

## Fourier expansion in variational quantum algorithms

Nikita A. Nemkov,<sup>\*</sup> Evgeniy O. Kiktenko, and Aleksey K. Fedorov<sup>†</sup>

*Russian Quantum Center, Skolkovo, Moscow 143026, Russia  
and National University of Science and Technology MISIS, Moscow 119049, Russia*



(Received 18 April 2023; accepted 11 August 2023; published 6 September 2023)

The Fourier expansion of the loss function in variational quantum algorithms (VQAs) contains a wealth of information yet is generally hard to access. We focus on the class of variational circuits where constant gates are Clifford gates and parametrized gates are generated by Pauli operators, which covers most practical cases while allowing much control due to the properties of stabilizer circuits. We give a classical algorithm that, for an  $N$ -qubit circuit and a single Pauli observable, computes coefficients of all trigonometric monomials up to a degree  $m$  in time bounded by  $O(N2^m)$ . Using the general structure and implementation of the algorithm, we reveal several aspects of Fourier expansions in Clifford plus Pauli VQAs such as (i) reformulating the problem of computing the Fourier series as an instance of the multivariate Boolean quadratic system, (ii) showing that the approximation given by a truncated Fourier expansion can be quantified by the  $L^2$ -norm and evaluated dynamically, (iii) the tendency of Fourier series to be rather sparse and Fourier coefficients to cluster together, and (iv) the possibility to compute the full Fourier series for circuits of nontrivial sizes, featuring tens to hundreds of qubits and parametric gates.

DOI: [10.1103/PhysRevA.108.032406](https://doi.org/10.1103/PhysRevA.108.032406)

### I. INTRODUCTION AND RESULTS

Variational quantum algorithms (VQAs) [1] are the leading candidates to make the most out of current NISQ devices [2,3]. While the scope of potential VQA applications is extremely broad, there are also many theoretical and practical limitations. Variational quantum algorithms are hybrid algorithms, using classical optimization to train parametrized quantum circuits, and in this sense they are similar to the classical machine learning models. The loss function of a VQA is defined as an average of some observable in the state, prepared by the parametrized quantum circuit. The structure of the VQA loss landscape is of central importance, because the efficiency of the classical optimization largely determines the quality of solutions obtained by the VQA. Accessible shapes of the loss function also determine the expressive power of the quantum machine learning models [4].

Typically, VQAs are trained by gradient-based methods and their local properties are of the most interest. At the same time, the structure of parametrized quantum circuits makes Fourier series representation a natural and rich language for the description of VQA loss functions. We now briefly survey some of the relevant results. Under the assumption that generators of parametric gates have commensurable eigenvalues, the Fourier series is in fact truncated to a trigonometric polynomial. As shown in Ref. [5], accessible frequencies in this expansion are determined by the spectrum of the generators, while the coefficients depend on the structure of the circuit and the observable. In Refs. [5,6] this observation

was used to highlight the importance of data encoding in quantum machine learning models. Recently, there has been an interest in quantitatively studying the expressive power of machine learning models based on the properties of their Fourier expansion [7,8]. In Refs. [9,10] the Fourier representation was used to dequantize a class of quantum machine learning models. Reference [11] investigates the case when the Fourier series is rather sparse, so the loss landscape can be efficiently recovered with limited experimental data. An interesting proposal made in Ref. [12] shows that noise in VQAs can be detected by observing inaccessible frequencies and mitigated by filtering them out. We also note that the fundamental result on the NP-hardness of training general VQAs [13] relies on their loss functions being trigonometric polynomials.

While the Fourier representation can be a very convenient tool to characterize variational loss functions, it has its limitations. When generators of the parametric gates square to identity, which is the most common case, the Fourier series for a VQA with  $M$  parameters is a multivariate trigonometric polynomial containing up to  $3^M$  terms. Exponential growth of accessible terms makes the Fourier series an impractical description, unless the actual distribution of coefficients is very sparse (e.g., the number of nonzero terms only grows polynomially with  $M$ ). Computing the Fourier coefficients is also a challenge. For example, evaluating the lowest-order constant Fourier term amounts to finding the loss function averaged over all parameter configurations, and there seems to be no efficient recipe for that in general.

In this paper we restrict our attention to a special class of parametrized quantum circuits, which we refer to as Clifford plus Pauli circuits. Parametric gates in Clifford plus Pauli circuits are exponentials of Pauli strings, while constant gates

<sup>\*</sup>nemkov@gmail.com

<sup>†</sup>akf@rqc.ru

are Clifford gates. This is in fact a very large class of circuits that includes the majority of most studied VQAs, such as the quantum approximate optimization algorithm (QAOA) [14], the hardware-efficient ansatz (HEA), [15], and the unitary coupled-cluster ansatz [16]. Special properties of stabilizer circuits [17,18] provide an essential technical advantage to study the Fourier expansion of Clifford plus Pauli circuits in quantitative details.

The interplay between properties of stabilizer circuits and VQAs have been explored previously, mainly in the context of quantum chemistry [19–21]. In particular, initialization methods based on perturbative expansion around Clifford points [22] or on the discrete search through the space of Clifford circuits [23,24] have been developed and ansatz structures [25,26] and partitioning schemes [27] based on the properties of Clifford gates have been proposed. In this work, however, we focus on a different scope of questions.

Our core technical contribution is an efficient classical algorithm computing all Fourier coefficients in the loss function up to level  $m$ , with time complexity bounded by  $O(N2^m)$ , where  $N$  is the number of qubits. Note that, in general, the complexity of the Fourier series is not limited directly by the number of qubits  $N$  but rather by the total number of parametric gates  $M$ . For Clifford plus Pauli VQAs this observation thus admits a concrete realization.

The algorithm has both theoretical and practical utility. On the theoretical side, we show that typical Fourier series are much sparser than anticipated in the general case. For Clifford plus Pauli circuits with a single Pauli observable, the number of coefficients is upper bounded by  $2^M$ , and for the worst-case behavior expected in practice we find  $(\frac{3}{2})^M$ . We also show that truncating the Fourier polynomial below the maximum degree gives an approximation that can be quantified by the  $L^2$ -norm of the loss function and evaluated dynamically as the algorithm proceeds. The number of terms contributing non-negligibly to the functional norm is typically an exponential fraction of all terms yet still growing exponentially itself.

On the practical side, we perform several case studies to probe the structure of Fourier expansions in more detail. In all examples we find that Fourier terms tend to cluster around some mean level, which is exactly  $\frac{M}{2}$  for nonlocal random circuits but much smaller for the local circuits with a special structure, such as the QAOA or HEA, making their Fourier expansion much sparser and easier to compute.

Our algorithm is based on a simple recursive expansion of the loss function. A similar approach was described in the context of the qubit coupled-cluster method [25] and in the context of the QAOA in Ref. [28]. However, there are some important distinctions with the previous work. First, we identify Clifford plus Pauli circuits as the class to which the method is universally applicable and treat the problem in general terms, as well as establish its direct relation to the Fourier series expansion. More importantly, we demonstrate how to significantly reduce the algorithm cost by pruning some branches of the recursive expansion early, based on filtering by expectation values. While this does not change the large- $M$  asymptotic, for practical cases the difference is essential. For example, for random Clifford plus Pauli circuits on  $N = 50$  qubits, it allows increasing the depth of circuits that can be handled from  $M = 30$  to  $M = 80$  without changing

the computational budget. Finally, we formulate the problem of computing all nonzero coefficients in the Fourier expansion as an instance of the multivariate Boolean quadratic problem, which allows us to argue that our algorithm is likely not far from optimal yet points towards potential improvements.

## II. PARAMETRIZED QUANTUM CIRCUITS

In this section we establish some notation, describe basic properties of variational circuits and their loss functions, and discuss how the Fourier expansion arises in this context.

### A. Trigonometric expansion of the unitary matrix

We will assume that a unitary matrix  $U(\phi)$  of a parametrized quantum circuit takes the following form:

$$U(\phi) = C_M P_M(\phi_M) \cdots C_2 P_2(\phi_2) C_1 P_1(\phi_1) C_0. \quad (1)$$

Here  $C_m$  are constant gates,  $P_m(\phi) = e^{-i(\phi/2)G_m}$  are single-parameter rotations, and  $\phi = (\phi_1, \dots, \phi_M)$  is a vector of parameters. We assume that Hermitian generators of the parametrized gates square to identity  $G_m^2 = \mathbb{1}$  (in particular,  $G_m$  are unitary), so

$$P_m(\phi) = \mathbb{1} \cos \frac{\phi}{2} - i G_m \sin \frac{\phi}{2}. \quad (2)$$

Applying this relation to each parametric gate in the circuit, we obtain the following formal trigonometric expansion containing  $2^M$  terms:

$$U(\phi) = \sum_{I \in \{0,1\}^M} U_I t_I \left( \frac{\phi}{2} \right). \quad (3)$$

Here  $I = (I_1, \dots, I_M)$  with  $I_m \in \{0, 1\}$  is a multi-index and  $t_I(\phi)$  is a multivariate trigonometric monomial of order  $M$ ,

$$t_I(\phi) = \prod_{m=1}^M t_{I_m}(\phi_m), \quad (4)$$

where each term in the product is defined by

$$t_i(\phi) = \cos^{1-i} \phi \sin^i \phi = \begin{cases} \cos \phi, & i = 0 \\ \sin \phi, & i = 1. \end{cases} \quad (5)$$

We note that coefficient matrices  $U_I$  correspond to the circuit's unitary matrix, evaluated at specific values

$$U_I = U(\phi = \pi I) = \alpha C_M G_M^{I_M} \cdots C_1 G_1^{I_1} C_0, \quad (6)$$

where  $\alpha$  is a phase factor  $\alpha = (-1)^{\sum_m I_m}$ .

