Quantum Machine Intelligence (2024) 6:9
https://doi.org/10.1007/s42484-024-00144-5

RESEARCH ARTICLE

The effects of quantum hardware properties on the performances of variational quantum learning algorithms

Giuseppe Buonaiuto[1] · Francesco Gargiulo[1] · Giuseppe De Pietro[1] · Massimo Esposito[1] · Marco Pota[1]

Received: 21 May 2023 / Accepted: 16 January 2024

Abstract
In-depth theoretical and practical research is nowadays being performed on variational quantum algorithms (VQAs), which have the potential to surpass traditional, classical, algorithms on a variety of problems, in physics, chemistry, biology, and optimization. Because they are hybrid quantum-classical algorithms, it takes a certain set of optimal conditions for their full potential to be exploited. For VQAs, the construction of an appropriate ansatz in particular is crucial, since it lays the ground for efficiently solving the particular problem being addressed. To prevent severe negative effects that hamper quantum computation, the substantial noise, together with the structural limitations, characteristic of currently available devices must be also taken into consideration while building the ansatz. In this work the effect of the quantum hardware structure, namely the topological properties emerging from the couplings between the physical qubits and the basis gates of the device itself, on the performances of VQAs is addressed. Specifically, it is here experimentally shown that a complex connectivity in the ansatz, albeit being beneficial for exploring wider sets of solutions, introduces an overhead of gates during the transpilation on a quantum computer that increases the overall error rate, thus undermining the quality of the training. It is hence necessary, when implementing a variation quantum learning algorithm, to find the right balance between a sufficiently parametrized ansatz and a minimal cost in terms of resources during transpilation. Moreover, the experimental finding allows to construct a heuristic metric function, which aids the decision-making process on the best possible ansatz structure to be deployed on a given quantum hardware, thus fostering a more efficient application of VQAs in realistic situations. The experiments are performed on two widely used variational algorithms, the VQE (variational quantum eigensolver) and the VQC (variational quantum classifier), both tested on two different problems, the first on the Markowitz portfolio optimization using real-world financial data, and the latter on a classification task performed on the Iris dataset.

Keywords Quantum machine learning · Quantum computing · Optimization

1 Introduction

A wide range of academic and business sectors have shown interest in quantum computing over the past few decades, due to its potential applications and uses. In theory, quantum computing enables the solution of now unsolvable or hard to solve problems, by utilizing quantum mechanical principles like entanglement and superposition. Recently, the first real quantum hardware was realized and made available for practical investigations (Clarke and Wilhelm 2008; Häffner et al. 2008; Slussarenko and Pryde 2019): even though they have limited computing power and are heavily subject to noise, they provide an impressive benchmark to test the capabilities of quantum computing and to explore its potential impact. Following the current growing trend (Global Quantum Computing Market Report 2022), it is expected that quantum devices, currently in the NISQ (noisy intermediate scale quantum) era, will explode in terms of number of qubits, noise resilience and programmability, opening the way for real-world applications (Quantum computing use n.d.). Although NISQ devices are subject to noise and decoherence, and have nowadays few qubits to be used for computation, they still might be useful to prove quantum advantage over several interesting tasks. For properly designed quantum algorithms, in fact, it is expected to witness the out-performance of NISQ devices over classical comput-

✉ Giuseppe Buonaiuto
giuseppe.buonaiuto@icar.cnr.it

[1] National Research Council of Italy (CNR), Institute for High Performance Computing and Networking (ICAR), Via Pietro Castellino, Naples 80131, Italy

Published online: 05 February 2024

ers. Among the most interesting algorithms that promise to show quantum advantage, the Variation Quantum Algorithms (VQAs) have recently attracted a lot of attention, due to their relative handy implementation and versatility of application (Cerezo et al. 2021). There are available in literature various examples of VQAs (Tilly et al. 2022; Zhou et al. 2020; Zhao and Wang 2021). These all share the same basic structure: the core of these algorithms is a variational quantum circuit containing parametrized gates, usually referred to as variational ansatz, which are updated iteratively by the means of an optimizer (either classical or quantum), aiming at the minimization of an objective function, built on the specific problem to be solved. Generally, VQAs are used to solve optimization problems (Kim and Oz 2022), to find the ground state energy of chemical or molecular compounds (Motta and Rice 2022) and, in the most recent field of quantum machine learning, to perform regression or classification (Biamonte et al. 2017). In the design of any VQA, particular care should be posed in defining a suitable parametrized ansatz. The variational quantum circuit in fact affects the performances of the algorithm in a rather fundamental way. Various approaches have been proposed aiming to define a set of suitable ansatz for quantum computers: ranging from those pointing at hardware efficiency (Kandala et al. 2017) to problem specific ansatz (Miki et al. 2022). The former might suffer overparametrization, which easily leads to vanishing gradients during training (Wang et al. 2021). The latter are carefully designed to map the intrinsic properties of the problem, like physical structures or graph configurations (Anselmetti et al. 2021). However, when there is no clear intuition about the underlying mathematical form of the possible solution, such as in many machine learning problems, a proper methodology for constructing an optimal ansatz is still to be found. One of the possible approaches, which is attracting recently a major attention, relies on using classical machine learning techniques to infer the optimal ansatz for quantum machine learning algorithms (Huang et al. 2022; Zhang et al. 2021). The search space of this neural algorithm is, however, vast and they might lack computational efficiency. Generally, in the design of a parametrized ansatz, one of the ideal characteristics required is to achieve good performances with limited circuit depth. Although this is desirable from many point of view, especially in terms of computational and resource cost, in this argument there is no account of the mapping of the ansatz on the underlying topology of the quantum hardware. In fact, when sent to the quantum computer for starting the computational pipeline, the parametrized circuit is properly transpiled on the basis gates of the quantum computer, in order to be accommodated on its structural geometry. For instance, entangling gates among qubits in the ansatz that are not physically connected on the hardware are transpiled by adding a set of SWAP gates that allow the connection among them, increasing the overall circuit depth (Younis and Iancu 2022). While this effect is not necessarily advantageous, the back side of the problem is that highly entangled ansatz might possess higher expressive power, thus being more suitable to tackle the specific problem to be solved. Hence, gathering information about the effect of the hardware characteristic, such as its topology, can in principle aid in designing an optimal set of ansatz, as was recently demonstrated and tackled (Cheng et al. 2022) for quantum neural networks, which possess as main feature a good balance between a sustained expressive power together with a minimal increase of the circuit depth. Furthermore, such effective mapping can decrease the runoff of the overall error rate, while not compromising drastically the performances and the accuracy of the results. It follows from the arguments above that is fundamental, for a performing implementation of VQAs, to understand the effect of the hardware topology during training, and thus to account for it when designing a variational ansatz.

In this work an extensive experimental study, on real quantum devices, of the effect of topology on the quality, i.e. the convergence rate, the accuracy of the result and the error, of two VQAs is performed, namely on VQE and VQC (Variational Quantum Classifier). To this end, experiments are performed over several quantum computers with varying hardware characteristics. The results demonstrate that the topology of the quantum computer significantly affects the quality of the results obtained from VQE and VQC algorithms. This study also reveals that the optimal choice of ansatz depends also on the particular topology of the quantum computer. For instance, certain hardware allows simple ansatz with a low connectivity between qubits to achieve similar performances as more complex and expressive ansatz. However, for other quantum hardware, a more complex and expressive ansatz is needed to obtain high-quality results. It is hence demonstrated that a compromise must be found for the ansatz, such that it needs to be expressive enough to allow the exploration of large portions of the space of the solutions, with a minimal amount of gates emerging from the transpilation on the real hardware. These experimental evidences allow to construct a heuristic function, which takes into account the transpilation requirements of the ansatz, the quantum volume of the quantum computer (Pelofske et al. 2022) and the number of qubits used for the quantum algorithm, ultimately helping the selection of an optimal configuration. The results can foster the research of topology-aware ansatz and provide a set of attributes to entail in the neural search algorithms of variational wavefunctions. Finally, it is experimentally demonstrated that, when a section of the entire NISQ device is used for computation, the effectiveness of training, and the effect of noise heavily depends on the interplay between the chosen qubit selection on the machine and the structure of the ansatz. This result is particularly relevant for the application of noisy devices in real-world problem, ultimately helping to miti-

gate detrimental effect for the training of quantum machine learning algorithms, such as barren plateaus.

The rest of the paper is organized as follows: Section 1 gives the basic background of the quantum algorithms in use (VQE and VQC) and states the motivation of the present work; Section 2 presents an overview of the problems selected for the experimental phase, the methods and data in use; Section 3 describes the experimental results and discuss the effect of topology; furthermore, the possible heuristic function that quantify the quality of the ansatz on the topology of the machine is presented and discussed.

2 Background

2.1 Variational quantum algorithms

The goal of VQAs is to find a model that fits on test data, after training on a given dataset. This process employs a classical optimizer that is used to find the set of optimal parameters θ that minimize a cost function $\tilde{C}(\theta)$, tailored on the problem to be solved. The cost function is defined, generally, as a function of the expectation value of a combination of Hermitian operators, evaluated on a variational ansatz, that is a parametrized quantum circuit. The parameters are iteratively updated until convergence is reached. Hence, in summary, a VQA is composed of three main ingredients: a quantum circuit with trainable parameters, a cost function that defines the specific task and an update rule for the parameter of the ansatz. In the following, two specific types of VQAs are investigated, the VQE and the VQC. These algorithms have the same structure but a different cost function, where the first aims at finding the minimum eigenvalue of a Hamiltonian modelling a class of problems, while the latter's objective function is one of the possible loss functions associated with multi-label classification, typically cross-entropy loss. In the following more details are provided about the aforementioned VQAs algorithms.

