Quantum Architecture Search: A Survey

Darya Martyniuk
Fraunhofer FOKUS
Berlin, Germany
darya.martyniuk@fokus.fraunhofer.de

Johannes Jung
Freie Universitat Berlin
Fraunhofer FOKUS
Berlin, Germany
johannes.jung@fokus.fraunhofer.de

Adrian Paschke
Freie Universitat Berlin
Fraunhofer FOKUS
Berlin, Germany
adrian.paschke@fokus.fraunhofer.de

***Abstract*—Quantum computing has made significant progress in recent years, attracting immense interest not only in research laboratories but also in various industries. However, the application of quantum computing to solve real-world problems is still hampered by a number of challenges, including hardware limitations and a relatively under-explored landscape of quantum algorithms, especially when compared to the extensive development of classical computing. The design of quantum circuits, in particular parameterized quantum circuits (PQCs), which contain learnable parameters optimized by classical methods, is a non-trivial and time-consuming task requiring expert knowledge. As a result, research on the automated generation of PQCs, known as quantum architecture search (QAS), has gained considerable interest. QAS focuses on the use of machine learning and optimization-driven techniques to generate PQCs tailored to specific problems and characteristics of quantum hardware. In this paper, we provide an overview of QAS methods by examining relevant research studies in the field. We discuss main challenges in designing and performing an automated search for an optimal PQC, and survey ways to address them to ease future research.**

***Index Terms*—Quantum architecture search, Quantum neural architecture circuit search, Automatic circuit generation, Variable ansatz, Quantum circuit structure search, QAS, AutoML**

I. Introduction

Advancing quantum computing involves developing and improving corresponding hardware and software systems. In recent years, research endeavors have led to rapid and impressive progress both in the physical realization of quantum computing concepts and in the development of software tools. However, there are still several serious challenges for the research community to overcome before quantum systems can be applied to real-world use cases. Machine learning (ML) and optimization algorithms can be used to open up the potential of hardware devices and expand the possibilities for programming these devices to effectively tackle complex problems. When addressing a particular task with quantum computing, the automation of algorithm design and its execution, including compilation and selection of a suitable device, are promising research directions. Quantum architecture search (QAS) [1] represents a variety of techniques tailored to automate the process of finding an optimal parametrized quantum circuit (PQC).

This work was partially funded by the BMWK project EniQmA (01MQ22007A) and and Einstein Research Unit on Quantum Digital Transformation (ERU-2020-607). *(Darya Martyniuk and Johannes Jung are co-first authors.)*

A PQC is a crucial component of variational quantum algorithms (VQAs), which have gained great interest in the community and have been successfully applied to various domains, such as chemical simulation [2], combinatorial optimization [3], and ML [4], [5]. It depends on learnable parameters $\vec{\theta}$, representing angles of quantum rotation gates. During the iterative execution of a VQA, a classical optimization algorithm adjusts $\vec{\theta}$ in order to minimize the underlying cost function [6]. In PQCs, a subroutine that consist of a sequential application of gates to specific qubits and thus dictates how quantum computation is performed, is called an *ansatz*. Despite the intuitive principle of PQCs, the manual design of a beneficial circuit ansatz is non-trivial, since the criteria for an optimal ansatz in a particular scenario is still an active research area. Moreover, PQCs may encounter trainability issues, such as barren plateaus [7]. To assists developers in creating powerful PQCs, several manually designed ansatz patterns have been presented in the literature, e.g., unitary coupled cluster (UCC) and hardware-efficient ansatz (HEA) [8]. Although the ansatz patterns are useful, they usually need to be tuned in terms of hyperparameters, which is a time-consuming process with no guarantee of success in fitting the particular scenario. Moreover, quantum noise and hardware constraints further impede the performance of ansatz patterns. To overcome these limitations, QAS methods aim to automatically find the optimal structure and the set of parameters for a PQC, tailored to both the underlying problem and the quantum hardware.

In the recent years, several promising techniques for the automated generation of PQCs have been presented. In this paper, we survey QAS methods and its open challenges.

We summarize **our contributions** as follows:

- We outline the relation of QAS to other fields.
- We provide a structured introduction in QAS methods.
- We discuss techniques to increase efficiency of QAS.
- We introduce ideas for further research direction.

This work includes six sections. Section II outlines a brief history of QAS and its connection to other fields. In Section III, components of QAS are introduced. Different search strategies are discussed in Section IV. For each strategy, a short overview is given, followed by summaries of selected studies. In Section V, we review techniques the improve the efficiency of QAS. Possible directions for future research are summarized in Section VI.

