Patterns for Quantum Circuit Cutting

Marvin Bechtold, Johanna Barzen, Martin Beisel, Frank Leymann, and Benjamin Weder

Institute of Architecture of Application Systems
University of Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany
http://www.iaas.uni-stuttgart.de

in: Proceedings of the 30th Conference on Pattern Languages of Programs (PLoP '23). See also BibTeX entry below.

BibTeX:

```
@inproceedings{Bechtold2023_CuttingPatterns,
    title = {{Patterns for Quantum Circuit Cutting}},
    author = {Bechtold, Marvin and Barzen, Johanna and Beisel, Martin and Leymann, Frank
              and Weder, Benjamin},
    year = 2023,
    month = oct,
    booktitle = {Proceedings of the 30\textsuperscript{th} Conference on
                 Pattern Languages of Programs (PLoP ’23)},
    publisher = {The Hillside Group},
    note = {in press}
}
```

Patterns for Quantum Circuit Cutting

Marvin Bechtold, IAAS, University of Stuttgart
Johanna Barzen, IAAS, University of Stuttgart
Martin Beisel, IAAS, University of Stuttgart
Frank Leymann, IAAS, University of Stuttgart
Benjamin Weder, IAAS, University of Stuttgart

Quantum computing holds the potential to deliver faster and more precise solutions to complex problems that remain intractable for classical computers. However, the limited number of qubits and the high error rates of current quantum devices restrict the size of computations that can be successfully performed. An approach to address this is quantum circuit cutting, which divides a quantum computation into multiple smaller parts executable on available quantum devices and classically combines their results to obtain the original computation's outcome. Even when quantum devices mature, the importance of circuit cutting may even increase because of the addition of quantum devices with limited numbers of qubits to the existing computing infrastructure. However, there is a lack of comprehensive surveys comparing current circuit cutting techniques, let alone providing abstract guidance for quantum software engineers in applying them. Moreover, to facilitate collaboration, quantum software engineers need a common understanding of circuit cutting. In this work, we introduce three patterns focusing on quantum circuit cutting, which describe proven solution strategies as a first step towards providing abstract guidance and fostering a common understanding in this domain. These patterns are integrated into an existing quantum computing pattern language, thereby supporting the comprehension and application of quantum circuit cutting for quantum software engineers and promoting its practical implementation.

Categories and Subject Descriptors: C.1.3 [**Computer systems organization**]: Architectures; Other architectures—*Quantum computing*; D.2.11 [**Software Engineering**]: Software Architectures—*Patterns*

General Terms: Documentation

Additional Key Words and Phrases: NISQ, Circuit Knitting, Quantum Circuit Decomposition, Distributed Quantum Computing

ACM Reference Format:

Bechtold, M. Barzen, J. Beisel, M. Leymann, F. and Weder, B. 2023. Patterns for Quantum Circuit Cutting. HILLSIDE Proc. of Conf. on Pattern Lang. of Prog. 30 (October 2023), 12 pages.

1. INTRODUCTION

Recent advancements in constructing quantum devices pave the way for realizing their potential for solving problems intractable for classical high-performance computers [Preskill 2018; Cao et al. 2018].Their potential stems from the use of quantum bits (qubits), enabling quantum devices to leverage unique quantum mechanical phenomena such as superposition and entanglement [Nielsen and Chuang 2009]. Quantum computations can be described using quantum circuits, consisting of gates that represent quantum operations manipulating qubits, wires that connect gates while representing the qubits themselves, and measurements that retrieve classical information