2.2 VQE

One of the most popular VQA algorithms is the VQE. VQE is a quantum algorithm used to find the eigenvalues and eigenvectors of a Hamiltonian operator that represents a quantum system. The goal of the VQE is to find the lowest eigenvalue of the Hamiltonian, which corresponds to the ground state energy of the system. The VQE algorithm can be mathematically described as follows: Let the Hamiltonian operator of a quantum system be represented by H. The exact ground state energy (E_0) is defined as the lowest eigenvalue of H. The eigenvalue problem can be written as:

$$H|\psi\rangle = E_0|\psi_0\rangle \tag{1}$$

where $|\psi_0$ is an eigenvector corresponding to the eigenvalue E_0. In VQE, the candidate eigenvector $|\psi$ is given by a parametrized circuit, i.e. the ansatz is composed of a set of parametrized unitary gates, which can be written as:

$$|\psi(\theta)\rangle = U(\theta)|0\rangle \tag{2}$$

where $U(\theta)$ is the quantum circuit, θ is a set of parameters, and $|0$ is the initial state. The goal of the VQE is to find the optimal values of the parameters θ that minimize the expectation value of the Hamiltonian:

$$E_{vqe}(\theta) = \langle\psi(\theta)H|\psi(\theta)\rangle, \tag{3}$$

such that, given the optimal set of parameters θ^*, the energy associated with the ansatz satisfies the properties $|E_0 - E_{vqe}(\theta^*)| < \epsilon$, where E_0 is the actual ground state of the Hamiltonian problem. The expectation value can be evaluated by repeatedly measuring the quantum circuit in the state $|\psi(\theta)$. This evaluation can be done classically, as the measurement outcomes are classical bits. The optimal values of θ can be found using classical optimization algorithms, such as gradient descent or gradient-free methods, which iteratively update the values of θ until a minimum of the expectation value is found. The final result is the approximation of the ground state energy. The success of the VQE algorithm depends critically on the choice of the ansatz. The ansatz must be able to accurately represent the ground state of the system, which is often a challenging task, especially for systems with large numbers of qubits. A good ansatz should be able to efficiently capture the relevant features of the ground state, such as symmetry and entanglement, in order to accurately represent the quantum system. Additionally, the ansatz should also be flexible enough to allow for the optimization of the parameters θ.

2.3 VQC

The second VQA implemented in this work is a supervised quantum classifier, which is the quantum analogue of classical classification machine learning algorithm, specifically a Quantum Neural Network. Quantum Neural Networks (QNNs) (Jia et al. 2019) are a type of machine learning model that combine the principles of quantum mechanics and neural networks. The idea behind QNNs is to leverage the quantum mechanical properties of quantum systems, such as superposition and entanglement, to improve the performance and capabilities of neural networks. In particular QNNs represent a subclass of VQAs with specific characteristic, such as an encoding function for classical data into quantum states, a trainable parametrized ansatz and a measurement scheme for decoding results. Several recent studies (Coles 2021; Abbas et al. 2021) show that when properly

defined, a QNN can outperform its classical counterpart, showing increasing trainability and robust inference capabilities. The quantum algorithm of a basic QNN consists of an encoder for classical data, which is a quantum circuit for representing data as complex vector and a variational circuit used for training. After the entire encoder-ansatz pipeline is constructed, measurements are performed, which lead to an output distribution function. This distribution is used to evaluate the updated parameters, via a gradient function. Given the peculiar structure of quantum circuit, gradients are evaluated making use of the parameter shift rules (Crooks 2019; Wierichs et al. 2022). Different classes of QNNs have been realized in the past years, being fully quantum, i.e. without a classical part of the computation, or hybrid. This work focuses on the VQC, which belongs to the hybrid class of QNNs. Like QNNs, the VQC algorithm is designed to learn a mapping from input data to output data. Specifically, the VQC algorithm is used for classification tasks, where the goal is to map an input vector to one of a finite number of output classes. The VQC algorithm uses a variational quantum circuit to classify the input data. The parameters of the variational quantum circuit can be optimized using classical optimization algorithms, such as the *Cobyla*, *SPSA* or *Adam* (Optimizers–IBM Quantum Documentation n.d.), which are available on the Qiskit package. In the VQC algorithm, the parameters of the variational quantum circuit are optimized to minimize a cost function, which measures the difference between the predicted and actual output labels. The performance of the VQC algorithm is heavily dependent on the choice of the variational quantum circuit architecture, i.e. the ansatz, and the underlying topology of the quantum computer used to implement the algorithm. The choice of the ansatz can impact the expressiveness of the circuit and the complexity of the computation (Saib et al. 2021). The underlying topology of the quantum computer can affect the way that the quantum gates are applied to the qubits, which can impact the performance of the algorithm, leading to an increase of the overall error rate. The importance of the quantum ansatz is highlighted by recent studies that have investigated the performance of the VQC algorithm on different architectures. For instance, it has been demonstrated (Farhi et al. 2014) that the VQC algorithm can be used to solve classification problems with high accuracy using a simple ansatz, basically consisting of single-qubit rotations and CNOT gates. However, more complex classification problem requires, in general, more complex and expressive ansatz. Furthermore, the performance of the VQC algorithm can be heavily influenced by the underlying topology of the quantum computer. In a recent study (Cheng et al. 2022) the impact of the topology of the quantum computer on the performance of a QNN algorithm has been investigated: it was found that the choice of topology can have a significant impact on the performance of the algorithm; hence, they provide an initial benchmark for designing an ansatz whose structure relies on the topology of the hardware, rather than being a black box parametrized circuit.

2.4 The role of hardware topology in a quantum algorithm pipeline

Topology of a quantum hardware refers to the way qubits are connected and arranged to form a quantum computer. To understand where and how the topology of the quantum computer plays a role in the computation it is here briefly outlined the computational pipeline of a quantum algorithm on a quantum hardware (Transpiler (qiskit.transpiler) n.d.; Cheng et al. 2022). It is worth to notice here that the quantitative analysis hereby presented focuses on superconducting qubits architectures, specifically the IBM quantum chips. Other types of quantum hardware will have different characteristics (Peruzzo et al. 2014; Zhao et al. 2023), i.e. more or less limitations depending on the specific feature under investigation. However, the findings and the analysis that follows can be easily generalized to all kinds of quantum computers nowadays available. Any quantum algorithm is first translated into a quantum assembly language, OpenQASM (Cross et al. 2017), through which it is re-expressed in terms of single- and two-qubit gates. At this point, the mapping of the connectivity structure of the circuit on the device takes place, by inserting a set of SWAP gates in the quantum circuit translated by OpenQASM. Such process makes the original circuit compatible with the topology of the computer (Murali et al. 2019; Acampora and Schiattarella 2021). This step is essential because different quantum computers can have drastically different qubit layouts and connectivity. For instance, some quantum computers may have a linear arrangement of qubits, while others may have a grid-like structure. Additionally, the mapping algorithm aims to minimize the number of SWAP gates inserted while maintaining the connectivity required by the program. This approach can reduce the program's execution time and optimize the program's performance on the quantum hardware (Kamaka 2020). A further translation step takes place, where the basis gates of the compiled circuit are translated in the basis gate of the quantum hardware, as they are generally different. It is worth to notice here that every quantum hardware will have different basis gates: the optimization will thus lead to different transpiled circuits. The performance and the computational cost of running the algorithm will heavily depend on the transpilation strategy: when running onto a real hardware in fact the number of SWAP gates inserted will generally increase the depth of the circuit, thus making the computation more complex in terms of running steps. Moreover, more gates means more gase noise, i.e. an overall greater error probability during the computation. It is hence crucial to account for the transpilation when evaluating the performances. Any quantum

hardware will have, in general, different basis gates: the optimization will thus lead to different transpiled circuits. At the end of the entire pipeline described above, the computational structure of algorithm is optimally or sub-optimally mapped onto the quantum computer: in the following the term topology is specifically referred to the entire protocol aforementioned, i.e. to the transpilation of the quantum algorithm on the hardware geometrical properties, being both the physical connectivity and the basis gates on which it needs to be expressed. In this work the hardware considered is from the IBM quantum experience: IBM in particular allow the user to manipulate and control, up to a certain level, the transpilation process of the program on the quantum machine, via Qiskit (Aleksandrowicz et al. 2019). Qiskit entails a default heuristic stochastic algorithm for the optimization of the number of SWAP gates that need to be inserted to map effectively the quantum program onto the machine (Li et al. 2018). Given its stochastic nature and the broadness of the associated distribution of SWAP gate inserted, for considering the effective depth of the transpiled circuit, it is thus necessary to consider the mean value over a set of possible realization of the optimization algorithm.

The arrangement and connections of qubits in nowadays available geometries are typically sparse compared to a fully connected quantum computer with N qubits, which has $O(N^2)$ connections. However, as the number of physical qubits in a quantum computer increases, with most of the available geometries (https://www.research.ibm.com/ibm-q/technology/devices/) the physical qubit connections become sparser, leading to the insertion of a considerable number of SWAP gates to compensate the sparsity of the connections. This trend is expected to continue, resulting in a larger number of extra SWAP gates in the near future. For this reason, it is crucial to take the effects of topologies with few connections into account while developing effective quantum programs.

The algorithms investigated in this experimental work, VQE and VQC, are particularly suitable to address the effect of the topology of the hardware, as they heavily rely on parametrized ansatz that needs to be mapped on the quantum computer. In VQE, the hardware topology affects the way that quantum gates are applied to the quantum state, and the resulting quantum state will depend on the basis gates of the quantum computer in use and on its underlying connectivity of the qubits. The VQE algorithm is designed to find the lowest eigenvalue of a Hamiltonian, which represents the energy of a quantum system. In order to achieve this, the VQE algorithm applies a series of quantum gates to the quantum state, and the specific set of quantum gates that are applied will depend on the hardware topology.

For VQC, the hardware topology affects the way that the quantum state is transformed during the training process. The goal of a VQC, and of QNN in general, is to learn a mapping from input data to output data, and this is achieved by adjusting the parameters of the quantum gates that are applied to the quantum state. The specific form of the quantum gates that are applied will depend on the connectivity of the qubits, and this can impact the overall performance of the algorithm. The understanding of the effect of the hardware topology on VQE and VQC fosters an informed decision-making process about the quantum hardware to use, which is particularly relevant in the NISQ era, and the parameters of the variational algorithms. This can lead to improved performance, reduced error rate, avoidance of barren plateau, and more efficient use of the available resources, as it will be shown in the following.