### B. Fourier expansion of the loss function

The loss function  $F(\phi)$  of a variational algorithm is defined by the average of some Hermitian operator  $H$ , often referred to as the Hamiltonian, in the state prepared by the circuit

$$F(\phi) = \langle 0 | U^\dagger(\phi) H U(\phi) | 0 \rangle. \quad (7)$$

Here and in the following  $|0\rangle = |0\rangle^{\otimes N}$  is the all-zero state of  $N$  qubits. Substituting the expansion (3) into the loss function gives

$$F(\phi) = \sum_{IJ} t_I \left( \frac{\phi}{2} \right) t_J \left( \frac{\phi}{2} \right) \langle 0 | U_I^\dagger H U_J | 0 \rangle. \quad (8)$$

In contrast to the expansion of the unitary matrix (3), which is homogeneous, expansion of the loss function contains trigonometric monomials of various degrees (see Appendix A for details). Let us organize the Fourier expansion of the loss function by level

$$F(\phi) = \sum_{m=0}^M F_m(\phi). \quad (9)$$

Each function  $F_m(\phi)$  is a homogenous trigonometric polynomial of order  $m$ . There are  $\binom{M}{m}$  possible parameter subsets at level  $m$ , each giving rise to  $2^m$  trigonometric monomials. Hence, the total number of independent coefficients in the Fourier expansion is

$$\sum_{m=0}^M 2^m \binom{M}{m} = 3^M. \quad (10)$$

### C. Fourier terms from averages

Computing the Fourier series for generic loss functions appears to be a formidable task. Indeed, we note that the constant term  $F_0$  in the Fourier expansion can be thought of as the loss function, averaged over all parameters

$$F_0 = \langle F(\phi) \rangle_\phi := \frac{1}{(2\pi)^M} \int_0^{2\pi} \prod_{m=1}^M d\phi_m F(\phi). \quad (11)$$

This relation holds because all higher levels  $F_{m>0}(\phi)$  in the Fourier series trivially vanish when averaged. Higher-level terms can be obtained similarly (see Appendix A).

The average in Eq. (11) can be expressed in a succinct form using orthogonality of trigonometric monomials  $t_I$  (4),

$$\langle t_I(\phi) t_J(\phi) \rangle_\phi = 2^{-M} \delta_{IJ}. \quad (12)$$

Hence, averaging (8) yields

$$F_0 = \frac{1}{2^M} \sum_{I \in \{0,1\}^M} \langle 0 | U_I^\dagger H U_I | 0 \rangle. \quad (13)$$

Evaluating this expression explicitly seems to be out of reach for generic circuits. In terms of a classical simulation, computing any single expectation value in Eq. (13) is difficult on its own for a sufficiently large number of qubits. Even when the averages can be computed efficiently, either classically or provided access to a quantum computer, Eq. (13) still requires summing  $2^M$  terms, infeasible for any significant number of parameters  $M$ . As we show in the next section, for Clifford plus Pauli quantum circuits evaluating  $F_0$  and in fact any particular monomial in the Fourier expansion is classically efficient.

## III. CLIFFORD PLUS PAULI VARIATIONAL CIRCUITS

### A. Definition and properties

Let us first establish some notation relevant for stabilizer circuits. A single-qubit Pauli operator is simply an  $X$ ,  $Y$ , or  $Z$  Pauli matrix or an identity, possibly with a phase  $\pm 1$  or  $\pm i$ . An  $n$ -qubit Pauli operator is a tensor product of  $n$  arbitrary single-qubit Pauli operators. Any two Pauli operators  $P_1$  and  $P_2$  either commute or anticommute:  $P_1 P_2 = \pm P_2 P_1$ . Clifford gates  $C$  are

operators that transform every Pauli gate  $P$  into some Pauli gate  $P'$ :  $C^\dagger P C = P'$ . The group of Clifford gates can be generated by the Hadamard gate  $H$ ,  $S = \sqrt{Z}$ , and controlled-NOT gate. Circuits consisting only of the Clifford gates applied to the stabilizer states (of which  $|0\rangle$  is an example) can be efficiently simulated classically due to the Gottesman-Knill theorem [18].

We define Clifford plus Pauli variational circuits as a subset of variational circuits (1), where generators of parametric gates are Pauli operators and all constant gates are Clifford gates. For clarity of exposition, we also assume that the Hamiltonian  $H$  is a Pauli operator. The case that is most relevant in practice, when the Hamiltonian is a polynomial-size sum of Pauli operators, can be handled by linearity. We stress that both the Pauli generators and the Hamiltonian are allowed to have arbitrary weight, i.e., be supported on any number of qubits. Note that Pauli rotations with generic angles are not Clifford gates, and hence Clifford plus Pauli circuits cannot be simulated efficiently using the stabilizer formalism.

Clifford plus Pauli circuits admit a simple canonical form where all Clifford gates are eliminated. First, one uses commutation properties of Clifford and Pauli operators to drag all the Clifford gates to the very end of the circuit. Generators of Pauli rotations will generally change during the process. The Clifford gate  $C$  accumulated at the end of the circuit is absorbed into the Hamiltonian  $H \rightarrow C^\dagger H C$ , which remains a Pauli operator. Hence, without loss of generality, we assume that Clifford plus Pauli variational circuit takes the Pauli form

$$U(\phi) = P_M(\phi_M) P_{M-1}(\phi_{M-1}) \cdots P_1(\phi_1), \quad (14)$$

where  $P(\phi) = e^{-i(\phi/2)P}$  for some Pauli string  $P$ . We will use notation  $(P_1 \cdots P_M | H)$  for a Clifford plus Pauli circuit in Pauli form. Note that this form is not unique, as Pauli rotations with commuting generators can be swapped.

### B. Computing averages

Now let us revisit the computation of the average loss function (13). Equation (13) now takes the form

$$F_0 = \frac{1}{2^M} \sum_{I \in \{0,1\}^M} \langle 0 | P_1^{I_1} \cdots P_M^{I_M} H P_M^{I_M} \cdots P_1^{I_1} | 0 \rangle. \quad (15)$$

We claim that this sum vanishes unless  $H$  commutes with every  $P_i$ . Indeed, suppose there is some  $P_m$  such that  $P_m H = -H P_m$ . Then it is straightforward to see that any two terms in the sum that only differ in the value of  $I_m$  give opposite contributions. Hence, for the circuit  $(P_1 \cdots P_M | H)$  we find

$$F_0 = \begin{cases} \langle 0 | H | 0 \rangle & \text{for } [H, P_i] = 0 \forall i \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

Therefore, computing the average loss function for Clifford plus Pauli circuits is a trivial task for any number of qubits and any number of parameters. In fact, as we show in the following, this applies to every individual term in the Fourier series. The difficulty of computing the full Fourier expansion then stems solely from the fact that the total number of non-vanishing coefficients can be exponentially large. In the next section we present an efficient classical algorithm to compute the Fourier expansion level by level.

Note that in the current section and in Sec. II C we focused on the constant term  $F_0$  for the sake of simplicity, with the goal of comparing the structure of Clifford plus Pauli and generic circuits. In what follows we will present an algorithm that computes all the coefficients in the Fourier expansion for Clifford plus Pauli circuits and is not limited to the constant term  $F_0$ .

#### IV. CLASSICAL ALGORITHM

##### A. Expansion of the dressed Hamiltonian

We introduce the following notation for an operator conjugated by the circuit's unitary matrix:

$$\mathcal{O}[\phi] = U^\dagger(\phi)\mathcal{O}U(\phi). \quad (17)$$

$$\mathcal{O}[\phi^{(m)}] = \begin{cases} \mathcal{O}[\phi^{(m-1)}], & [\mathcal{O}, P_m] = 0 \\ \mathcal{O}[\phi^{(m-1)}] \cos \phi_m + i(P_m \mathcal{O})[\phi^{(m-1)}] \sin \phi_m, & \{\mathcal{O}, P_m\} = 0. \end{cases} \quad (20)$$

Here  $\phi^{(m)}$  is the subset of the first  $m \leq M$  parameters  $\phi^{(m)} := (\phi_1, \dots, \phi_m)$  and  $\mathcal{O}[\phi^{(m)}]$  is defined as in (17) with the conjugating unitary matrix  $U(\phi^{(m)}) := P_m(\phi_m) \cdots P_1(\phi_1)$ .

Repeatedly applying (20) to  $H[\phi] \equiv H[\phi^{(M)}]$  represents the dressed Hamiltonian as a sum of Pauli strings multiplied by trigonometric monomials, i.e., it gives the Fourier expansion of the dressed Hamiltonian with operator coefficients. The recurrent expansion can be conveniently visualized as a binary tree (see Fig. 1 for an example). The tree is constructed as follows. The nodes, which we refer to as the computational nodes, correspond to variational circuits, specified by a list of Pauli generators

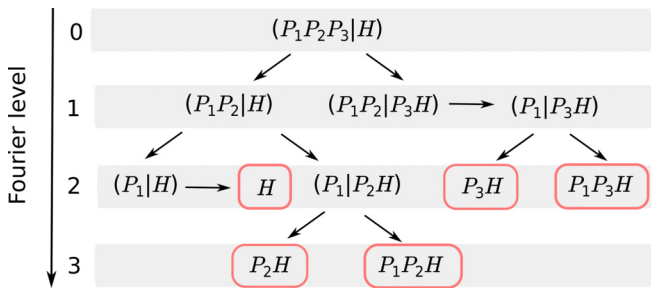


FIG. 1. Sample binary graph representing recursive expansion of a dressed Hamiltonian. The tree consists of nodes  $(P_1 \cdots P_m | \mathcal{O})$ , each representing a variational circuit with generator  $P_i$  and observable  $\mathcal{O}$ . The nodes are constructed and organized by levels, corresponding to the order of Fourier modes. The root node is the original circuit. Each node where the last generator  $P_m$  anticommutes with the observable  $\mathcal{O}$  is split in two at the subsequent level. The last generators commuting with the corresponding observables are simply removed, without branching or increasing the level. Eliminating all Pauli generators in this recursive way leaves the nodes each containing a single operator coefficient in the Fourier expansion of the dressed Hamiltonian. In the example presented  $\{P_3, H\} = \{P_2, H\} = \{P_1, P_3H\} = \{P_1, P_2H\} = \{P_2, P_3H\} = \{P_1, H\} = 0$ .

