



A semi-agnostic ansatz with variable structure for variational quantum algorithms

M. Bilkis^{1,2} · M. Cerezo^{2,3} · Guillaume Verdon^{4,5} · Patrick J. Coles² · Lukasz Cincio²

Received: 30 January 2023 / Accepted: 4 October 2023
© The Author(s) 2023

Abstract

Quantum machine learning—and specifically Variational Quantum Algorithms (VQAs)—offers a powerful, flexible paradigm for programming near-term quantum computers, with applications in chemistry, metrology, materials science, data science, and mathematics. Here, one trains an ansatz, in the form of a parameterized quantum circuit, to accomplish a task of interest. However, challenges have recently emerged suggesting that deep ansatzes are difficult to train, due to flat training landscapes caused by randomness or by hardware noise. This motivates our work, where we present a variable structure approach to build ansatzes for VQAs. Our approach, called VAns (Variable Ansatz), applies a set of rules to both grow and (crucially) remove quantum gates in an informed manner during the optimization. Consequently, VAns is ideally suited to mitigate trainability and noise-related issues by keeping the ansatz shallow. We employ VAns in the variational quantum eigensolver for condensed matter and quantum chemistry applications, in the quantum autoencoder for data compression and in unitary compilation problems showing successful results in all cases.

Keywords Quantum machine learning · Variational quantum algorithms · Quantum circuit discovery

1 Introduction

Quantum computing holds the promise of providing solutions to many classically intractable problems. The availability of Noisy Intermediate-Scale Quantum (NISQ) devices (Preskill 2018) has raised the question of whether these devices will themselves deliver on such a promise, or whether they will simply be a stepping stone to fault-tolerant architectures.

Parameterized quantum circuits have emerged as one of the best hopes to make use of NISQ devices. Variational

quantum algorithms (VQAs) (Bharti et al. 2022; Cerezo et al. 2021a; Peruzzo et al. 2014) train such circuits to minimize a cost function and consequently accomplish a task of interest. Examples of such tasks are finding ground-states (Peruzzo et al. 2014), solving linear systems of equations (Bravo-Prieto et al. 2019; Huang et al. 2019; Xu et al. 2021), simulating dynamics (Cirstoiu et al. 2020; Commeau et al. 2020; Gibbs et al. 2021; Yuan et al. 2019), factoring (Anshuetz et al. 2019), compiling (Khatri et al. 2019; Sharma et al. 2020), enhancing quantum metrology (Beckey et al. 2022; Koczor et al. 2020), and analyzing principle components (Cerezo et al. 2022; LaRose et al. 2019). More generally, one may employ multiple input states to train the parameterized quantum circuit, and many data science applications have been envisioned for VQAs (Abbas et al. 2021; Biamonte et al. 2017; Schuld et al. 2014; Verdon et al. 2019b).

Despite recent relatively large-scale implementations of VQAs (Arute et al. 2020; Harrigan et al. 2021; Ollitrault et al. 2020), there are still several issues that need to be addressed to ensure that VQAs can provide a quantum advantage on NISQ devices. One issue is trainability. For instance, it has been shown that several VQA architectures become untrainable for large problem sizes due to the existence of the so-called barren plateau phenomenon (Cerezo et al. 2021b; Holmes

✉ M. Bilkis
matias.bilkis@uab.cat; matiasbilkis@gmail.com

¹ Física Teòrica: Informació i Fenòmens Quàntics, Departament de Física, Universitat Autònoma de Barcelona, ES-08193 Bellaterra (Barcelona), Spain

² Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

³ Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM, USA

⁴ Sandbox@Alphabet, Mountain View, CA, USA

⁵ Institute for Quantum Computing, University of Waterloo, Waterloo, ON, Canada

et al. 2021; McClean et al. 2018; Pesah et al. 2021; Thanasilp et al. 2021; Zhao and Gao 2021), which can be linked to circuits having large expressibility (Holmes et al. 2022) or generating large quantities of entanglement (Marrero et al. 2021; Patti et al. 2021; Sharma et al. 2022). The exponential scaling caused by such barren plateaus cannot simply be escaped by changing the optimizer (Arrasmith et al. 2021; Cerezo and Coles 2021). However, some promising strategies have been proposed to mitigate barren plateaus, such as correlating parameters (Volkoff and Coles 2021), layerwise training (Skolik et al. 2021), and clever parameter initialization (Grant et al. 2019; Verdon et al. 2019a).

The other major issue is quantum hardware noise, which accumulates with the circuit depth (Stilck França and Garcia-Patron 2021; Wang et al. 2021). This of course reduces the accuracy of observable estimation, e.g., when trying to estimate a ground state energy. However, it also leads to a more subtle and detrimental issue known as noise-induced barren plateaus (Wang et al. 2021). Here, the noise corrupts the states in the quantum circuit and the cost function exponentially

concentrates around its mean value. Similar to other barren plateaus, this phenomenon leads to an exponentially large precision being required to train the parameters. Currently, no strategies have been proposed to deal with noise-induced barren plateaus. Hence, developing such strategies is a crucial research direction.

Circuit depth is clearly a key parameter for both of these issues. It is, therefore, essential to construct ansatzes that maintain a shallow depth to mitigate noise and trainability issues, but also that have enough expressibility to contain the problem solution. Two different strategies for ansatzes can be distinguished: either using a fixed (Bartlett and Musiał 2007; Cao et al. 2019; Farhi et al. 2014; Hadfield et al. 2019; Kandala et al. 2017) or a variable structure (Chivilikhin et al. 2020; Cincio et al. 2018, 2021; Du et al. 2020; Grimsley et al. 2019; Rattew et al. 2019; Tang et al. 2021; Zhang et al. 2021, 2020b). While the former is the traditional approach, the latter has recently gained considerable attention due to its versatility to address the aforementioned challenges. In variable structure ansatzes, the overall strategy consists of employing

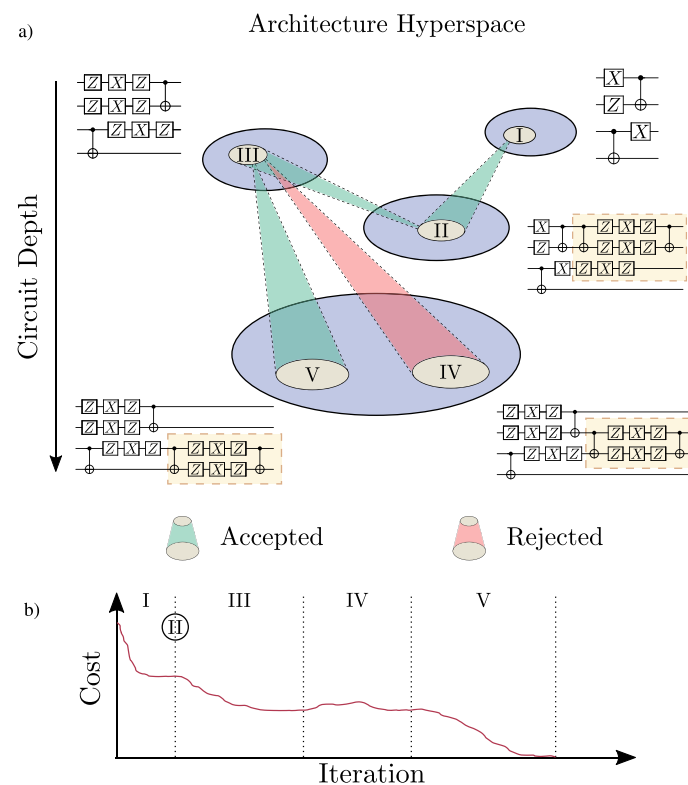


Fig. 1 Schematic diagram of the VAns algorithm. **a** VAns explores the hyperspace of architectures of parametrized quantum circuits to create short depth ansatzes for VQA applications. VAns takes a (potentially non-trivial) initial circuit (step I) and optimizes its parameters until convergence. At each step, VAns inserts blocks of gates into the circuit which are initialized to the identity (indicated in a box in the figure), so that the ansatzes at contiguous steps belong to an equivalence class of circuits leading to the same cost value (step II). VAns then employs a classical algorithm to simplify the circuit by eliminating gates and

finding the shortest circuit (step II to III). The ovals represent subspaces of the architecture hyperspace connected through VAns. While some regions may be smoothly connected by placing identity resolutions, VAns can also explore regions that are not smoothly connected via a gate-simplification process. VAns can either reject (step IV) or accept (step V) modifications in the circuit structure. Here $Z(X)$ indicates a rotation about the $z(x)$ axis. **b** Schematic representation of the cost function value versus the number of iterations for a typical VAns implementation which follows the steps in a)

a machine learning protocol to iteratively grow the quantum circuit by placing gates that empirically lower the cost function. While these approaches address the expressibility issue by exploring specific regions of the ansatz architecture hyperspace, their depth can still grow and lead to noise-induced issues, and they can still have trainability issues from accumulating a large number of trainable parameters.

In this work, we combine several features of recently proposed methods to introduce the Variable Ansatz (VAns) algorithm to generate variable structure ansatzes for generic VQA applications. As shown in Fig. 1, VAns iteratively grows the parameterized quantum circuit by adding blocks of gates initialized to the identity, but also prevents the circuit from over-growing by removing gates and compressing the circuit at each iteration. In this sense, VAns produces shallow circuits that are more resilient to noise, and that have less trainable parameters to avoid trainability issues. Our approach provides a simple yet effective way to address the ansatz design problem, without resorting to resource-expensive computations.

This article is fix-structured as follows. In Sec. 2, we provide background on VQA and barren plateaus, and we present a comprehensive literature review of recent variable ansatz design. We then turn to Sec. 3, where we present the VAns algorithm. In Sec. 4, we present numerical results where we employ VAns to obtain the ground state of condensed matter and quantum chemistry systems. Here, we also show how VAns can be used to build ansatzes for quantum autoencoding applications, a paradigmatic VQA implementation, and demonstrate how VAns can be applied to compile quantum circuits. Moreover, we study the noise resilience of VAns by benchmarking ground-state preparation tasks in the presence of noise. Finally, in Sec. 5, we discuss our results and present potential future research directions employing the VAns algorithm.

2 Background

2.1 Theoretical framework

In this work, we consider generic Variational Quantum Algorithm (VQA) tasks where the goal is to solve an optimization problem encoded into a cost function of the form

$$C(\mathbf{k}, \boldsymbol{\theta}) = \sum_i f_i \left(\text{Tr}[O_i U(\mathbf{k}, \boldsymbol{\theta}) \rho_i U^\dagger(\mathbf{k}, \boldsymbol{\theta})] \right). \quad (1)$$

Here, $\{\rho_i\}$ are n -qubit states forming a training set, and $U(\mathbf{k}, \boldsymbol{\theta})$ is a quantum circuit parametrized by continuous parameters $\boldsymbol{\theta}$ (e.g., rotation angles) and by discrete parameters \mathbf{k} (e.g., gate placements). Moreover, O_i are observables and f_i are functions that encode the opti-

mization task at hand. For instance, when employing the Variational Quantum Eigensolver (VQE) algorithm, we have $f_i(x) = x$ and the cost function reduces to $C(\mathbf{k}, \boldsymbol{\theta}) = \text{Tr}[H U(\mathbf{k}, \boldsymbol{\theta}) \rho U^\dagger(\mathbf{k}, \boldsymbol{\theta})]$, where ρ is the input state (and the only state in the training set) and H is the Hamiltonian whose ground state one seeks to prepare. Alternatively, in a binary classification problem where the training set is of the form $\{\rho_i, y_i\}$, with $y_i \in \{0, 1\}$ being the true label, the choice $f_i(x) = (x - y_i)^2$ leads to the least square error cost.

Given the cost function, a quantum computer is employed to estimate each term in Eq. 1, while the power of classical optimization algorithms is leveraged to solve the optimization task

$$\underset{\mathbf{k}, \boldsymbol{\theta}}{\text{argmin}} C(\mathbf{k}, \boldsymbol{\theta}). \quad (2)$$

The success of a VQA algorithm in solving Eq. 2 hinges on several factors. First, the classical optimizer must be able to efficiently train the parameters, and in the past few years, there has been a tremendous effort in developing quantum-aware optimizers (Arrasmith et al. 2020; Fontana et al. 2020; Koczor and Benjamin 2019; Kübler et al. 2020; Nakanishi et al. 2020; Stokes et al. 2020; Verdon et al. 2018). Moreover, while several choices of observables $\{O_i\}$ and functions $\{f_i\}$ can lead to different faithful cost functions (i.e., cost functions whose global optima correspond to the solution of the problem), it has been shown that global cost functions can lead to barren plateaus and trainability issues for large problem sizes (Cerezo et al. 2021b; Sharma et al. 2022). Here, we recall that global cost functions are defined as ones where O_i acts non-trivially on all n qubits. Finally, as discussed in the next section, the choice of ansatz for $U(\mathbf{k}, \boldsymbol{\theta})$ also plays a crucial role in determining the success of the VQA scheme.

2.2 Barren plateaus

The barren plateau phenomenon has recently received considerable attention as one of the main challenges to overcome for VQA architectures to outperform their classical counterparts. Barren plateaus were first identified in McClean et al. (2018), where it was shown that deep random parametrized quantum circuits that approximate 2-designs have gradients that are (in average) exponentially vanishing with the system size. That is, one finds that

$$\text{Var} \left[\frac{\partial C(\mathbf{k}, \boldsymbol{\theta})}{\partial \theta} \right] \leq F(n), \quad \text{with } F(n) \in \mathcal{O} \left(\frac{1}{2^n} \right), \quad (3)$$

where $\theta \in \boldsymbol{\theta}$. From Chebyshev's inequality, we have that $\text{Var} \left[\frac{\partial C(\mathbf{k}, \boldsymbol{\theta})}{\partial \theta} \right]$ bounds the probability that the cost-function

partial derivative deviates from its mean value (of zero) as

$$\Pr \left[\left| \frac{\partial C(\mathbf{k}, \boldsymbol{\theta})}{\partial \theta} \right| \geq c \right] \leq \frac{\text{Var}[\frac{\partial C(\mathbf{k}, \boldsymbol{\theta})}{\partial \theta}]}{c^2}, \quad (4)$$

for any $c > 0$. Hence, when the cost exhibits a barren plateau, an exponentially large precision is needed to determine a cost minimizing direction and navigate the flat landscape (Arrasmith et al. 2021; Cerezo and Coles 2021).

This phenomenon was generalized in Cerezo et al. (2021b) to shallow circuits, and it was shown that the locality of the operators O_i in Eq. 1 play a key role in leading to barren plateaus. Barren plateaus were later analyzed and extended to the context of dissipative (Sharma et al. 2022) and convolutional quantum neural networks (Pesah et al. 2021; Zhao and Gao 2021), and to the problem of learning scramblers (Holmes et al. 2021). A key aspect here is that circuits with large expressibility (Holmes et al. 2022; Larocca et al. 2022) (i.e., which sample large regions of the unitary group (Sim et al. 2019)) and which generate large amounts of entanglement (Marrero et al. 2021; Patti et al. 2021; Sharma et al. 2022) will generally suffer from barren plateaus. While several strategies have been developed to mitigate the randomness or entanglement in ansatzes prone to barren plateaus (Bharti and Haug 2021; Cerezo et al. 2022; Grant et al. 2019; Pesah et al. 2021; Skolik et al. 2021; Verdon et al. 2019a; Volkoff and Coles 2021; Zhang et al. 2020a), it is widely accepted that designing smart ansatzes which prevent altogether barren plateaus is one of the most promising applications.