Author's address: M. Bechtold, Universitätsstraße 38, 70569 Stuttgart, Germany; email: bechtold@iaas.uni-stuttgart.de; J. Barzen, Universitätsstraße 38, 70569 Stuttgart, Germany; email: barzen@iaas.uni-stuttgart.de; M. Beisel, Universitätsstraße 38, 70569 Stuttgart, Germany; email: beisel@iaas.uni-stuttgart.de; F. Leymann, Universitätsstraße 38, 70569 Stuttgart, Germany; email: leymann@iaas.uni-stuttgart.de; B. Weder, Universitätsstraße 38, 70569 Stuttgart, Germany; email: weder@iaas.uni-stuttgart.de

from the qubits. Executing a quantum circuit requires a quantum device providing the necessary number of qubits. However, today's devices, known as *Noisy Intermediate-Scale Quantum (NISQ)* devices, are limited in their ability to successfully execute quantum circuits due to their restricted capabilities, including a limited number of qubits and high error rates stemming from imperfect gate implementations, flawed measurements, and short decoherence times [Preskill 2018; Leymann and Barzen 2020].

One approach to circumvent these issues is quantum circuit cutting [Peng et al. 2019; Mitarai and Fujii 2021a]. Quantum circuit cutting divides a quantum circuit into several smaller circuits with fewer qubits and gates. By executing these smaller circuits on quantum devices and processing their results classically, the outcome of the original, larger circuit can be determined. While these smaller circuits primarily tackle the challenge of limited qubits, they can also reduce the impact of errors on the result [Leymann and Barzen 2020]. Even when quantum devices mature, the significance of circuit cutting may actually rise due to the incorporation of quantum devices with limited qubit numbers to the existing computing infrastructure [Furutanpey et al. 2023]. Cutting a quantum circuit can be accomplished using different techniques, namely cutting wires or cutting gates [Brenner et al. 2023].

Given the importance of circuit cutting, it is essential to bring this technique into quantum applications, which are typically developed within interdisciplinary teams of quantum software engineers, including physicists, mathematicians, and computer scientists. However, each development team member brings a distinct perspective and understanding of quantum applications. For example, a physicist understands which quantum mechanical phenomena must be considered when cutting a circuit, whereas a computer scientist enforces software engineering principles for quantum applications. These differences in focus and the lack of a common understanding often lead to misunderstandings and comprehension errors, hindering the development process. Hence, establishing a common understanding of the core concepts of circuit cutting and their implications on applications becomes indispensable. Achieving this common foundation fosters successful collaboration, where differences are not only overcome but also leveraged to advantage.

To address this, a well-established approach for documenting proven solutions to recurring problems within a specific context are patterns [Alexander et al. 1977]. Patterns ease the understanding of these solutions by presenting them in a well-structured and technology-independent manner. Particularly in an interdisciplinary and intricate domain like quantum computing, patterns serve as an effective tool for explaining the mechanism and rationale of a solution [Meszaros and Doble 1997]. Furthermore, patterns are interconnected in a pattern language, which facilitates the combination of related patterns and alleviates the understanding of similar problems and their respective solutions. In the domain of quantum computing, a pattern language has been introduced [Leymann 2019] and continuously expanded, e.g. [Weigold et al. 2021b; Beisel et al. 2022; Georg et al. 2023]. However, despite the growing importance of circuit cutting, the existing pattern language has not yet incorporated patterns for quantum circuit cutting. Therefore, we expand the quantum computing pattern language in this work with three new patterns that revolve around circuit cutting, with the goal of making this knowledge more accessible to quantum software engineers and facilitating a common understanding among them.

2. FUNDAMENTALS

This section introduces the pattern format and briefly describes the pattern authoring method. Afterward, the quantum computing fundamentals relevant to the quantum circuit cutting patterns are presented.

2.1 Pattern Format and Authoring Method

The pattern format is based on earlier research on quantum computing patterns [Leymann 2019; Weigold et al. 2021b; Weigold et al. 2021c; Beisel et al. 2022; Georg et al. 2023] and follows best practices in pattern research [Gamma et al. 1994]. Each pattern is identified by a unique *name*, which encapsulates its gist, and a mnemonic *icon*. The *problem* addressed by the pattern is concisely posed as a question, followed by a comprehensive explanation of the *context*, defining the circumstances under which the problem typically emerges. The *forces* describe the requirements or constraints restricting the available solutions. Subsequently, a *solution* addressing and reconciling all identified forces is presented, accompanied by an illustrative sketch. Afterward, the *result* section discusses the state achieved from applying the solution, along with its resulting positive and negative consequences. Lastly, the *related patterns* section outlines connections with other patterns in the language, while the *known uses* enumerates instances where the pattern has been applied.

