was proposed in Ref. [107] and achieves a quadratic speed-up over classical sampling; that is, the required number of calls  $N_{\text{calls}}^{(\text{QAE})}$  to the algorithm  $A$  to achieve an estimate  $\tilde{a}$  of the amplitude  $a$  to within an error  $\varepsilon$  scales as

$$N_{\text{calls}}^{(\text{QAE})} \in O\left(\frac{1}{\varepsilon}\right). \quad (17)$$

It was later shown that practical implementations of a QAE algorithm would have the potential to speed up a wide range of important applications; in particular, it was shown in Ref. [108] that QAE can be used to achieve a quadratic (up to logarithmic factors) speedup for Monte Carlo algorithms by enhancing the sampling convergence rate. Given the importance of Monte Carlo algorithms in a broad range of areas, from chemistry [109] to statistical physics [110] to computational finance [111], practical implementations of QAE algorithms would have a significant impact.

The QAE algorithm of Ref. [107] is a purely quantum algorithm, relying on the phase-estimation procedure similar to that shown in Fig. 2, and involves many complicated controlled unitary operations that may be difficult to implement. After the work in Ref. [108] showed potential uses of QAE, many new forms of QAE were developed [112–115] that replaced the phase-estimation procedure with classical post-processing, but the core workings of the algorithm remained the same. All forms of QAE involve constructing Grover-like iteration operators [42]

$$Q = (A(\mathbb{1} - 2|0\rangle_n\langle 0|_n)A^\dagger)O, \quad (18)$$

similar to Eq. (2). The algorithms work by repeating the Grover iteration operator different numbers of times to amplify the amplitude  $a$  by different amounts and detecting

which numbers of iterations produce a large probability of measuring good states. The original algorithm of Ref. [107] uses phase estimation to do this, while more recent QAE algorithms make use of classical postprocessing instead.

An illustrative example of a hybrid QAE algorithm is the maximum-likelihood QAE (MLQAE), proposed by Suzuki *et al.* [113]. In MLQAE, Grover-type circuits are run at various different numbers  $m$  of iterations to produce the states

$$|\psi^{(m)}\rangle \equiv (Q^m)A|\psi_0\rangle \quad (19)$$

$$\begin{aligned} &= \sin((2m+1)\theta_a) \frac{\Pi_G|\psi_0\rangle}{\|\Pi_G|\psi_0\rangle\|} \\ &\quad + \cos((2m+1)\theta_a) \frac{\Pi_B|\psi_0\rangle}{\|\Pi_B|\psi_0\rangle\|}, \end{aligned} \quad (20)$$

where the angle  $\theta_a$  is defined by  $\sin^2 \theta_a \equiv a$  and  $\Pi_G$  ( $\Pi_B$ ) is a projection operator on to the good (bad) subspace of the computational basis. Then, these output states are measured, and it is recorded whether the outcome is good or bad. This process is repeated for some number  $N_{\text{shot}}$  of shots, and a measurement record  $\mathbf{h} = (h_m)_m$  is constructed, where  $h_m$  is the number of good states produced by  $N_{\text{shot}}$  shots of the circuit with  $m$  Grover iterations.

A likelihood function  $L(\theta_{\tilde{a}}; \mathbf{h})$  for the measurement record is then built,

$$L(\theta_{\tilde{a}}; \mathbf{h}) = \prod_m L_m(\theta_{\tilde{a}}; h_m) \quad (21)$$

$$\begin{aligned} &= \prod_m (\sin^2((2m+1)\theta_{\tilde{a}}))^{h_m} \\ &\quad \times (\cos^2((2m+1)\theta_{\tilde{a}}))^{(N_{\text{shot}}-h_m)}. \end{aligned} \quad (22)$$

By classically maximizing this likelihood function (or its logarithm), the angle  $\theta_a$ , most likely to produce the measurement record can be determined, and hence an estimate  $\tilde{a}$  for amplitude  $a$  can be found. This is illustrated in Fig. 5. Figure 5(a) shows an illustration of the likelihood  $L_m(\theta_{\tilde{a}}; h_m)$  of a Grover-like QAE circuit with  $m$  Grover iterations ( $m$  larger for lower plots) to produce  $h_m$  good states from  $N_{\text{shot}}$  runs, with the angle  $\theta_{\tilde{a}}$ . It can be seen that smaller numbers  $m$  of iterations produce a small number of broad peaks (a small number of rough regions where the true  $\theta_a$  might be), while larger  $m$  values produce many sharp peaks (which can help identify an accurate estimate if the rough region is known). The overall likelihood function, illustrated in Fig. 5(b), can be maximized to find an accurate estimate of the angle  $\theta_a$  (and hence the amplitude  $a$ ). It can be shown [113] that for some sequences of choices of  $m$ , MLQAE can achieve the same quadratic speedup [Eq. (17)] compared to the scaling of classical sampling [Eq. (16)].

Various modifications to MLQAE have been developed, including dealing with limited circuits depths and with noisy circuits [116–118], but all versions offer a clear illustration of the power of (nontrivial) classical and quantum processing working together to produce speedups over what is possible by classical processing alone.