Here, we remark that there exists a second method leading to barren plateaus which can even affect smart ansatzes with no randomness or entanglement-induced barren plateaus. As shown in Wang et al. (2021), the presence of certain noise models acting throughout the circuit maps the input state toward the fixed point of the noise model (i.e., the maximally mixed state) (Stilck França and Garcia-Patron 2021; Wang et al. 2021), which effectively implies that the cost function value concentrates exponentially around its average as the circuit depth increases. Explicitly, in a noise-induced barren plateau one now finds that

$$\left| \frac{\partial C(\mathbf{k}, \boldsymbol{\theta})}{\partial \theta} \right| \leq g(n), \quad \text{with } g(l) \in \mathcal{O}\left(\frac{1}{q^l}\right), \quad (5)$$

where $q > 1$ is a noise parameter and l the number of layers. From Eq. 5, we see that noise-induced barren plateaus will be critical for circuits whose depth scales (at least linearly) with the number of qubits. It is worth remarking that Eq. 5 is no longer probabilistic as the whole landscape flattens. Finally, we note that strategies aimed at reducing the randomness of the circuit cannot generally prevent the cost from hav-

ing a noise-induced barren plateau, since here reducing the circuit noise (improving the quantum hardware) and employing shallow circuits seem to be the only viable and promising strategies to prevent these barren plateaus.

2.3 Ansatz for parametrized quantum circuits

Here, we analyze different ansatzes strategies for parametrized quantum circuits and how they can be affected by barren plateaus. Without loss of generality, a parametrized quantum circuit $U(\mathbf{k}, \boldsymbol{\theta})$ can always be expressed as

$$U(\mathbf{k}, \boldsymbol{\theta}) = \prod_j U_{k_j}(\theta_j) W_{k_j}, \quad (6)$$

where W_{k_j} are fixed gates, and where $U_{k_j}(\theta_j) = e^{-i\theta_j G_{k_j}}$ are unitaries generated by a Hermitian operator G_{k_j} and parametrized by a continuous parameter $\theta_j \in \boldsymbol{\theta}$. In a *fixed structure* ansatz, the discrete parameters $k_j \in \mathbf{k}$ usually determine the type of gate, while in a *variable structure* ansatz they can also control the gate placement in the circuit.

2.3.1 Fixed structure ansatz

Let us first discuss fixed structure ansatzes. A common architecture with fixed structure is the layered Hardware Efficient Ansatz (HEA) (Kandala et al. 2017), where the gates are arranged in a brick-like fashion and act on alternating pairs of qubits. One of the main advantages of this ansatz is that it employs gates native to the specific device used, hence avoiding unnecessary depth overhead arising from compiling non-native unitaries into native gates. This type of ansatz is *problem-agnostic*, in the sense that it is expressible enough so that it can be generically employed for any task. However, its high expressibility (Holmes et al. 2022) can lead to trainability and barren plateau issues.

As previously mentioned, designing smart ansatzes can help in preventing barren plateaus. One such strategy are the so-called *problem inspired* ansatzes. Here the goal is to encode information of the problem into the architecture of the ansatz so that the optimal solution of Eq. 2 exists within the parameter space without requiring high expressibility. Examples of these fixed structure ansatzes are the Unitary Coupled Cluster (UCC) Ansatz (Bartlett and Musiał 2007; Cao et al. 2019) for quantum chemistry and the Quantum Alternating Operator Ansatz (QAOA) for optimization (Farhi et al. 2014; Hadfield et al. 2019). However, while these ansatzes might not exhibit expressibility-induced barren plateaus, they usually require deep circuits to be implemented, and hence are very prone to be affected by noise-induced barren plateaus (Wang et al. 2021).

2.3.2 Variable structure ansatzes

To avoid some of the limitations of these fixed structure ansatzes, there has recently been great effort put forward towards developing variable ansatz strategies for parametrized quantum circuits (Chivilikhin et al. 2020; Cincio et al. 2018, 2021; Du et al. 2020; Grimsley et al. 2019; Rattew et al. 2019; Tang et al. 2021; Zhang et al. 2021, 2020b). Here, the overall strategy consists of iteratively changing the quantum circuit by placing (or removing) gates that empirically lower the cost function. In what follows, we briefly review some of these variable ansatz proposals.

The first proposal for variable ansatzes for quantum chemistry was introduced in Grimsley et al. (2019) under the name of ADAPT-VQE. Here, the authors follow a circuit structure similar to that used in the UCC ansatz and propose to iteratively grow the circuit by appending gates that implement fermionic operators chosen from a pool of single and double excitation operators. At each iteration, one decides which operator in the pool is to be appended, which can lead to a considerable overhead if the number of operators in the pool is large. Similarly, to fix structure UCC ansatzes, the mapping from fermions to qubits can lead to prohibitively deep circuits. This issue can be overcome using the qubit-ADAPT-VQE (Tang et al. 2021) algorithm, where the pool of operators is modified in such a way that only easily implementable gates are considered. However, the size of the pool still grows with the number of qubits. In this context, trainability issues have been tackled in Sim et al. (2021), where the parameter optimization is turned into a series of optimizations each performed on a subset of the whole parameter set; an heuristic method is proposed in order to remove gates and add new ones, which in turn allowed the authors to optimize circuits naively containing more than a thousand parameters. In addition, Ref. (Zhang et al. 2021) studies how the pool of operators can be further reduced by computing the mutual information between the qubits in classically approximated ground state. We remark that estimations of the mutual information have also been recently employed to reduce the depth of fixed structured ansatzes (Tkachenko et al. 2021). We refer the reader to Claudino et al. (2020) for a detailed comparison between ADAPT-VQE and UCC ansatzes. Despite constituting a promising approach, it is unclear whether ADAPT-VQE and its variants will be able to overcome typical noise-induced trainability problems as the systems under study are increased in size. Moreover, due to its specific quantum chemistry scope, the application of these schemes is limited.

A different approach to variable ansatzes that has gained considerable attention are machine-learning-aided evolutionary algorithms that upgrade individuals (quantum circuits) from a population. Noticeably, the presence of quantum correlations makes it so that it is not straightforward to combine

features between circuits during the evolution, as simply merging two promising circuits does not necessarily lead to low cost function values. Thus, only random mutations have been considered so far. An example of this method is found in the Evolutionary VQE (EVQE) (Rattew et al. 2019), where one smoothly explores the Hilbert space by growing the circuit with identity-initialized blocks of gates and randomly removing sequences of gates. As suggested by the authors, this might avoid entering regions leading to barren plateaus. Another example of an evolutionary algorithm is the Multi-objective Genetic VQE (MoG-VQE) (Chivilikhin et al. 2020), where one uses block-structured ansatz and simultaneously optimizes both the energy and number of entangling gates. Evolutionary algorithms constitute a promising approach to ansatzes design, they nevertheless come at the cost of high quantum-computational resources to evolve populations of quantum circuits.

A different machine learning approach to discover ansatz structures has been considered in Cincio et al. (2018, 2021) where the goal is to obtain a short depth version of a given unitary. Given specific quantum hardware constraints (such as connectivity, noise-model as represented by quantum channels or available gates), an algorithm grows and modifies the structure of a parametrized quantum circuit to best match the action of the trained circuit with that of the target unitary. At each iteration, a parallelization and compression procedure is applied. This method was able to discover a short-depth version of the swap test to compute state overlaps (Cincio et al. 2018), and in Cincio et al. (2021) it was shown to drastically improve the discovered circuit performance in the presence of noise. Moreover, this technique has recently been tested in large-scale numerics for combinatorial optimization problems (Liu et al. 2021). In addition, in Ostaszewski et al. (2021) the authors present a different iterative algorithm where single-qubit rotations are used for growing the circuit, hence leading to a scheme with limited expressibility power.

Finally, in the recent works of Refs. (Du et al. 2020; Pirhooshyaran and Terlaky 2020; Zhang et al. 2020b) the authors employ tools from auto-machine learning to build variable ansatzes. Specifically, in Du et al. (2020) the authors make use of the supernet and weight sharing strategies from neural network architecture search (Elsken et al. 2019), while in Zhang et al. (2020b) the proposal is based on a generalization of the differentiable neural architecture search (Liu et al. 2018) for quantum circuits. In Pirhooshyaran and Terlaky (2020) the authors study the design of quantum circuits for multi-label classification: policy gradient methods are used to train a neural network, so to decide which gates will compose candidate quantum circuits. We remark that while promising, the latter methods require quantum-computational resources which considerably grow with the problem sizes.

In the next section, we present a task-oriented NISQ-friendly approach to the problem of ansatzes design. In the

context of the literature, the approach presented here generalizes the work in Cincio et al. (2018, 2021) as a method to build potentially trainable short-depth ansatz for Variational Quantum Algorithm tasks. Unlike previous methods, the algorithm introduced here not only grows the circuit but more importantly employs classical routines to remove quantum gates in an informed manner during the optimization. In addition, our method can be used in a wide class of problems within the realm of variational quantum algorithms (where the cost function is usually defined as the expectation of a Hamiltonian), but also within more general contexts (where the loss function can take more general forms such as mean-squared error and log likelihoods). Finally, below, we showcase the performance of VAns in the presence of hardware noise, a benchmark notably absent in many other variable ansatz proposals.

3 The variable ansatz (VAns) algorithm

3.1 Overview

The goal of the VAns algorithm is to adaptively construct shallow and trainable ansatzes for generic quantum machine learning applications using parametrized quantum circuits. Let us define as \mathcal{C}_l the architecture hyperspace of quantum circuits of depth l , where a single layer is defined as gates acting in parallel. VAns takes as input:

- A cost function $C(\mathbf{k}, \boldsymbol{\theta})$ to minimize.
- A dictionary \mathcal{D} of parametrized gates that compile to the identity. That is, for $V(\boldsymbol{\gamma}) \in \mathcal{D}$ there exists a set of parameters $\boldsymbol{\gamma}^*$ such that $V(\boldsymbol{\gamma}^*) = \mathbb{1}$.
- An initial circuit configuration $U^{(0)}(\mathbf{k}, \boldsymbol{\theta}) \in \mathcal{C}_{l_0}$ of depth l_0 .
- Circuit Insertion rules which stochastically take an element $V(\boldsymbol{\gamma}^*) \in \mathcal{D}$ and append it to the circuit. The insertion is a map $\mathcal{I} : \mathcal{C}_l \rightarrow \mathcal{C}_{l'}$ with $l' \geq l$.

- Circuit Simplification rules to eliminate unnecessary gates, redundant gates, or gates that do not have a large effect on the cost. The simplification is a map $\mathcal{S} : \mathcal{C}_l \rightarrow \mathcal{C}_{l'}$ with $l' \leq l$.
- An optimization algorithm for the continuous parameters $\boldsymbol{\theta}$, and an optimization algorithm for the discrete parameters \mathbf{k} .

Given these inputs, VAns outputs a circuit architecture and set of parameters that approximately minimize Eq. 2. In what follows, we describe the overall structure of VAns (presented in Algorithm 1), and in the next sections, we provide additional details for the Insertion and Simplification modules. In all cases, the steps presented here are aimed at giving a general overview of the method and are intended to be used as building blocks for more advanced versions of VAns.

The first ingredient of VAns (besides the cost function, which is defined by the problem at hand) is a dictionary \mathcal{D} of parametrized gates that can compile to identity and which VAns will employ to build the ansatz. A key aspect here is that \mathcal{D} can be composed of any set of gates, so that one can build a dictionary specifically tailored for a given application. For instance, for problems with a given symmetry, \mathcal{D} can contain gates preserving said symmetry (Gard et al. 2020). In addition, it is usually convenient to have the unitaries in \mathcal{D} expressed in terms of gates native to the specific quantum hardware employed, as this avoids compilation depth overheads.

Once the gate dictionary is set, the ansatz is initialized to a given configuration $U^{(0)}(\mathbf{k}, \boldsymbol{\theta})$. As shown in Algorithm 1, one then employs an optimizer to train the continuous parameters $\boldsymbol{\theta}$ in the initial ansatz until the optimization algorithm converges. In Fig. 2, we show two non-trivial initialization strategies employed in our numerical simulations (see Sect. 4). In Fig. 2a, the circuit is initialized to a separable product ansatz which generates no entanglement between the qubits, while in Fig. 2b, one initializes to a shallow alternating Hardware

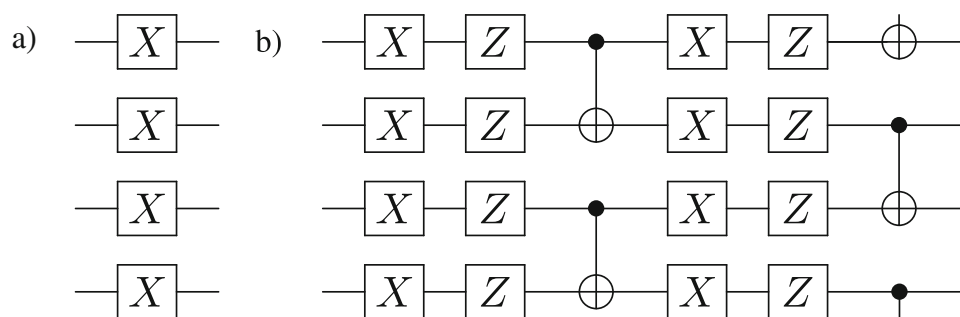


Fig. 2 Examples of initial circuit configurations for VAns. VAns take as input an initial structure for the parametrized quantum circuit. In **a**, we depict a separable product ansatz which generates no entanglement between the qubits. On the other hand, **b** shows two layers of a shallow

alternating Hardware Efficient Ansatz where neighboring qubits are initially entangled. Here, Z (X) indicates a rotation about the z (x) axis

Efficient Ansatz such that neighboring qubits are entangled. While the choice of an appropriate initial ansatz can lead to faster convergence, VAns can in principle transform a simple initial ansatz into a more complex one as part of its architecture search.

From this point, VAns enters a nested optimization loop. In the outer loop, VAns explores the architecture hyperspace to optimize the ansatz's discrete parameters k that characterize the circuit structure. Then, in the inner loop, the ansatz structure is fixed and the continuous parameters θ are optimized.

At the start of the outer loop, VAns employs its **Insertion** rules to stochastically grow the circuit. The fact that these rules are stochastic guarantees that different runs of VAns explore distinct regions of the architecture hyperspace. As previously mentioned, the gates added to the circuit compile to the identity so that circuits that differ by gate insertions belong to an equivalence class of circuits leading to the same cost function value. As discussed below, the **Insertion** rules can be such that they depend on the current circuit they act upon. For instance, VAns can potentially add entangling gates to qubits that were not previously connected via two-qubit gates.

To prevent the circuit from constantly growing each time gates are inserted, VAns follows the **Insertion** step by a **Simplification** step. Here, the goal is to determine if the circuit depth can be reduced without significantly modifying the cost function value in a systematic way, as proposed in Maslov et al. (2008). This is a fundamental step of VAns as it allows the algorithm to explore and jump between different regions of the architecture hyperspace which might not be trivially connected. Moreover, unlike other variable ansatz strategies which continuously increase the circuit depth or which randomly remove gates, the **Simplification** step allows VAns to find short depth ansatzes by deleting gates in an informed manner.