Following the quantum chemistry literature, we call  $H(\phi)$  the dressed Hamiltonian. The loss function is the average of the dressed Hamiltonian in the all-zero state

$$F(\phi) = \langle 0 | H[\phi] | 0 \rangle. \quad (18)$$

Next we make the following simple observation: For an arbitrary Pauli string  $\mathcal{O}$  it holds that

$$P(\phi)^\dagger \mathcal{O} P(\phi) = \begin{cases} \mathcal{O}, & [P, \mathcal{O}] = 0 \\ \mathcal{O} \cos \phi + i P \mathcal{O} \sin \phi, & \{P, \mathcal{O}\} = 0, \end{cases} \quad (19)$$

i.e., when the conjugating Pauli rotation  $P(\phi)$  commutes with  $\mathcal{O}$ , it cancels out, while for an anticommuting Pauli rotation the result can be written as a sum of two Pauli operators. This gives a recurrence procedure to expand the dressed Hamiltonian. Indeed, it follows from (19) that for an arbitrary Pauli string  $\mathcal{O}$ ,

and an observable  $(P_1 \cdots P_m | \mathcal{O})$ . If the observable  $\mathcal{O}$  anticommutes with the last Pauli generator  $P_m$ , the node branches into two  $(P_1 \cdots P_m | \mathcal{O}) \rightarrow \cos \phi_m (P_1 \cdots P_{m-1} | \mathcal{O}) + i \sin \phi_m (P_1 \cdots P_{m-1} | P_m \mathcal{O})$ . For brevity, we omit coefficients in the diagram. Branching increases the Fourier level by one. If the last Pauli generator instead commutes with  $\mathcal{O}$ , it is simply removed  $(P_1 \cdots P_m | \mathcal{O}) \rightarrow (P_1 \cdots P_{m-1} | \mathcal{O})$  and the Fourier level remains unchanged. We depict this by horizontal arrows in the diagram. When there are no Pauli generators left, the node contains the final observable encoding a single operator coefficient in the Fourier expansion of the dressed Hamiltonian.

The graphical representation makes several distinctive features of Fourier series for Clifford plus Pauli circuits manifest. Let  $n(m)$  be the number of resulting Fourier modes at level  $m$ . Introduce

$$\delta(m) = 2^{-m} n(m), \quad \Delta(m) = \sum_{k=0}^m \delta(k). \quad (21)$$

For any Clifford plus Pauli circuit and any Pauli Hamiltonian it holds that

$$\Delta(M) = \sum_{m=0}^M 2^{-m} n(m) = 1, \quad (22)$$

i.e., the weighted sum of populations at all Fourier levels is an invariant. The statement can be proved by induction. The base case  $M = 0$  is trivial. Next consider a computational tree with  $M + 1 > 0$  levels. Since the root has two children nodes, each giving rise to their own subtree, we can write

$$\Delta(M + 1) = \frac{1}{2} \sum_{m=0}^M 2^{-m} n_1(m + 1) + \frac{1}{2} \sum_{m=0}^M 2^{-m} n_2(m + 1). \quad (23)$$

Here  $n_1(m)$  and  $n_2(m)$  are the numbers of resulting Fourier modes at the level  $m$  within the two subtrees. Each subtree has



at most  $M$  levels, so  $\sum_{m=0}^M 2^{-m} n_i(m+1) = 1$ , completing the proof.

The relation (22) implies certain constraints on the distribution of Fourier terms. For example, the maximum number of Fourier terms  $\sum_m n(m)$  is upper bounded by  $2^M$  (when the last level is fully populated) [cf. the bound for generic circuits  $3^M$  (10)]. Importantly, the presence of any single Fourier term at level  $m < M$  reduces the maximum possible number of terms at other levels. For instance, if  $n(0) = 1$ , i.e.,  $F_0 \neq 0$ , all other Fourier terms vanish.

Let us discuss the complexity of the algorithm. For a single Pauli observable, it amounts to constructing the computational tree in Fig. 1, which has no more than  $\sum_{m=0}^M 2^m = O(2^M)$  nodes. For  $N$ -qubit circuits, generating each node only involves multiplying several Pauli strings of length  $N$ , and hence has complexity  $O(N)$ . Therefore, the time complexity of the algorithm can be bounded by  $O(N2^M)$ . Finally, when the problem Hamiltonian is a linear combination of  $D$  Pauli strings, the algorithm processes each of them independently, giving rise to complexity  $O(DN2^M)$ . This is the worst-case scaling; we will give more detailed estimates for specific problems in the following.

### B. Accounting for expectation values

So far, we have discussed the expansion of the dressed Hamiltonian. In turn, the loss function is given by its expectation value in the all-zero state (18). A likely scenario is that the majority of the final Pauli observables have vanishing expectations, and hence do not contribute to the loss function. This observation allows significantly increasing the efficiency of the computation by pruning unfit branches in advance.

Let  $\mathbb{F}_2^n$  be a vector space of binary strings of length  $n$ . For  $\mathbf{k} = (k_1, \dots, k_n) \in \mathbb{F}_2^n$  define  $\mathbf{Z}(\mathbf{k}) = \bigotimes_{i=1}^n Z^{k_i}$  [ $\mathbf{X}(\mathbf{k})$  is defined similarly]. For any  $n$ -qubit Pauli operator  $P$ , one can define two vectors  $P_Z, P_X \in \mathbb{F}_2^n$  such that

$$P = \alpha \mathbf{Z}(P_Z) \mathbf{X}(P_X), \quad (24)$$

where  $\alpha$  is a phase factor. One can think of  $P_Z$  and  $P_X$  as coordinates of  $P$  in the basis of Pauli  $Z$  and  $X$  strings, respectively. With the notation in place, we can explain how expectation values can be taken into account during the expansion of the dressed Hamiltonian.

A Pauli string  $P$  has a nonzero expectation value  $\langle 0 | P | 0 \rangle \neq 0$  if and only if  $P_X = (0, \dots, 0)$ . First assume, for simplicity, that  $X$ -vectors of the first  $N$  Pauli generators  $(P_1)_X, \dots, (P_N)_X$  are linearly independent and constitute a basis in  $\mathbb{F}_2^N$ . This implies that for every  $\mathcal{O}$  there is a unique vector  $\mathbf{k} \in \mathbb{F}_2^N$  such that

$$\mathcal{O}_X = k_1(P_1)_X + \dots + k_N(P_N)_X. \quad (25)$$

Therefore, among all  $2^N$  possible observables of the form  $P_1^{t_1} \dots P_N^{t_N} \mathcal{O}$ , which can be produced during recursive expansion of  $(P_1 \dots P_N | \mathcal{O})$ , only a single one with  $t_i = k_i$  can yield a nonzero expectation value (all other terms will have a nonzero  $X$  component). Thus, instead of generating the full recursive expansion of  $(P_1 \dots P_N | \mathcal{O})$ , which can contain up to  $2^N$  nodes, we can first find  $\mathbf{k}$  from (25) and then check if this operator actually appears in the dressed Hamiltonian, i.e., is compatible

with the branching rules. This yields an exponential saving for large  $N$ .

Now let us lift the restriction of the first  $N$  Pauli generators forming the basis. The necessary condition for  $(P_1 \dots P_m | \mathcal{O})$  to have a nonzero expectation is that  $\mathcal{O}_X$  is contained in the span of  $(P_1)_X, \dots, (P_m)_X$ . Therefore, for each newly generated computational node, we can test if this condition is satisfied. If it is not, all final observables stemming from the expansion of this node have zero expectation values and the node can be disregarded.

As discussed in Appendix B, there is some room for further optimization based on the freedom to permute commuting Pauli generators. Also, at this point we would like to spell out explicitly an elementary observation about the Fourier series of Clifford plus Pauli circuits. For a single Pauli observable, all nonzero coefficients are given by averages of Pauli strings and hence equal to  $\pm 1$ .

### C. Truncated Fourier series as an approximation

Having many terms at low Fourier levels appears to be convenient, because this partially reduces a proliferation of coefficients at subsequent levels. This is further reinforced by the observation that each individual term at a lower level contributes exponentially more to the loss function than a term at a higher level. Intuitively, this is because the average absolute value of a trigonometric monomial of order  $m$  is  $(\pi/2)^{-m}$  and decays exponentially with degree  $m$  (see Appendix A3). At the same time, there can be exponentially more terms at higher levels. We can quantify this trade-off by evaluating the  $L^2$ -norm of the loss function (the choice of  $L^2$ -norm is simply a matter of convenience, as it is naturally related to the Fourier coefficients). From orthogonality of trigonometric monomials (12) it follows that

$$\|F\|^2 := \langle |F(\phi)|^2 \rangle_\phi = \sum_{m=0}^M 2^{-m} l(m), \quad (26)$$

where  $l(m)$  is the number of nonzero Fourier terms in the expansion of the loss function at level  $m$ , which is upper bounded by the number of nonzero terms in the expansion of the dressed Hamiltonian  $l(m) \leq n(m)$ . Note that  $n(m) - l(m)$  is the number of operators in the dressed Hamiltonian at level  $m$  with zero expectation value (i.e., with nontrivial  $X$  coordinate). Using (22), we can then bound the  $L^2$ -norm of the loss function:

$$\|F\|^2 \leq 1. \quad (27)$$

Similarly, we can bound the approximation error of a truncated the Fourier expansion. Let  $F^{(m)}(\phi)$  denote the Fourier series truncated to the first  $m$  levels. Then

$$\|F^{(m)} - F\|^2 \leq \sum_{k=m+1}^M 2^{-k} l(k) \leq 1 - \Delta(m). \quad (28)$$

If  $\Delta(m)$  is close to 1, i.e., sufficiently many terms are concentrated up to level  $m$ , the truncated Fourier series  $F^{(m)}(\phi)$  gives a good approximation to the full loss function. Note that our recursive expansion generates  $F^{(m)}(\phi)$  level by level, so the quality of the approximation can be gauged dynamically and the computation stopped when the necessary accuracy

is reached. The derivation of the relations (26) and (28) is explicated in Appendix A.3.

We need to mention two caveats related to the approximation result stated. First, while closeness in  $L^2$ -norm guarantees a good approximation for most parameter configurations, it does not translate into pointwise convergence for all parameters. Indeed, while the average absolute value of higher-level monomials (4) is exponentially suppressed, their maximum values are independent of the order  $\max_{\phi} |t_l(\phi)| = 1$ . Note that since  $F(\phi)$  is an expectation value of a Pauli string,  $|F(\phi)| \leq 1$  and hence the contribution of any individual monomial cannot be neglected for every parameter configuration. A more refined truncation procedure compatible with pointwise convergence might be possible, but we do not pursue this question here.