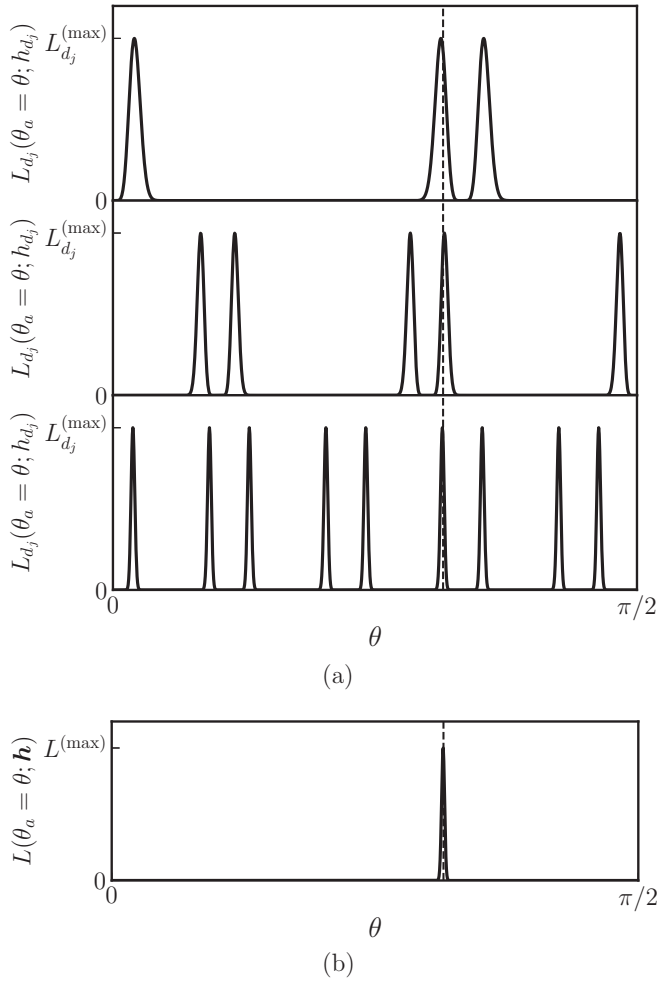


FIG. 5. (a) Illustrations of the likelihood  $L_m(\theta_a; h_m)$  of a Grover-like QAE circuit with  $m$  Grover iterations ( $m$  larger for lower plots) to produce  $h_m$  good states from  $N_{\text{shot}}$  runs, with the angle  $\theta_a$ . The actual angle  $\theta_a$  that generated the outcomes is indicated by the dashed line. (b) The overall likelihood  $L_m(\theta_a; \mathbf{h})$  produced by multiplying together the likelihoods for each  $m$ .

#### D. Quantum subroutines within quantum chemistry and materials simulations

A promising direction of research for early quantum algorithms is the development of quantum methods for solving problems in quantum chemistry. Systems for which quantum mechanics must be taken into account to calculate important properties tend in the general case to be hard to treat with purely classical techniques; this is due to the exponential growth of the size of the Hilbert space compared to the system size. Directly exploiting the quantum nature of quantum computers could lead to very natural techniques for these problems.

One approach that is the subject of much recent research [76,119] is to apply the hybrid VQE algorithm described in Subsec. III B 1. By choosing the Hamiltonian  $\hat{H}$  used in the VQE loop to be the Hamiltonian of the chemical system under study (as well as suitable VQE ansatz), the VQE procedure can be used to find the ground state (or other eigenstates) of the Hamiltonian, from which other important properties can

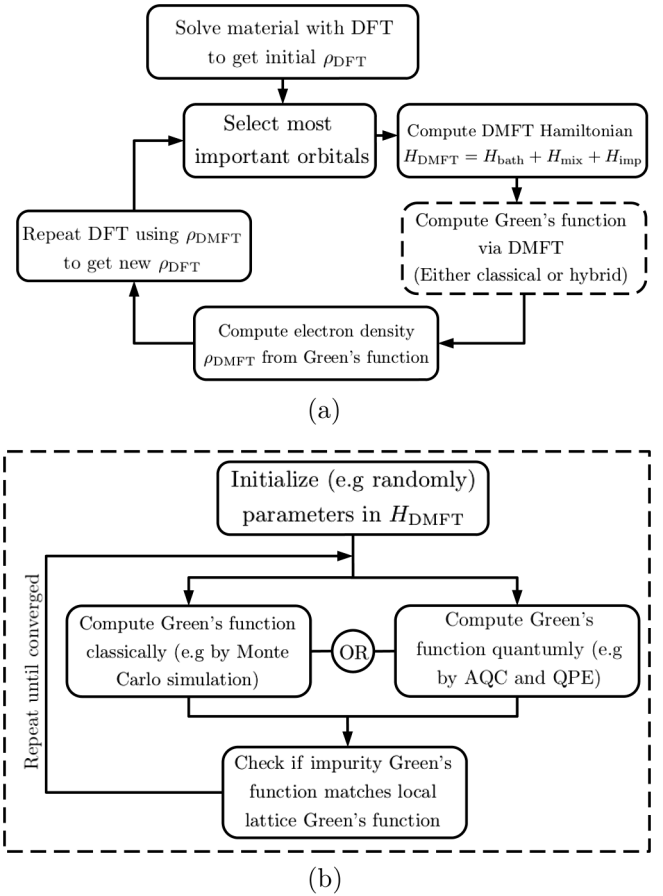


FIG. 6. Schematics showing the high-level structure of the hybrid algorithm for correlated materials proposed in Ref. [120]. (a) The outer loop in which a DFT solution is iteratively improved. The step in which the Green's function for the impurity model is calculated is shown with a dashed border to indicate that it is *this* piece of the algorithm that can be swapped out for a quantum subroutine. (b) The expanded version of the impurity model Green's function calculation from Fig. 6(a), indicating that it can be achieved either purely classically or in a hybrid quantum-classical way.

be calculated (e.g., the Green's function [120]), or various chemical properties.

While the direct VQE approach may be suitable for problems that are small enough to fit on near-term quantum devices (but are nonetheless large enough to be difficult for classical methods), for larger problems (such as for correlated materials simulations) a more involved approach must be taken. One proposal from Ma *et al.* [121] considers separating a full system into a small active space, with an effective quantum Hamiltonian, and its environment, which can be treated less accurately via density functional theory (DFT). The small active space can then be treated with VQE to learn about the electronic properties of the whole system.

Many purely classical algorithms for quantum chemistry and materials rely on quantum Monte Carlo [122,123], a family of classical algorithms which can, to some extent, emulate quantum systems in thermal equilibrium. Quantum Monte Carlo is limited for fermionic systems, however, by what is known as the sign problem, where accurate sampling can

become computationally infeasible below certain temperatures [124]. These algorithms have the advantage of providing ready-made classical wrappers to add in quantum components, and are therefore natural candidates for early hybrid quantum-classical algorithms, such as the example presented in Ref. [120]. While a full review of hybrid quantum-classical materials and chemistry algorithms, including methods which are based on time evolution rather than ground-state preparation is beyond the scope of our current work, Bauer *et al.* review the algorithmic opportunities in Ref. [119].