2.5 Barren plateau

The barren plateau phenomenon is a well-known problem that arises in variational quantum circuits when optimizing the parameters using a classical optimization algorithm. This phenomenon occurs when the optimization landscape becomes increasingly complex and the gradients become exponentially small as the number of qubits and parameters in the circuit increases (Cerezo et al. 2022). Formally, consider the cost function $\tilde{C}(\theta)$: this function is said to experience a barren plateau if for all $\theta_i \in \theta$, the expectation value of the partial derivates, $E_i[\partial_{\theta_i}\tilde{C}(\theta)] = 0$ and the variance tends exponentially to zero whit a growing number of qubits. The occurrence and the severity of barren plateaus can be propelled by various co-occurring phenomena, such as quantum noise, circuit complexity and by the connectivity of the qubits in the quantum computer (McClean et al. 2018). If the qubits are highly entangled, the optimization landscape can become more complex, leading to the appearance of more severe barren plateaus. There is an ongoing active exploration of various strategies to mitigate barren plateau, including using more efficient optimization algorithms, optimizing the circuits in a hierarchical manner, or designing circuits that have fewer entangling gates (Grant et al. 2018; Cerezo et al. 2021; Tuysuz et al. 2022). Noise too (Wang et al. 2021) can have a detrimental effect during training, leading to a complex optimization landscape and ultimately to barren plateau. The key mechanism in this case is that the quantum states involved in the computation get destroyed or deeply modified by the noise, giving rise to a flat optimization landscape. These noise-induced plateaus are conceptually different from standard barren plateaus: while the latter emerge from random parameter initialization, the former are caused by the convergence of the quantum state to a mixed state, thus flatting the probability distribution of the output. Hence, the combined effect of qubit connectivity and of noisy computation is the main phenomenon to take into account when facing vanishing gradients on real quantum devices. Further, it has been recently demonstrated that the appearance of barren plateau is non-exclusive of gradient-based method for the parameter

updates of the circuits, but gradient-free optimizers too suffer exponential suppression of convergence in the optimization landscape (Arrasmith et al. 2021). These results are validated in the present work, where a gradient-free optimizer, namely Cobyla (Powell 1994), has been used for the experimental phase. This implies that the barren plateaus are ubiquitous in quantum machine learning and hence their functional mechanism, together with a set of practices to avoid or mitigate the phenomenon, need to be addressed for the efficient usage of quantum machine learning in real-world application.

3 Methods and materials

The performance of VQE and VQC are evaluated using different quantum computers, with various topologies, which are defined by the arrangement of qubits, their connectivity together with the whole transpilation process of the circuit onto the quantum hardware selected. The experiments are conducted on real IBM quantum computers, through the IBM quantum experience (IBM quantum n.d.). Both VQE and VQC are realized by means of the Qiskit framework, which is an open-source software development kit (SDK) for quantum computing, on real IBM quantum computers. VQE was used to estimate the optimal portfolio, out of a collection of financial assets, by minimizing the expected value of the portfolio's risk while maximizing the return. On the other hand, VQC was trained on the Iris dataset for classification. The quantum circuits for both VQE and VQC were constructed using a set of quantum gates that define the ansatz of the circuits. Different ansatzes were used, with various entanglement topologies, i.e. various connectivity maps among each qubit. The Qiskit package contains several pre-designed ansatz. In this work two are selected for the experimental phase: *Two Local* ansatz, which consists in alternating single-qubit rotations and entangling gates (controlled rotations) in pairs, and *Real Amplitude* ansatz, where the single-qubit rotations are predefined around the Y axis and the entangling gates are made of controlled X gates. The amplitude of the emerging quantum state will always have zero imaginary part, thus being always made of real numbers. For all the ansatz considered, four possible entanglement structures are checked in the experiments: the *full* entanglement, the *linear* entanglement, the *circular* and the *pairwise* entanglement. Fundamentally, these entanglement assumptions differ in sparsity and connectivity, where qubits may be entangled in pairs with a fully connected geometry (full entanglement), or only respect to the nearest neighbour without circular boundary conditions (linear entanglement) and with circular boundary conditions (circular entanglement). The pairwise entanglement instead has a multilayer structure, where on the first layer just the even-numbered qubits are entangled, while in the second layer all the odd-numbered qubits are connected through entangling gates. To investigate the effect of quantum computer topology on the performance of the algorithms, different quantum computers are used for the experiments. These carry various coupling maps, which define a specific topology: Furthermore, these computers will have different basis gates; hence, the overall transpilation process of the circuit on the hardware considered may heavily vary. Specifically, for the sake of the experimental phase, for any hardware and ansatz considered a medium level of optimization has been considered, i.e. each circuit is transpiled considering a sub-optimal number of SWAP gates for each ansatz (Transpiler – IBM n.d.) . The quantum computer used in the experiments were IBM's *ibmq toronto*, *ibm geneva*, *ibmq guadalupe*, *ibm hanoi*, *ibm auckland*, *ibm cairo*, *ibmq montreal*, *ibmq mumbai* and *ibmq kolkata*. These computers are different in terms of the number of qubits, their connectivity, and the number of gates required for transpilation. Transpilation consists in rewriting a given input circuit so that the topology of the specified quantum device is properly matched. This process is crucial for understanding the effect of topology on the computation: the ansatz used for constructing the variational algorithm is transpiled on the real machine, i.e. it is deployed on the real quantum hardware and expressed, structurally, in terms of a compatible set of unitary gates representing the specific structure of the ansatz itself, in particular its entanglement map. To evaluate the performance of the algorithms, the convergence rate of the objective functions is computed. For VQC, a metric of the quality of the classification is further considered to evaluate the performance of the algorithm; specifically, the metric employed is the *F1-score*. The parameters of the experimental phase, described in detail in the following section, have been selected after a grid search in the parameter space: specifically among various classical optimizers (Cobyla, Adam, SPSP and NFT) Cobyla showed the best performances, i.e. convergence rate and global optimum, for various initial conditions. The same search was performed on the learning rate: considering the values 0.1, 0.01, 0.001, the optimal choice for different initial conditions was found to be 0.001.

3.1 Data

For the portfolio optimization task performed with VQE, data are collected from Yahoo!@finance (Yahoo Finanza n.d.) using *yfinance* (https://pypi.org/project/yfinance). Small-scale examples are realized, by selecting $N = 2$ diverse assets, (Apple, IBM), to examine the effectiveness of the suggested technique. These are exemplary global assets are quite interesting as their dynamics determined by monetary and social phenomena. For each asset i, with $1 \leq i \leq N$,

the temporal range between 2011/12/23 and 2022/10/21 is considered. The asset performance for each day t in the selected range ($0 \leq t \leq T$) is properly captured by its closing price p_i^t.

The *Iris dataset*, on the other hand, is utilized for the classification task with the VQC. It is a widely used benchmark dataset in machine learning and is publicly accessible on the *Scikit-learn* (Pedregosa et al. 2011) Python library. The Iris dataset feature vectors were preprocessed and normalized to be used in the algorithms.

3.2 Portfolio optimization

In the following, the portfolio optimization problem and its quantum formulation are briefly outlined: a more detailed discussion about the formalism hereby described can be found in Buonaiuto et al. (2023).

Portfolio Optimization (PO) may be expressed in a variety of ways, each of which approximates the real-world problem (Markowitz 1952) to varying degrees of complexity. Here, the emphasis is on multi-objective portfolio optimization, which makes use of the budget at hand to be invested, while attempting to simultaneously maximize return and reduce risk. The set of investments d_i (measured as a percentage of the budget or number of asset units) allotted for each ith asset on the market is referred to as a portfolio. A PO seeks to maximize the portfolio's return, $\mu^{\mathrm{T}} d$, while minimizing risk $d^{\mathrm{T}} \Sigma d$, which states the portfolio variance, where μ denotes the vector of average asset returns, Σ is the covariance matrix and d denotes the vector of investments expressed as fractions of the total available budget. Hence, for PO the objective function can be written as follows:

$$\mathcal{L}(d) : \mu^{\mathrm{T}} d - q d^{\mathrm{T}} \Sigma d, \tag{4}$$

where the risk aversion parameter q expresses the propensity to risk of the investor. In a realistic scenario, the available budget B of an investor is fixed, within a certain time window of investments. Therefore, the sum of all investments d_i must be equal to 1. Furthermore, only buying is going to be allowed in the model, thus imposing $d_i \geq 0$. The optimization problem can be thus written as:

$$\begin{aligned} &\max_d \mathcal{L}(d) : \max_d (\mu^{\mathrm{T}} d - q d^{\mathrm{T}} \Sigma d), \\ &\text{s.t.} \quad \sum_{i=1}^{N} d_i = 1 \end{aligned} \tag{5}$$

The product $d_i B$ must be an integer number, multiple of the price P_i: these constraints restrict the solutions of the continuous problem to a subset of acceptable possibilities. Hence, it is convenient to restate the entire problem as follows:

$$\begin{aligned} &\max_m \mathcal{L}(m) : \max_m (\mu'^{\mathrm{T}} m - q m^{\mathrm{T}} \Sigma' m), \\ &\text{s.t.} \quad P'^{\mathrm{T}} m = 1 \end{aligned} \tag{6}$$

where m is the vector of integer units of every asset, while $P' = P/B$, $\mu' = P' \circ \mu$ and $\Sigma' = (P' \circ \Sigma)^{\mathrm{T}} \circ P'$ are appropriate transformations of μ and Σ. With the above formulation Eq. (6) the problem is transformed into an integer quadratic unconstrained optimization.