Second, the bound (28) may be too weak in practice. By using the inequality  $l(m) \leq n(m)$  we effectively assume that all the final observables in the dressed Hamiltonian expansion above level  $m$  have nonzero expectation values. In practice, we expect that only a small fraction of observables contribute to the loss function norm, i.e.,  $l(m) \ll n(m)$ . Properly taking this into account can significantly strengthen the bound but requires accounting for the structure of a particular circuit at hand. We illustrate this in a random circuit model discussed in Sec. V.

#### D. Is there a more efficient algorithm?

As sketched in Fig. 2(a), our discussion features four different scales. The largest scale (I) is set by the number of terms in the expansion of the dressed Hamiltonian. It depends only on the structure of the circuit and the Hamiltonian and can contain up to  $2^M$  terms. Note that the total number of nodes in the computational tree Fig. 1 can only exceed the number of final observables by a constant factor, so the recursive expansion algorithm is optimal for computing the Fourier series of the dressed Hamiltonian.

Another scale (III) corresponds to the number of nonzero terms in the Fourier expansion of the loss function. It quantifies the very complexity of describing the loss function by its Fourier series and implies a limit to when such a description can be practical. Also, as discussed in Sec. IV C, the truncated Fourier series can furnish a good approximation to the full loss function while containing only a tiny fraction of all nonzero terms. Hence, we associate a separate scale (IV) with it.

The most relevant scale in practice, however, is set by the number of computational nodes (II). It quantifies the complexity of the algorithm. Without accounting for the expectation values, it simply coincides with the number of terms in the dressed Hamiltonian. In Sec. IV B we explained how to prune the branches of the dressed Hamiltonian expansion, with the expected saving being exponential in the number of qubits. Still, in general, this leaves a large gap between the number of computational nodes and the number of nonzero terms in the loss function. Indeed, assume for simplicity that the first  $N$  Pauli generators span an  $X$  basis and consider a computational node  $(P_1, \dots, P_N | \mathcal{O})$ . For generic  $P_i$  and  $\mathcal{O}$  this node is exponentially unlikely to make a nonzero contribution to the loss function. Indeed, there is a unique combination  $P_1^{k_1} \dots P_N^{k_N} \mathcal{O}$  that has a nonzero expectation value in the all-zero state, but

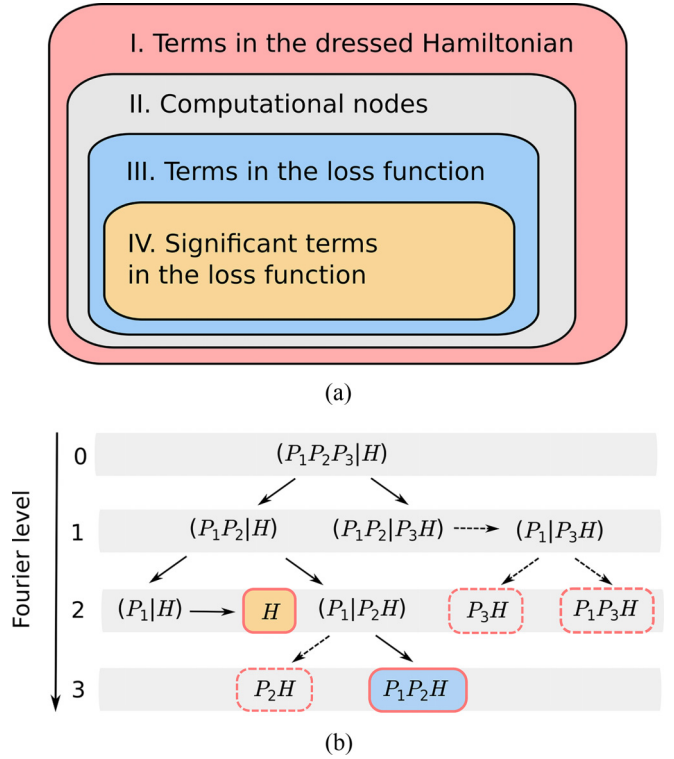


FIG. 2. (a) Sketch of the hierarchy of scales in the problem. (b) Version of the sample computational tree from Fig. 1, highlighting different properties of the nodes. Dashed arrows lead to nodes  $(P_1 \dots P_m | \mathcal{O})$  for which  $\mathcal{O}_X$  does not lie in the span of  $(P_i)_X$ . These nodes do not contribute to the loss function and are not actually generated by the algorithm. The number of remaining nodes quantifies the complexity of the algorithm. Note that the structure of the computational tree does not determine which final observables have nonzero expectation values. In the diagram, we assume that only  $H$  and  $P_1 P_2 H$  contribute to the loss function. If we choose to truncate the Fourier expansion at level 2, only the  $H$  node contributes.

this very combination is unlikely to satisfy all the branching rules and hence actually appears in the recursive expansion. Hence, most nodes of the type  $(P_1, \dots, P_N | \mathcal{O})$  do not contribute to the loss function and are not necessary to generate in the first place. Can a more efficient pruning algorithm be developed?

Let us formalize the question. For Pauli strings  $P_i$  and  $P_j$ , set  $\langle P_i, P_j \rangle = 1$  if they anticommute and  $\langle P_i, P_j \rangle = 0$  otherwise. Note that for any three Pauli strings  $P_i$ ,  $P_j$ , and  $P_k$  it holds that  $\langle P_i, P_j P_k \rangle = \langle P_i, P_j \rangle + \langle P_i, P_k \rangle$  (here and in the following mod2 is implied). All possible final observables in the expansion of the dressed Hamiltonian are of the form

$$\mathcal{O}(k) = P_1^{k_1} \dots P_M^{k_M} H, \quad (29)$$

with some  $k \in \mathbb{F}_2^M$ . In order for  $\mathcal{O}(k)$  to actually appear in the set of final observables,  $k$  has to be consistent with the branching rules. If  $k_i = 1$ , the Pauli string  $P_i$  must anticommute with  $P_{i+1}^{k_{i+1}} \dots P_M^{k_M} H$ , while  $k_i = 0$  implies no constraints. These conditions can be expressed as  $M$  equations ( $i = 1, \dots, M$ )

$$k_i = k_i \langle P_i, H \rangle + \sum_{j=i+1}^M k_i k_j \langle P_i, P_j \rangle. \quad (30)$$

Indeed, for  $k_i = 0$  the relation is trivially satisfied, while for  $k_i = 1$  it is equivalent to the statement that  $P_i$  anticommutes the preceding observable  $\langle P_i, \prod_{j=i+1}^M P_j^{k_j} H \rangle = 1$ .

In addition to satisfying the branching rules, we need to impose that  $\mathcal{O}(k)$  has a nonzero expectation value

$$H_X + \sum_{i=1}^M k_i (P_i)_X = 0. \quad (31)$$

This relation contains  $N$  constraints for an  $N$ -qubit problem.

Together, branching constraints (30) and  $X$  constraints (31) present an instance of a Boolean multivariate quadratic problem (Boolean MQ), which is known to be NP-hard. State-of-the-art algorithms [29,30] have worst-case time complexities around  $2^{0.69M}$  to find a solution or prove one does not exist. Due to the recurrent structure of Eq. (30), our Boolean MQ instances are significantly simpler than the general case. As shown in the next section, for random circuits, which are expected to capture the worst-case behavior in practice, time complexity around  $2^{0.59M-N}$  is sufficient to find all solutions. Thus, while generic algorithms for the Boolean MQ problem are unlikely to be useful directly, there is a possibility that more efficient pruning techniques can be adopted in our scheme, narrowing the gap between the number of computational nodes and nonzero coefficients in the loss function.

## V. CASE STUDIES

So far, we have discussed general properties of the Fourier expansion for Clifford plus Pauli circuits. In this section we consider several specific examples that showcase how the expansions are structured in practice. We will both make analytic estimates and put the classical algorithm to work in numeric simulations.

### A. Random circuits

We first study the case where all the Pauli generators, as well as the Hamiltonian, are random Pauli operators with the support on all of the qubits. In this setup, it is simple to give probabilistic estimates for the expected distribution of Fourier terms in the loss function.

In fact, the assumption that the Pauli generators are random is not necessary as long as the observable is random. Therefore, we expect this behavior to capture well the asymptotic limit of most sufficiently deep circuits. Indeed, even if the original Hamiltonian is local, as we go down the computational tree (see Fig. 1), the observables at the intermediate computational nodes become ever more scrambled and eventually behave as random Pauli operators. The argument is not rigorous, of course, and circuits that do not conform to this pattern can be constructed. Nevertheless, the behavior of random circuits should give a useful reference point for generic circuits.

#### 1. Distribution of terms in the dressed Hamiltonian

We will first look at the coarse-grained characteristics of the dressed Hamiltonian expansion, such as the number of nonzero terms at level  $m$ , denoted by  $n_M(m)$ , and the total number of terms  $n_M = \sum_m n_M(m)$  (here we add a subscript

$M$  to emphasize the dependence on the total number of parametrized gates). When all  $P_m$  and  $H$  are random, the probability of branching at each computational node is  $\frac{1}{2}$ . Therefore, on average, each iteration of the algorithm increases the total number of nodes  $n_M$  by a factor of  $\frac{1}{2} \times 1 + \frac{1}{2} \times 2 = \frac{3}{2}$ , leading to

$$n_M = \left(\frac{3}{2}\right)^M. \quad (32)$$

The same reasoning applies to the number of nonzero computational nodes at each level  $n_M(m)$ , which hence satisfies the recurrence relation  $n_{M+1}(m) = \frac{1}{2}n_M(m) + n_M(m-1)$ . Solving it yields

$$n_M(m) = 2^{m-M} \binom{M}{m}. \quad (33)$$

One can check that  $\sum_{m=0}^M n_M(m) = n_M$ .