To give a flavor of how these algorithms work, we focus on work by Bauer *et al.* [120], which addresses the significant computational challenge to simulate the properties of strongly correlated materials where the independent-electron approximation intrinsic to the standard DFT breaks down. This work explores the dynamical mean field theory (DMFT) method (see Ref. [125] for a thorough review), which can be applied to materials expressible as lattice models. The DMFT approach which recasts the material as a local site (known as an ‘‘impurity’’) interacting with the rest of the system expressed as a noninteracting bath. The total Hamiltonian is written as

$$H_{\text{DMFT}} = H_{\text{imp}} + H_{\text{bath}} + H_{\text{mix}}, \quad (23)$$

where

$$H_{\text{imp}} = \sum_{a,b=1}^N t_{ab} f_a^\dagger f_b + \sum_{a,b,c,d=1}^N U_{abcd} f_a^\dagger f_b^\dagger f_c f_d \quad (24)$$

is the impurity Hamiltonian that will be treated carefully, where  $f^\dagger$  ( $f$ ) operators create (annihilate) a fermion in one of  $N$  local orbitals and the hopping and interaction integrals  $t_{ab}$  and  $U_{abcd}$  are determined from the underlying material. The noninteracting bath Hamiltonian has the form

$$H_{\text{bath}} = \sum_j \epsilon_j b_j^\dagger b_j, \quad (25)$$

$$(26)$$

where the  $b^\dagger$  ( $b$ ) operators create (annihilate) a fermion in one of the bath modes (considered to be infinitely many, but can be truncated to a finite number of modes if necessary). The remaining term,

$$H_{\text{mix}} = \sum_{a=1}^N \sum_j (V_{aj} f_a^\dagger b_j + V_{aj}^* b_j^\dagger f_a), \quad (27)$$

$$(28)$$

describes the interaction between the impurity and the bath. While at this level of description the bath mode energies  $\epsilon_j$  and interactions  $V_{aj}$  are free parameters, the core of the DMFT algorithm is to iteratively update these parameters in a self-consistent way until the Green’s function of the impurity model and the relevant local terms of the lattice Green’s function converge to match. This DMFT scheme is used within an additional outer loop that uses the information it provides about the correlated electrons to iteratively improve a DFT solution.

The structure of the outer loop is shown in Fig. 6(a). The step in which the impurity Green’s function is calculated is shown with a dashed border to indicate that it is

this piece of the algorithm that can be swapped out for a hybrid quantum-classical subroutine. This step is expanded in Fig. 6(b). Classically, there are various ways to calculate the Green’s function, such as by a quantum Monte Carlo algorithm, as discussed previously in this section. However, to overcome the inherent computational challenges of classical approaches, Bauer *et al.* [120] propose using a combination of adiabatic state preparation (using AQC to compute approximations to Hamiltonian eigenstates) and quantum phase estimation [126] to prepare the ground state, from which the Green’s function can be measured.

## IV. HARDWARE CONSIDERATIONS

### A. Larger computations on smaller hardware:

#### Entanglement forging

Since near-term quantum processors will have only a limited number of qubits available, it is worthwhile to consider how classical processing can be used to apply hybrid algorithms to problems that may not otherwise fit on small devices. An illustrative example of such a method is the recent *entanglement forging*, due to Ref. [127]. In that work, Eddins *et al.* show how a quantum system mapped to  $2N$  qubits can be studied on a device with only  $N$  qubits, by breaking the  $2N$ -qubit system into two  $N$ -qubit partitions and expressing the density operator  $\rho = |\Psi\rangle\langle\Psi|$  of the system in terms of the Schmidt decomposition

$$|\Psi\rangle = (U \otimes V) \sum_{k=1}^{2^N-1} \lambda_k |b_k\rangle \langle b_k|^{\otimes 2}, \quad (29)$$

where  $|b_k\rangle$  are computational basis states for an  $N$  qubit and  $U, V$  are unitary operators local to each half of the bipartition. The non-negative, real-value Schmidt coefficients  $\lambda_k$  are treated as variational parameters for a VQE circuit (see Subsec. III B 1). In this representation, expectation values  $\langle\Psi|O|\Psi\rangle$  of  $2N$ -qubit operators of the form  $O = O_1 \otimes O_2$  can be decomposed into terms of the form  $\langle\phi_1|UO_1U^\dagger|\phi_1\rangle\langle\phi_2|VO_2V^\dagger|\phi_2\rangle$ , where  $|\phi\rangle$  can be single-basis states  $|b_k\rangle$  or uniform superpositions of two basis states. These expectation values can be computed on  $N$  qubits, and are then sampled in a weighted way according to the coefficients  $\lambda_k \lambda_j$ .

While this will not scale well in the general case, systems that are only weakly entangled across the bipartition will have only a small number of dominant Schmidt coefficients  $\lambda_k$ ; thus, the expectation value  $\langle\Psi|O|\Psi\rangle$  can be well approximated with a manageable number of samples. In Ref. [127], entanglement forging (along with some heuristic enhancements) is used to find ground-state energies of various geometries of the water molecule with high accuracy, via a ten-qubit mapping, using only five superconducting qubits on real quantum hardware.

### B. Additional controls

In the case of QA and other nonuniversal analog processors, we see that sometimes hybrid techniques required additional controls. The most prototypical example here is reverse annealing, as discussed in Subsec. III A 1, which allows for hybrid techniques. These controls are important because

they allow a much richer variety of techniques which have demonstrated real gains in experiments. In particular, in this example the controls were actually already available within the underlying device (the protocol described in Ref. [55] is essentially reverse annealing combined with some other techniques), but had not been made available to external users. It is therefore highly important to think about how existing experimental techniques could be used in computation for different purposes than they were originally intended.

### C. Co-locating the quantum and classical computing power

When discussing computations at the abstract level we have so far, it is easy to forget that communication between classical and quantum systems is not free. While the currently dominant model of cloud access to quantum computers is very useful for proof-of-concept experiments, the time required to send signals over the internet between quantum and classical components could prove to be a major barrier. There are already efforts to colocate quantum computing resources with powerful classical high-performance computing, for example through the Jülich supercomputing center's JUNIQ program [128].

Some computation may require even faster communication between quantum and classical components; for qubit technology such as superconducting qubits which operate on chip substrates at ultralow temperatures, this means that classical computation at cryogenic temperatures (although possibly not as cold as the quantum computer) could be beneficial. Inspired by the quantum computing use cases, there have been numerous studies in this direction [129–133]. As we discuss in Sec. V, there are also many types of classical processors, each with their strengths and weaknesses; any of these could potentially be useful candidates to colocate with quantum computers.

## V. THE BIG PICTURE: SPECIALIZED ACCELERATORS

Throughout this work, we have offered many examples showing how classical and quantum processors can potentially be used together to perform computational tasks much more efficiently than what is possible with a classical processor alone. While a quantum processor is certainly quite different, both practically and theoretically, from any kind of classical processor, different types of processors working together is already a common structure within purely classical computing; there are many classical examples of specialized coprocessors, to which specific kinds of processing are off loaded, and this is sometimes known as *heterogeneous* or *heterotic*. A thorough discussion of heterotic computing is available in Ref. [134]. In this section, we will briefly describe some examples of classical heterotic computing and offer some intuition for how quantum processors can naturally fit into this framework.