3.2.1 Quantum formulation of PO

To solve the PO Eq. (6) above onto a quantum computer it is first necessary to transform the target vector into a binary string f, i.e. into vectors composed of zeros and ones. This is a necessary step to encode the problems variables into a quantum state expressed in the computational basis. As a result, for each asset i, depending on its price P_i, a certain amount of binarizing elements k_i are used to construct a binary conversion matrix C for the vector m. More formally:

$$m_i^{\max} = Int\left(\frac{B}{P_i}\right), \tag{7}$$

and

$$k_i = Int\left(\log_2 m_i^{\max}\right), \tag{8}$$

such that

$$m_i = \sum_{j=0}^{k_i} 2^j f_{i,j}. \tag{9}$$

With such conversion mechanism, the binarized target vector has dimension $f = [f_{1,0}, \ldots, f_{1,k_1}, \ldots, f_{N,0}, \ldots, f_{N,k_N}]$, is $\dim(f) = \sum_{i=1}^{N} (k_i + 1)$, lower than the standard Qiskit implementation (Portfolio Optimization n.d.). It is convenient to define the conversion matrix C as follows:

$$C = \begin{pmatrix} 2^0 \ldots 2^{k_1} & 0 \ldots 0 & \ldots & 0 \ldots 0 \\ 0 \ldots 0 & 2^0 \ldots 2^{k_2} & \ldots & 0 \ldots 0 \\ \vdots \ddots \vdots & \vdots \ddots \vdots & \ddots & \vdots \ddots \vdots \\ 0 \ldots 0 & 0 \ldots 0 & \ldots & 2^0 \ldots 2^{k_N} \end{pmatrix}, \tag{10}$$

and thus, the conversion can be written in short notation as $m = Cf$. It is possible to redefine the problem Eq. (6), in terms of the binary vector b, applying the encoding matrix

by $\mu'' = C^{\mathrm{T}}\mu'$, $\Sigma'' = C^{\mathrm{T}}\Sigma' C$ and $P'' = C^{\mathrm{T}}P'$:

$$\max_f \mathcal{L}(f) : \max_f \left(\mu''^{\mathrm{T}} f - q f^{\mathrm{T}}\Sigma'' f\right), \quad \text{s.t.} \quad P''^{\mathrm{T}} f = 1 \quad f_i \in \{0,1\} \quad \forall i \in [1,\ldots,dim(f)]. \tag{11}$$

It is now necessary to convert the problem Eq. (11) into a QUBO by performing a transformation onto the constraints, turning them into a penalty term in the objective function. Different constraints can be turned in specific penalties (Glover et al. 2019), and specifically the one considered in this work, a linear equality term, is transformed into $\lambda(P''^{\mathrm{T}} f - 1)^2$. Hence, the problem Eq. (11) is rewritten as a QUBO of the form:

$$\max_f \mathcal{L}(f) : \max_f \left(\mu''^{\mathrm{T}} f - q f^{\mathrm{T}}\Sigma'' f - \lambda(P''^{\mathrm{T}} f - 1)^2\right), \tag{12}$$

where λ is the penalty strength. It is now possible to convert the QUBO problem of the PO into an Ising Hamiltonian (Lucas 2014). Since the Ising model is written in terms of spin decision variables z_i with values $\{-1, 1\}$, the components of the binary vector are first converted according to the rule $f_i \to \frac{1+z_i}{2}$, thus obtaining:

$$\min_z \mathcal{L}(z) : \min_z \left(\sum_i h_i z_i + \sum_{i,j} J_{i,j} z_i z_j + \lambda(\sum_i \pi_i z_i - \beta)^2\right), \quad \text{s.t.} \quad z_{i,j} \in \{-1,1\} \quad \forall i, \tag{13}$$

with $J_{i,j}$ being the coupling term between two spin variables. The quantum Hamiltonian can be now constructed introducing the Pauli operators Z, with eigenvalues ± 1:

$$H = \sum_i h_i Z_i + \sum_{i,j} J_{i,j} Z_i \otimes Z_j + \lambda(\sum_i \pi_i Z_i - \beta)^2. \tag{14}$$

Following the steps above, the PO problem Eq. (6) is now formulated as a search problem of a ground state, i.e. the minimum energy eigenstate, of the Ising Hamiltonian Eq. (14). In particular, the ground eigenstate corresponds to the solution of the problem, i.e. to the optimal vector of assets allocation. It is therefore possible to use the VQE, to build a variational wavefunction that iteratively converges to the target ground state, which corresponds to the optimal portfolio.

3.3 Classification

Classification is a machine learning task that involves predicting a discrete target variable based on a set of input features. The Iris dataset is a classic example of a classification problem: it is made of a collection of measurements of the sepal length, width, petal length and petal width of three species of Iris. The general task of classification on the Iris dataset can be expressed mathematically as follows: let $\boldsymbol{x} = [x_1, x_2, x_3, x_4]$ be the vector of input features, where x_1 represents the sepal length, x_2 represents the sepal width, x_3 represents the petal length, and x_4 represents the petal width. Let $y \in 0, 1, 2$ be the target variable, where 0, 1, and 2 represent the three different species of iris flowers: *setosa*, *versicolor*, and *virginica*, respectively.

A function $f : \mathbb{R}^4 \to 0, 1, 2$ is learned that maps an input feature vector $\boldsymbol{x}$ to a predicted species label y. The goal is to obtain a trained model that is able to predict with accuracy the species of an Iris flower based on its measurements. To achieve this goal, various classification algorithms such as logistic regression, decision trees, random forests, support vector machines, or neural networks can be used. These algorithms learn a decision boundary in the input feature space that separates the different species of iris flowers. The decision boundary is determined by a set of model parameters that are learned from the training data. The trained model can be used to predict the species labels y for new input feature vectors $\boldsymbol{x}$ that were not seen during training. The accuracy of the classifier can be measured using various evaluation metrics, such as accuracy, precision, recall, F1 score, or area under the ROC curve.

3.3.1 VQC on the Iris dataset

The input features of the Iris dataset are the sepal length, sepal width, petal length, and petal width of the flowers. The VQC algorithm requires that these data need to be encoded in a quantum state. This is achieved by applying a feature map, which is a quantum circuit that transforms the input features into a quantum state. One example of a feature map, used extensively in this work, is the *ZZ*-Feature Map (Havlicek et al. 2019), $U_{\xi(\boldsymbol{x})}$, which is defined as follows:

$$U_{\xi(\boldsymbol{x})} = \exp\left(-i \sum_D \xi_D(x) \prod_{i \in D} Z_i\right), \tag{15}$$

where $\boldsymbol{x}$ is the input data vector, D is the sublist of indices related to train data entries, and Z_i are Pauli-Z operators applied to the i^{th} qubits of the quantum circuit. The funcion ξ_D is defined as $\xi_i(x) = x_i$ and $\xi_{i,j}(x) = (\pi - x_i)(\pi - x_j)$. The *ZZ* feature map is a two-qubit entangling gate that generates a pairwise interaction between each pair of qubits. The pairwise interaction is proportional to the product of the corresponding input features.

After applying the feature map, the resulting quantum state is passed through an ansatz. The parametrized quantum cir-

cuit associated with the ansatz is used to encode the target labels of the classification problem. The parameters of the ansatz are optimized during the training process to minimize a cost function. The cost function measures the difference between the predicted labels and the actual labels of the training data.

The output of the ansatz is a quantum state that represents the predicted label for the input feature vector. The predicted label is obtained by measuring the expectation value of a Hermitian operator M on the final quantum state:

$$y = \text{argmax}_k \langle \psi_{\boldsymbol{\theta}} | M_k | \psi_{\boldsymbol{\theta}} \rangle$$

where $\psi_{\boldsymbol{\theta}}$ is the final quantum state after applying the feature map and the ansatz with parameters $\boldsymbol{\theta}$, M_k is a Hermitian operator that corresponds to the k^{th} target label, and y is the predicted label. During the training process, the parameters of the ansatz are optimized using classical optimization algorithms such as gradient descent or Bayesian optimization. The optimization process seeks to find the parameters that minimize the cost function, which measures the difference between the predicted labels and the actual labels of the training data. Traditionally, ansatzes are designed specially for different applications. For instance, the hardware-efficient ansatz (Kandala et al. 2017) and the molecular physics-inspired UCCSD (Barkoutsos et al. 2018; Sokolov et al. 2020) are designed for specific problems to be tackled with VQE algorithms. In the case of QNNs, or quantum machine learning algorithms like VQC, a general procedure about the finding of optimal ansatz is lacking. Hence, it is customary to make use of general purpose parametrized circuit, which do not find a clear interpretation for the problem, but have a number of parameters and a rich connectivity structure, sufficient to tackle the complexity of the task. Once the parameters of the VQC are optimized, the trained model can be used to classify new input feature vectors. The new input feature vector is first encoded into a quantum state using the feature map again. The resulting quantum state is then passed through the ansatz with the optimized parameters to obtain the predicted label.