Taken together, **Insertion** and **Simplification** provide a set of discrete parameters k . However, to determine if this new circuit structure can improve the cost function value it is necessary to enter the inner optimization loop and train the continuous parameters θ . When convergence in the optimization is reached, the final cost function value is compared to the cost in the previous iteration. Updates that lead to equi-cost values or to smaller costs are accepted, while updates leading to higher cost functions are accepted with exponentially decaying probability in a manner similar to a Metropolis-Hastings step (Hastings 1970). Here one accepts an update which increases the cost value with a probability given by $\exp(-\beta \frac{\Delta C}{C_0})$, with $\frac{\Delta C}{C_0}$ being increment in the cost function with respect to the initial value, and $\beta > 0$ a “temperature” factor. The previous optimizations in inner and outer loops are repeated until a termination condition f_{Term} is reached, e.g., distance to a target cost function value (if a

Algorithm 1 Pseudo-code for VAns.

Input: Cost function $C(k, \theta)$; initial circuit $U^{(0)}(k, \theta)$; dictionary of gates \mathcal{D} ; **Insertion** rules which take gates from \mathcal{D} and appends them to a circuit; **Simplification** rules; optimization algorithm Opt_C for continuous parameters θ ; optimization algorithm Opt_D for discrete parameters which accepts or rejects an ansatz update given changes in the cost function value; termination condition function $f_{\text{Term}}(n, C(k, \theta), U(k, \theta))$, $n \in \mathbb{N}$

Output: Optimized ansatz $U^{(f)}$.

Init: Randomly initialize the parameters θ ; initialize the ansatz $U^{(f)} \leftarrow U^{(0)}(k, \theta)$; $C^{(f)} \leftarrow 0$; $k^{(f)} \leftarrow k$; $\theta^{(f)} \leftarrow \theta$; $\mathcal{U}(k, \theta) \leftarrow 1$; **Term** \leftarrow **false**; $n \leftarrow 0$.

- 1 Optimize θ with Opt_C and store result in $\theta^{(f)}$; $C^{(f)} \leftarrow C(k, \theta)$.
- 2 **while** **Term** is **false** **do**
- 3 $n \leftarrow n + 1$
- 4 **Accept** \leftarrow **false**
- 5 **while** **Accept** is **false** **do**
- 6 Use **Insertion** in $U^{(f)}$ and store new sets of discrete parameters, continuous parameters and ansatz in θ , k , and $\mathcal{U}(k, \theta)$, respectively.
- 7 Use **Simplification** rules 1-5 on $\mathcal{U}(k, \theta)$ and store new sets of discrete parameters, continuous parameters and ansatz in θ , k , and $\mathcal{U}(k, \theta)$, respectively.
- 8 Optimize continuous parameters in $\mathcal{U}(k, \theta)$ with Opt_C and store result in θ ; $C^{(f)} \leftarrow C(k, \theta)$.
- 9 Use **Simplification** rule 6
- 10 Given $C(k, \theta)$ and $C^{(f)}$, optimize discrete parameters in $\mathcal{U}(k, \theta)$ with Opt_D and store result in **Accept**.
- 11 $k^{(f)} \leftarrow k$.
- 12 $\theta^{(f)} \leftarrow \theta$.
- 13 $U^{(f)} \leftarrow \mathcal{U}(k, \theta)$.
- 14 $C^{(f)} \leftarrow C(k, \theta)$.
- 15 **Term** $\leftarrow f_{\text{Term}}(n, C^{(f)}, U^{(f)})$

lower bound is available), maximum VAns iteration number, or an user-specified function that might depend on variables such as circuit structure and cost value reached.

3.2 Insertion method

As previously mentioned, the **Insertion** step stochastically grows the circuit by inserting into the circuit a parametrized block of gates from the dictionary \mathcal{D} which compiles to the identity. It is worth noting that in practice one can allow some deviation from the identity to reach a larger gate dictionary \mathcal{D} . In turn, this permits VAns to explore regions of the architecture hyperspace that could otherwise take several iterations to be reached. In Fig. 3, we show examples of two parametrized quantum circuits that can compile to the identity.

There are many choices for how VAns determines which gates are chosen from \mathcal{D} at each iteration, and where they should be placed. When selecting gates, we have here taken

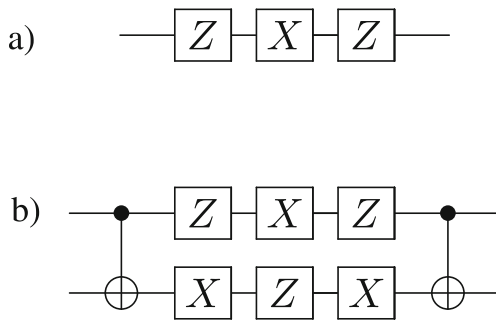
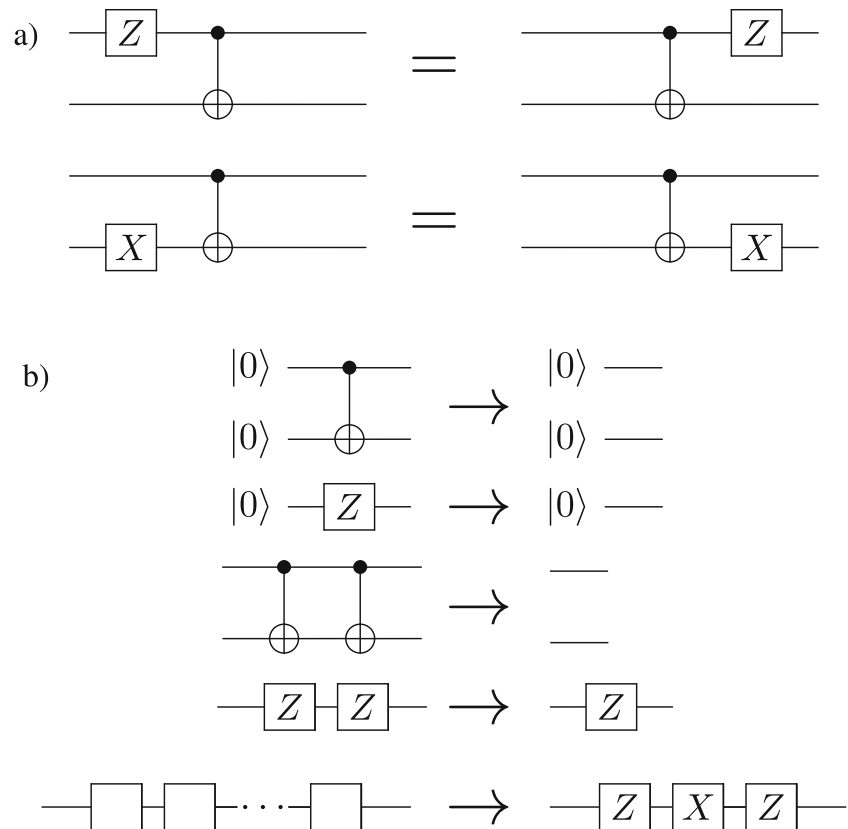


Fig. 3 Circuits from the dictionary \mathcal{D} used during the **Insertion steps**. Here, we show two types of the parametrized gate sequences composed of CNOTs and rotations about the z and x axis. Specifically, one obtains the identity if the rotation angles are set to zero. Using the circuit in **a**, one inserts a general unitary acting on a given qubit, while the circuit in **b** entangles the two qubits it acts upon

a uniform sampling approach, where every sequence of gates in \mathcal{D} has an equal probability to be selected. While one could follow a similar approach for determining where said gates should be inserted, this can lead to deeper circuits with regions containing an uneven number of CNOTs. In our heuristics, VAns has a higher probability to place two-qubit gates acting on qubits that were otherwise not previously connected or shared a small number of entangling gates.

Fig. 4 Rules for the **Simplification steps**. **a** Commutation rules used by VAns to move gates in the circuit. As shown, one can commute a CNOT with a rotation Z (X) about the z (x) axis acting on the control (target) qubit. **b** Example of simplification rules used by VAns to reduce the circuit depth. Here, we assume that the circuit is initialized to $|0\rangle^{\otimes n}$



3.3 Simplification method

The **Simplification** steps in VAns are aimed at eliminating unnecessary gates, redundant gates, or gates that do not have a large effect on the cost. For this purpose, **Simplification** moves gates in the circuit using the commutation rules shown in Fig. 4a to group single qubits rotations and CNOTs together. Once there are no further commutations possible, the circuit is scanned and a sequence of simplification rules are consequently applied. For instance, assuming that the input state is initialized to $|0\rangle^{\otimes n}$, we can define the following set of simplification rules.

1. CNOT gates acting at the beginning of the circuit are removed.
2. Rotations around the z -axis acting at the beginning of the circuit are removed.
3. Consecutive CNOT sharing the same control and target qubits are removed.
4. Two or more consecutive rotations around the same axis and acting on the same qubit are compiled into a single rotation (whose value is the sum of the previous values).
5. If three or more single-qubit rotations are sequentially acting on the same qubit, they are simplified into a general single-qubit rotation of the form $R_z(\theta_1)R_x(\theta_2)R_z(\theta_3)$ or $R_x(\theta_1)R_z(\theta_2)R_x(\theta_3)$ which has the same action as the

previous product of rotations—note that in the figures, we denote such rotations by X or Z —.

6. Gates whose presence in the circuit does not considerably reduce the cost are removed.

Rules (1)–(5) are schematically shown in Fig. 4b. We remark that a crucial feature of these *Simplification* rules is that they can be performed using a classical computer that analyzes the circuit structure and hence do not lead to additional quantum-computation resources.

As indicated by step (6), the *Simplification* steps can also delete gates whose presence in the circuit does not considerably reduce the cost. Here, given a parametrized gate, one can remove it from the circuit and compute the ensuing cost function value. If the resulting cost is increased by more than some threshold value, the gate under consideration is removed and the simplification rules (1)–(5) are again implemented. Here, one can use information from the inner optimization loop to find candidate gates for removal. For instance, when employing a gradient descent optimizer, one may attempt to remove gates whose parameters lead to small gradients. Note that, unlike the simplification steps (1)–(5) in Fig. 4b, when using the deletion process in (6) one needs to call a quantum computer to estimate the cost function, and hence these come at an additional quantum-resource overhead which scales linearly with the number of gates one is attempting to remove.

An interesting aspect of the *Simplification* method is that it allows VAns to obtain circuit structures that are not contained in the initial circuit $U^{(0)}(\mathbf{k}, \boldsymbol{\theta})$ or in the gate dictionary \mathcal{D} , and hence to explore new regions of the architecture hyperspace. For instance, using the gate dictionary in Fig. 3, VAns can obtain a gate structure as the one shown in Fig. 5.

3.4 Scaling of VAns

With the previous overview of the VAns method in mind, let us now discuss the computational complexity arising from

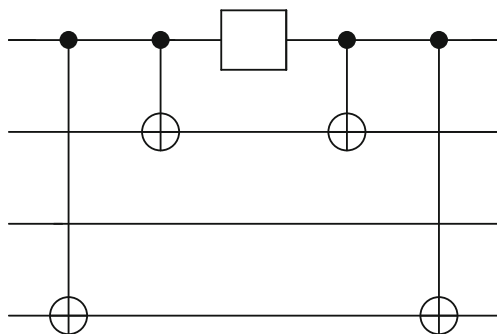


Fig. 5 Circuit obtained from VAns. Shown is a non-trivial circuit structure that can be obtained by VAns using the *Insertion* and *Simplification* steps and the gate dictionary in Fig. 3

using VAns versus that of using a standard fix architecture scheme. In the following discussion, we will not include any computational cost or complexity of the optimizer as we assume that the same tools could be used for fixed or variable ansatzes.

Firstly, we note that any additional computational cost comes due to circuit manipulations, meaning that we should study the scaling of the *Insertion* and *Simplification* methods. On the one hand, adding gates via *Insertion* is stochastic, and independent of the number of qubits or the current circuit depth, that is: its complexity is always in $\mathcal{O}(1)$. Then, removing gates via *Simplification* has a cost which increases with the number of gates in the circuit. If we have M gates, then running the *Simplification* scheme has a cost $\mathcal{O}(M)$. We note that such computational complexity is similar to that of using gradient-free versus gradient-based methods, as the computational cost of the latter also scale as $\mathcal{O}(M)$. Notably, since the goal of VAns is to produce short-depth circuits, the algorithm itself tries to reduce its own computational cost during training. As we will see below in our numerical examples, VAns is always able to find short-depth circuits whose solutions are better than those arising from fixed structure ansatzes, meaning that the extra complexity of VAns can be justified in terms of its performance.

3.5 Mitigating the effect of barren plateaus

3.5.1 General considerations

Let us now discuss why VAns is expected to mitigate the impact of barren plateaus.

First, consider the type of barren plateaus that are caused by the circuit approaching an approximate 2-design (McClean et al. 2018). Approximating a 2-design requires a circuit that both has a significant number of parameters and also has a significant depth (Brandao et al. 2016; Dankert et al. 2009; Haferkamp 2022; Harrow and Low 2009; Harrow and Mehraban 2018). Hence, reducing either the number of parameters or the circuit depth can combat these barren plateaus. Fortunately, VAns attempts to reduce both the number of parameters and the depth. Consequently, VAns actively attempts to avoid approximating a 2-design.

Second, consider the barren plateaus that are caused by hardware noise (Wang et al. 2021). For such barren plateaus, it was shown that the circuit depth is the key parameter, as the gradient vanishes exponentially with the depth. As VAns actively attempts to reduce the number of CNOTs, it also reduces the circuit depth. Hence, VAns will mitigate the effect of noise-induced barren plateaus by keeping the depth shallow during the optimization. Evidence of such a mechanism can be found in the next section, where we consider simulations under the presence of noise. In such a case, VAns

automatically adjusts the circuit layout in such a way that the cost function reaches a minima, which translates to short-depth circuits in noisy scenarios.

3.5.2 Specific applications

While in the previous subsection, we have presented general arguments as to why VAns can improve trainability, here, we instead present a practical method that combines VAns with the recent techniques of Ref. Sack et al. (2022) for mitigating barren plateaus using classical shadows. For completeness, we briefly review the results in Sack et al. (2022). Firstly, as noted in Sect. 2.2 it is known that the presence of barren plateaus is intrinsically related to the entanglement generated in the circuit (Marrero et al. 2021; Patti et al. 2021; Sharma et al. 2022). That is, circuits generating large amounts of entanglement are prone to barren plateaus. With this remark in mind, the authors in Sack et al. (2022) propose to detect the onset of a barren plateau by monitoring the entanglement in the state at the output of the circuit. This can be achieved by computing, via classical shadows (Huang et al. 2020), the second Rényi entanglement entropy $S_2(\rho_R) = -\log(\text{Tr}[\rho_R]^2)$, where $\rho_R = \text{Tr}_{\bar{R}}[U(\mathbf{k}, \boldsymbol{\theta})\rho_i U^\dagger(\mathbf{k}, \boldsymbol{\theta})]$ denotes a reduced state on a subset of R qubits. As such, if $S_2(\rho_R)$ approaches the maximal possible entanglement of the S qubits, given by the so-called Page value $S^{\text{Page}} \sim k \log(2) - \frac{1}{2^{n-2k+1}}$, one knows that the optimization is leading to a region of high entanglement, and thus of barren plateaus. The key proposal in Sack et al. (2022) is to tune the optimizer (e.g., by controlling the gradient step) so that regions of large entropies are avoided. This technique is shown to work well with an identity block initialization (Grant et al. 2019), whereby the parameters in the trainable unitary are chosen such that $U(\mathbf{k}, \boldsymbol{\theta}) = \mathbb{1}$ at the start of the algorithm. Note that, in principle, this is still a fixed ansatz method, as some circuit structure has to be fixed beforehand, and as no gates are ever removed. Hence, the methods in Sack et al. (2022) can be readily combined with VAns to variationally explore the architecture hyperspace while keeping track of the reduced state entropy. In practice, this means that one can modify the VAns update rule to allow for steps that do not significantly increase entropy, while favoring steps that keep the entropy constant, or even that reduce it (e.g., by removing gates during the Simplification modules).