### A. Graphics processing unit

The clearest example of classical heterogeneous computing is that of a computer's central processing unit (CPU) working together with graphics processing unit (GPU) [135]. A typical GPU consists of, among other things, a much larger number

of cores than a typical CPU, and these cores are particularly suited to the kinds of three-dimensional vector-matrix operations needed for graphics processing, such as translations and rotations in space. Typically, the same calculation needs to happen many different times on independent inputs; for example, every point on a three-dimensional object needs to move in the same way, with only the starting conditions different for each point. The large number of small cores on a GPU allows this to happen at a greatly accelerated pace [135] compared to the relatively small number of (more general purpose) cores on a CPU can manage. Typically, the computation is arranged such that the CPU hands off these specific tasks to the GPU when needed.

More recently, GPU technology has been used for applications beyond graphics processing but which also require the same large-scale parallelization of small numerical operations, such as machine learning calculations (e.g., via TensorFlow [136]).

### B. Application-specific integrated circuit

An application-specific integrated circuit (ASIC) [137] can be thought of as a processor designed and manufactured to carry out a specific computational task much more quickly and with much more energy efficiency than can be achieved by a programming a general-purpose processor such as a CPU. An example of an ASIC with these properties is Intel's Quick Sync video technology; these circuits are used for rapidly converting video between different formats and are integrated with Intel CPUs to handle these specific workloads [138]. Similar hardware, known as field-programmable gate arrays (FPGAs) [139], can be reprogrammed at the hardware level after being deployed.

#### 1. Neuromorphic processors

Many modern classical algorithms are based mimicking the behavior of the human brain through the use of software-defined neural networks. While this approach is proving successful in many areas, standard processors are proving to be limited by particular bottlenecks, such as the communication between the core and memory. To avoid these limitations, the emerging development of neuromorphic processors [140] aim to emulate the behavior of the human brain directly at the hardware level, leading to hardware that can run artificial intelligence applications much more efficiently than standard processors.

#### 2. Quantum computers as specialized coprocessors

The specialized accelerators described here all show how hardware designed and optimized for a specific type of computational work can operate in tandem with a standard processor, such as a CPU, to enhance overall computational efficiency. Hybrid quantum-classical algorithms, including the ones described in this work, fit naturally into this framework; a quantum processing unit (QPU) can be called by a general purpose classical CPU to accelerate specific computational subroutines, such as the phase estimation component of Shor's algorithm (see Subsec. II B) or the ansatz preparation component of a VQE algorithm (see Subsec. III B 1).

From this perspective, hybrid quantum-classical algorithms are a continuation of a traditional pattern in the development of computer technology. While the operating principles of a quantum computer are quite different from their classical counterparts, the way they fit into larger architectures (and the surrounding concerns) are not fundamentally something which has not been encountered before.

## VI. THE FUTURE OF HYBRID ALGORITHMS

The development path of hybrid algorithms is not universal and does not always involve a nonhybrid precursor; while hybrid forms of QAE grew out of the original pure form of QAE (and indirectly out of Grover's search, a pure quantum algorithm), many others came about in different and varied ways. Shor's algorithm, for example, emerged from the fact that a quantum algorithm fulfilled a niche application which allowed for a very powerful computational tool. This is similar to many hybrid materials and chemistry algorithms, which have developed out of classical algorithms by replacing classical methods such as quantum Monte Carlo with quantum computing calls. In the case of variational algorithms, as well as hybrid forms of quantum annealing, the capabilities of the hardware has driven the development of algorithms. A key lesson to draw from these stories is that there is not one way to develop hybrid algorithms: They can emerge in a multitude of different ways.

An immediate question which arises from this discussion is whether hybrid algorithms are just a phase in the evolution of quantum computing, and when (and if) large, fully fault-tolerant quantum computers come into existence, will hybrid algorithms still be used? If we look at the evolution of classical computing technologies, we can find a hint. As new, advanced coprocessors have been developed classically they have not displaced older paradigms, but rather been added to them to perform specialized tasks. We do not see any reason to believe that the evolution of quantum computers will be any different. Quantum computers are not intrinsically faster at simple applications like adding numbers, and even if large

numbers of high-quality qubits became readily available, it is doubtful that they would ever be better than classical bits by all metrics, in which case heterotic computing would be the most sensible paradigm, as it currently is in the world of classical computing. We find it unlikely that once quantum hardware advances to the stage of genuine usefulness nonhybrid algorithms will ever be deployed in a "production" setting, for the simple reason that it is likely that further improvements could be made by adding a hybrid component.

From our discussion so far, we have seen that hybrid algorithms are most meaningfully classified in terms not of how much classical computing power they use, but how that computation fits into the overall model. While there are notable algorithms which are both hybrid and nonhybrid, a pattern which has appeared both in terms of Grover's algorithm and subsequent QAE techniques, and quantum annealing and subsequent hybrid versions, is that nonhybrid algorithms often evolve into richer, more powerful, hybrid variants. This pattern is likely not incidental; it is often easier to first imagine algorithms in a more "pure" setting and then allow them to evolve. This suggests an interesting relationship between hybrid and nonhybrid algorithms: The nonhybrid setting is a useful incubator for hybrid techniques which will later be implemented. As such, we are not suggesting that hybrid algorithms are the only class of quantum algorithms which are worthwhile to study, but that it is probably most appropriate to think of nonhybrid algorithms as the foundations of new algorithms which are yet to be developed.

## ACKNOWLEDGMENTS

The authors thank R. Dridi, V. Kendon, S. Stepney, and D. Browne for useful discussions and references. Both authors were supported by the UK Engineering and Physical Science Research Council (EPSRC) Grant No. EP/T026715/1 for Collaborative Computational Project in Quantum Computing (CCP-QC). N.C. was additionally supported by EPSRC Grant No. EP/W00772X/1, and A.C. was supported by the EPSRC UK Quantum Technology Hub in Computing and Simulation (EP/T001062/1).