4 Results and discussion

4.1 VQE for PO

The results of the convergence of the objective function (energy) for the VQE on the PO problem are presented in Fig. 1. The energy of the parametrized ansatz for each quantum hardware reaches a convergence value that changes according to the ansatz selected and to the machine considered. This fact simply states the co-occurrence of two effects: the intrinsic quantum noise, both gate noise and readout noise, which is different in magnitude and form among quantum

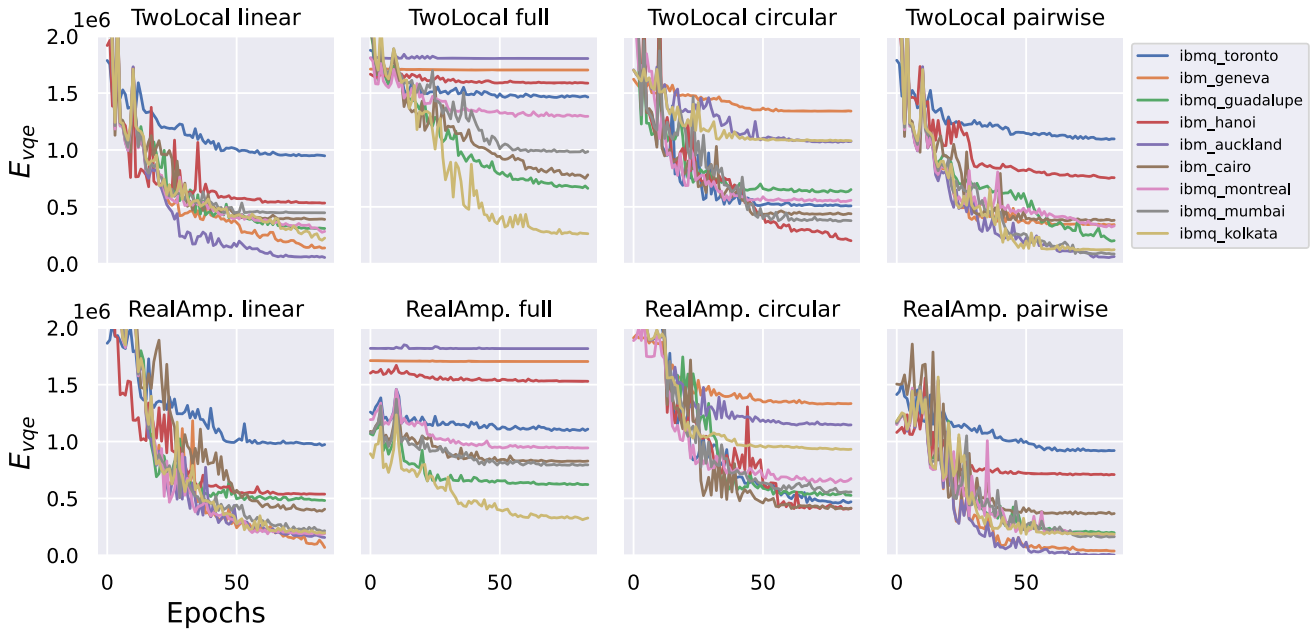

Fig. 1 Convergence of the objective function for the PO problem with VQE. The problem is formulated for three assets and two encoding bits. A total budget of $B = 1000$ is considered. In particular, the convergence of various ansatz on different quantum hardware is explored. It can be seen that the minimum reached changes dramatically depending on where, in terms of quantum hardware, the ansatz is transpiled. Furthermore, the convergence speed, i.e. the number of epochs required for reaching the minimum, also depends on the combination of the topology of the hardware and the ansatz in use

devices, and the computational overheads introduced by the transpilation of the ansatz on each device. The latter consideration is made clear by inspecting the behaviour of objective function for each machine for a given ansatz: the result can differ drastically, ranging from an almost zero decrease of the energy (such as for the *TwoLocal full* with *ibm auckland*) to a nearly optimal minimum (*TwoLocal full* with *ibm kolkata*). It is here argued that the problem of vanishing gradient, i.e. zero or nearly zero changes in the loss function, often referred to barren plateau in quantum machine learning, it is a consequence of the combined effect mentioned above. This point will be addressed for the VQC later on in this work. To find out which are the key figures of merits to take into account when mapping each ansatz on the hardware topology, data about the value at convergence from every training performed with VQE are aggregated with respect to the quantum machine used. The results are shown in Fig. 2, where the aggregated data correspond to the ratio between the classically calculated minimum of the objective function and the minimum found via the VQE, so that they are upper bounded by 1, i.e. a perfect agreement of the variational solution with the classical results. The classical results are found via the branch-and-bound method (Marinescu and Dechter 2006; Niu et al. 2008). While each hardware, as previously stated, shows high variability depending on the ansatz transpiled on it, there is a clear trend in the mean value and the median of the ratio E_{clas}/E_{vqe}, over the ansatz. Specifically, the hardware is ordered with growing quantum volume. It is possible to notice that, in fact, higher quantum volumes not only give better convergence for a larger set of variational wavefunctions, but the variability of the results tends to be more contained. These findings are in line with the fact that hardware with higher quantum volumes allows deeper circuits to be executed efficiently; hence, the transpilation of more structured ansatz, such as the *TwoLocal full* or *RealAmplitude full*, take please without hindering the convergence rate. Conversely, it can be noticed that hardware with lower quantum volume, in particular *ibm geneva*, has high variability in the results. However, effectively, ansatz with a lower circuit depth after transpilation, i.e. ansatz with a limited connectivity among qubits, manages to converge to the ground state of the problem. These results provide the first heuristic procedure in selecting a proper ansatz for a given quantum hardware: the quantum volume of the machine needs to be high enough to allow an effective computation of deep circuit, thus enabling to use ansatz with a complex entanglement structure. When this is not the case, hence for limited quantum volume, a linear or pairwise entanglement structure of the ansatz is advisable, as it hinders the excessive growth of the error rate and fosters the convergence during training.

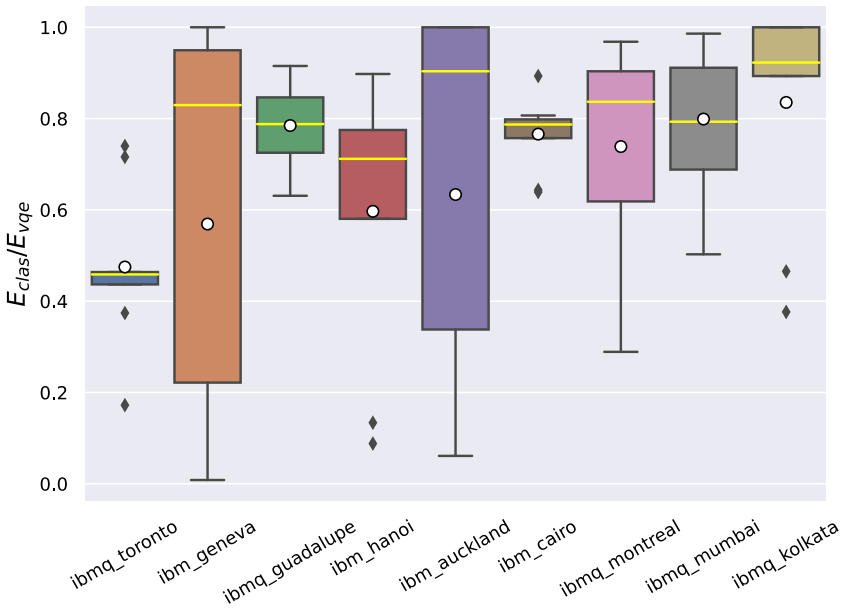

Fig. 2 Boxplot of the ratio between the classically obtained energy and the minimum energy with VQE for the PO problem. Outliers are represented by black squares. White dots indicate the mean of the result distribution for each quantum hardware. The convergence value of minimum energies are grouped over the set of ansatzes considered, for each quantum hardware. Quantum hardware is listed in terms of increasing quantum volume. A clear trend emerges, as the quantum volume increases. However, specific ansatzes perform better on smaller machines. Notably, the *ibm guadalupe* machine shows performances comparable to hardware with higher quantum volume. Quantum computers with lower quantum volume on the contrary experience a higher variability of the solution, thus fluctuating performances

4.2 VQC for Iris classification

The results of training for the VQC on the Iris dataset are shown in Fig. 3. A Cobyla optimizer is employed here too, as the VQC inherently transform the classification problem into an optimization. In line with what observed previously with the VQE, a varying behaviour of the convergence rate is observed, depending on the hardware in use and on the ansatz. Additionally, here the transpilation of the *ZZ*-feature map takes place too, adding an additional layer of gates to the final quantum circuit to be trained. Compared to the VQE, here the behaviour of the objective function is nearly identical for all the quantum hardware considered, given a specific ansatz. The convergence value shifts as a consequence of the different error rate probabilities associated with each machine, and for the overhead of noise introduced by the transpilation of the variational circuit structure. Similar features in the behaviour of the objective function, compared to the previous case, can be found: most strictly, some hardware with limited number of qubits and low quantum volume allows an efficient training of a subset of the ansatz. This validates further the statements previously made. Furthermore, there is an evident lack of trainability for a single machine, namely the *ibm cairo*, which find itself locked in a barren plateau for virtually any configuration. In the next subsection this occurrence will be investigated in higher detail. In these experiments also, data from every training of the VQC are grouped to visualize the effectiveness of each quantum hardware to foster the convergence on the entire collection of variational ansatz. Results can be visualized in Fig. 4. Here, both the last epoch of the objective function (Fig. 4a)) and the F1-score are plotted (Fig. 4b)). In particular the F1-score measures the quality of the classification on the test, weighting the classified and the correctly classified data for each of the three classes in the Iris dataset (cit). From both panels in Fig. 4a trend, although slightly less pronounced than the VQE case, is visible. Higher quantum volume fosters a reduction in the objective function and on average higher performances. Here too, for some smaller quantum hardware, such as *ibm guadalupe* and *ibm geneva*, good performances are obtained, at least for some ansatz. The lack of, or poor, training for the *ibm cairo* is here spotted too, both in terms of loss function and for the evaluation metric. The heuristic previously outline is overall confirmed by this set of experiments too, on a different dataset and with a different algorithm: higher quantum volume allows the efficient deployment of a vast class of parametrized ansatz with high connectivity between qubits, without causing a significant increase of the noise and thus