#### 2. Distribution of terms in the loss function

So far we have discussed the distribution of Fourier terms in the dressed Hamiltonian and now we turn to the loss function. Since every final observable of the dressed Hamiltonian is again a random Pauli operator, it has  $2^{-N}$  probability of having a nonzero expectation value. Therefore, the distribution of Fourier terms by level  $l_M(m)$  is simply

$$l_M(m) = 2^{-N} n_M(m). \quad (34)$$

The expected number of all nonzero terms in the loss function is

$$l_M = 2^{-N} \left(\frac{3}{2}\right)^M \approx 2^{0.59M-N}. \quad (35)$$

Now we recall that each term at level  $m$  contributes exactly  $2^{-m}$  to the  $L^2$ -norm. From the distribution of Fourier coefficients by level, we can derive the distribution of the norm by the level  $v_M(m) = 2^{-m} l_M(m)$ , explicitly given by

$$v_M(m) = 2^{-N-M} \binom{M}{m}. \quad (36)$$

Note that  $\sum_{m=0}^M v_M(m) = 2^{-N}$ . In Fig. 3(a) we plot the results of numerical simulations for random circuits, which convincingly confirm our estimates. Details of numerical simulations are specified in Appendix C 1, where the distributions (34) and (36) are also shown for several different values of  $M$ .

#### 3. Accuracy of a truncated expansion

We can now address the question of how many terms need to be included in the loss function to give a good  $L^2$  approximation. Since the binomial distribution (36) is symmetric around  $m = \frac{M}{2}$ , including Fourier terms up to level  $\frac{M}{2}$  will on average account for 50% of the norm. The total number of nodes at the included levels can be estimated as  $l_M(\frac{M}{2}) \sim 2^{M/2-N}$ . While in the large- $M$  limit this is an exponentially small fraction of all terms (35), the number of relevant terms itself is still an exponential in  $M$ .

#### 4. Complexity of the algorithm and simulation limits

An exponential increase in the number of relevant terms with  $M$  clearly limits the depth of the circuits we can address.

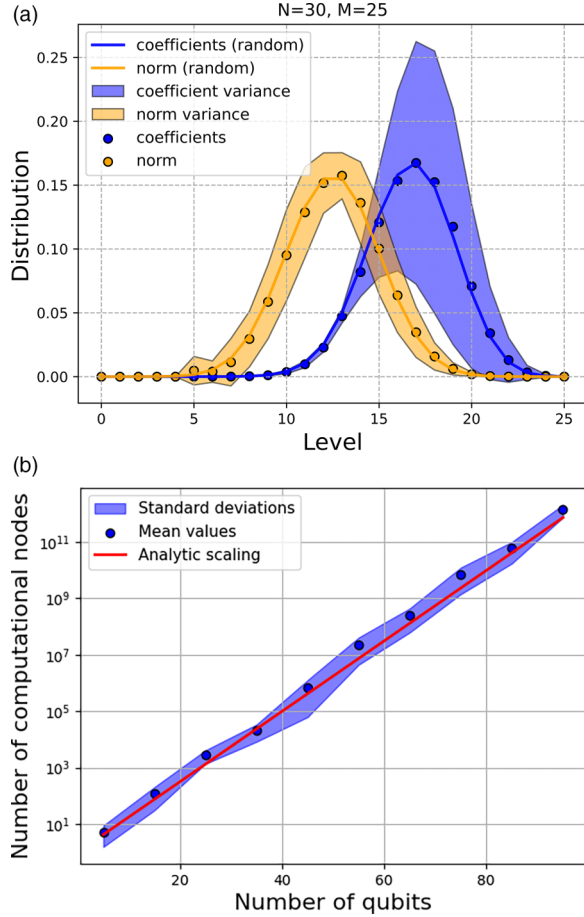


FIG. 3. (a) Level distribution of the number of terms  $l(m)$  and norm  $v(m)$  (both normalized) in Fourier expansion of the dressed Hamiltonian for random circuits. Scatter plots are mean values computed from the simulations; filled areas quantify standard deviations. Solid curves are theoretical predictions. (b) Complexity of the algorithm for random Pauli circuits as a function of the number of qubits  $N$  with depth  $M = N/\log_2 \frac{3}{2}$ . Details of numerical experiments are discussed in Appendix C 1.

Importantly, the property that most final observables have zero expectation values in turn limits the number of qubits  $N$  we can meaningfully simulate. While in principle the number of qubits is only limited by the simulation cost of Clifford circuits, to yield a nonzero loss function the number of Pauli rotation gates  $M$  needs to increase with  $N$ .

Requiring the number of nonzero terms in the loss function (35) to stay of  $O(1)$  as we increase  $N$  requires scaling the depth of the circuit as  $M \sim N/\log_2 \frac{3}{2}$ . Due to the subroutine filtering by the expectation value, the algorithm only branches during processing of the first  $M - N$  gates, leading to a number of computational nodes around approximately  $(\frac{3}{2})^{M-N} \simeq 2^{0.59(M-N)}$ . Therefore, the number of computational nodes generated per nonzero term in the loss function grows with the number of qubit roughly as

$$2^{0.41N} \approx 10^{N/8}. \quad (37)$$

As reported in Fig. 3(b), this scaling is confirmed numerically.

With a computational budget to process  $10^6$  nodes, which takes minutes with a basic implementation run on a laptop, the

Fourier expansion of the loss function for a 50-qubit random circuit with 85 Pauli rotation gates can be computed exactly. With resources to process  $10^{12}$  nodes, which should be feasible with an efficient implementation on a computational cluster, 100-qubit circuits with about 170 Pauli gates can be handled.

## B. QAOA

Variational circuits appearing in practice are far from the random Pauli model described above. Instead, they typically involve only local gates and observables. If two local Pauli strings are supported on different subsets of qubits they necessarily commute, and hence the probability of two generic local operators anticommuting is much smaller than  $\frac{1}{2}$ . Although we expect the random Pauli model to describe well the large depth asymptotic, circuits of shorter depth may behave quite differently. Indeed, until the observables at computational nodes become sufficiently scrambled, the branching probability is much smaller than  $\frac{1}{2}$  and the complexity growth is much slower. This allows us to compute Fourier expansions for circuits with much higher depths than anticipated for the random model.

As a case study, we consider instances of the QAOA for the max-cut problem on regular graphs [14], which is the most studied approach to combinatorial optimization. Two key characteristics of a particular QAOA circuit are the degree of a graph  $d$  and the number of layers  $p$  (see Appendix C 2 for details).

Due to locality, shallow instances of the QAOA allow for efficient classical computation of the loss function independently of the number of qubits. With every observable one associates a reverse light cone, containing all qubits that are connected to the observable by the entangling gates. For graphs of bounded degree, the size of the reverse light cone stays constant in the large- $N$  limit and classical computation of the loss function never involves simulating quantum circuits larger than that size. For a QAOA on a graph of degree  $d$  with  $p$  layers, the size of the reverse light cone is bounded by [14]

$$N_c = 2 \frac{(d-1)^{p+1} - 1}{d-2}. \quad (38)$$

Note that  $N_c$  scales exponentially with the level  $p$ , so the large- $p$  regime is the most difficult to simulate.

Large  $p$  also implies a large number of parametric gates, which is the key limiting factor for our algorithm. At the same time, while the locality of gates is not necessary in our approach, it certainly helps. Moreover, the benefits of locality are incorporated automatically. Indeed, the Pauli generators supported outside the reverse light cone of the Hamiltonian commute with all observables at the computational nodes and are trivially eliminated. Using a basic Monte Carlo sampling, we estimate the expected complexity of our algorithm to compute the full Fourier expansion of the loss function for several values of  $d$  and  $p$ . Results are reported in Table I; details of numerical simulations are specified in Appendix C 2. Though estimates are very crude, they provide useful anchors for expected complexities.

We stress here that the special structure of QAOA circuits and observables makes it possible to handle much larger num-



TABLE I. Estimated number of computational nodes to exactly compute Fourier expansion for QAOA circuits of varying degree and level. Statistics are collected over different graphs and observables (see Appendix C2 for details).

Degree \ Level	1	2	3	4
2	$10^{0.7 \pm 0.08}$	$10^{1.5 \pm 0.9}$	$10^{3.4 \pm 1.7}$	$10^{4.7 \pm 2.6}$
3	$10^{1.1 \pm 0.1}$	$10^{3.6 \pm 0.4}$	$10^{8.6 \pm 0.9}$	$10^{17.5 \pm 2.2}$
4	$10^{1.4 \pm 0.3}$	$10^{5.6 \pm 0.5}$	$10^{16.4 \pm 1.3}$	$10^{36.6 \pm 3.0}$

bers of parameters compared to random circuits. Indeed, for a  $d$ -regular graph with  $N$  nodes there are exactly  $|E| = Nd/2$  edges, so the number of parametric gates in our simulations is given by

$$M = p(N_c + |E|) = pN_c \left( \frac{d}{2} + 1 \right). \quad (39)$$

For instance, for  $p = 3$  and  $d = 3$  this gives  $N_c = 30$  and  $M = 225$ , and a random circuit in this setup would require processing approximately  $(\frac{3}{2})^{M-N_c} \simeq 2.18 \times 10^{34}$  computational nodes, clearly an unmanageable amount. In contrast, for the actual QAOA instances, we find about  $10^9$  computational nodes to be sufficient. The distribution of Fourier terms in the QAOA appears to be somewhat irregular although with clustering characteristics similar to the random circuits of equivalent complexity (see Appendix C2 for details).

### C. Hardware-efficient circuits

One frequently studied design of variational algorithms is the hardware-efficient form [15], where the circuit is constructed to give the maximal expressivity with limited depth, efficiently using native hardware gates. We consider hardware-efficient circuits with brick wall architecture, where each block is built of an entangling controlled-Z (CZ) gate and four single-qubit Pauli rotations, and observables of weight 2 (see Appendix C3 for details). In this setting, we estimate the complexity of our algorithm to exactly compute the loss function and report the results in Fig. 4. The takeaway is similar to the QAOA case: Locality of the circuit and the observable strongly reduces the number of Fourier terms compared to the random case, so the computational budget needed to process random circuits with  $M \sim 180$  should be sufficient to fully characterize the Fourier series for circuits with  $M \sim 600$  parametric gates in the current local setup.