- 
- [1] R. P. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* **21**, 467 (1982).
- [2] P. Benioff, The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines, *J. Stat. Phys.* **22**, 563 (1980).
- [3] D. Deutsch and R. Penrose, Quantum theory, the Church-Turing principle, and the universal quantum computer, *Proc. R. Soc. London, Ser. A* **400**, 97 (1985).
- [4] D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, *Proc. R. Soc. London, Ser. A* **439**, 553 (1992).
- [5] D. R. Simon, On the power of quantum computation, *SIAM J. Comput.* **26**, 1474 (1997).
- [6] E. Bernstein and U. Vazirani, Quantum complexity theory, *SIAM J. Comput.* **26**, 1411 (1997).
- [7] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA* (IEEE, New York, 1994), pp. 124–134.
- [8] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Rev.* **41**, 303 (1999).
- [9] I. L. Chuang, N. Gershenfeld, and M. Kubinec, Experimental Implementation of Fast Quantum Searching, *Phys. Rev. Lett.* **80**, 3408 (1998).
- [10] J. Brooke, D. Bitko, T. F. Rosenbaum, and G. Aeppli, Quantum annealing of a disordered magnet, *Science* **284**, 779 (1999).
- [11] R. Marx, A. F. Fahmy, J. M. Myers, W. Bermel, and S. J. Glaser, Approaching five-bit NMR quantum computing, *Phys. Rev. A* **62**, 012310 (2000).
- [12] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Experimental realization of Shor's quantum factoring algorithm using

- nuclear magnetic resonance, *Nature (London)* **414**, 883 (2001).
- [13] S. Gulde, M. Riebe, G. P. T. Lancaster, C. Becher, J. Eschner, H. Häffner, F. Schmidt-Kaler, I. L. Chuang, and R. Blatt, Implementation of the Deutsch–Jozsa algorithm on an ion-trap quantum computer, *Nature (London)* **421**, 48 (2003).
- [14] M. S. Anwar, J. A. Jones, D. Blazina, S. B. Duckett, and H. A. Carteret, Implementation of NMR quantum computation with parahydrogen-derived high-purity quantum states, *Phys. Rev. A* **70**, 032324 (2004).
- [15] C. Negrevergne, T. S. Mahesh, C. A. Ryan, M. Ditty, F. Cyr-Racine, W. Power, N. Boulant, T. Havel, D. G. Cory, and R. Laflamme, Benchmarking Quantum Control Methods on a 12-Qubit System, *Phys. Rev. Lett.* **96**, 170501 (2006).
- [16] C.-Y. Lu, X.-Q. Zhou, O. Gühne, W.-B. Gao, J. Zhang, Z.-S. Yuan, A. Goebel, T. Yang, and J.-W. Pan, Experimental entanglement of six photons in graph states, *Nat. Phys.* **3**, 91 (2007).
- [17] M. S. Tame, R. Prevedel, M. Paternostro, P. Böhi, M. S. Kim, and A. Zeilinger, Experimental Realization of Deutsch’s Algorithm in a One-Way Quantum Computer, *Phys. Rev. Lett.* **98**, 140501 (2007).
- [18] C.-Y. Lu, W.-B. Gao, O. Gühne, X.-Q. Zhou, Z.-B. Chen, and J.-W. Pan, Demonstrating Anyonic Fractional Statistics with a Six-Qubit Quantum Simulator, *Phys. Rev. Lett.* **102**, 030502 (2009).
- [19] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom *et al.*, Quantum annealing with manufactured spins, *Nature (London)* **473**, 194 (2011).
- [20] S. J. Devitt, Performing quantum computing experiments in the cloud, *Phys. Rev. A* **94**, 032329 (2016).
- [21] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney *et al.*, Quantum supremacy using a programmable superconducting processor, *Nature (London)* **574**, 505 (2019).
- [22] 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, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang *et al.*, Quantum computational advantage using photons, *Science* **370**, 1460 (2020).
- [23] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
- [24] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse Ising model, *Phys. Rev. E* **58**, 5355 (1998).
- [25] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, A variational eigenvalue solver on a photonic quantum processor, *Nat. Commun.* **5**, 4213 (2014).
- [26] P. J. J. O’Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus *et al.*, Scalable Quantum Simulation of Molecular Energies, *Phys. Rev. X* **6**, 031007 (2016).
- [27] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, Error mitigation extends the computational reach of a noisy quantum processor, *Nature (London)* **567**, 491 (2019).
- [28] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, *Nature (London)* **567**, 209 (2019).
- [29] R. LaRose, A. Tikku, É. O’Neel-Judy, L. Cincio, and P. J. Coles, Variational quantum state diagonalization, *npj Quantum Inf.* **5**, 57 (2019).
- [30] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, D. A. Buell, B. Burkett, N. Bushnell, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney *et al.*, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, *Nat. Phys.* **17**, 332 (2021).
- [31] A. Perdomo-Ortiz, S. E. Venegas-Andraca, and A. Aspuru-Guzik, A study of heuristic guesses for adiabatic quantum computation, *Quant. Info. Proc.* **10**, 33 (2011).
- [32] Q.-H. Duan, S. Zhang, W. Wu, and P.-X. Chen, An alternative approach to construct the initial Hamiltonian of the adiabatic quantum computation, *Chin. Phys. Lett.* **30**, 010302 (2013).
- [33] N. Chancellor, Modernizing quantum annealing using local searches, *New J. Phys.* **19**, 023024 (2017).
- [34] T. Graß, Quantum Annealing with Longitudinal Bias Fields, *Phys. Rev. Lett.* **123**, 120501 (2019).
- [35] N. Chancellor, Modernizing quantum annealing II: Genetic algorithms with the inference primitive formalism, [arXiv:1609.05875](https://arxiv.org/abs/1609.05875).
- [36] D. Venturelli and A. Kondratyev, Reverse quantum annealing approach to portfolio optimization problems, *Quantum Machine Intelligence* **1**, 17 (2019).
- [37] J. King, M. Mohseni, W. Bernoudy, A. Fréchet, H. Sadeghi, S. V. Isakov, H. Neven, and M. H. Amin, Quantum-assisted genetic algorithm, [arXiv:1907.00707](https://arxiv.org/abs/1907.00707).
- [38] J. Roffe, Quantum error correction: An introductory guide, *Contemp. Phys.* **60**, 226 (2019).
- [39] N. P. Breuckmann, K. Duivenvoorden, D. Michels, and B. M. Terhal, Local decoders for the 2D and 4D toric code, *Quantum Info. Comput.* **17**, 181 (2017).
- [40] D. Horsman, V. Kendon, and S. Stepney, *Abstraction/Representation Theory and the Natural Science of Computation* (Cambridge University Press, Cambridge, UK, 2018), pp. 127–150.
- [41] C. Horsman, S. Stepney, R. C. Wagner, and V. Kendon, When does a physical system compute? *Proc. R. Soc. A* **470**, 20140182 (2014).
- [42] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing* (ACM, New York, 1996), pp. 212–219.
- [43] A. B. Fimmel, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, Quantum annealing: A new method for minimizing multidimensional functions, *Chem. Phys. Lett.* **219**, 343 (1994).
- [44] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106).
- [45] J. Roland and N. J. Cerf, Quantum search by local adiabatic evolution, *Phys. Rev. A* **65**, 042308 (2002).