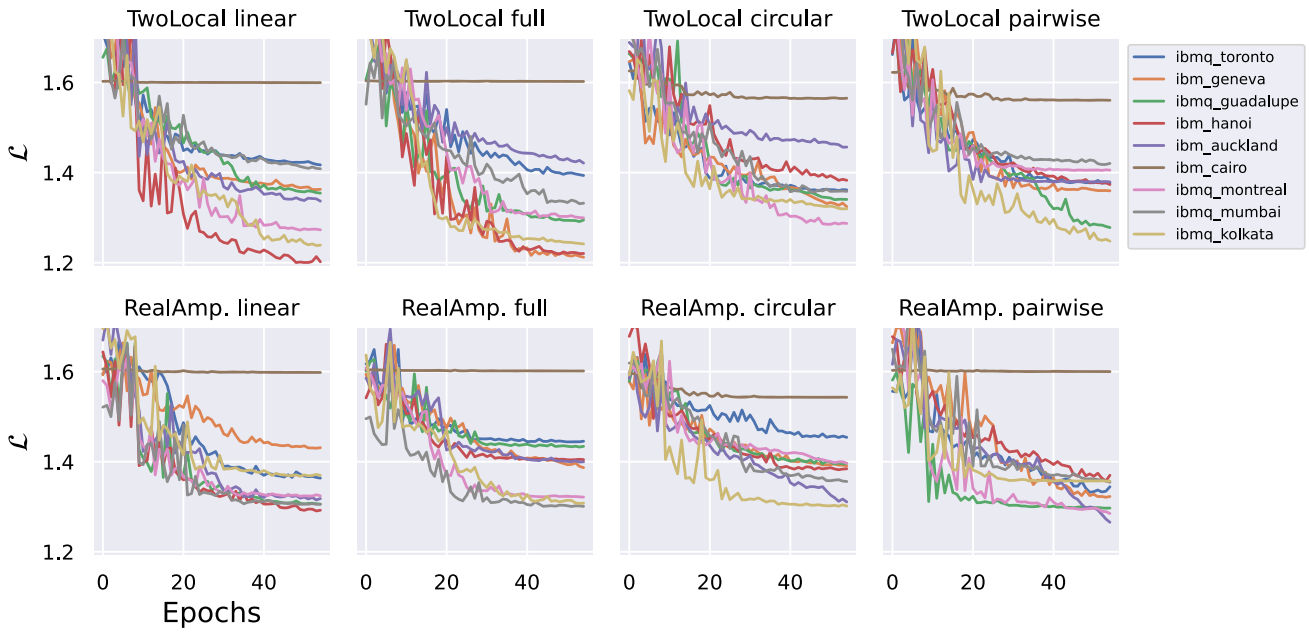

Fig. 3 Convergence of the objective function for the Iris classification problem with VQC. The classification is performed on the entire Iris dataset. The quantum feature map used is the *ZZ* Feature Map, identical for each experiment. The loss function is the *Cross entropy*, which is a standard for multi-class classification tasks, while the optimizer for the classical component of the computation is the *Cobyla*. In particular, the convergence of various ansatzes on different quantum hardware is explored. It can be seen that the minimum reached changes dramatically depending on where, in terms of quantum hardware, the ansatz is transpiled. Furthermore, the convergence speed, i.e. the number of epochs required for reaching the minimum, also depends on the combination of the topology of the hardware and the ansatz in use. Finally, a barren plateau can be observed, i.e. the loss get stucked in a sector of the parameter space with zero gradients, thus making the train unfeasible. In this experiment the barren plateau appears in the training of the Cairo quantum computer, for every ansatz, indicating that this behaviour is given specifically by the mapping of the problem on the topology of the device

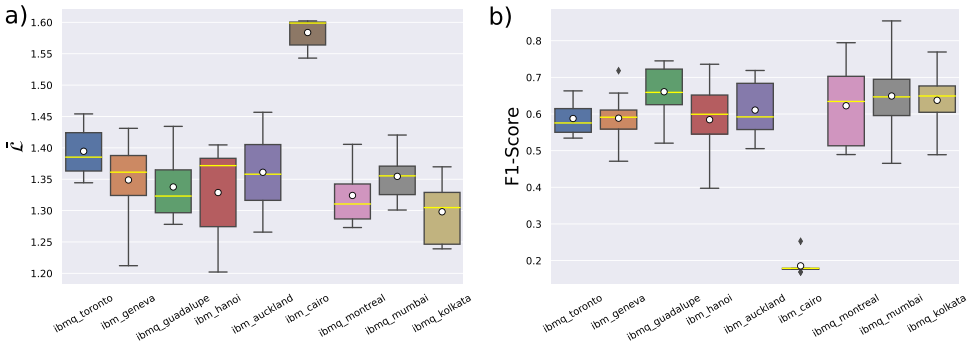

Fig. 4 Boxplot of loss function distribution for VQC and F1-score on the test set. Outliers are represented by black squares. White dots indicate the mean of the result distribution for each quantum hardware. **a** Distribution of the convergence value of the loss function, over the set of ansatzes considered, for each quantum hardware. Quantum hardware is listed in terms of increasing quantum volume. A clear trend emerges, as the quantum volume increases. However, specific ansatzes perform better on smaller machines. The outlier box, corresponding to the *Cairo* quantum computer, found itself in a barren plateaus, which hinder a proper training. **b** F-1 score on the Iris test set assessing the quality of the classification. The mean behaviour of each quantum hardware maps the trend of the loss function, with higher quantum volume giving higher median results. The lack of training for the *Cairo* machine, given by the barren plateau, effects systematically the quality of the classification

a detrimental effect on the performances. However, poorly connected ansatz can be as performative as the highly connected one when deployed on smaller machines, i.e. with less quantum volume and number of qubits.

4.3 A heuristic for the decision-making process

Based on the results of the experimental phase discussed above, it is possible to construct a heuristic function that aids in the selection of an ansatz, or in the definition of a connectivity map for an ansatz, given the underlying topology of the quantum hardware. It has been shown that quantum volume is a favourable figure of merit for this purpose. This by itself is not sufficient, as the length of the transpiled circuit needs to be taken into account. As mentioned in the previous section, the transpilation process adds SWAP gates to the circuit, increasing the computational cost and the probability of error: hence rather than the entire quantum volume, it is the ratio between the mean depth of the transpiled circuit D_c and the quantum volume of the hardware Q_V that plays an important role in defining the perspective performances of the algorithm. This ratio represents the resource cost of the deployment of the parametrized circuit on the specific machine. To construct the heuristic measure though, another feature, emerging from the experiment, needs to be accounted for. In principle, computers with higher number of qubits are more costly to manipulate and the connection between each computational unit harder to optimize, as the interaction between qubits becomes sparser. Hence, another quantity of interest is the ratio between the number of qubits to be used (N_q) and the number of qubits in the hardware (N_T). When this ratio is too large, optimization of the number of SWAP gates during transpilation becomes harder and the probability of ending up in a sub-optimal configuration is higher. Putting these considerations together, a candidate for the heuristic measure that aids the decision-making process for the deployment of the ansatz reads:

$$\xi_q = \frac{N_q}{N_T} \log\left(\frac{Q_V^2}{D_c} + 1\right), \tag{16}$$

where the log function has been inserted for favouring a higher quantum volume with small transpilation depth. The higher the value of the ξ_q function, the better, on average, the performance of the ansatz on the quantum hardware. Values of the ξ_q function for the quantum machines and the ansatz presented in this work are shown in Fig. 5: machines with higher quantum volumes, such as *ibm montreal*, *mumbai*, and *kolkata*, are clearly favoured for each and every ansatz, in line with the experimental results for VQE and VQC. Furthermore, *ibm guadalupe*, which has lower number of qubits (16 qubits) compared to the other hardware (27 qubits), is expected to be quite optimal, according to the result of the heuristic function. This expectation is confirmed again by the experimental results: both in the VQE case and for the VQC, the *ibm guadalupe* performs better of hardware with similar quantum volumes and for a larger set of variational ansatz. It is worth to notice that such function is problem independent, as it deals directly with the characteristic of the machine, with the number of qubits to be used and with the transpilation process. Furthermore, notice that, as the depth of the transpiled circuit, specifically the number of SWAP gates, is generated by a stochastic optimization algorithm, it is necessary to evaluate the function over a set of run, taking the mean value of the distribution as a figure of merit. In the results shown in Fig. 5 each ansatz was transpiled 200 times using the optimization algorithm contained in the Qiskit package.

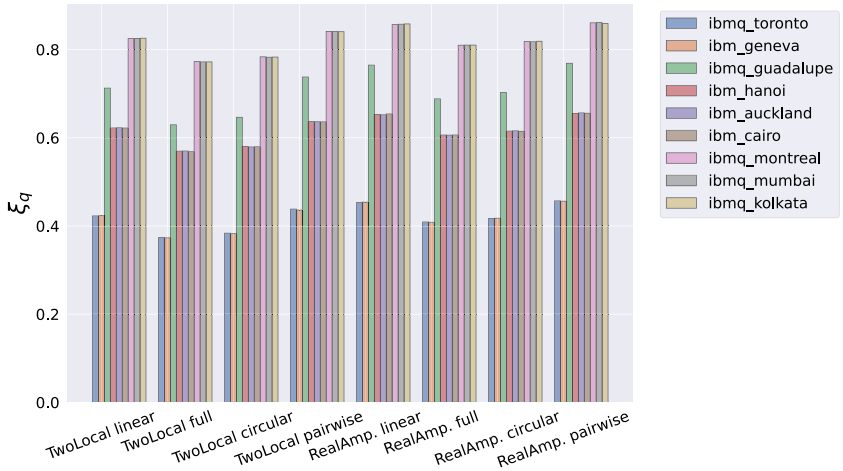

Fig. 5 Heuristic function for evaluating the effectiveness of an ansatz on a quantum hardware. The definition of the function is given in Eq. 16. Notice that, while in general higher QV are favoured for deploying virtually any ansatz, this is not a straight rule. The number of qubit in the hardware plays a role: specifically for *ibm guadalupe*, which possesses 16 qubits, the heuristic function is well performing. The result is validated by the experimental phase in previous section