4 Numerical results

In this section, we present heuristic results obtained from simulating VAns to solve paradigmatic problems in condensed matter, quantum chemistry, and quantum machine learning. We first use VAns in the Variational Quantum Eigensolver

(VQE) algorithm (Peruzzo et al. 2014) to obtain the ground state of the Transverse Field Ising model (TFIM), the XXZ Heisenberg spin model, and the H_2 and H_4 molecules. We then apply VAns to a quantum autoencoder (Romero et al. 2017) task for data compression. We then move to use VAns to compile a Quantum Fourier Transform unitary in systems up to 10 qubits. Finally, we benchmark the performance of VAns under the presence of noisy channels, which are unavoidable in quantum hardware, and demonstrate that it successfully finds cost-minimizing circuits on a wide range of noise-strength levels.

The simulations presented here were performed using Tensorflow quantum (Broughton et al. 2020). Adam (Kingma and Ba 2015) and qFactor (Younis and Cincio 2020) were employed to optimize the continuous parameters $\boldsymbol{\theta}$. While the results shown in noiseless scenarios were obtained from a single instance of the algorithm for each problem, noisy simulations required several instances of the algorithm to reach a minima (here, we present results obtained after 50 iterations), due to the fact that the optimization landscape is considerably more challenging in the latter case. The dictionary of gates used consisted on single-qubit rotation around x and z axis, and CNOTs gates between all qubits in the circuit. Moreover, we have assumed no connectivity constraints under the quantum circuits under consideration. In the following examples, VAns was initialized to either a separable ansatz or an ℓ -layer HEA, with $\ell < 3$ (see Fig. 2 for a depiction of separable and 2-layer HEA circuits); in our heuristics, we observed that, as expected, varying the initial circuit helps to attain a quicker convergence towards optimal cost-values. Nevertheless, the optimal choice of initial circuit is subtle: highly expressible initial circuits are not always the best choice, since they might bias the search and even diminish the convergence rate due to the appearance of multiple local minima in the optimization landscape. Moreover, under the presence of noise, long-depth initial circuits such as HEA potentially increase the cost-value function instead of diminishing it, due to noise accumulation. In such a case, it takes VAns a higher amount of iterations to shorten circuit's depth. For the noisy simulations, we have considered a one-layered HEA as initial circuit for VAns.

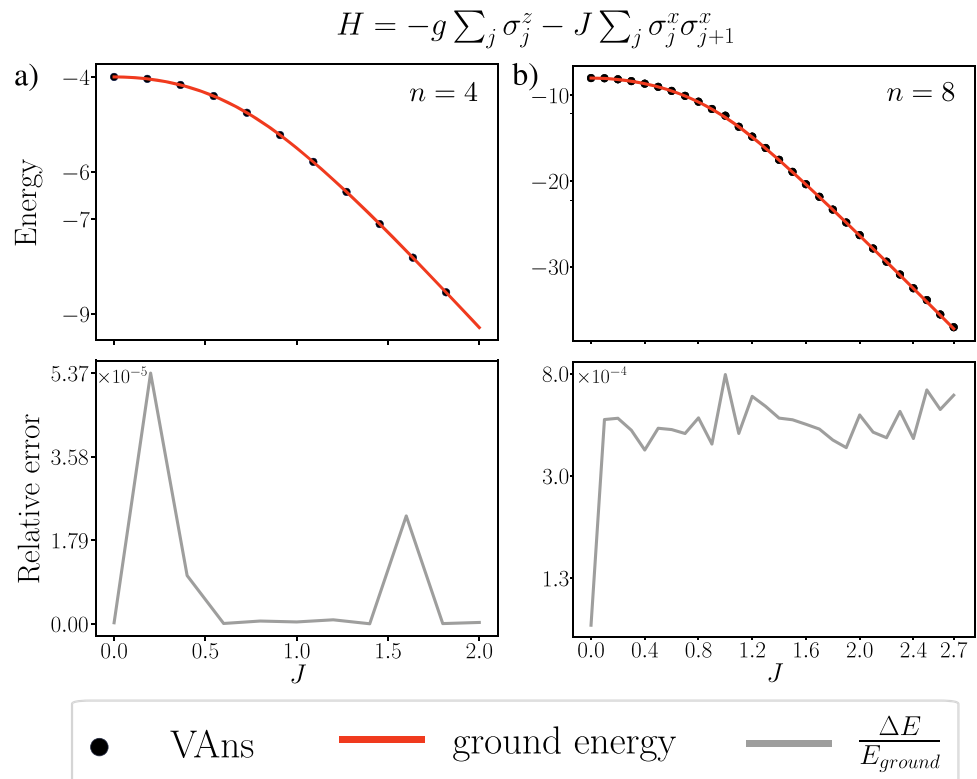
4.1 Transverse field ising model

Let us now consider a cyclic TFIM chain. The Hamiltonian of the system reads

$$H = -J \sum_{j=1}^n \sigma_j^x \sigma_{j+1}^x - g \sum_{j=1}^n \sigma_j^z, \quad (7)$$

where $\sigma_j^{x(z)}$ is the Pauli x (z) operator acting on qubit j , and where $n+1 \equiv 1$ to indicate periodic boundary con-

Fig. 6 Results of using VAns to obtain the ground state of a Transverse Field Ising model. Here, we use VAns in the VQE algorithm for the Hamiltonian in Eq. 7 with **a** $n = 4$ qubits and **b** $n = 8$ qubits, field $g = 1$, for different values of the interaction J . Top panels: solid lines indicate the exact ground state energy, and the markers are the energies obtained using VAns. Bottom panels: Relative error in the energy for the same interaction values



ditions. Here, J indicates the interaction strength, while g is the magnitude of the transverse magnetic field. As mentioned in Sect. 2, when using the VQE algorithm the goal is to optimize a parametrized quantum circuit $U(\mathbf{k}, \boldsymbol{\theta})$ to prepare the ground state of H so that the cost function becomes $E(\mathbf{k}, \boldsymbol{\theta}) = \text{Tr}[HU(\mathbf{k}, \boldsymbol{\theta})\rho U^\dagger(\mathbf{k}, \boldsymbol{\theta})]$, where one usually employs $\rho = |\mathbf{0}\rangle\langle\mathbf{0}|$ with $|\mathbf{0}\rangle = |0\rangle^{\otimes n}$. Note that, we here employ E as the cost function label to keep with usual notation convention.

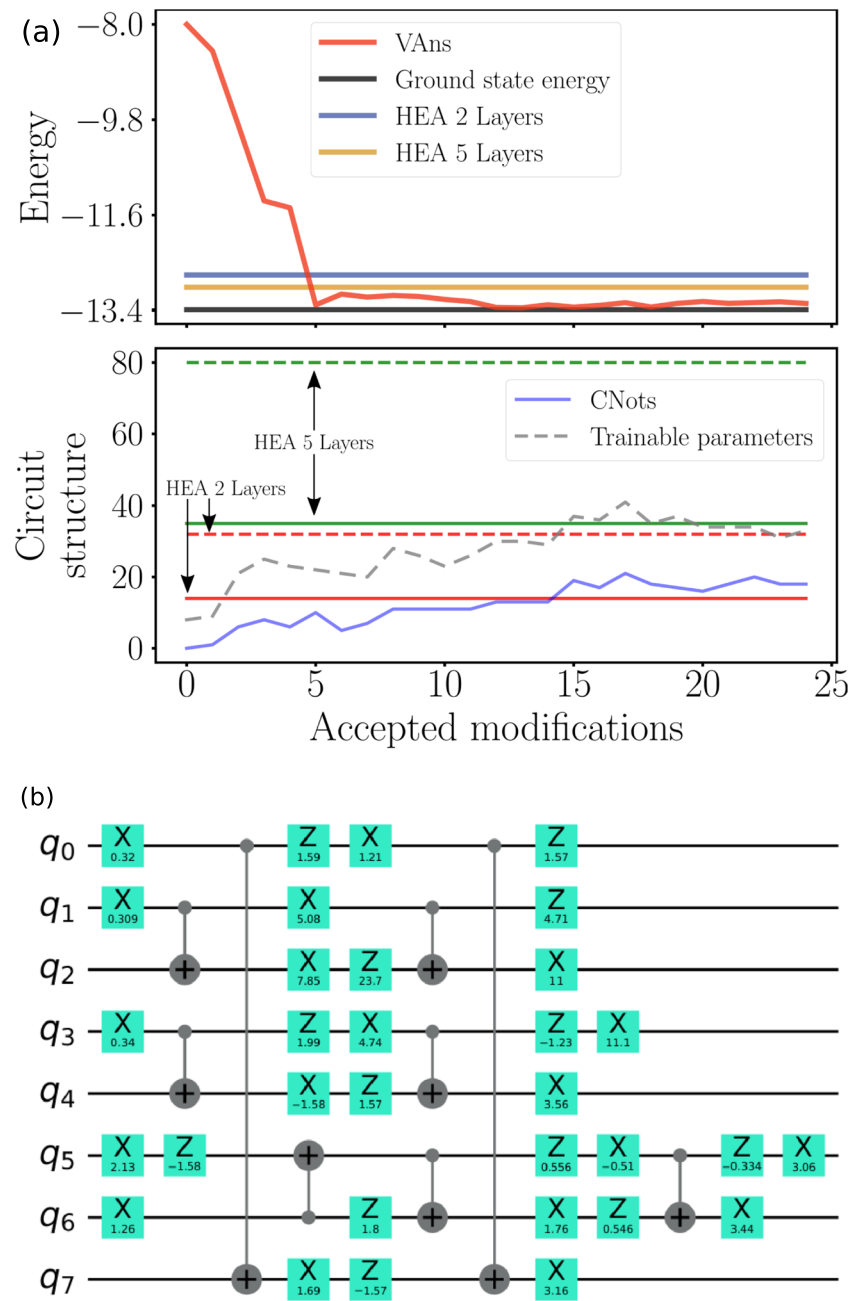
In Fig. 6, we show results obtained from employing VAns to find the ground state of a TFIM model of Eq. 7 with $n = 4$ qubits (a) and with $n = 8$ qubits (b), field $g = 1$, and different interactions values. To quantify the performance of the algorithm, we additionally show the relative error $|\Delta E/E_0|$, where E_0 is the exact ground state energy E_0 , $\Delta E = E_{\text{VAns}} - E_0$, and E_{VAns} the best energy obtained through VAns. For 4 qubits, we see from Fig. 6 that the relative error is always smaller than 6×10^{-5} , showing that the ground state was obtained for all cost values. Then, for $n = 8$ qubits, VAns obtains the ground state of the TFIM with relative error smaller than 8×10^{-4} .

To gain some insight into the learning process, in Fig. 7, we show the cost function value, number of CNOTs, and the number of trainable parameters in the circuit discov-

ered by VAns as different modifications of the ansatz are accepted to minimize the cost in an $n = 8$ TFIM VQE implementation. Specifically, in Fig. 7(top), we see that as VAns explores the architecture hyperspace, the cost function value continually decreases until one can determine the ground state of the TFIM. Figure 7(bottom) shows that initially VAns increases the number of trainable parameters and CNOTs in the circuit via the Insertion step. However, as the circuit size increases, the action of the Simplification module becomes more relevant as we see that the number of trainable parameters and CNOTs can decrease throughout the computation. Moreover, here, we additionally see that reducing the number of CNOTs and trainable parameters can lead to improvements in the cost function value. The latter indicates that VAns can indeed lead to short depth ansatzes which can efficiently solve the task at hand, even without the presence of noise.

In Fig. 7, we also compare the performance of VAns with that of the layered Hardware Efficient Ansatz of Fig. 2b with 2 and 5 layers. We specifically compare against those two fixed structure ansatzes as the first (latter) has a number of trainable parameters (CNOTs) comparable to those obtained in the VAns circuit. In all cases, we see that VAns can produce better results than those obtained with the Hardware Efficient Ansatz.

Fig. 7 VAns learning process. **a** Here, we show an instance of running the algorithm for the Hamiltonian in Eq. 7 with $n = 8$ qubits, field $g = 1$, and interaction $J = 1.5$. The top panel shows the cost function value and the bottom panel depicts the number of CNOTs, and the number of trainable parameters versus the number of modifications of the ansatz accepted in the VAns algorithm. Top: As the number of iterations increases, VAns minimizes the energy until one finds the ground state of the TFIM. Here, we also show the best results obtained by training a fixed structure layered Hardware Efficient Ansatz (HEA) with 2 and 5 layers, and in both cases, VAns outperforms the HEA. Bottom: While initially the number of CNOTs and number of trainable parameters increases, the Simplification method in VAns prevents the circuit from constantly growing, and can even lead to shorter depth circuits that achieve better solutions. Here, we also show the number of CNOTs (solid line) and parameters (dashed line) in the HEA ansatzes considered, and we see that VAns can obtain circuits with less entangling and trainable gates. **b** We show a low-depth, ground-state preparing circuits found by VAns during the learning process; here Z (X) indicates a rotation about the z (x) axis, about the corresponding value appearing below



4.2 XXZ Heisenberg model

Here, we use VAns in a VQE implementation to obtain the ground state of a periodic XXZ Heisenberg spin chain in a transverse field. The Hamiltonian of the system is

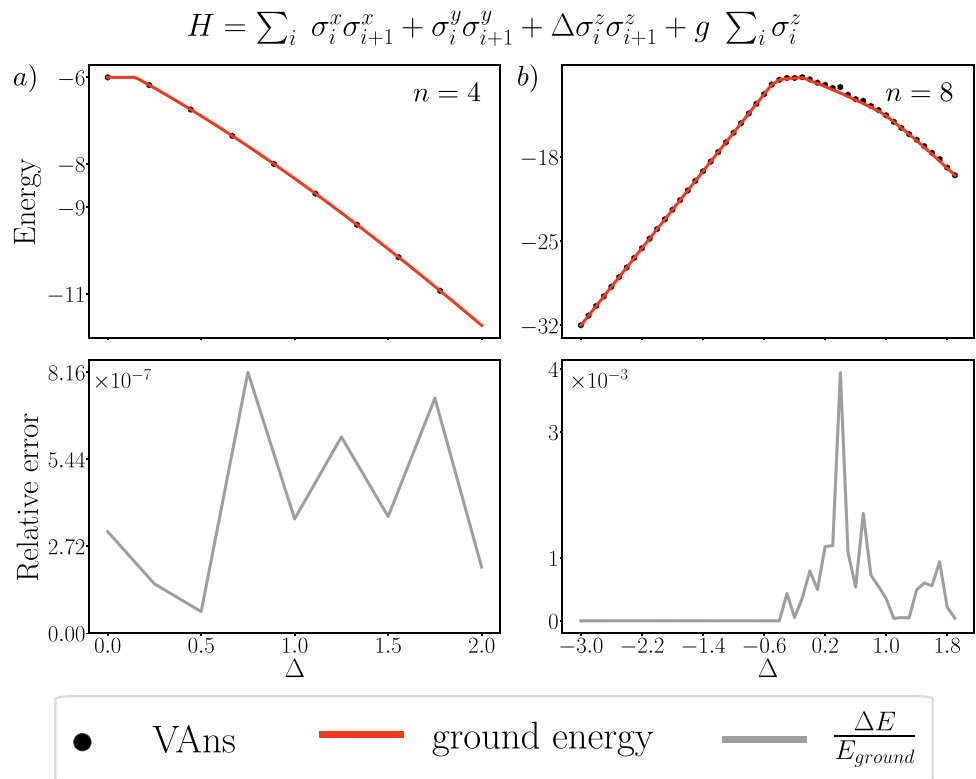
$$H = \sum_{j=1}^n \sigma_j^x \sigma_{j+1}^x + \sigma_j^y \sigma_{j+1}^y + \Delta \sigma_j^z \sigma_{j+1}^z + g \sum_{j=1}^n \sigma_j^z, \quad (8)$$

where again σ_j^μ are the Pauli operator (with $\mu = x, y, z$) acting on qubit j , $n + 1 \equiv 1$ to indicate periodic boundary

conditions, and where Δ is the anisotropy. We recall that H commutes with the total spin component $S_z = \sum_j \sigma_j^z$, meaning that its eigenvectors have definite magnetization M_Z along z (Cerezo et al. 2017).