- [46] G. E. Santoro, R. Martoňák, E. Tosatti, and R. Car, Theory of quantum annealing of an Ising spin glass, *Science* **295**, 2427 (2002).
- [47] T. Albash and D. A. Lidar, Adiabatic quantum computation, *Rev. Mod. Phys.* **90**, 015002 (2018).
- [48] S. Yarkoni, E. Raponi, S. Schmitt, and T. Bäck, Quantum annealing for industry applications: Introduction and review, [arXiv:2112.07491](https://arxiv.org/abs/2112.07491).
- [49] E. J. Crosson and D. A. Lidar, Prospects for quantum enhancement with diabatic quantum annealing, *Nat. Rev. Phys.* **3**, 466 (2021).
- [50] Y. Yamashiro, M. Ohkuwa, H. Nishimori, and D. A. Lidar, Dynamics of reverse annealing for the fully connected  $p$ -spin model, *Phys. Rev. A* **100**, 052321 (2019).
- [51] J. Marshall, D. Venturelli, I. Hen, and E. G. Rieffel, Power of Pausing: Advancing Understanding of Thermalization in Experimental Quantum Annealers, *Phys. Rev. Appl.* **11**, 044083 (2019).
- [52] M. Ohkuwa, H. Nishimori, and D. A. Lidar, Reverse annealing for the fully connected  $p$ -spin model, *Phys. Rev. A* **98**, 022314 (2018).
- [53] A. Callison, M. Festenstein, J. Chen, L. Nita, V. Kendon, and N. Chancellor, Energetic perspective on rapid quenches in quantum annealing, *PRX Quantum* **2**, 010338 (2021).
- [54] T. Graß, Quantum annealing sampling with a bias field, [arXiv:2205.15820](https://arxiv.org/abs/2205.15820).
- [55] T. Lanting, A. J. Przybysz, A. Yu. Smirnov, F. M. Spedalieri, M. H. Amin, A. J. Berkley, R. Harris, F. Altomare, S. Boixo, P. Bunyk, N. Dickson, C. Enderud, J. P. Hilton, E. Hoskinson, M. W. Johnson, E. Ladizinsky, N. Ladizinsky, R. Neufeld, T. Oh, I. Perminov *et al.*, Entanglement in a Quantum Annealing Processor, *Phys. Rev. X* **4**, 021041 (2014).
- [56] J. Machta, Population annealing with weighted averages: A Monte Carlo method for rough free-energy landscapes, *Phys. Rev. E* **82**, 026704 (2010).
- [57] W. Wang, J. Machta, and H. G. Katzgraber, Population annealing: Theory and application in spin glasses, *Phys. Rev. E* **92**, 063307 (2015).
- [58] R. H. Swendsen and J.-S. Wang, Replica Monte Carlo Simulation of Spin Glasses, *Phys. Rev. Lett.* **57**, 2607 (1986).
- [59] D. J. Earl and M. W. Deem, Parallel tempering: Theory, applications, and new perspectives, *Phys. Chem. Chem. Phys.* **7**, 3910 (2005).
- [60] H. Neven, AQC 2016—Opening remarks: Why we believe quantum annealing will succeed, <https://www.youtube.com/watch?v=mSEkxjN0I4>.
- [61] D-Wave Systems Inc., Reverse quantum annealing for local refinement of solutions, [https://www.dwavesys.com/media/5hsjmvom/14-1018a-a\\_reverse\\_quantum\\_annealing\\_for\\_local\\_refinement\\_of\\_solutions.pdf](https://www.dwavesys.com/media/5hsjmvom/14-1018a-a_reverse_quantum_annealing_for_local_refinement_of_solutions.pdf).
- [62] D. Ottaviani and A. Amendola, Low rank non-negative matrix factorization with D-wave 2000Q, [arXiv:1808.08721](https://arxiv.org/abs/1808.08721).
- [63] J. Golden and D. O'Malley, Reverse annealing for nonnegative/binary matrix factorization, *PLOS ONE* **16**, e0244026 (2021).
- [64] N. Chancellor, Fluctuation-guided search in quantum annealing, *Phys. Rev. A* **102**, 062606 (2020).
- [65] N. Chancellor and V. Kendon, Experimental test of search range in quantum annealing, *Phys. Rev. A* **104**, 012604 (2021).
- [66] A. D. King, J. Carrasquilla, J. Raymond, I. Ozfidan, E. Andriyash, A. Berkley, M. Reis, T. Lanting, R. Harris, F. Altomare, K. Boothby, P. I. Bunyk, C. Enderud, A. Fréchet, E. Hoskinson, N. Ladizinsky, T. Oh, G. Poulin-Lamarre, C. Rich *et al.*, Observation of topological phenomena in a programmable lattice of 1,800 qubits, *Nature (London)* **560**, 456 (2018).
- [67] A. D. King, J. Raymond, T. Lanting, S. V. Isakov, M. Mohseni, G. Poulin-Lamarre, S. Ejtemaee, W. Bernoudy, I. Ozfidan, A. Yu. Smirnov, M. Reis, F. Altomare, M. Babcock, A. J. Baron, C. Berkley, K. Boothby, P. I. Bunyk, H. Christiani, C. Enderud, B. Evert *et al.*, Scaling advantage over path-integral Monte Carlo in quantum simulation of geometrically frustrated magnets, *Nat. Commun.* **12**, 1113 (2021).
- [68] G. Passarelli, Ka-W. Yip, D. A. Lidar, H. Nishimori, and P. Lucignano, Reverse quantum annealing of the  $p$ -spin model with relaxation, *Phys. Rev. A* **101**, 022331 (2020).
- [69] G. Passarelli, K.-W. Yip, D. A. Lidar, and P. Lucignano, Standard quantum annealing outperforms adiabatic reverse annealing with decoherence, *Phys. Rev. A* **105**, 032431 (2022).
- [70] D. Guéry-Odelin, A. Ruschhaupt, A. Kiely, E. Torrontegui, S. Martínez-Garaot, and J. G. Muga, Shortcuts to adiabaticity: Concepts, methods, and applications, *Rev. Mod. Phys.* **91**, 045001 (2019).
- [71] A. Hartmann and W. Lechner, Rapid counter-diabatic sweeps in lattice gauge adiabatic quantum computing, *New J. Phys.* **21**, 043025 (2019).
- [72] G. Passarelli, V. Cataudella, R. Fazio, and P. Lucignano, Counterdiabatic driving in the quantum annealing of the  $p$ -spin model: A variational approach, *Phys. Rev. Research* **2**, 013283 (2020).
- [73] G. Passarelli, R. Fazio, and P. Lucignano, Optimal quantum annealing: A variational shortcut-to-adiabaticity approach, *Phys. Rev. A* **105**, 022618 (2022).
- [74] Y.-Q. Chen, Y. Chen, C.-K. Lee, S. Zhang, and C.-Y. Hsieh, Optimizing quantum annealing schedules with Monte Carlo tree search enhanced with neural networks, *Nat. Mach. Intell.* **4**, 269 (2022).
- [75] P. R. Hegde, G. Passarelli, A. Scocco, and P. Lucignano, Genetic optimization of quantum annealing, *Phys. Rev. A* **105**, 012612 (2022).
- [76] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, *Nat. Rev. Phys.* **3**, 625 (2021).
- [77] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, Hybrid quantum-classical algorithms and quantum error mitigation, *J. Phys. Soc. Jpn.* **90**, 032001 (2021).
- [78] J. A. Nelder and R. Mead, A simplex method for function minimization, *Comput. J.* **7**, 308 (1965).
- [79] Y. Cao, J. Romero, J. P. Olson, M. Degroote, Peter 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).
- [80] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth *et al.*, The variational quantum eigensolver: A review of methods and best practices, [arXiv:2111.05176](https://arxiv.org/abs/2111.05176).