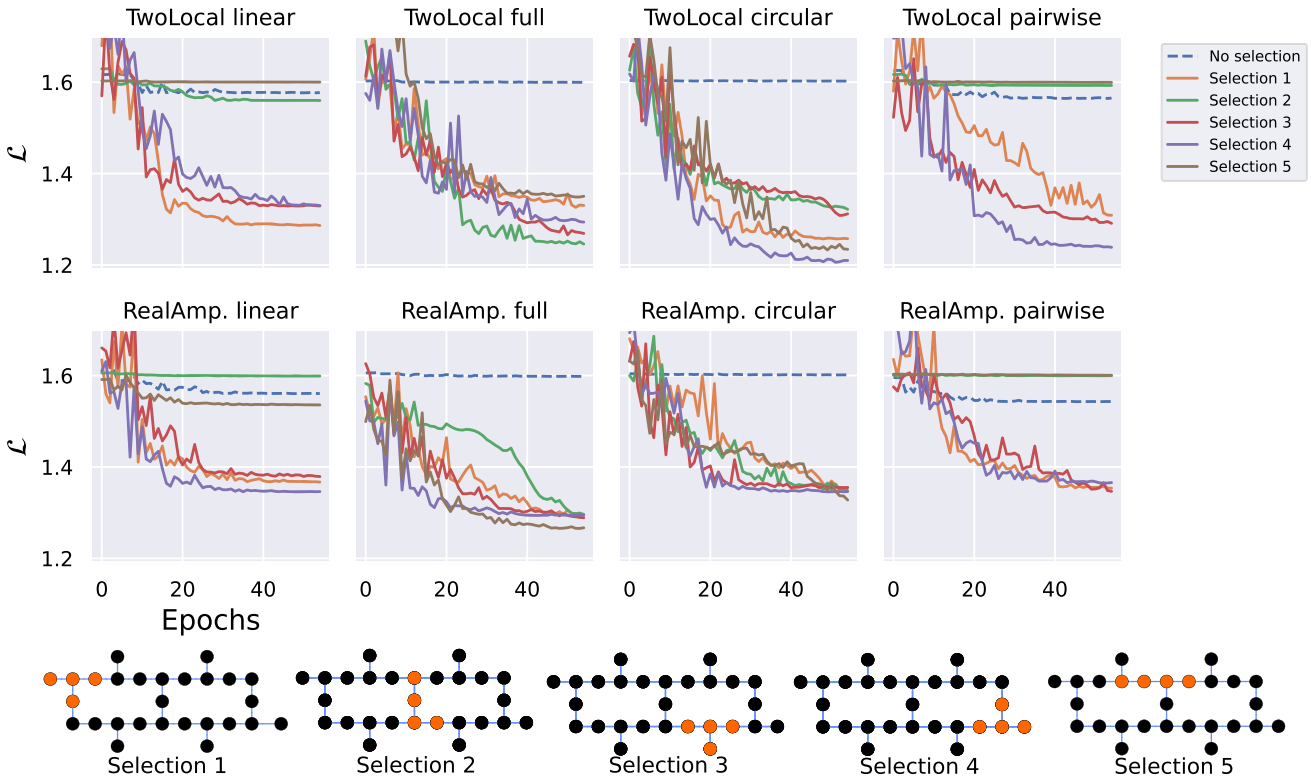

Fig. 6 Experiment on various local clusters of qubits of *ibm cairo* for Iris Classification. The classification is performed on the entire Iris dataset. The quantum feature map used is the ZZ Feature Map, identical for each experiment. The loss function is the *Cross entropy*, which is a standard for multi-class classification tasks, while the optimizer for the classical component of the computation is the *Cobyla*. In particular, the convergence of various ansatzes on different local clusters of qubits of the *ibm cairo* hardware is explored. It can be seen that, when no explicit selection is indicated (dashed line), the optimization is stacked in the barren plateau, in analogy with the original experiment. On the contrary, different selections allow the training to be performed with various degrees of efficiency. This is due to the combined effect of noise rate and different qubit connectivity, as the result strongly depends not only on the selection of qubit performed but on the ansatz in use as well, i.e. on the specific transpilation requirement of the variational wavefunction

4.4 Analysis of the *ibm Cairo* plateau

As explained in Section 2, the combination of non-efficient connectivity between qubits and quantum noise can foster the presence of noise-induced barren plateau during training. This is precisely what happens to the *ibm Cairo* machine in the previous experiments (VQC on Iris). As the problem is formulated on a subset of qubits (namely four) of the entire machine, it is possible to address the occurrence of the plateau, and to design possible way out of it, by considering different local clusters of qubits during training. These have, intrinsically, different average error rates associated with the logical operation (gates) that need to be performed to build up the computational pipeline. Furthermore, the efficiency of the connection between local clusters of qubits, i.e. the quality of the transpilation of interaction terms in the ansatz, will strongly depend on the proximity of the qubits involved and on the error associated with single-qubit gates transpilation. In Fig. 6 an extensive set of experiments on different cluster selections of the *ibm cairo* is performed. Each selection is used for training a VQC on the Iris dataset for the usual set of parametrized ansatz considered so far. The selections differ for the gate noise level and/or for their linear connectivity: in this way the combined effects can be addressed. There are in fact two main sources of noise, the one associated with the local level on the single qubit and the one emerging from the transpiled structure on the quantum hardware, which if not optimized increases the number of gates required and involved other qubits with their noise level. The first striking effect of selecting a cluster, compared to a random selection, is that very often the training escapes the barren plateau. This fact proves that both the connectivity of real qubits and their noise have an effect on training, given the same initial set of parameters and hyperparameters. Another important feature emerging from the experimental results is that Selections 1, 3, and 4, topologically identical but different in terms of single gate noise, give rise to a similar behaviour in the loss function for most of the ansatz. However, for the *TwoLocal full* and *RealAmplitude full*, Selections 2 and 5 are either favoured or comparatively similar to the others. This states the importance of the topology once again: the aforementioned ansatz is accommodated more efficiently, in terms of Swap gate overhead during transpilation and in terms or overall noise rate, by the selected cluster of qubits. This is not the case for the other ansatz, where the linear arrangement of the qubits for Selections 2 and 5 hinders the training, giving rise once more to barren plateaus. It is hence demonstrated that a possible way to mitigate the vanishing gradients in the training of a quantum machine learning algorithm such as the VQC, is to take into account various qubits arrangements, so that the most efficient mapping of the ansatz on the device can take place.

5 Conclusion and perspectives

In conclusion, this work has investigated the impact of quantum hardware topology on the performance of variational quantum algorithms (VQAs), taking VQE and VQC as main benchmarks. Specifically, it has been here experimentally demonstrated that complex connectivity in the ansatz can introduce an overhead of gates during transpilation on a quantum computer, which may increase the overall error rate and degrade the quality of results. It is important to strike a balance between a sufficiently parametrized ansatz and minimal cost in terms of resources during transpilation to obtain the best performance from VQAs. The experiments have been carried out on two widely used variational algorithms, VQE and VQC, for two different tasks — Markowitz portfolio selection and classification on the Iris dataset, respectively. The results show a remarkable dependence on the machine in use, hence on its underlying structure, for the quality of the convergence of the algorithm during training. This allows to construct a heuristic function to determine, a priori, which ansatz is expected to be deployed more efficiently on the quantum hardware. These findings provide valuable insights into the design of optimal ansatz and the use of VQA for practical applications, highlighting the importance of considering the noise and limitations of current quantum hardware.

Author Contributions G.B. and M.P. constructed the main research ideas. G.B. and M.E. and G.D. wrote the manuscript, F.G. revised it and contributed to the realization of the experiments.

Funding Open access funding provided by Consiglio Nazionale Delle Ricerche (CNR) within the CRUI-CARE Agreement.

Availability of data and materials All data used in the manuscript are publicly available on the referenced repositories. Code from the author is available upon reasonable request.

Declarations

Conflict of interest The authors declare no competing interests.

References

Abbas A, Sutter D, Zoufal C, Lucchi A, Figalli A, Woerner S (2021) The power of quantum neural networks. Nat Comput Sci 1(6):403–409

Acampora G, Schiattarella R (2021) Deep neural networks for quantum circuit mapping. Neural Comput Applic 33(20):13723–13743. https://doi.org/10.1007/s00521-021-06009-3

Aleksandrowicz G, Alexander T, Barkoutsos P, Bello L, Ben-Haim Y, Bucher D, Cabrera-Hernández F, Carballo-Franquis J, Chen A, Chen C et al (2019) Qiskit: An open-source framework for quantum computing. Accessed on: Mar. **16**

Anselmetti G-LR, Wierichs D, Gogolin C, Parrish RM (2021) Local, expressive, quantum-number-preserving vqe ansatze for fermionic systems. N J Phys 23(11):113010. https://doi.org/10.1088/1367-2630/ac2cb3

Arrasmith A, Cerezo M, Czarnik P, Cincio L, Coles PJ (2021) Effect of barren plateaus on gradient-free optimization. Quantum 5:558. https://doi.org/10.22331/q-2021-10-05-558

Barkoutsos PK, Gonthier JF, Sokolov I, Moll N, Salis G, Fuhrer A, Ganzhorn M, Egger DJ, Troyer M, Mezzacapo A et al (2018) Quantum algorithms for electronic structure calculations: Particle-hole hamiltonian and optimized wave-function expansions. Phys Rev A 98(2):022322

Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S (2017) Quantum machine learning. Nature 549(7671):195–202. https://doi.org/10.1038/nature23474

Buonaiuto G, Gargiulo F, De Pietro G, Esposito M, Pota M (2023) Best practices for portfolio optimization by quantum computing, experimented on real quantum devices. https://doi.org/10.1038/s41598-023-45392-w

Cerezo M, Sone A, Volkoff T, Cincio L, Coles PJ (2021) Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nat Commun 12(1):1791. https://doi.org/10.1038/s41467-021-21728-w

Cerezo M, Arrasmith A, Babbush R, Benjamin SC, Endo S, Fujii K, McClean JR, Mitarai K, Yuan X, Cincio L, Coles PJ (2021) Variational quantum algorithms. Nature Reviews. Physics. 3(9):625–644. https://doi.org/10.1038/s42254-021-00348-9

Cerezo M, Verdon G, Huang H-Y, Cincio L, Coles PJ (2022) Challenges and opportunities in quantum machine learning. Nat Comput Sci 2(9):567–576. https://doi.org/10.1038/s43588-022-00311-3