In Fig. 8, we show numerical results for finding the ground state of Eq. 8 with $n = 4$ and $n = 8$ qubits, field $g = 1$, and for different anisotropy values. For 4 qubits, we see that VAns can obtain the ground state with relative errors which are always smaller than 9×10^{-7} . In the $n = 8$ qubits case, the relative error is of the order 10^{-3} , with error increasing in the region $0 < \Delta < 1$. We remark that a similar phenomenon is observed in Cervera-Lierta et al. (2020), where errors in

Fig. 8 Results of using VAns to obtain the ground state of a Heisenberg XXZ model. Here, we use VAns in the VQE algorithm for the Hamiltonian in Eq. 8 with **a** $n = 4$ and **b** $n = 8$ qubits, field $g = 1$, and indicated anisotropies Δ . Top panels: The solid line indicates the exact ground state energy, and the markers are the energies obtained using VAns. Bottom panels: Relative error in the energy for the same anisotropy values



preparing the ground state of the XXZ chain increase in the same region. The reason behind this phenomenon is that the optimizer can get stuck in a local minimum where it prepares excited states instead of the ground state. Moreover, it can be verified that while the ground state and the three first excited states all belong to the same magnetization sub-space of state with magnetization $M_Z = 0$, they have in fact different local symmetries and structure. Several of the low-lying excited states have a Néel-type structure of spins with non-zero local magnetization along z of the form $|\uparrow\downarrow\uparrow\downarrow\cdots\rangle$. On the other hand, the state that becomes the ground-state for $\Delta > 1$ is a state where all spins have zero local magnetization along z , meaning that the local states are in the xy plane of the Bloch sphere. Since there is a larger number of excited states with a Néel-type structure (and with different translation symmetry) variational algorithms tend to prepare such states when minimizing the energy. Moreover, since mapping a state with non-zero local magnetization along z to a state with zero local magnetization requires a transformation acting on all qubits, any algorithm performing local updates will have a difficult time finding such mapping.

4.3 Molecular hamiltonians

Here, we show results for using VAns to obtain the ground state of the Hydrogen molecule and the H_4 chain. Molecular

electronic Hamiltonians for quantum chemistry are usually obtained in the second quantization formalism in the form

$$H = \sum_{mn} h_{mn} a_m^\dagger a_n + \frac{1}{2} \sum_{mnpq} h_{mnpq} a_m^\dagger a_n^\dagger a_p a_q, \quad (9)$$

where $\{a_m^\dagger\}$ and $\{a_n\}$ are the fermionic creation and annihilation operators, respectively, and where the coefficients h_{mn} and h_{mnpq} are one- and two-electron overlap integrals, usually computed through classical simulation methods (Aspuru-Guzik et al. 2005). To implement Eq. 9 in a quantum computer one needs to map the fermionic operators into qubits operators (usually through a Jordan Wigner or Bravyi-Kitaev transformation). Here, we employed the OpenFermion package (McClean et al. 2019) to map Eq. 9 into a Hamiltonian expressed as a linear combination of n -qubit Pauli operators of the form

$$H = \sum_z c_z \sigma^z, \quad (10)$$

with $\sigma^z \in \{\mathbb{1}, \sigma^x, \sigma^y, \sigma^z\}^{\otimes n}$, c_z real coefficients, and $z \in \{0, x, y, z\}^{\otimes n}$.

In all cases, the basis set used to approximate atomic orbitals was STO-3g and a neutral molecule was considered. The Jordan-Wigner transformation was used in all cases.

While for the H_2 the number of qubits required is four ($n = 4$), this number is doubled for the H_4 chain ($n = 8$).

4.3.1 H_2 Molecule

Figure 9 shows the results obtained for finding the ground state of the Hydrogen molecule at different bond lengths. As shown, VAns is always able to accurately prepare the ground state within chemical accuracy. Moreover, as seen in Fig. 9(bottom), VAns usually requires less than 15 iterations until achieving convergence, showing that the algorithm quickly finds a way through the architecture hyperspace towards a solution.

4.3.2 H_4 Molecule

Figure 10 shows the results obtained for finding the ground state of the H_4 chain, where equal bond distances are taken. Noticeably, the dictionary of gates \mathcal{D} chosen here is not a *chemical-inspired* one (e.g., it does not contain single and double excitation operators nor its hardware-efficient implementations), yet VAns is able to find ground-state preparing circuits within chemical accuracy (McArdle et al. 2020).

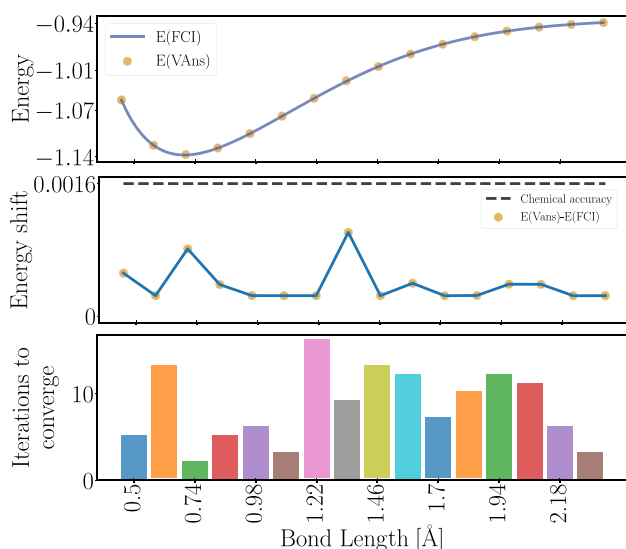


Fig. 9 Results of using VAns to obtain the ground state of a Hydrogen molecule, at different bond lengths. Here, we use VAns in the VQE algorithm for the molecular Hamiltonian obtained after a Jordan-Wigner transformation, leading to a 4-qubit circuit. Top: solid lines correspond to ground state energy as computed by the Full Configuration Interaction (FCI) method, whereas points correspond to energies obtained using VAns. Middle: differences between exact and VAns ground state energies are shown. Dashed line corresponds to chemical accuracy, which stands for the ultimate accuracy experimentally reachable in such systems. Bottom: number of iterations required by VAns until convergence are shown

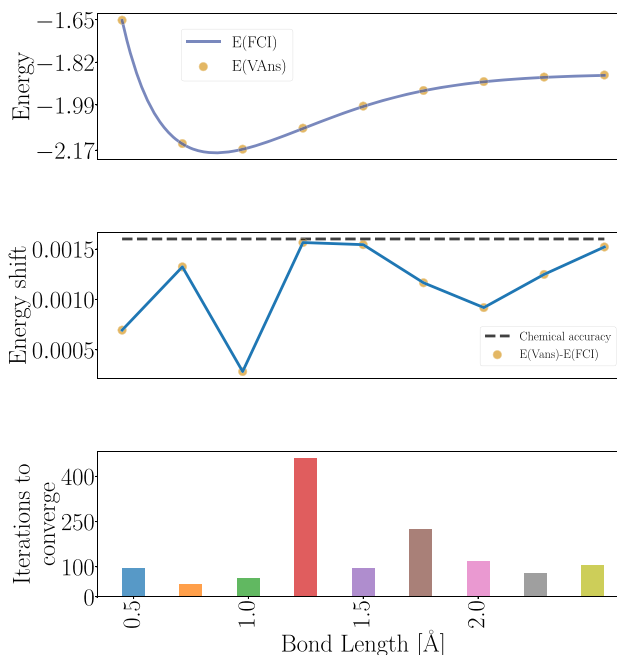


Fig. 10 Results of using VAns to obtain the ground state of a H_4 chain with a linear geometry, at different bond lengths. Here, we use VAns in the VQE algorithm for the molecular Hamiltonian obtained after a Jordan-Wigner transformation, leading to an 8-qubit circuit. Top: solid lines correspond to ground state energy as computed by the Full Configuration Interaction (FCI) method, whereas points correspond to energies obtained using VAns. Middle: differences between exact and VAns ground state energies are shown. Dashed line corresponds to chemical accuracy. Bottom: number of iterations required by VAns until convergence are shown

4.4 Quantum autoencoder

Here, we discuss how to employ VAns and the results from the previous section for the Hydrogen molecule to train the quantum autoencoder introduced in Romero et al. (2017). For the sake of completeness, we now describe the quantum autoencoder algorithm for compression of quantum data.

Consider a bipartite quantum system AB of n_A and n_B qubits, respectively. Then, let $\{p_i, |\psi_i\rangle\}$ be a training set of pure states on AB . The goal of the quantum autoencoder is to train an encoding parametrized quantum circuit $V(\mathbf{k}, \boldsymbol{\theta})$ to compress the states in the training set onto subsystem A , so that one can discard the qubits in subsystem B without losing information. Then, using the decoding circuit (simply given by $V^\dagger(\mathbf{k}, \boldsymbol{\theta})$) one can recover each state $|\psi_i\rangle$ in the training set (or in a testing set) from the information in subsystem A . Here, $V(\mathbf{k}, \boldsymbol{\theta})$ is essentially decoupling subsystem A from subsystem B , so that the state is completely compressed into subsystem A if the qubits in B are found in a fixed target state (which we here set as $|0\rangle_B = |0\rangle^{\otimes n_B}$).

As shown in Romero et al. (2017), the degree of compression can be quantified by the cost function

$$C_G(\mathbf{k}, \boldsymbol{\theta}) = 1 - \sum_i p_i \text{Tr} \left[(|\mathbf{0}\rangle\langle\mathbf{0}|_B \otimes \mathbb{1}_A) V |\psi_i\rangle\langle\psi_i| V^\dagger \right],$$

where $\mathbb{1}_B$ is the identity on subsystem A , and where we have omitted the \mathbf{k} and $\boldsymbol{\theta}$ dependence in V for simplicity of notation. Here, we see that if the reduced state in B is $|\mathbf{0}\rangle_B$ for all the states in the training set, then the cost is zero. Note that, as proved in Cerezo et al. (2021b), this is a global cost function (as one measures all the qubits in B simultaneously) and hence can have barren plateaus for large problem sizes. This issue can be avoided by considering the following local cost function where one instead measures individually each qubit in B (Cerezo et al. 2021b)

$$C_L(\mathbf{k}, \boldsymbol{\theta}) = 1 - \sum_{k=1}^{n_B} \sum_i \frac{p_i}{n_B} \text{Tr} \left[(|\mathbf{0}\rangle\langle\mathbf{0}|_k \otimes \mathbb{1}_{A,\bar{k}}) V |\psi_i\rangle\langle\psi_i| V^\dagger \right]. \quad (11)$$

Here, $\mathbb{1}_{A,\bar{k}}$ is the identity on all qubits in A and all qubits in B except for qubit k . We remark that this cost function is faithful to $C_G(\mathbf{k}, \boldsymbol{\theta})$, meaning that both cost functions vanish under the same conditions (Cerezo et al. 2021b).

As shown in Fig. 11, we employ the ground states $|\psi_i\rangle$ of the H_2 molecule (for different bond lengths) to create a training set of six states and a test set of ten states. Here, the circuits obtained through VAns in the previous section serve as (fixed) state preparation circuits for the ground states of the H_2 molecule. We then use VAns to learn an encoding circuit $V(\mathbf{k}, \boldsymbol{\theta})$ which can compress the states $|\psi_i\rangle$ into a subsystem of two qubits.

Figure 12 presents results obtained by minimizing the local cost function of Eq. 11 for a single run of the VAns algorithm. As seen, within 15 accepted architecture modifi-

Table 1 Fidelities for VAns applied to the quantum autoencoder task

Set	$-\log_{10}(1 - \mathcal{F})$
Training	6.49 (5.05–6.95)
Testing	6.17 (3.61–6.95)

The fidelities (\mathcal{F}) are reported as Mean value (Min-Max) across both training and testing sets

cations, VAns can decrease the training cost function down to 10^{-7} , by departing from a separable product ansatz (see Fig. 2). We here additionally show results obtained by training the Hardware Efficient Ansatz of Fig. 2b with 4 and 15 layers (as they have a comparable number of trainable parameters and CNOTs, respectively, compared to those obtained with VAns). In all cases, VAns achieves the best performance. In particular, it is worth noting that VAns has much fewer parameters (~ 45 versus 180) than the 15-layer HEA, while also achieving a cost value that is lower by two orders of magnitude. Hence, VAns obtains better performance with fewer quantum resources.

Moreover, the fidelities obtained after the decoding process at the training set are reported in Table 1. Here, \mathcal{F} is used to denote the fidelity between the input state and the state obtained after the encoding/decoding circuit. As one can see, VAns obtains very high fidelities for this task.