## VI. DISCUSSION AND OUTLOOK

We looked at some qualitative and quantitative aspects of Fourier series expansion of VQA loss functions. The main observation is that restricting our study to the class of Clifford plus Pauli circuits allows us to provide a much finer picture than possible for generic VQAs. We presented an efficient classical algorithm for computing the Fourier expansion level by level, with the worst-case complexity bounded by  $O(N2^M)$  for a single Pauli observable. We estimated the complexity of the algorithm and characterized the distribution of Fourier terms in several examples, including random nonlocal Clifford plus Pauli circuits as well as more conventional local

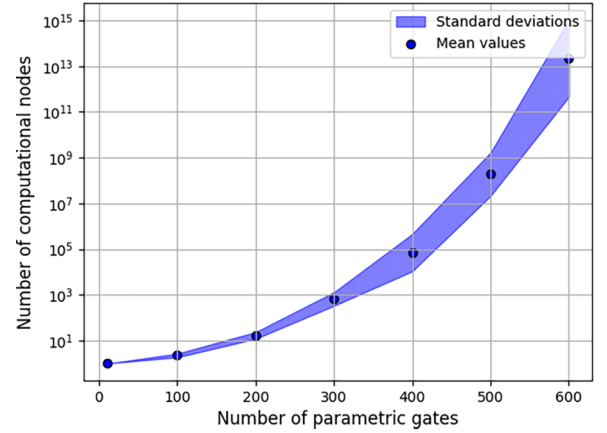


FIG. 4. Estimated algorithm complexity for computing the full Fourier series of a two-local hardware efficient circuit with 50 qubits and Pauli Hamiltonian of weight 2, as a function of the number of parametric gates. Details are specified in Appendix C3.

circuits such as the QAOA and HEA. We anticipate our findings will provide useful tools for further study of the interplay between the properties of VQAs and the structure of their Fourier series expansion.

One major issue facing VQAs is trainability, where two crucial obstacles are barren plateaus [31,32] and local minimums [13,33]. Although many ansatz structures and initialization and optimization heuristics have been proposed to mitigate these problems (we refer the reader to [34] for a summary and to [35,36] for a discussion of the overparametrized setting), they persist in many practical scenarios. Interestingly, the Fourier series contains a wealth of global information about the loss landscape and may hence give a useful perspective of these problems. For instance, we note that the variance of the loss function gradient, frequently used to diagnose barren plateaus, is naturally related to the Fourier coefficients

$$\text{var}_\phi[\nabla F(\phi)]^2 = - \int d\phi F(\phi) \Delta F(\phi) = \sum_{m=0}^M m l(m). \quad (40)$$

[To arrive at this expression first integrate by parts and then use the fact that for every trigonometric monomial  $t_m(\phi)$  of degree  $m$  it holds that  $\Delta t_m(\phi) = -m t_m(\phi)$ .] We note an interesting recent work [37] that uses a related technique to diagnose barren plateaus.

Therefore, already coarse-grained characteristics of the Fourier expansion, such as the distribution of terms by level  $l(m)$ , may provide a very useful input. Note that this distribution can be estimated by a Monte Carlo sampling, even when the full computation of the loss function may be out of reach. Also, it would be very interesting to evaluate the role of lower- vs higher-order Fourier modes in shaping the loss landscape. Superficially, since the higher-order modes are more oscillatory yet typically contribute less to the  $L^2$ -norm of the loss function, they may be a justified suspect in creating the majority of spurious local minima.

Also, it looks promising to explore the potential of our algorithm for classical computation of quantum mean values. For circuits with local gates and constant depth, classical algorithms exist that scale favorably with the number of qubits

[38]. While gate locality helps in practice, our algorithm is based on the properties of stabilizer circuits and it is not principally constrained by the gate locality. Nor is it constrained by the degree of entanglement [39,40] or the circuit's graph tree width [41], which can be a bottleneck for simulators based on tensor networks. Apparently, our approach is most similar to simulation schemes for circuits dominated by Clifford gates. The parameter ranges that can be handled look similar, involving 50–100 qubits and dozens to hundreds of non-Clifford gates [42,43], but detailed benchmarks are necessary to make a reasonable comparison. We should also stress that our approach is neither a simulation algorithm (in a weak or strong sense) nor an algorithm exclusively computing mean values. Instead, it returns the full (or truncated) Fourier series of a VQA and hence has an interesting character.

We would also like to mention two related works, which appeared recently. In [44] the authors focused on a heuristic algorithm for pointwise approximation of expectation values of Clifford plus Pauli circuits, something we briefly touched on in Sec. IV C. Reference [45], using a technique similar to ours, considered the computation of expectation values in noisy Clifford plus Pauli circuits. A remarkable conclusion of the study is that, in the noisy setting, the simulation cost is polynomial in both the number of qubits and the number of parametric gates.

## ACKNOWLEDGMENTS

We thank Vsevolod Yashin and V. Vijendran for useful discussions. We are grateful for the Priority 2030 program at the National University of Science and Technology MISIS, Project No. K1-2022-027.

## APPENDIX A: STRUCTURE OF A GENERIC FOURIER EXPANSION

### 1. Level expansion and number of terms

For the sake of clarity, here we give a more detailed description of the general structure of the Fourier expansion for the loss function introduced in Sec. II. Let us first illustrate the trigonometric expansion of the unitary matrix (3) for the case with  $M = 2$  angles

$$U(\phi_1, \phi_2) = U_{00} \cos \frac{\phi_1}{2} \cos \frac{\phi_2}{2} + U_{01} \cos \frac{\phi_1}{2} \sin \frac{\phi_2}{2} + U_{10} \sin \frac{\phi_1}{2} \cos \frac{\phi_2}{2} + U_{11} \sin \frac{\phi_1}{2} \sin \frac{\phi_2}{2}. \quad (\text{A1})$$

Note that trigonometric monomials here have the same degree and period  $\omega = 4\pi$  in each variable. Substituting such expansions into the definition of the loss function (7) leads to the Fourier series expansion of the loss function. Applying identities  $\cos^2 \frac{\phi}{2} = \frac{1+\cos \phi}{2}$ ,  $\sin^2 \frac{\phi}{2} = \frac{1-\cos \phi}{2}$ , and  $\cos \frac{\phi}{2} \sin \frac{\phi}{2} = \frac{\sin \phi}{2}$  leads to an expression involving trigonometric monomials of a smaller period  $\frac{\omega}{2} = 2\pi$  and degrees up to  $M$ . For instance,

$$\begin{aligned} F(\phi_1, \phi_2) &= \langle 0|U^\dagger(\phi_1, \phi_2)HU(\phi_1, \phi_2)|0\rangle \\ &= \frac{1}{4} \langle 0|U_{00}^\dagger HU_{00}|0\rangle (1 + \cos \phi_1 + \cos \phi_2 \\ &\quad + \cos \phi_1 \cos \phi_2) + \dots \end{aligned} \quad (\text{A2})$$

and we wrote explicitly only the contribution from the first term.

More generally, terms in the expansion (9) assume the form

$$F_0 = \text{const}, \quad F_1(\phi) = \sum_{i=1}^M F_i(\phi_i), \quad F_2(\phi) = \sum_{i,j=1}^M F_{ij}(\phi_i, \phi_j), \quad F_3(\phi) = \sum_{i,j,k=1}^M F_{ijk}(\phi_i, \phi_j, \phi_k), \quad (\text{A3})$$

etc. Here

$$\begin{aligned} F_i(\phi_i) &= A_i \cos \phi_i + B_i \sin \phi_i, \\ F_{ij}(\phi_i, \phi_j) &= A_{ij} \cos \phi_i \cos \phi_j + B_{ij} \cos \phi_i \sin \phi_j \\ &\quad + C_{ij} \sin \phi_i \cos \phi_j + D_{ij} \sin \phi_i \sin \phi_j, \end{aligned} \quad (\text{A4})$$

etc. At each level  $m$  there are  $\binom{M}{m}$  subsets of parameters, enumerating possible indices of homogeneous polynomials  $F_{i_1 \dots i_m}$ . Defining the polynomial for each parameter configuration requires specifying  $2^m$  coefficients. This leads to the counting (10) for the total number of coefficients in the Fourier series.

### 2. Coefficients from averages

Trivially,  $F_0 = \langle F(\phi) \rangle_\phi$ . Higher-order terms can be obtained similarly. For instance,  $F_i(\phi_i) = \langle F(\phi) - F_0 \rangle_{\phi \neq \phi_i}$ , i.e., averaging over all angles except  $\phi_i$  leaves only first-order monomials involving a given angle  $\phi_i$ . Higher-order terms can be found recursively. Note that, in this prescription, comput-

ing terms at level  $m$  requires one to first compute and subtract the contribution of all levels below  $m$ .

### 3. Fourier series and $L^2$ -norm

Here we fill in some of the technical detail omitted in Sec. IV C. First, let us evaluate the average absolute value  $\langle |t_I(\phi)| \rangle_\phi$  of an order- $m$  trigonometric monomial  $t_I(\phi) = \prod_{j=1}^m t_{I_j}(\phi_j)$  with  $I = (I_1, \dots, I_m) \in \{0, 1\}^m$  [see Eq. (4)]. Since  $\langle |t_I(\phi)| \rangle_\phi = \prod_{j=1}^m \langle |t_{I_j}(\phi_j)| \rangle_{\phi_j}$  and  $\langle |\sin(\phi)| \rangle_\phi = \langle |\cos(\phi)| \rangle_\phi = \frac{1}{2\pi} \int_0^{2\pi} d\phi |\cos \phi| = \frac{2}{\pi} \int_0^{\pi/2} d\phi \cos \phi = \frac{2}{\pi}$ , we have  $\langle |t_I(\phi)| \rangle_\phi = (\frac{2}{\pi})^m$ , as claimed in the main text.