- [81] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [arXiv:1411.4028](#).
- [82] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem, [arXiv:1412.6062](#).
- [83] E. Farhi and A. W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm, [arXiv:1602.07674](#).
- [84] Z.-C. Yang, A. Rahmani, A. Shabani, H. Neven, and C. Chamon, Optimizing Variational Quantum Algorithms Using Pontryagin's Minimum Principle, *Phys. Rev. X* **7**, 021027 (2017).
- [85] Z. Jiang, E. G. Rieffel, and Z. Wang, Near-optimal quantum circuit for Grover's unstructured search using a transverse field, *Phys. Rev. A* **95**, 062317 (2017).
- [86] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, Quantum approximate optimization algorithm for MaxCut: A fermionic view, *Phys. Rev. A* **97**, 022304 (2018).
- [87] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, *Algorithms* **12**, 34 (2019).
- [88] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, XY mixers: Analytical and numerical results for the quantum alternating operator ansatz, *Phys. Rev. A* **101**, 012320 (2020).
- [89] M. M. Wauters, E. Panizon, G. B. Mbeng, and G. E. Santoro, Reinforcement-learning-assisted quantum optimization, *Phys. Rev. Research* **2**, 033446 (2020).
- [90] M.-H. Yung, J. Casanova, A. Mezzacapo, J. McClean, L. Lamata, A. Aspuru-Guzik, and E. Solano, From transistor to trapped-ion computers for quantum chemistry, *Sci. Rep.* **4**, 3589 (2014).
- [91] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz, *Quantum Sci. Technol.* **4**, 014008 (2018).
- [92] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, *Nature (London)* **549**, 242 (2017).
- [93] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, Quantum computational chemistry, *Rev. Mod. Phys.* **92**, 015003 (2020).
- [94] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting ansatz expressibility to gradient magnitudes and barren plateaus, *PRX Quantum* **3**, 010313 (2022).
- [95] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, Noise-induced barren plateaus in variational quantum algorithms, *Nat. Commun.* **12**, 6961 (2021).
- [96] A. Arrasmith, M. Cerezo, P. Czarnik, L. Cincio, and P. J. Coles, Effect of barren plateaus on gradient-free optimization, *Quantum* **5**, 558 (2021).
- [97] T. Volkoff and P. J. Coles, Large gradients via correlation in random parameterized quantum circuits, *Quantum Sci. Technol.* **6**, 025008 (2021).
- [98] S. Wang, P. Czarnik, A. Arrasmith, M. Cerezo, L. Cincio, and P. J. Coles, Can error mitigation improve trainability of noisy variational quantum algorithms? [arXiv:2109.01051](#).
- [99] L. T. Brady, C. L. Baldwin, A. Bapat, Y. Kharkov, and A. V. Gorshkov, Optimal Protocols in Quantum Annealing and Quantum Approximate Optimization Algorithm Problems, *Phys. Rev. Lett.* **126**, 070505 (2021).
- [100] L. T. Brady, L. Kocia, P. Bienias, A. Bapat, Y. Kharkov, and A. V. Gorshkov, Behavior of analog quantum algorithms, [arXiv:2107.01218](#).
- [101] M. Larocca, P. Czarnik, K. Sharma, G. Muraleedharan, P. J. Coles, and M. Cerezo, Diagnosing barren plateaus with tools from quantum optimal control, [arXiv:2105.14377](#).
- [102] J. Berwald, N. Chancellor, and R. Dridi, Understanding domain-wall encoding theoretically and experimentally, [arXiv:2108.12004](#).
- [103] N. P. D. Sawaya, G. G. Guerreschi, and A. Holmes, On connectivity-dependent resource requirements for digital quantum simulation of  $d$ -level particles, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2020), pp. 180–190.
- [104] F. G. Fuchs, H. Ø. Kolden, N. H. Aase, and G. Sartor, Efficient encoding of the weighted MAX  $k$ -CUT on a quantum computer using QAOA, *SN Comput. Sci.* **2**, 89 (2021).
- [105] C. Campbell and E. Dahl, QAOA of the highest order, in *2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C)* (2022), pp. 141–146.
- [106] J. Plewa, J. Sieńko, and K. Rycerz, Variational algorithms for workflow scheduling problem in gate-based quantum devices, *Comput. Informatics* **40**, 897 (2021).
- [107] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, in *Quantum Computation and Information (Washington, DC, 2000)*, Contemporary Mathematics Vol. 305 (American Mathematical Society, Providence, RI, 2002), pp. 53–74.
- [108] A. Montanaro, Quantum speedup of Monte Carlo methods, *Proc. R. Soc. London, Ser. A* **471**, 20150301 (2015).
- [109] B. L. Hammond, W. A. Lester, and P. J. Reynolds, *Monte Carlo Methods in ab initio Quantum Chemistry* (World Scientific, Singapore, 1994).
- [110] W. Krauth, *Statistical Mechanics: Algorithms and Computations* (Oxford University Press, Oxford, UK, 2006).
- [111] P. Glasserman, *Monte Carlo Methods in Financial Engineering* (Springer, New York, 2004).
- [112] S. Aaronson and P. Rall, in *Proc. of the Symposium on Simplicity in Algorithms (SOSA): Quantum Approximate Counting, Simplified* (SIAM, Philadelphia, PA, 2020), pp. 24–32.
- [113] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, Amplitude estimation without phase estimation, *Quant. Info. Proc.* **19**, 75 (2020).
- [114] K. Nakaji, Faster amplitude estimation, *Quantum Info. Comput.* **20**, 1109 (2020).
- [115] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, Iterative quantum amplitude estimation, *npj Quantum Inf.* **7**, 52 (2021).
- [116] T. Giurgica-Tiron, I. Kerenidis, F. Labib, A. Prakash, and W. Zeng, Low depth algorithms for quantum amplitude estimation, *Quantum* **6**, 745 (2022).
- [117] E. G. Brown, O. Goktas, and W. K. Tham, Quantum amplitude estimation in the presence of noise, [arXiv:2006.14145](#).
- [118] S. Uno, Y. Suzuki, K. Hisanaga, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, Modified Grover operator for quantum amplitude estimation, *New J. Phys.* **23**, 083031 (2021).