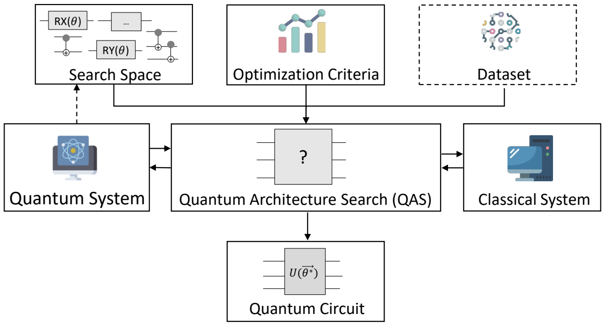

Fig. 1. General overview of QAS: Given a search space (which can be constrained by hardware characteristics), performance criteria, and, in some applications, a dataset, QAS aims to automatically find an optimal PQC with parameters $\vec{\theta}^*$ that maximizes the performance. Quantum and classical systems can be accessed to perform the evaluation of candidate circuits.

II. QAS Research and Relations to Other Fields

Initial research into the automatic design of quantum circuits dates back to the late 1990s and early 2000s. In early work, researchers used evolutionary algorithms to discover alternative architectures for quantum circuits that are simpler than those constructed manually [9], [10], more efficient than classical [11], and designed without the need for in-depth knowledge of quantum physics [12]. With the growing accessibility of quantum hardware, simulation environments, and software tools, the interest on the automatic generation of quantum circuits for specific problems has been increased. Since then, a wide range of ML techniques such as deep reinforcement learning [13]–[16], differentiable algorithms [1], [17]–[20], and Bayesian optimization [21]–[23] have been used to generate or optimize quantum circuits in various application areas, e.g., chemistry [24]–[30], ML [18], [31], [32], and optimization [15], [33]. The motivation driving the research remains in encoding problems into quantum systems without requiring extensive expertise in quantum computing, designing efficient and noise-resilient circuits in light of limitations of noisy intermediate-scale quantum (NISQ) hardware, and overcoming algorithmic-specific issues, e.g., trainability issues. It should be noted that over the years, the automated design of quantum circuits has been termed in the scientific literature also as quantum circuit design search [9], [10], [22], quantum circuit search [34], [35], ansatz architecture search [36], adaptive variational quantum algorithms [37], quantum circuit learning [38], [39], quantum neural architecture search (QNAS) [21], and hybrid optimization [40]. We used these notations to identify relevant studies for this survey. However, some of the publications cover the generation of quantum circuits in general rather than PQCs. Therefore, although these studies have influenced existing QAS methods, they are outside the scope of this survey.

Many QAS techniques introduced in recent years have been strongly inspired by *neural architecture search (NAS)*, which is the process of automated engineering of neural network architectures for a given task. NAS methods have already been successfully applied to a variety of tasks, such as large-scale image classification [41], segmentation [42], and text classification [43]. NAS aligns closely with *hyperparameter optimization (HPO)*, which aims to automate the search for optimal hyperparameters of a ML model, i.e., parameters used for the setup of the model or the optimizer (e.g., learning rate, type of optimizer) [44]. NAS and HPO methods can be considered as subfields of *automated machine learning (AutoML)*. AutoML intends to automate the entire pipeline of a ML model including data preparation and processing, feature engineering, algorithm and architecture selection (e.g., with NAS), and HPO. In contrast to related classical fields, QAS extends the idea of architecture search beyond (Q)ML models because of the variety of VQA applications. Moreover, due to the current high impact of the concrete hardware on the success of a circuit execution, a lot of QAS studies take into account hardware characteristics developing hardware-aware QAS methods [8], [25], [33], [34]. The resilience of QAS solutions to noise is another crucial property in the current NISQ-era. It has been shown that the presence of noise causes the estimated values of the cost function for given parameters to differ from those in a noiseless environment [45]. Thus, a number of QAS approaches [34], [39], [45] define noise resilience as an important property of candidate circuits.

Several works survey techniques and breakthroughs in automated generation of quantum circuits. Reference [46] provide a survey on evolving quantum algorithms using genetic programming. However, as VQAs had not yet been introduced at the time the work was created, the automated generation of PQCs is not taken into account here. A brief overview of QAS techniques with emphasis on search strategies is provided in [47]. Some of QAS methodologies are outlined in [48] in the context of a broader review of ansatz designing techniques. However, given the active research effort on this topic and novel ideas offered in the recent years, there is a need of a comprehensive overview of QAS methods and challenges.

III. Background

In this work, we define *QAS* as follows. Given a task to be solved, QAS aims to automatically design a PQC optimized against specified performance criteria. Besides to the *task* that specifies the search objective, the process of QAS can incorporate input *data*, e.g., in case of quantum ML applications, or additional *constraints*, e.g., on properties of quantum hardware such as type of supported gates, number of qubits, or qubit connectivity, as illustrated on Fig. 1. QAS methods can be delineated along four dimensions depicted on Fig. 2. Note that this categorization is heavily inspired by NAS-related surveys [49], [50].

Search space specifies the set of potential architectures that can be discovered. Due to the large search space, it is typical for QAS studies to constrain it, e.g., by manually constructing the set of gates available for the circuit design. The constraints can be derived from the authors' experience or task's requirements. Alternatively, the constrained gate set may be derived from hardware properties, such as the native gate set supported by a particular quantum machine.