Next let us explain the relations (26) and (28) in more detail. A general Fourier expansion in our case can be written as

$$F(\phi) = \sum_{m=0}^M \sum_{I_m \in \{0,1\}^m} c_{I_m} t_{I_m}(\phi), \quad (\text{A5})$$

i.e., we split the sum over levels  $m$ , and within each level trigonometric monomials  $t_{I_m}$  have the same order. Trigonometric monomials  $t_{I_m}$  of different orders are

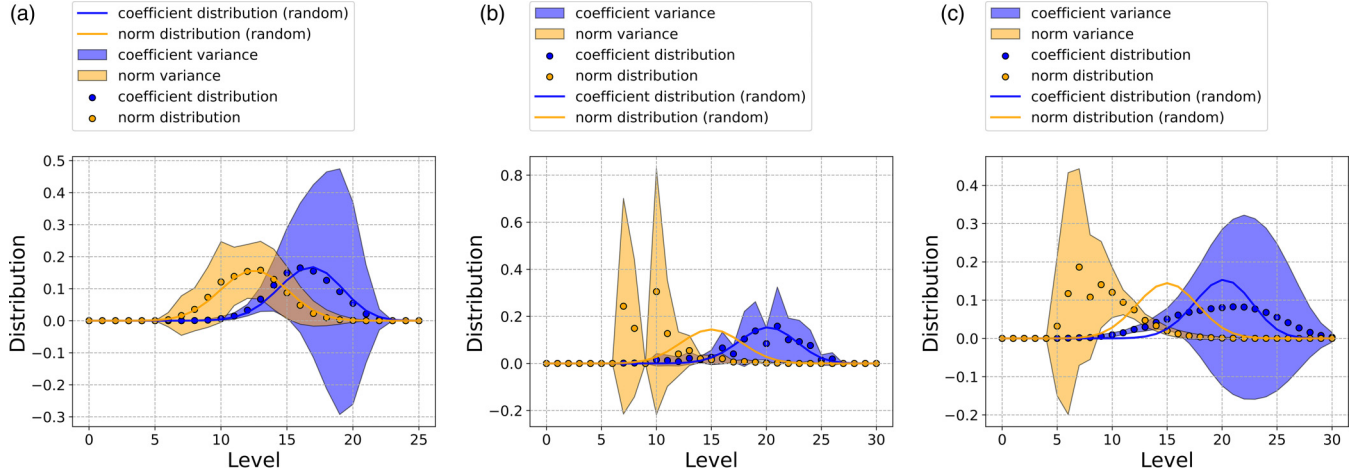


FIG. 5. Normalized distribution of nonzero Fourier coefficients in (a) local circuits with random observables, (b) QAOA circuits with  $d = 3$  and  $p = 2$ , and (c) brick wall HEA circuits with  $N = 50$  and  $M = 304$ . Statistics are collected for random circuits structures (except for the HEA, where it is fixed) and random observables. Solid lines are theoretical curves for random circuits of the equivalent complexity.

orthogonal and those of the same order satisfy (12). The coefficients  $c_{I_m}$  are averages of Pauli strings, so either  $c_{I_m} = 0$  or  $|c_{I_m}| = 1$ . It follows that

$$\|F\|^2 = \langle |F(\phi)|^2 \rangle_\phi = \sum_{m=0}^M 2^{-m} \sum_{I_m \in \{0,1\}^m} |c_{I_m}|^2. \quad (\text{A6})$$

By definition,  $\sum_{I_m \in \{0,1\}^m} |c_{I_m}|^2$  is equal to  $l(m)$ , the number of nonzero terms at level  $m$ . This leads to (26).

To arrive at (28), we repeat this computation with

$$F(\phi) - F^{(k)}(\phi) = \sum_{m=k}^M \sum_{I_m \in \{0,1\}^m} c_{I_m} t_{I_m}(\phi), \quad (\text{A7})$$

yielding

$$\|F(\phi) - F^{(k)}(\phi)\|^2 = \sum_{m=k}^M 2^{-m} l(m). \quad (\text{A8})$$

Recall that  $l(m)$  is the number of nonzero coefficients in the loss function at level  $m$ , which is upper bounded by  $n(m)$ , the number of terms in the dressed Hamiltonian at that level. Using  $l(m) \leq n(m)$  and the definition (21) of  $\Delta(m)$ , we obtain

$$\|F(\phi) - F^{(k)}(\phi)\|^2 \leq \sum_{m=k}^M 2^{-m} n(m) = 1 - \Delta(k). \quad (\text{A9})$$

## APPENDIX B: OPTIMIZING PAULI ORDER

A Pauli form of a Clifford plus Pauli circuit is not unique, because adjacent Pauli rotations with commuting generators can be swapped. In principle, the number of computational nodes may be sensitive to this ordering. We briefly mention two possible optimizations along these lines.

(i) *Delayed branching.* The first strategy may be to delay branching as long as possible, by swapping the anticommuting Pauli to the left. For illustration, consider a Pauli circuit  $(P_1 \cdots P_N P'_1 \cdots P'_N | \mathcal{O})$  where all Pauli generators mutually commute and in addition  $[P_i, \mathcal{O}] = 0$ . Generators  $P'_i$  may not

commute with  $\mathcal{O}$ , while processing operators  $P'_i$  up to  $2^N$  nodes may be generated. After that, assuming that the  $P_i$  span an  $X$  basis, no new nodes will be produced. Had we started with the circuit  $(P'_1 \cdots P'_N P_1 \cdots P_N | \mathcal{O})$  instead, which is equivalent by assumption, generators  $P_i$  would be eliminated right away and no branching would be required (assuming the  $P'_i$  also span an  $X$  basis).

(ii) *Early pruning.* The freedom to swap commuting Pauli generators can also be used to enforce early pruning. As an illustration, consider the circuit  $(PP'_1 \cdots P'_m | \mathcal{O})$ , where  $P$  commutes with all  $P'_i$  and, moreover,  $P$  is an independent generator, in the sense that  $P_X$  does not lie in the span of  $(P'_i)_X$ . If we process the circuit directly, up to  $2^m$  computational nodes are generated and then tested against compatibility with the last generator  $P$ . However, in the equivalent setting  $(P'_1 \cdots P'_m P | \mathcal{O})$  compatibility with  $P$  is tested right away and on average half of the observables  $\mathcal{O}$  will be pruned, leading on average to  $2^{m-1}$  computation nodes. If there are  $n$  generators of  $P$  type, savings up to  $2^{-n}$  can be expected.

Note that both optimizations rely on the presence of large streaks of commuting Pauli generators. Therefore, for the non-local random Pauli circuits very little is to be gained, while for structured local circuits (e.g., the QAOA) the savings might be substantial.

## APPENDIX C: DETAILS OF NUMERICAL COMPUTATIONS

An implementation of the algorithm and the data presented in the paper are available at the GitHub repository [46].

### 1. Random circuits

The statistics in Fig. 3(a) are collected from 20 random nonlocal Pauli circuits with  $N = 30$  qubits and depth  $M = 25$ . Note that the distribution of the norm  $v(m)$  is not an independent characteristic but is computed from the distribution of nodes  $l(m)$  according to  $v(m) = 2^{-m} l(m)$ . In Fig. 5(a) we also plot the distribution of Fourier terms for random circuits,

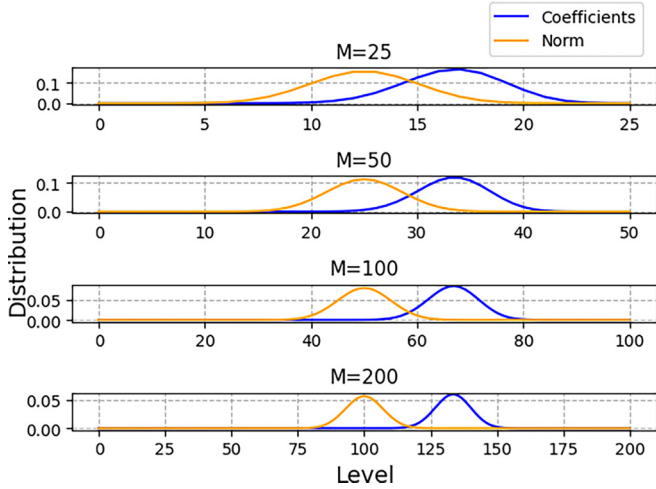


FIG. 6. Distribution of coefficients and norm (both normalized) for various numbers of the parameter  $M$ .

where only the observable is nonlocal, while the circuit consists of random Pauli exponentials of weight 2. As expected, the average values are the same as for the nonlocal random circuits, but the fluctuations around the mean are much higher.

We also depict the theoretical prediction for the average distribution of coefficients and nodes for several values of  $M$  in Fig. 6. Adjusted for the number of levels, the distributions do not shift as  $M$  varies but become more concentrated. Note that the overlap between the plots of coefficient and norm distributions quickly decreases with  $M$ . However, as explained in Sec. V A, the total number of terms contributing non-negligibly to the norm still grows exponentially with  $M$ .

## 2. QAOA

Given a graph  $G$  with edges  $E_{ij}$ , a single layer of the QAOA circuit consists of two-qubit Pauli rotation gates  $Z_i Z_j (\gamma_{ij}) = \exp(-i Z_i Z_j \gamma_{ij} / 2)$  (here  $Z_i$  stands for a Pauli string with  $Z$  on the  $i$ th position and identities on all others) for each edge  $E_{ij}$ , followed by the sequence of single-qubit  $X_i (\beta_i)$  gates placed on every qubit [see Fig. 7(a) for an example]. Note that in the standard formulation of the QAOA,  $\gamma_{ij} = \gamma$  and  $\beta_i = \beta$ , i.e., all  $ZZ$  gates and all  $X$  gates have the same parameters within each layer. However, taking this correlation into account does not simplify our analysis and we will not impose it. A single layer repeated  $p$  times gives an instance of the QAOA with  $p$  layers. The Hamiltonian is given by the sum of all Pauli  $Z$  generators  $H = \sum_{E_{ij}} Z_i Z_j$ .