Cheng J, Wang H, Liang Z, Shi Y, Han S, Qian X (2022) Topgen: Topology-aware bottom-up generator for variational quantum circuits. arXiv:2210.08190

Clarke J, Wilhelm FK (2008) Superconducting quantum bits. Nature 453(7198):1031–1042

Coles PJ (2021) Seeking quantum advantage for neural networks. Nature Computational Science 1(6):389–390. https://doi.org/10.1038/s43588-021-00088-x

Crooks, G.E.: Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition. arXiv preprint arXiv:1905.13311. (2019)

Cross AW, Bishop LS, Smolin JA, Gambetta JM (2017) Open quantum assembly language. arXiv:1707.03429

Farhi E, Goldstone J, Gutmann S (2014) A quantum approximate optimization algorithm. arXiv:1411.4028

Global Quantum Computing Market Report (2022) Rising Investments in Quantum Technology Fuel Growth - ResearchAndMarkets.com. https://www.businesswire.com/news/home/20221129005908/en/Global-Quantum-Computing-Market-Report-2022-Rising-Investments-in-Quantum-Technology-Fuel-Growth---ResearchAndMarkets.com

Glover, F., Kochenberger, G., Du, Y. (2019) Quantum bridge analytics i: a tutorial on formulating and using qubo models. 4OR 17(4):335–371. https://doi.org/10.1007/s10288-019-00424-y

Grant E, Benedetti M, Cao S, Hallam A, Lockhart J, Stojevic V, Green AG, Severini S (2018) Hierarchical quantum classifiers. Npj Quantum Inf 4(1):65. https://doi.org/10.1038/s41534-018-0116-9

Häffner H, Roos CF, Blatt R (2008) Quantum computing with trapped ions. Phys Rep 469(4):155–203. https://doi.org/10.1016/j.physrep.2008.09.003

Havlicek V, Corcoles AD, Temme K, Harrow AW, Kandala A, Chow JM, Gambetta JM (2019) Supervised learning with quantum-enhanced feature spaces. Nature. 567(7747):209–212. https://doi.org/10.1038/s41586-019-0980-2

Huang Y, Li Q, Hou X, Wu R, Yung M-H, Bayat A, Wang X (2022) Robust resource-efficient quantum variational ansatz through an evolutionary algorithm. Phys Rev A 105:052414. https://doi.org/10.1103/PhysRevA.105.052414

IBM quantum. https://quantum-computing.ibm.com/

Jia Z-A, Yi B, Zhai R, Wu Y-C, Guo G-C, Guo G-P (2019) Quantum neural network states: A brief review of methods and applications. Adv Quantum Technol 2(7–8):1800077

Kamaka BK (2020) Quantum transpiler optimization: On the development, implementation, and use of a Quantum Research testbed. https://scholar.afit.edu/etd/3590/

Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow JM, Gambetta JM (2017) Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. Nature 549(7671):242–246

Kim J, Oz Y (2022) Quantum energy landscape and circuit optimization. Phys Rev A 106:052424. https://doi.org/10.1103/PhysRevA.106.052424

Li, G., Ding, Y., Xie, Y.: Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices. arXiv e-prints, 1809–02573 (2018) https://doi.org/10.48550/arXiv.1809.02573, arXiv:1809.02573 [cs.ET]

Lucas A (2014) Ising formulations of many np problems. Front Phys 2. https://doi.org/10.3389/fphy.2014.00005

Marinescu R, Dechter R (2006) And/or branch-and-bound search for pure 0/1 integer linear programming problems. In: Beck JC, Smith BM (eds) Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Springer, Berlin, pp 152–166

Markowitz H (1952) Portfolio selection. J Finance 7(1): 77–91. Accessed 2022-11-25

McClean JR, Boixo S, Smelyanskiy VN, Babbush R, Neven H (2018) Barren plateaus in quantum neural network training landscapes. Nat Commun 9(1):4812. https://doi.org/10.1038/s41467-018-07090-4

Miki T, Okita R, Shimada M, Tsukayama D, Shirakashi J-i (2022) Variational ansatz preparation to avoid cnot-gates on noisy quantum devices for combinatorial optimizations. AIP Adv 12(3):035247. https://doi.org/10.1063/5.0077706

Motta M, Rice JE (2022) Emerging quantum computing algorithms for quantum chemistry. WIREs Comput Mol Sci 12(3):1580. https://doi.org/10.1002/wcms.1580

Murali P, Baker JM, Javadi-Abhari A, Chong FT, Martonosi M (2019) Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers. In: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. ASPLOS 19. ACM, ??? . https://doi.org/10.1145/3297858.3304075

Niu S-f, Wang G-x, Sun X-l (2008) A branch-and-bound algorithm for discrete multi-factor portfolio optimization model. J Shanghai Univ 12:26–30. https://doi.org/10.1007/s11741-008-0105-3

Optimizers–IBM Quantum Documentation — qiskit.org. https://qiskit.org/documentation/stubs/qiskit.algorithms.optimizers.html. [Accessed 15-01-2024]

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E

(2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830

Pelofske E, Brtschi A, Eidenbenz S (2022) Quantum volume in practice: What users can expect from nisq devices. IEEE Trans Quantum Eng 3:1–19. https://doi.org/10.1109/TQE.2022.3184764

Peruzzo A, McClean J, Shadbolt P, Yung M-H, Zhou X-Q, Love PJ, Aspuru-Guzik A, O'Brien JL (2014) A variational eigenvalue solver on a photonic quantum processor. Nat Commun 5(1):4213. https://doi.org/10.1038/ncomms5213

Portfolio Optimization, Qiskit 0.26.2 documentation — qiskit.org. https://qiskit.org/documentation/stable/0.26/tutorials/finance/01_portfolio_optimization.html. [Accessed 15-01-2024]

Powell MJD (1994) In: Gomez S, Hennart J-P (eds.) A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation, pp. 51–67. Springer, Dordrecht. https://doi.org/10.1007/978-94-015-8330-5_4

Quantum computing use cases are getting real-what you need to know — mckinsey.com. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/quantum-computing-use-cases-are-getting-real-what-you-need-to-know. [Accessed 09-Mar-2023]

Saib W, Wallden P, Akhalwaya I (2021) The effect of noise on the performance of variational algorithms for quantum chemistry. In: 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 42–53. https://doi.org/10.1109/QCE52317.2021.00020

Slussarenko S, Pryde GJ (2019) Photonic quantum information processing: A concise review. Applied Phys Rev 6(4):041303. https://doi.org/10.1063/1.5115814

Sokolov IO, Barkoutsos PK, Ollitrault PJ, Greenberg D, Rice J, Pistoia M, Tavernelli I (2020) Quantum orbital-optimized unitary coupled cluster methods in the strongly correlated regime: Can quantum algorithms outperform their classical equivalents? J Chem Phys 152(12):124107

Tilly J, Chen H, Cao S, Picozzi D, Setia K, Li Y, Grant E, Wossnig L, Rungger I, Booth GH, Tennyson J (2022) The variational quantum eigensolver: A review of methods and best practices. Phys Rep **986**, 1–128. https://doi.org/10.1016/j.physrep.2022.08.003 . The Variational Quantum Eigensolver: a review of methods and best practices

Transpiler – IBM Quantum Documentation — docs.quantum.ibm.com. https://docs.quantum.ibm.com/api/qiskit/transpiler. [Accessed 15-01-2024]

Transpiler (qiskit.transpiler). https://qiskit.org/documentation/apidoc/transpiler.html

Tuysuz C, Clemente G, Crippa A, Hartung T, Kuhn S, Jansen K (2022) Classical Splitting of Parametrized Quantum Circuits. arXiv e-prints, 2206–09641. https://doi.org/10.48550/arXiv.2206.09641, arXiv:2206.09641 [quant-ph]

Wang S, Fontana E, Cerezo M, Sharma K, Sone A, Cincio L, Coles PJ (2021) Noise-induced barren plateaus in variational quantum algorithms. Nature Communications. 12(1):6961. https://doi.org/10.1038/s41467-021-27045-6

Wierichs, D., Izaac, J., Wang, C., Lin, C.Y.-Y.: General parameter-shift rules for quantum gradients. Quantum 6:677 (2022) https://doi.org/10.22331/q-2022-03-30-677

Yahoo Finanza - Mercato azionario in tempo reale, quotazioni e notizie di economia e finanza — it.finance.yahoo.com. https://it.finance.yahoo.com/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAIHdcyeGT4HED4q_5ThVGfe8xfcUwEGfx2gqSvUGHTaH-eGpoUdnNhin27d1jA6rGEe7tq2HLkwDdzJN7rh8yAsN4V07R4Suk7Jv91ApJ5ksWOqS1mTGP_8bxMwYpseCrAjhkJkGqCItNVrKgvm2JjfGTIa8MbAwEQ7fZDzHlkFe. [Accessed 15-01-2024]

Younis E, Iancu C (2022) Quantum Circuit Optimization and Transpilation via Parameterized Circuit Instantiation. arXiv e-prints, 2206–07885. https://doi.org/10.48550/arXiv.2206.07885. [quant-ph]

Zhang S-X, Hsieh C-Y, Zhang S, Yao H (2021) Neural predictor based quantum architecture search. arXiv:2103.06524

Zhao, R., Wang, S.: A review of Quantum Neural Networks: Methods, Models, Dilemma. arXiv e-prints, 2109–01840 (2021) https://doi.org/10.48550/arXiv.2109.01840, arXiv:2109.01840 [cs.ET]

Zhao L, Goings J, Shin K, Kyoung W, Fuks JI, Kevin Rhee J-K, Rhee YM, Wright K, Nguyen J, Kim J, Johri S (2023) Orbital-optimized pair-correlated electron simulations on trapped-ion quantum computers. Npj Quantum Inf 9(1):60. https://doi.org/10.1038/s41534-023-00730-8

Zhou L, Wang S-T, Choi S, Pichler H, Lukin MD (2020) Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. Phys. Rev. X. 10:021067. https://doi.org/10.1103/PhysRevX.10.021067