4.5 Unitary compilation

Unitary compilation is a task in which a target unitary is decomposed into a sequence of quantum gates that can be implemented on a given quantum computer. Current quantum computers are limited by the depth of quantum circuits that can be executed on them, which makes the compilation

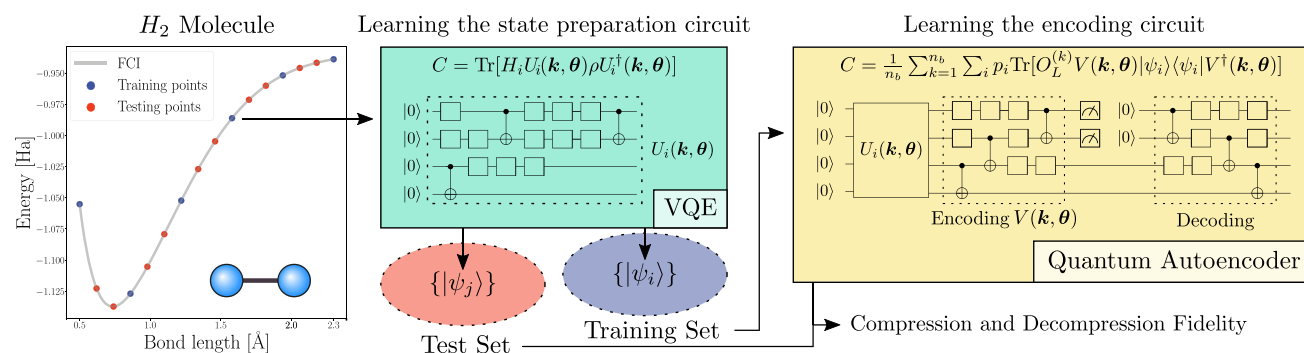


Fig. 11 Schematic diagram of the quantum autoencoder implementation. We first employ VAns to learn the circuits that prepare the ground states $\{|\psi_i\rangle\}$ of the H_2 molecule for different bond lengths. These ground states are then used to create a training set and test set for the quantum autoencoder implementation. The goal of the autoencoder is to train an encoding parametrized quantum circuit $V(\mathbf{k}, \boldsymbol{\theta})$ to compress each

$|\psi_i\rangle$ into a subsystem of two qubits so that one can recover $|\psi_i\rangle$ from the reduced states. The performance of the autoencoder can be quantified by computing the compression and decompression fidelities, i.e., the fidelity between the input state and the output state to the encoding/decoding circuit

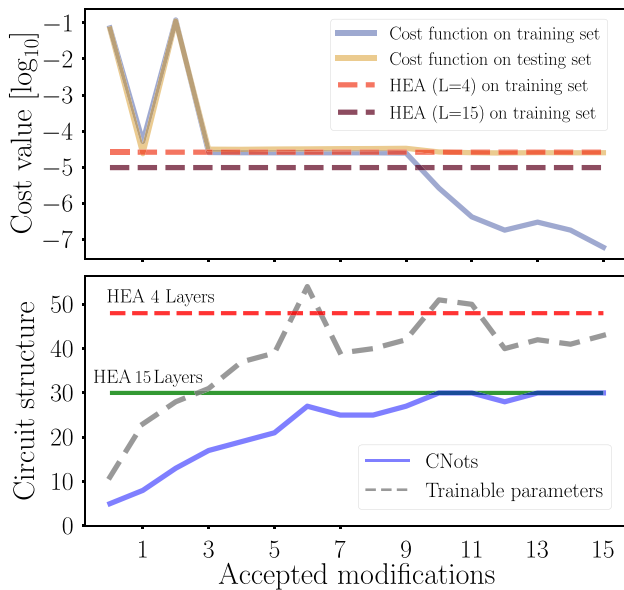


Fig. 12 Results of using VAns to train a quantum autoencoder. Here, we use VAns to train an encoding parametrized quantum circuit by minimizing Eq. 11 on a training set comprised of six ground states of the hydrogen molecule. We here also show the lowest cost function obtained for a HEA of 4 and 15 layers. Top panel: the cost function evaluated at both versus accepted VAns circuit modifications. In addition, we also show results of evaluating the cost on the testing set. Bottom panel: number of CNOTs, and number of trainable parameters versus the number of modifications of the ansatz accepted in the VAns algorithm. Here, we additionally show the number of CNOTs (solid line) and parameters (dashed line) in the HEA ansatzes considered. We remark that for $L = 15$, the HEA has 180 parameters, and hence the curve is not shown as it would be off the scale

task very important in the near term. Indeed, one wants to decompose a given unitary using as few gates as possible to maximally reduce the effect of noise. In this section, we show that VAns is capable of finding very short decompositions as compared with other techniques.

We will illustrate our approach by compiling Quantum Fourier Transform (QFT) on systems up to $n = 10$ qubits. Apart from VAns, we also compile the QFT unitary using standard HEA and compare the performance of both methods.

The cost function for unitary compilation is defined as follows. First, a training set is selected

$$\{(|\psi_j\rangle, U_{\text{QFT}}^{(n)}|\psi_j\rangle)\}_{j=1}^N, \quad (12)$$

where $U_{\text{QFT}}^{(n)}$ is a target QFT unitary on n qubits and $|\psi_j\rangle$ are N , randomly selected input states. We assume that the states $|\psi_j\rangle$ are pairwise orthogonal to avoid potential optimization problems caused by similarities in the training set. The cost

function takes the form

$$C(\mathbf{k}, \boldsymbol{\theta}) = \sum_{j=1}^N \|U_{\text{QFT}}^{(n)}|\psi_j\rangle - V(\mathbf{k}, \boldsymbol{\theta})|\psi_j\rangle\|^2. \quad (13)$$

Note that the cost function introduced in Eq. 13 becomes equivalent to a more standard one, $C'(\mathbf{k}, \boldsymbol{\theta}) = \|U_{\text{QFT}}^{(n)} - V(\mathbf{k}, \boldsymbol{\theta})\|^2$, when $N = 2^n$. While $C(\mathbf{k}, \boldsymbol{\theta})$ measures the distance between the exact output of QFT and the one returned by $V(\mathbf{k}, \boldsymbol{\theta})$ only on selected input states, $C'(\mathbf{k}, \boldsymbol{\theta})$ measures the discrepancy between full unitaries $U_{\text{QFT}}^{(n)}$ and $V(\mathbf{k}, \boldsymbol{\theta})$.

It has recently been shown (Caro et al. 2022) that $N \ll 2^n$ is sufficient to accurately decompose $U_{\text{QFT}}^{(n)}$. More precisely, a constant number of training states N (independent of n) can be used to ensure small value of $C'(\mathbf{k}, \boldsymbol{\theta})$, while minimizing the cost function in Eq. 13. This observation provides an exponential speedup in evaluating the cost function for unitary compilation. Indeed, the cost of evaluating $C(\mathbf{k}, \boldsymbol{\theta})$ in Eq. 13 is $N \cdot 2^n$ (assuming the circuit $V(\mathbf{k}, \boldsymbol{\theta})$ consists of few body gates and the states $U_{\text{QFT}}^{(n)}|\psi_j\rangle$ are given in computational basis), while the cost of computing $C'(\mathbf{k}, \boldsymbol{\theta})$ is 4^n .

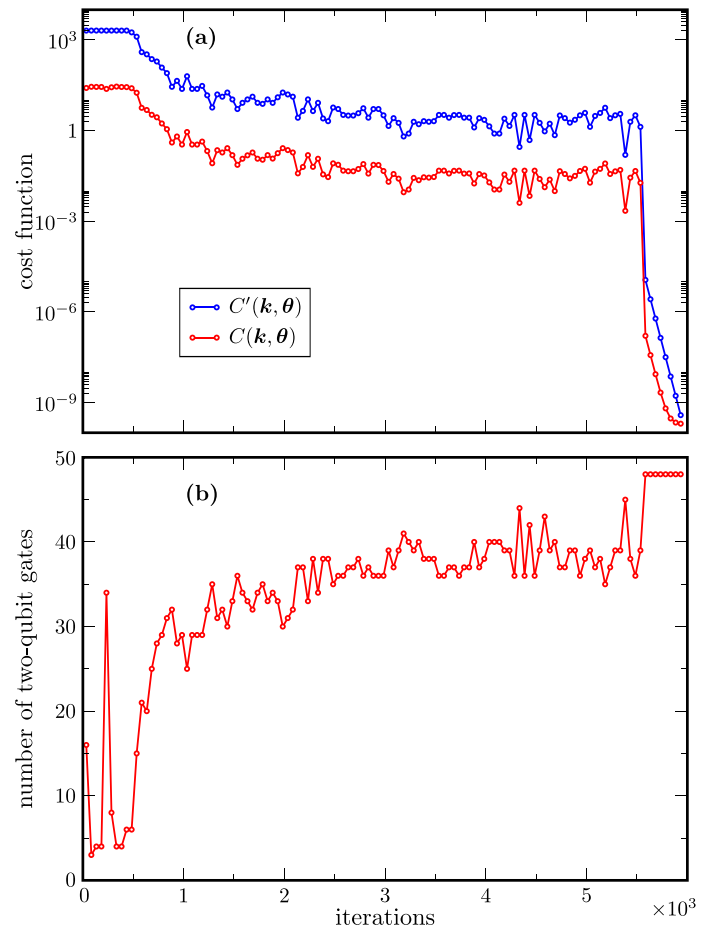
The number of training states N which lead to small value of $C'(\mathbf{k}, \boldsymbol{\theta})$ depends on the number of independent variational parameters in $V(\mathbf{k}, \boldsymbol{\theta})$. Suboptimal decompositions $V(\mathbf{k}, \boldsymbol{\theta})$ (in terms of number of parametrized gates) will require larger N to achieve good compilation accuracy. We have used $N = 15$ for $n = 10$ qubit compilation with VAns, and a much larger training set in an approach that uses HEA; the increase in N is necessary since the latter approach needed a much deeper circuit, as discussed below.

We have used most general two-qubit gates as the building block in VAns and to construct HEA. This is a slight generalization to the Insertion and Simplification steps in the VAns algorithm discussed above. General two-qubit gates can be decomposed down to CNOTs and one-qubit rotations using standard methods.

The method based on HEA requires very deep circuits (at $n = 10$). They consists of so many gates that the regular optimization has very small success probability. We therefore modify the method based on HEA and utilize the recursive structure of $U_{\text{QFT}}^{(n)}$. In the modified approach, we use HEA to compile $U_{\text{QFT}}^{(n-1)}$ and then use it to create an ansatz for $U_{\text{QFT}}^{(n)}$. The ansatz for larger system size additionally consists of several layers of HEA. We apply the above growth technique starting from $n = 3$ to eventually build the ansatz for $n = 10$. We stress that VAns does not require such simplification and is capable of finding the decomposition with high success probability directly at $n = 10$ while initialized randomly.

Figure 13 shows VAns results for $n = 10$ qubit QFT compilation. Panel (a) depicts how the value of the cost func-

Fig. 13 Results of using VAns for unitary compilation. Here, we use VAns to find a decomposition of QFT unitary defined on $n = 10$ qubits, by minimizing a cost function $C(\mathbf{k}, \boldsymbol{\theta})$ (red line in panel **a**) defined in Eq. (13). The cost evaluates a difference between exact output of QFT and the one returned by a current circuit, on a small number of input states only ($N = 15$). The blue line shows corresponding difference between full unitaries, $C'(\mathbf{k}, \boldsymbol{\theta}) = \|U_{\text{QFT}}^{(n)} - V(\mathbf{k}, \boldsymbol{\theta})\|^2$. We observe a high correlation between those two cost functions. Panel **b** shows how VAns modifies the number of two-qubit gates as it approaches the minimum of $C(\mathbf{k}, \boldsymbol{\theta})$. The minimum is found with 48 gates, which is ~ 4.5 times less than the decomposition found with HEA (not shown)



tion $C(\mathbf{k}, \boldsymbol{\theta})$ is minimized over the iterations. We also show the corresponding value of $C'(\mathbf{k}, \boldsymbol{\theta}) = \|U_{\text{QFT}}^{(n)} - V(\mathbf{k}, \boldsymbol{\theta})\|^2$. We observe strong correlation between both cost functions. $C'(\mathbf{k}, \boldsymbol{\theta})$ is eventually minimized below 10^{-9} at the end of the optimization. Panel (b) shows how the number of two-qubit gates evolves as VAns optimization is performed. Excluding the initial “warm-up” period, during which the cost function has very large (close to maximal possible) value, the number of two-qubit gates is steadily grown reaching 48 at the end of the optimization.

The approach based on HEA requires 219 general two-qubit gates to decompose $n = 10$ QFT, which is over 4 times more than the best circuit found by VAns. The HEA approach uses recursive structure of QFT to find accurate decomposition, while VAns does not rely on that property and finds a solution in fewer number of iterations. Finally, VAns takes advantage of the generalization bound (Caro et al. 2022), finding solution with near optimal number of variational parameters in $V(\mathbf{k}, \boldsymbol{\theta})$; the training set size N required for small generalization error is small resulting in fast cost function evaluations.

4.6 Noisy simulations

The results previously shown were obtained without considering hardware noise. We observed that VAns was able to better exploit the quantum resources at hand (i.e., attain a lower cost-function value) as compared to its fix-structure counterpart (e.g., HEA).

We now consider the case where noisy channels are present in the quantum circuit, an unavoidable situation in current experimental setups, with noise essentially forbidding large-depth quantum circuits to preserve quantum coherence. In the context of VQAs, the overall effect of noise is that of degrading the cost-function value and, if its strength is sufficiently high, then short-depth circuits turn to be favored even at the cost of expressibility. For instance, increasing the number of layers in HEA ansatz might not reduce the cost function and even increase it, since noise accumulates due to the presence of gates.

There are several sources of noise in quantum computers. For instance, experimental implementation of a quantum gate takes a finite amount of time, which in turn depends on

the physical qubit at hand, the latter subjected to thermal relaxation errors. Relaxation and dephasing errors depend, in general, on each particular qubit (i.e., *the qubit label*). The overall effect of the gate implementation is often modeled by a depolarizing quantum channel, followed by phase flips and amplitude damping channels, whose strength depends on the aforementioned parameters (gate implementation time, qubit label), gate type and environment temperature. For instance, an entangling gate such as a CNOT injects considerably more noise to the circuit than a single-qubit rotation. Moreover, state preparation and readout errors should be taken into account. We refer the reader to find further details on noise modeling in Ref. Georgopoulos et al. (2021). We also note that additional sources of noise should ultimately be considered, such as idle noise and cross-talk effects (LaRose et al. 2022; Murali et al. 2020). Because the complexity of noise modelling in NISQ devices is particularly high, we here propose a sufficiently simple model that yet captures the essential noise sources.

The λ -model While a noise-model is ultimately linked to the specific quantum hardware at hand and depends on several factors, we propose a unifying and simplified one that depends on a single parameter. The main motivation behind this is that of benchmarking the performance of different ansatzes in the presence of noise. In more complex scenarios, one should consider specific process matrices obtained from, e.g., process tomography (O'Brien et al. 2004; Yuen-Zhou et al. 2014), which would here obscure the benchmark and also bias it towards specific quantum hardware. In particular, our model is inspired by Refs Blank et al. (2020); Georgopoulos et al. (2021), which in turn are implemented in the *aer* noise simulator of IBM, and consists of the following models. State preparation and measurement errors are modeled via bit flip channels acting on each qubit, with strength $\lambda \cdot 10^{-2}$, happening before the circuit $U(k, \theta)$ and measurements respectively. Noise due to gate implementation is modeled as a depolarizing channel, followed by a phase flip and amplitude damping channels acting on the target qubit right after the gate. In principle, the strength of the channel should depend on the specific qubit, and gate type, but in order to keep the model simple enough we have assumed no noise dependence on the qubit label. Moreover, a two-qubit gate is considerably more noisy than a single-qubit one, which in the λ -model is reflected by the fact that noise strengths are an order of magnitude higher, in the depolarizing channel, than in single-qubit gates, the latter being $\lambda \cdot 10^{-5}$. Finally, the strengths of the phase flip and amplitude damping channels are set to $\lambda \cdot 10^{-3}$. We note that, while not considered here, different situations can easily be incorporated such as qubit connectivity constraints, or differences in qubits' quality (some qubits might be noisier than others). In such cases, we expect VAns to find circuits which auto-

matically balance the trade-offs at hand, i.e., minimize the number of gates acting on such noisier qubits.