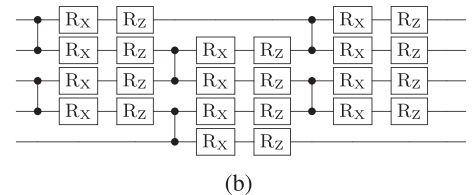
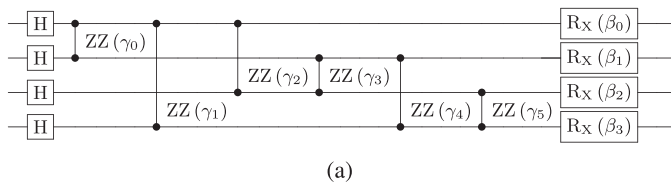


FIG. 7. Circuit layouts for (a) the single-layer QAOA instance on a 3-regular graph with four nodes and (b) the hardware-efficient brick wall circuit with five qubits (all rotation gates have different parameters, not indicated explicitly).

To estimate the complexity of computing the Fourier series for instances of the QAOA of degree  $d$  with  $p$  layers, it is sufficient to consider circuits of size  $N_c$  (38). The statistics presented in Table I were collected in the following way. For each configuration  $(d, p)$  we generate 20 random regular graphs of degree  $d$  and for each graph choose a single observable  $Z_i Z_j$  at random. Then for each circuit-observable pair we estimate the complexity of the algorithm using the Monte Carlo technique (Appendix C 4) with  $10^4$  samples. Fluctuations are significant both due to inaccuracies of the Monte Carlo sampling but more importantly due to inhomogeneous nature of data gathered over different graphs and observables. To represent large fluctuations more meaningfully, we compute average values and deviations in logarithmic scale, i.e., for the exponents of the expected number of computations nodes, hence the format used in Table I. We stress that the numbers reported are estimated numbers of computational nodes, i.e., the expected complexity of the algorithm. The number of nonzero terms in the loss function is less by orders of magnitude, but also harder to estimate with reasonable precision.

Besides the overall complexity, it might be of interest to look at the particular distributions of the Fourier terms in the loss function. To this end, in Fig. 5(b) we plot distributions of nonzero Fourier coefficients by level. The statistics are gathered for 100 different instances of  $d = 3$  and  $p = 2$  QAOA circuits, with a single randomly chosen QAOA observable for each circuit. The theoretical curve for random circuits with  $M = 30$  is plotted for comparison. We note that while the distribution of the coefficients looks close enough to the random case, the distribution of the norm has significant differences and large fluctuations. In particular, the norm appears to be concentrated at lower Fourier levels than expected for random circuits (after adjusting for the distribution of terms), which might make the truncated schemes quite useful.

## 3. Hardware-efficient circuits

An example of a brick wall hardware-efficient circuit that we consider is shown in Fig. 7(b). The statistics for Fig. 4 are collected for  $N = 50$  qubit circuits with up to  $M = 600$  rotation gates. For each  $M$  we take ten random observables of weight 2 and estimate the number of computational nodes with  $10^4$  Monte Carlo samples. Similarly to the QAOA case, due to large individual fluctuations, we compute the means and deviations in logarithmic scale.

In Fig. 5(c) we also plot the distribution of Fourier coefficients for  $N = 50$  circuits with  $M = 308$  parametric gates (corresponding to 77 CZ gates), averaged over 100 random



Pauli observables of weight 2. Similarly to the QAOA case, the most interesting part is the norm distribution, which shows large fluctuations and also the tendency to cluster at lower levels (compared to a random circuit with a similar distribution of coefficients).

#### 4. Estimating complexity from Monte Carlo sampling

It is useful to estimate the runtime or complexity of the algorithm without performing the full computation. To do this, we can use a simple version of Monte Carlo sampling to probe the structure of the computational tree in Fig. 1. First, we describe a version that allows us to estimate the number of terms in the expansion of the dressed Hamiltonian. To this end, we can use the basic algorithm to traverse the computational tree, but instead of keeping both branches at each split, we choose one at random and disregard the other. This procedure produces a single complete branch from the computational tree. The probability to generate any particular branch is  $2^{-m}$ , with  $m$  the number of splits performed in the

process. Let  $s(m)$  be the number of samples with  $m$  splits. We estimate the total number of terms  $n$  as

$$n \approx \frac{\sum_{m=0}^M 2^m s(m)}{\sum_{m=0}^M s(m)}. \quad (\text{C1})$$

Note that in this case the number of splits  $m$  is the same as the Fourier level of the resulting term.

To estimate the complexity of the actual algorithm, i.e., the number of computation nodes generated during the computation, we need to take into account the pruning based on the expectation values (see Sec. IV B). The sampling prescription above is modified in a simple way. We traverse the computational tree and choose branches at random. Importantly, we choose only from the branches compatible with the pruning restrictions. If one of the branches is not admissible, the actual algorithm will not generate additional nodes and the probability of the sample does not need to be updated. In this case, the Fourier level of the resulting term does not correspond to the probability of sampling. The prescription (C1) remains unchanged.

- 
- [1] 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, *Nat. Rev. Phys.* **3**, 625 (2021).
  - [2] J. Preskill, *Quantum* **2**, 79 (2018).
  - [3] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, *Rev. Mod. Phys.* **94**, 015004 (2022).
  - [4] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, *Nature (London)* **549**, 195 (2017).
  - [5] M. Schuld, R. Sweke, and J. J. Meyer, *Phys. Rev. A* **103**, 032430 (2021).
  - [6] F. J. Gil Vidal and D. O. Theis, *Front. Phys.* **8**, 297 (2020).
  - [7] P. Atchade-Adelomou and K. Larson, *arXiv:2302.00105*
  - [8] B. Casas and A. Cervera-Lierta, *Phys. Rev. A* **107**, 062612 (2023).
  - [9] F. J. Schreiber, J. Eisert, and J. J. Meyer, *arXiv:2206.11740*.
  - [10] J. Landman, S. Thabet, C. Dalyac, H. Mhiri, and E. Kashefi, *arXiv:2210.13200*.
  - [11] E. Fontana, I. Rungger, R. Duncan, and C. C. Cîrstoiu, *arXiv:2208.05958*.
  - [12] E. Fontana, I. Rungger, R. Duncan, and C. C. Cîrstoiu, *arXiv:2206.08811*.
  - [13] L. Bittel and M. Kliesch, *Phys. Rev. Lett.* **127**, 120502 (2021).
  - [14] E. Farhi, J. Goldstone, and S. Gutmann, *arXiv:1411.4028*.
  - [15] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, *Nature (London)* **549**, 242 (2017).
  - [16] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, *Quantum Sci. Technol.* **4**, 014008 (2018).
  - [17] D. Gottesman, Ph.D. thesis, California Institute of Technology, 1997.
  - [18] S. Aaronson and D. Gottesman, *Phys. Rev. A* **70**, 052328 (2004).
  - [19] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, *Chem. Rev.* **119**, 10856 (2019).
  - [20] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, *Chem. Rev.* **120**, 12685 (2020).
  - [21] M. D. Sapova and A. K. Fedorov, *Commun. Phys.* **5**, 199 (2022).
  - [22] K. Mitarai, Y. Suzuki, W. Mizukami, Y. O. Nakagawa, and K. Fujii, *Phys. Rev. Res.* **4**, 033012 (2022).
  - [23] M. H. Cheng, K. E. Khosla, C. N. Self, M. Lin, B. X. Li, A. C. Medina, and M. S. Kim, *arXiv:2207.01539*.
  - [24] G. S. Ravi, P. Gokhale, Y. Ding, W. M. Kirby, K. N. Smith, J. M. Baker, P. J. Love, H. Hoffmann, K. R. Brown, and F. T. Chong, *arXiv:2202.12924*.
  - [25] I. G. Ryabinkin, R. A. Lang, S. N. Genin, and A. F. Izmaylov, *J. Chem. Theory Comput.* **16**, 1055 (2020).
  - [26] J. Brown, M. P. Coons, E. Lloyd, A. Fleury, K. Bieniasz, V. Senicourt, and A. Zaribafian, *arXiv:2211.10501*.
  - [27] P. Schleich, J. Boen, L. Cincio, A. Anand, J. S. Kottmann, S. Tretiak, P. A. Dub, and A. Aspuru-Guzik, *J. Chem. Theor. Comput.* **19**, 4952 (2023).
  - [28] S. Hadfield, Ph.D. thesis, Columbia University, 2018.
  - [29] I. Dinur, in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms*, edited by D. Marx (SIAM, Philadelphia, 2021), pp. 2550–2564.
  - [30] S. Barbero, E. Bellini, C. Sanna, and J. Verbel, *Discrete Appl. Math.* **309**, 13 (2022).
  - [31] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, *Nat. Commun.* **9**, 4812 (2018).
  - [32] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, *Nat. Commun.* **12**, 1791 (2021).
  - [33] E. R. Anschuetz and B. T. Kiani, *Nat. Commun.* **13**, 7760 (2022).
  - [34] Y. Wang, B. Qi, C. Ferrie, and D. Dong, *arXiv:2302.06858*.

- [35] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M. Cerezo, *Nat. Comput. Sci.* **3**, 542 (2023).
- [36] J. Kim, J. Kim, and D. Rosa, *Phys. Rev. Res.* **3**, 023203 (2021).
- [37] V. Heyraud, Z. Li, K. Donatella, A. L. Boité, and C. Ciuti, [arXiv:2302.04649](https://arxiv.org/abs/2302.04649).
- [38] S. Bravyi, D. Gosset, and R. Movassagh, *Nat. Phys.* **17**, 337 (2019).
- [39] G. Vidal, *Phys. Rev. Lett.* **91**, 147902 (2003).
- [40] Y. Zhou, E. M. Stoudenmire, and X. Waintal, *Phys. Rev. X* **10**, 041038 (2020).
- [41] I. L. Markov and Y. Shi, *SIAM J. Comput.* **38**, 963 (2008).
- [42] S. Bravyi and D. Gosset, *Phys. Rev. Lett.* **116**, 250501 (2016).
- [43] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard, *Quantum* **3**, 181 (2019).
- [44] T. Begušić, K. Hejazi, and G. K.-L. Chan, [arXiv:2306.04797](https://arxiv.org/abs/2306.04797).
- [45] E. Fontana, M. S. Rudolph, R. Duncan, I. Rungger, and C. Cîrstoiu, [arXiv:2306.05400](https://arxiv.org/abs/2306.05400).
- [46] N. Nemkov, E. Kiktenko, and A. Fedorov, FourierVQA (2023), <https://github.com/idnm/FourierVQA>.