- [119] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, Quantum algorithms for quantum chemistry and quantum materials science, *Chem. Rev.* **120**, 12685 (2020).
- [120] B. Bauer, D. Wecker, A. J. Millis, M. B. Hastings, and M. Troyer, Hybrid Quantum-Classical Approach to Correlated Materials, *Phys. Rev. X* **6**, 031045 (2016).
- [121] H. Ma, M. Govoni, and G. Galli, Quantum simulations of materials on near-term quantum computers, *npj Comput. Mater.* **6**, 85 (2020).
- [122] E. Gull, A. J. Millis, A. I. Lichtenstein, A. N. Rubtsov, M. Troyer, and P. Werner, Continuous-time Monte Carlo methods for quantum impurity models, *Rev. Mod. Phys.* **83**, 349 (2011).
- [123] B. M. Austin, D. Y. Zubarev, and W. A. Lester, Quantum Monte Carlo and related approaches, *Chem. Rev.* **112**, 263 (2012).
- [124] E. Y. Loh, J. E. Gubernatis, R. T. Scalettar, S. R. White, D. J. Scalapino, and R. L. Sugar, Sign problem in the numerical simulation of many-electron systems, *Phys. Rev. B* **41**, 9301 (1990).
- [125] A. Georges, G. Kotliar, W. Krauth, and M. J. Rozenberg, Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions, *Rev. Mod. Phys.* **68**, 13 (1996).
- [126] A. Y. Kitaev, Quantum measurements and the abelian stabilizer problem, [arXiv:quant-ph/9511026](https://arxiv.org/abs/quant-ph/9511026).
- [127] A. Eddins, M. Motta, T. P. Gujarati, S. Bravyi, A. Mezzacapo, C. Hadfield, and S. Sheldon, Doubling the size of quantum simulators by entanglement forging, *PRX Quantum* **3**, 010309 (2022).
- [128] JUNIQ, Launch of JUNIQ, <https://www.fz-juelich.de/SharedDocs/Pressemitteilungen/UK/EN/2019/2019-10-25-juniq.html>.
- [129] H. Homulle and E. Charbon, Cryogenic low-dropout voltage regulators for stable low-temperature electronics, *Cryogenics* **95**, 11 (2018).
- [130] B. Patra, R. M. Incandela, J. P. G. van Dijk, H. A. R. Homulle, L. Song, M. Shahmohammadi, R. B. Staszewski, A. Vladimirescu, M. Babaie, F. Sebastiano, and E. Charbon, Cryo-CMOS circuits and systems for quantum computing applications, *IEEE J. Solid-State Circuits* **53**, 309 (2018).
- [131] S. Weinreb, J. C. Bardin, and H. Mani, Design of cryogenic SiGe low-noise amplifiers, *IEEE Trans. Microwave Theory Tech.* **55**, 2306 (2007).
- [132] H. Homulle and E. Charbon, Performance characterization of Altera and Xilinx 28 nm FPGAs at cryogenic temperatures, in *2017 International Conference on Field Programmable Technology (ICFPT)* (IEEE, 2017), pp. 25–31.
- [133] I. D. Conway Lamb, J. I. Colless, J. M. Hornibrook, S. J. Pauka, S. J. Waddy, M. K. Frechtling, and D. J. Reilly, An FPGA-based instrumentation platform for use at deep cryogenic temperatures, *Rev. Sci. Instrum.* **87**, 014701 (2016).
- [134] S. Stepney, V. Kendon, P. Hines, and A. Sebald, A framework for heterotic computing, *EPTCS* **95**, 263 (2012).
- [135] C. McClanahan, History and evolution of GPU architecture: A paper survey), <http://mcclanahoochie.com/blog/wp-content/uploads/2011/03/gpu-hist-paper.pdf>.
- [136] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, TensorFlow: Large-scale machine learning on heterogeneous distributed systems, [arXiv:1603.04467](https://arxiv.org/abs/1603.04467).
- [137] M. J. Sebastian Smith, *Application-Specific Integrated Circuits* (Addison-Wesley Professional, Boston, MA, 1997).
- [138] C. Angelini, Intel's second-gen core CPUs: The Sandy Bridge review, <https://www.tomshardware.com/reviews/sandy-bridge-core-i7-2600k-core-i5-2500k,2833-4.html>.
- [139] S. M. S. Trimberger, Three ages of FPGAs: A retrospective on the first thirty years of FPGA technology, *IEEE Solid-State Circuits Mag.* **10**, 16 (2018).
- [140] C. D. James, J. B. Aimone, N. E. Miner, C. M. Vineyard, F. H. Rothganger, K. D. Carlson, S. A. Mulder, T. J. Draelos, A. Faust, M. J. Marinella, J. H. Naegle, and S. J. Plimpton, A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications, *Biologically Inspired Cognitive Architectures* **19**, 49 (2017).