With this model at hand, we have explored the region of λ in which the action of the noise becomes *interesting*. The results of running VAns under the λ -model for ground state preparation (VQE) of TFIM 8-qubit system are shown in Fig. 14. Here, the noise strength is sufficiently high so to affect the ground-state energy (which can otherwise be attained by either VAns or a 3-layered HEA). We thus sweep the value of λ by two orders of magnitude, and compare the results that VAns reaches with those of HEA (varying the number of layers of the latter). We see that for a sufficiently high noise strength, increasing the number of gates (e.g., the number of layers in HEA) eventually degrades the cost-function value, as opposed to the noise-less scenario. On the contrary, we observe that even if the noise-strength is sufficiently high, VAns considerably outperforms HEA by automatically adjusting the depth of the circuit to the noise-strength at hand. Thus, if the noise is large, shallow circuits

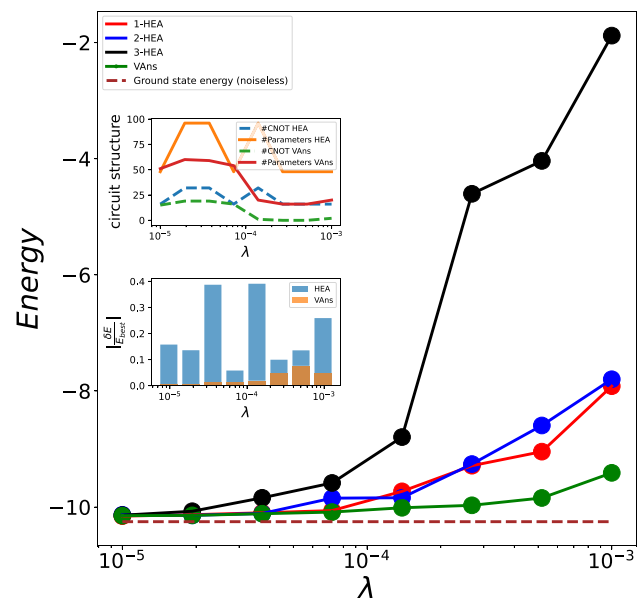


Fig. 14 Results of using VAns for VQE under the λ -model. Here, we consider the TFIM for 8 qubits, with $g = J = 1$. The results are obtained after repeating 50 iterations of optimizations with VAns and HEA respectively. We observe that VAns discovers much more efficient quantum circuits as compared to HEA. As shown in the upper inset, VAns automatically adjusts the circuit layout according to the noise strength at hand, a feature that fix-structure ansatzes lack. In the lower inset, we show the relative errors (e.g., standard deviation over optimal cost found) for both ansatzes, across the 50 iterations; we observe that VAns is more precise in reaching a minimum as compared to HEA. We note that in this experiment, we have initialized VAns to a 1-layered HEA, which is in turn inconvenient for a sufficiently high value of λ . Yet, VAns learns how to adapt the ansatz (in this case, finding a separable one) so to reach the lowest cost value. In all cases VAns termination criteria was set to a maximum number of 30 iterations

are found, whereas if the noise-strength is low, deeper circuits are allowed to be explored, since the penalty of adding new gates is smaller. In general, we observe that VAns is capable of finding the best possible circuit under given conditions, which is something that HEA simply can not accomplish.

5 Discussion

In this work, we have introduced the VAns algorithm, a semi-agnostic method for building variable structure parametrized quantum circuits. We expect VAns to be especially useful for abstract applications, such as linear systems (Bravo-Prieto et al. 2019; Huang et al. 2019; Xu et al. 2021), factoring (Anshuetz et al. 2019), compiling (Khatri et al. 2019; Sharma et al. 2020), metrology (Beckey et al. 2022; Koczor et al. 2020), and data science (Abbas et al. 2021; Biamonte et al. 2017; Cerezo et al. 2022; LaRose et al. 2019; Schuld et al. 2014; Verdon et al. 2019b), where physically motivated ansatzes are not readily available. In addition, VAns will likely find use even for physical applications such as finding ground states of molecular and condensed matter systems, as it provides a shorter depth alternative to physically motivated ansatzes for mitigating the impact of noise, as shown in our noisy simulations.

At each iteration of the optimization, VAns stochastically grows the circuit to explore the architecture hyperspace. More crucially, VAns also compresses and simplifies the circuit by removing redundant gates and unimportant gates. This is a key aspect of our method, as it differentiates VAns from other variable ansatz alternatives and allows us to produce short-depth circuits, which can mitigate the effect of noise-induced barren plateaus (NIBPs). We will further investigate this mitigation of NIBPs in future work.

To showcase the performance of VAns, we simulated our algorithm for several paradigmatic problems in VQAs. Namely, we implemented VAns to find ground states of condensed matter systems and molecular Hamiltonians, for a quantum autoencoder problem and for 10-qubit QFT compilation. In all cases, VAns was able to satisfactorily create circuits that optimize the cost. Moreover, as expected, these optimal circuits contain a small number of trainable parameters and entangling gates. Here, we also compared the results of VAns with results obtained using a Hardware Efficient Ansatz with either the same number of entangling gates, or the same number of parameters, and in all cases, we found that VAns could achieve the best performance. This point is crucial for the success of VAns in the presence of noisy channels, as it automatically adapts the circuit layout to the situation at hand (e.g., noise strength). For instance, under the λ -model (which is the noise model we have implemented),

VAns notably outperforms HEA under ground-state preparation tasks.

While we provided the basic elements and structure of VAns (i.e., the gate Insertion and gate Simplification rules), these should be considered as blueprints for variable ansatzes that can be adapted and tailored to more specific applications. For instance, the gates that VAns inserts can preserve a specific symmetry in the problem. Moreover, one can cast the VAns architecture optimization (e.g., removing unimportant gates) in more advanced learning frameworks. Examples of such frameworks include supervised learning or reinforced learning schemes, which could potentially be employed to detect which gates are the best candidates for being removed.

Acknowledgements Some of the simulations presented in this paper were done using Sandbox@Alphabet's TPU-based Floq simulator. MB is thankful to Josep Flix, John Calsamiglia Costa and Carles Acosta for their kindness in facilitating computational resources at Port d'Informació Científica, Barcelona. MB acknowledges support from the U.S. Department of Energy (DOE) through a quantum computing program sponsored by the Los Alamos National Laboratory (LANL) Information Science & Technology Institute, and support from the Spanish Agencia Estatal de Investigación, project PID2019-107609GB-I00, Generalitat de Catalunya CIRIT 2017-SGR-1127. MC was initially supported by the Laboratory Directed Research and Development (LDRD) program of LANL under project number 20180628ECR, and also supported by the Center for Nonlinear Studies at LANL. PJC acknowledges initial support from the LANL ASC Beyond Moore's Law project. LC was initially supported by the LDRD program of LANL under project number 20200056DR. PJC and LC were also supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, under the Accelerated Research in Quantum Computing (ARQC) program.

Author contribution All authors contributed equally to this work.

Funding Open Access Funding provided by Universitat Autònoma de Barcelona.

Data availability Data supporting the claims made in this work is available upon reasonable request.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abbas A, Sutter D, Zoufal C et al (2021) The power of quantum neural networks. *Nat Comput Sci* 1(6):403–409. <https://doi.org/10.1038/s43588-021-00084-1>
- Anschuetz E, Olson J, Aspuru-Guzik A et al (2019) Variational quantum factoring. In: International workshop on quantum technology and optimization problems. Springer, pp 74–85. https://link.springer.com/chapter/10.1007/978-3-030-14082-3_7
- Arrasmith A, Cerezo M, Czarnik P et al (2021) Effect of barren plateaus on gradient-free optimization. *Quantum* 5:558. <https://doi.org/10.22331/q-2021-10-05-558>. <https://quantum-journal.org/papers/q-2021-10-05-558/>
- Arrasmith A, Cincio L, Somma RD et al (2020) Operator sampling for shot-frugal optimization in variational algorithms. *arXiv:2004.06252*
- Arute F, Arya K, Babbush R et al (2020) Hartree-fock on a superconducting qubit quantum computer. *Science* 369(6507):1084–1089. <https://doi.org/10.1126/science.abb9811>. <https://science.sciencemag.org/content/369/6507/1084>
- Aspuru-Guzik A, Dutoi AD, Love PJ et al (2005) Simulated quantum computation of molecular energies. *Science* 309(5741):1704–1707. <https://doi.org/10.1126/science.1113479>. <https://science.sciencemag.org/content/309/5741/1704>
- Bartlett RJ, Musiał M (2007) Coupled-cluster theory in quantum chemistry. *Rev Mod Phys* 79(1):291. <https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.79.291>
- Beckey JL, Cerezo M, Sone A et al (2022) Variational quantum algorithm for estimating the quantum Fisher information. *Phys Rev Res* 4(1):013083. <https://doi.org/10.1103/PhysRevResearch.4.013083>. <https://journals.aps.org/prresearch/abstract/10.1103/PhysRevResearch.4.013083>
- Bharti K, Cervera-Lierta A, Kyaw TH et al (2022) Noisy intermediate-scale quantum algorithms. *Rev Mod Phys* 94(1):015004. <https://doi.org/10.1103/RevModPhys.94.015004>. <https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.94.015004>
- Bharti K, Haug T (2021) Quantum-assisted simulator. *Phys Rev A* 104(4):042418. <https://doi.org/10.1103/PhysRevA.104.042418>. <https://journals.aps.org/pra/abstract/10.1103/PhysRevA.104.042418>
- Biamonte J, Wittek P, Pancotti N et al (2017) Quantum machine learning. *Nature* 549(7671):195–202. <https://doi.org/10.1038/nature23474>. <https://www.nature.com/articles/nature23474>
- Blank C, Park DK, Rhee JKK et al (2020) Quantum classifier with tailored quantum kernel. *npj Quantum Inf* 6(1):41. <https://doi.org/10.1038/s41534-020-0272-6>
- Brandao FG, Harrow AW, Horodecki M (2016) Local random quantum circuits are approximate polynomial-designs. *Commun Math Phys* 346(2):397–434. <https://doi.org/10.1007/s00220-016-2706-8>. <https://link.springer.com/article/10.1007/s00220-016-2706-8>
- Bravo-Prieto C, LaRose R, Cerezo M et al (2019) Variational quantum linear solver. *arXiv:1909.05820*
- Broughton M, Verdon G, McCourt T et al (2020) Tensorflow quantum: a software framework for quantum machine learning. *arXiv:2003.02989*
- Cao Y, Romero J, Olson JP et al (2019) Quantum chemistry in the age of quantum computing. *Chem Rev* 119(19):10856–10915. <https://pubs.acs.org/doi/10.1021/acs.chemrev.8b00803>
- Caro MC, Huang HY, Cerezo M et al (2022) Generalization in quantum machine learning from few training data. *Nat Commun* 13:4919. <https://doi.org/10.1038/s41467-022-32550-3>
- Cerezo M, Sharma K, Arrasmith A et al (2022) Variational quantum state eigensolver. *npj Quantum Inf* 8(1):1–11. <https://doi.org/10.1038/s41534-022-00611-6>
- Cerezo M, Arrasmith A, Babbush R et al (2021) Variational quantum algorithms. *Nat Rev Phys* 3(1):625–644. <https://doi.org/10.1038/s42254-021-00348-9>. <https://www.nature.com/articles/s42254-021-00348-9>
- Cerezo M, Coles PJ (2021) Higher order derivatives of quantum neural networks with barren plateaus. *Quantum Sci Technol* 6(2):035006. <https://doi.org/10.1088/2058-9565/abf51a>. <https://iopscience.iop.org/article/10.1088/2058-9565/abf51a>
- Cerezo M, Rossignoli R, Canosa N et al (2017) Factorization and criticality in finite xxz systems of arbitrary spin. *Phys Rev Lett* 119(22):220605. <https://doi.org/10.1103/PhysRevLett.119.220605>. <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.119.220605>
- Cerezo M, Sone A, Volkoff T et al (2021) Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nat Commun* 12(1):1–12. <https://doi.org/10.1038/s41467-021-21728-w>. <https://www.nature.com/articles/s41467-021-21728-w>
- Cervera-Lierta A, Kottmann JS, Aspuru-Guzik A (2020) The meta-variational quantum eigensolver (meta-vqe): learning energy profiles of parameterized hamiltonians for quantum simulation. *arXiv:2009.13545*
- Chivilikhin D, Samarin A, Ulyantsev V et al (2020) Mog-vqe: multiobjective genetic variational quantum eigensolver. *arXiv:2007.04424*
- Cincio L, Subaşı Y, Sornborger AT et al (2018) Learning the quantum algorithm for state overlap. *New J Phys* 20(11):113022. <https://doi.org/10.1088/1367-2630/aae94a>
- Cincio L, Rudinger K, Sarovar M et al (2021) Machine learning of noise-resilient quantum circuits. *PRX Quantum* 2:010324. <https://doi.org/10.1103/PRXQuantum.2.010324>. <https://link.aps.org/doi/10.1103/PRXQuantum.2.010324>
- Cirstoiu C, Holmes Z, Iosue J et al (2020) Variational fast forwarding for quantum simulation beyond the coherence time. *npj Quantum Inf* 6(1):1–10. <https://doi.org/10.1038/s41534-020-00302-0>. <https://www.nature.com/articles/s41534-020-00302-0>
- Claudino D, Wright J, McCaskey AJ et al (2020) Benchmarking adaptive variational quantum eigensolvers. *Front Chem* 8:1152. <https://doi.org/10.3389/fchem.2020.606863>. <https://www.frontiersin.org/articles/10.3389/fchem.2020.606863/full>
- Commeau B, Cerezo M, Holmes Z et al (2020) Variational hamiltonian diagonalization for dynamical quantum simulation. *arXiv:2009.02559*
- Dankert C, Cleve R, Emerson J et al (2009) Exact and approximate unitary 2-designs and their application to fidelity estimation. *Phys Rev A* 80(1):012304. <https://journals.aps.org/pra/abstract/10.1103/PhysRevA.80.012304>
- Du Y, Huang T, You S et al (2020) Quantum circuit architecture search: error mitigation and trainability enhancement for variational quantum solvers. *arXiv:2010.10217*
- Elsken T, Metzen JH, Hutter F et al (2019) Neural architecture search: a survey. *J Mach Learn Res* 20(55):1–21. <https://lilianweng.github.io/lil-log/2020/08/06/neural-architecture-search.html>
- Farhi E, Goldstone J, Gutmann S (2014) A quantum approximate optimization algorithm. *arXiv:1411.4028*
- Fontana E, Cerezo M, Arrasmith A et al (2020) Optimizing parametrized quantum circuits via noise-induced breaking of symmetries. *arXiv:2011.08763*
- Gard BT, Zhu L, Barron GS et al (2020) Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm. *npj Quantum Inf* 6(1):1–9. <https://doi.org/10.1038/s41534-019-0240-1>. <https://www.nature.com/articles/s41534-019-0240-1>
- Georgopoulos K, Emary C, Zuliani P (2021) Modeling and simulating the noisy behavior of near-term quantum computers. *Phys Rev A* 104:062432. <https://doi.org/10.1103/PhysRevA.104.062432>. <https://link.aps.org/doi/10.1103/PhysRevA.104.062432>

- Gibbs J, Gili K, Holmes Z et al (2021) Long-time simulations with high fidelity on quantum hardware. [arXiv:2102.04313](https://arxiv.org/abs/2102.04313)
- Grant E, Wossnig L, Ostaszewski M et al (2019) An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum* 3:214. <https://doi.org/10.22331/q-2019-12-09-214>. <https://quantum-journal.org/papers/q-2019-12-09-214/>
- Grimsley HR, Economou SE, Barnes E et al (2019) An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nat Commun* 10(1):1–9. <https://doi.org/10.1038/s41467-019-10988-2>. <https://www.nature.com/articles/s41467-019-10988-2>
- Hadfield S, Wang Z, O’Gorman B et al (2019) From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* 12(2):34. <https://doi.org/10.3390/a12020034>. <https://www.mdpi.com/1999-4893/12/2/34>
- Haferkamp J (2022) Random quantum circuits are approximate unitary t -designs in depth $O(n^{5+o(1)})$. *Quantum* 6:795. <https://doi.org/10.22331/q-2022-09-08-795>
- Harrigan MP, Sung KJ, Neeley M et al (2021) Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat Phys* 1–5. <https://doi.org/10.1038/s41567-020-01105-y>. <https://www.nature.com/articles/s41567-020-01105-y>
- Harrow AW, Low RA (2009) Random quantum circuits are approximate 2-designs. *Commun Math Phys* 291(1):257–302. <https://doi.org/10.1007/s00220-009-0873-6>. <https://link.springer.com/article/10.1007/s00220-009-0873-6>
- Harrow A, Mehraban S (2018) Approximate unitary t -designs by short random quantum circuits using nearest-neighbor and long-range gates. [arXiv:1809.06957](https://arxiv.org/abs/1809.06957)
- Hastings WK (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57(1):97–109. <https://doi.org/10.2307/2334940>. <http://www.jstor.org/stable/2334940>
- Holmes Z, Arrasmith A, Yan B et al (2021) Barren plateaus preclude learning scramblers. *Phys Rev Lett* 126(19):190501. <https://doi.org/10.1103/PhysRevLett.126.190501>. <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.126.190501>
- Holmes Z, Sharma K, Cerezo M et al (2022) Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *PRX Quantum* 3:010313. <https://doi.org/10.1103/PRXQuantum.3.010313>. <https://link.aps.org/doi/10.1103/PRXQuantum.3.010313>
- Huang HY, Bharti K, Rebentrost P (2019) Near-term quantum algorithms for linear systems of equations. [arXiv:1909.07344](https://arxiv.org/abs/1909.07344)
- Huang HY, Kueng R, Preskill J (2020) Predicting many properties of a quantum system from very few measurements. *Nat Phys* 16(10):1050–1057. <https://doi.org/10.1038/s41567-020-0932-7>. <https://www.nature.com/articles/s41567-020-0932-7>
- Kandala A, Mezzacapo A, Temme K et al (2017) Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* 549(7671):242–246. <https://doi.org/10.1038/nature23879>
- Khatri S, LaRose R, Poremba A et al. (2019) Quantum-assisted quantum compiling. *Quantum* 3:140. <https://doi.org/10.22331/q-2019-05-13-140>. <https://quantum-journal.org/papers/q-2019-05-13-140/>
- Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: *Proceedings of the 3rd international conference on learning representations (ICLR)*. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
- Koczor B, Benjamin SC (2019) Quantum natural gradient generalised to non-unitary circuits. [arXiv:1912.08660](https://arxiv.org/abs/1912.08660)
- Koczor B, Endo S, Jones T et al (2020) Variational-state quantum metrology. *New J Phys*. <https://doi.org/10.1088/1367-2630/ab965e>. <https://iopscience.iop.org/article/10.1088/1367-2630/ab965e>
- Kübler JM, Arrasmith A, Cincio L et al (2020) An adaptive optimizer for measurement-frugal variational algorithms. *Quantum* 4:263. <https://doi.org/10.22331/q-2020-05-11-263>. <https://quantum-journal.org/papers/q-2020-05-11-263/>
- Larocca M, Czarnik P, Sharma K et al. (2022) Diagnosing barren plateaus with tools from quantum optimal control. *Quantum* 6:824. <https://doi.org/10.22331/q-2022-09-29-824>
- LaRose R, Mari A, Russo V et al (2022) Error mitigation increases the effective quantum volume of quantum computers. [arXiv:2203.05489](https://arxiv.org/abs/2203.05489)
- LaRose R, Tikku A, O’Neel-Judy É et al (2019) Variational quantum state diagonalization. *npj Quantum Inf* 5(1):1–10. <https://doi.org/10.1038/s41534-019-0167-6>. <https://www.nature.com/articles/s41534-019-0167-6>
- Liu X, Angone A, Shaydulin R et al (2021) Layer vqe: a variational approach for combinatorial optimization on noisy quantum computers. [arXiv:2102.05566](https://arxiv.org/abs/2102.05566)
- Liu H, Simonyan K, Yang Y (2018) Darts: differentiable architecture search. [arXiv:1806.09055](https://arxiv.org/abs/1806.09055)
- Marrero CO, Kieferová M, Wiebe N (2021) Entanglement-induced barren plateaus. *PRX Quantum* 2(4):040316. <https://doi.org/10.1103/PRXQuantum.2.040316>. <https://journals.aps.org/prxquantum/abstract/10.1103/PRXQuantum.2.040316>
- Maslov D, Dueck GW, Miller DM et al (2008) Quantum circuit simplification and level compaction. *IEEE Trans Comput-Aided Des Integr Circuits Syst* 27(3):436–444. <https://doi.org/10.1109/TCAD.2007.911334>
- McArdle S, Endo S, Aspuru-Guzik A et al (2020) Quantum computational chemistry. *Rev Mod Phys* 92(1):015003. <https://doi.org/10.1103/RevModPhys.92.015003>. <https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.92.015003>
- McClean JR, Boixo S, Smelyanskiy VN et al (2018) Barren plateaus in quantum neural network training landscapes. *Nat Commun* 1:1–6. <https://doi.org/10.1038/s41467-018-07090-4>. <https://www.nature.com/articles/s41467-018-07090-4>
- McClean JR, Sung KJ, Kivlichan ID et al (2019) Openfermion: the electronic structure package for quantum computers. [arXiv:1710.07629](https://arxiv.org/abs/1710.07629)
- Murali P, McKay DC, Martonosi M et al (2020) Software mitigation of crosstalk on noisy intermediate-scale quantum computers. In: *Proceedings of the twenty-fifth international conference on architectural support for programming languages and operating systems*. Association for Computing Machinery, New York, NY, USA, ASPLOS ’20, pp 1001–1016. <https://doi.org/10.1145/3373376.3378477>
- Nakanishi KM, Fujii K, Todo S (2020) Sequential minimal optimization for quantum-classical hybrid algorithms. *Phys Rev Res* 2(4):043158. <https://doi.org/10.1103/PhysRevResearch.2.043158>. <https://journals.aps.org/prresearch/abstract/10.1103/PhysRevResearch.2.043158>
- O’Brien JL, Pryde GJ, Gilchrist A et al (2004) Quantum process tomography of a controlled-not gate. *Phys Rev Lett* 93:080502. <https://doi.org/10.1103/PhysRevLett.93.080502>. <https://link.aps.org/doi/10.1103/PhysRevLett.93.080502>
- Ollitrault PJ, Kandala A, Chen CF et al (2020) Quantum equation of motion for computing molecular excitation energies on a noisy quantum processor. *Phys Rev Res* 2(4):043140. <https://doi.org/10.1103/PhysRevResearch.2.043140>. <https://journals.aps.org/prresearch/abstract/10.1103/PhysRevResearch.2.043140>
- Ostaszewski M, Grant E, Benedetti M (2021) Structure optimization for parameterized quantum circuits. *Quantum* 5:391. <https://doi.org/10.22331/q-2021-01-28-391>. <https://quantum-journal.org/papers/q-2021-01-28-391/>
- Patti TL, Najafi K, Gao X et al (2021) Entanglement devised barren plateau mitigation. *Phys Rev Res* 3(3):033090. <https://doi.org/10.1103/PhysRevResearch.3.033090>. <https://par.nsf.gov/servlets/purl/10328786>

- Peruzzo A, McClean J, Shadbolt P et al (2014) A variational eigenvalue solver on a photonic quantum processor. *Nat Commun* 5(1):1–7. <https://doi.org/10.1038/ncomms5213>. <https://www.nature.com/articles/ncomms5213#citeas>
- Pesah A, Cerezo M, Wang S et al (2021) Absence of barren plateaus in quantum convolutional neural networks. *Phys Rev X* 11(4):041011. <https://doi.org/10.1103/PhysRevX.11.041011>. <https://journals.aps.org/prx/abstract/10.1103/PhysRevX.11.041011>
- Pirhooshayan M, Terlaky T (2020) Quantum circuit design search. [arXiv:2012.04046](https://arxiv.org/abs/2012.04046)
- Preskill J (2018) Quantum computing in the nisy era and beyond. *Quantum* 2:79. <https://doi.org/10.22331/q-2018-08-06-79>. <https://quantum-journal.org/papers/q-2018-08-06-79/>
- Rattew AG, Hu S, Pistoia M et al (2019) A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver. [arXiv:1910.09694](https://arxiv.org/abs/1910.09694)
- Romero J, Olson JP, Aspuru-Guzik A (2017) Quantum autoencoders for efficient compression of quantum data. *Quantum Sci Technol* 2(4):045001. <https://doi.org/10.1088/2058-9565/aa8072>. <https://iopscience.iop.org/article/10.1088/2058-9565/aa8072>
- Sack SH, Medina RA, Michailidis AA et al (2022) Avoiding barren plateaus using classical shadows. *PRX Quantum* 3(2):020365. <https://doi.org/10.1103/PRXQuantum.3.020365>. <https://journals.aps.org/prxquantum/abstract/10.1103/PRXQuantum.3.020365>
- Schuld M, Sinayskiy I, Petruccione F (2014) The quest for a quantum neural network. *Quantum Inf Process* 13(11):2567–2586. <https://doi.org/10.1007/s11128-014-0809-8>
- Sharma K, Cerezo M, Cincio L et al (2022) Trainability of dissipative perceptron-based quantum neural networks. *Phys Rev Lett* 128(18):180505. <https://doi.org/10.1103/PhysRevLett.128.180505>
- Sharma K, Khatri S, Cerezo M et al (2020) Noise resilience of variational quantum compiling. *New J Phys* 22(4):043006. <https://doi.org/10.1088/1367-2630/ab784c>. <https://iopscience.iop.org/article/10.1088/1367-2630/ab784c>
- Sim S, Romero J, Gonthier JF et al (2021) Adaptive pruning-based optimization of parameterized quantum circuits. *Quantum Sci Technol* 6(2):025019. <https://doi.org/10.1088/2058-9565/abe107>
- Sim S, Johnson PD, Aspuru-Guzik A (2019) Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Adv Quantum Technol* 2(12):1900070. <https://doi.org/10.1002/qute.201900070>. <https://onlinelibrary.wiley.com/doi/abs/10.1002/qute.201900070>
- Skolik A, McClean JR, Mohseni M et al (2021) Layerwise learning for quantum neural networks. *Quantum Mach Intell* 3(1):1–11. <https://doi.org/10.1007/s42484-020-00036-4>
- Stilck França D, Garcia-Patron R (2021) Limitations of optimization algorithms on noisy quantum devices. *Nat Phys* 17(11):1221–1227. <https://doi.org/10.1038/s41567-021-01356-3>
- Stokes J, Isaac J, Killoran N et al (2020) Quantum natural gradient. *Quantum* 4:269. <https://doi.org/10.22331/q-2020-05-25-269>. <https://quantum-journal.org/papers/q-2020-05-25-269/>
- Tang HL, Shkolnikov V, Barron GS et al (2021) qubit-adapt-vqe: an adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor. *PRX Quantum* 2(2):020310. <https://doi.org/10.1103/PRXQuantum.2.020310>
- Thanasilp S, Wang S, Nghiem NA et al (2021) Subtleties in the trainability of quantum machine learning models. [arXiv:2110.14753](https://arxiv.org/abs/2110.14753)
- Tkachenko NV, Sud J, Zhang Y et al (2021) Correlation-informed permutation of qubits for reducing ansatz depth in vqe. *PRX Quantum* 2(2):020337. <https://doi.org/10.1103/PRXQuantum.2.020337>. <https://journals.aps.org/prxquantum/abstract/10.1103/PRXQuantum.2.020337>
- Verdon G, Broughton M, McClean JR et al (2019a) Learning to learn with quantum neural networks via classical neural networks. [arXiv:1907.05415](https://arxiv.org/abs/1907.05415)
- Verdon G, Marks J, Nanda S et al (2019b) Quantum Hamiltonian-based models and the variational quantum thermalizer algorithm. [arXiv preprint arXiv:1910.02071](https://arxiv.org/abs/1910.02071)
- Verdon G, Pye J, Broughton M (2018) A universal training algorithm for quantum deep learning. [arXiv:1806.09729](https://arxiv.org/abs/1806.09729)
- Volkoff T, Coles PJ (2021) Large gradients via correlation in random parameterized quantum circuits. *Quantum Sci Technol* 6(2):025008. <https://doi.org/10.1088/2058-9565/abd891>. <https://iopscience.iop.org/article/10.1088/2058-9565/abd891>
- Wang S, Fontana E, Cerezo M et al (2021) Noise-induced barren plateaus in variational quantum algorithms. *Nat Commun* 12(1):1–11. <https://doi.org/10.1038/s41467-021-27045-6>. <https://www.nature.com/articles/s41467-021-27045-6>
- Xu X, Sun J, Endo S et al (2021) Variational algorithms for linear algebra. *Sci Bull* 66(21):2181–2188. <https://doi.org/10.1016/j.scib.2021.06.023>. <https://www.sciencedirect.com/science/article/pii/S2095927321004631>
- Younis E, Cincio L (2020) Quantum fast circuit optimizer (qFactor). <https://github.com/BQSKit/qfactor>
- Yuan X, Endo S, Zhao Q et al (2019) Theory of variational quantum simulation. *Quantum* 3:191. <https://doi.org/10.22331/q-2019-10-07-191>. <https://quantum-journal.org/papers/q-2019-10-07-191/>
- Yuen-Zhou J, Krich JJ, Kassal I et al (2014) The process matrix and how to determine it: quantum process tomography. In: *Ultrafast spectroscopy*. 2053–2563, IOP Publishing, pp 1–1 to 1–9. <https://doi.org/10.1088/978-0-750-31062-8ch1>
- Zhang SX, Hsieh CY, Zhang S et al (2020b) Differentiable quantum architecture search. [arXiv:2010.08561](https://arxiv.org/abs/2010.08561)
- Zhang ZJ, Kyaw TH, Kottmann J et al (2021) Mutual information-assisted adaptive variational quantum eigensolver. *Quantum Sci Technol*. <https://doi.org/10.1088/2058-9565/abdca4>. <https://iopscience.iop.org/article/10.1088/2058-9565/abdca4>
- Zhang K, Hsieh MH, Liu L et al (2020a) Toward trainability of quantum neural networks. [arXiv:2011.06258](https://arxiv.org/abs/2011.06258)
- Zhao C, Gao XS (2021) Analyzing the barren plateau phenomenon in training quantum neural networks with the ZX-calculus. *Quantum* 5:466. <https://doi.org/10.22331/q-2021-06-04-466>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.