To identify the circuit cutting patterns, we investigated state-of-the-art techniques in the scientific literature, assessing their theoretical descriptions, associated implementations, and implementations from well-established libraries such as Pennylane [Xanadu 2023] and Qiskit [IBM 2023b]. As a result, we assembled and analyzed recurring solution approaches that are ultimately compiled into the quantum circuit cutting patterns.

2.2 Quantum Computing

Quantum computing leverages quantum mechanical phenomena to perform calculations, utilizing qubits as its fundamental information units. Unlike classical bits, which are 0 or 1, qubits exist in a superposition of states $|0\rangle := (1,0)^T$ and $|1\rangle := (0,1)^T$, forming linear combinations of these states. Additionally, qubits can be entangled, i.e., the state of one qubit cannot be described independently of the others. This unique quantum phenomenon is essential for achieving exponential speedups compared to classical computations [Jozsa and Linden 2003]. Furthermore, qubits are manipulated through quantum operations that alter their states while maintaining their inherent structure. These operations, typically represented as unitary matrices, enable the creation and manipulation of superpositions and entanglement. Finally, the measurement of a qubit extracts classical information, i.e., a classical bit, and is defined by a Hermitian operator called an observable. Its eigenvectors constitute a set of orthogonal quantum states, the measurement basis. Upon measuring, the quantum system probabilistically collapses into one of the observable's eigenvectors, with the corresponding eigenvalue indicating the result. The probability of each outcome is determined by the overlap between the original quantum state and the states in the measurement basis. An important measurement concept is the expectation value that represents the average result from repeated measurements on identical quantum states. To approximate the expectation value in practice, a quantum computation is executed multiple times. Each of these executions, referred to as a *shot*, produces a bitstring as output. Subsequently, classical post-processing of the bitstrings from these runs is used to compute the observable's expectation value.

Quantum computations are often schematically represented as quantum circuits comprising wires, gates, and measurements. Gates describe the transformation of qubits through unitary quantum operations and are categorized as single-qubit or multi-qubit gates. A wire represents a qubit evolution, from left to right, starting with its initial state and typically ending with its measurement. Throughout the circuit, the wire connects gates, describing the manipulation of the qubit over time. The size of a quantum circuit is defined by its width, indicating the number of qubits involved, and its depth, referring to the number of gate layers applied. To successfully execute a quantum circuit, a device must provide sufficient qubits for its width and maintain an error rate compatible with its depth, as deeper circuits risk accumulating too many errors and experiencing quantum state decay [Leymann and Barzen 2020].

3. PATTERNS FOR QUANTUM CIRCUIT CUTTING

The quantum computing patterns provide solutions to effectively address various recurring problems that quantum software engineers encounter during the design, implementation, and execution of quantum algorithms, e.g., state initialization, circuit execution, and error handling [Leymann 2019; Beisel et al. 2022]. Before introducing the new patterns in detail, we briefly outline the existing quantum computing pattern language and explain how our contribution extends it.

Figure 1 offers a comprehensive overview of the existing pattern language, with patterns categorized by the application area they address. Since most quantum applications are hybrid, involving both classical and quantum tasks, their computation must be divided between quantum and classical resources [Leymann and Barzen 2020]. To this end, the *Program Flow* patterns aid with task allocation between these distinct resources [Weigold et al. 2021c]. These patterns feature the VARIATIONAL QUANTUM ALGORITHM (VQA), which uses a classical optimizer to iteratively train a quantum circuit, a common approach for addressing optimization and simulation problems. Prominent VQAs include the VARIATIONAL QUANTUM EIGENSOLVER (VQE) and QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM (QAOA), which are defined as patterns. To execute these hybrid quantum algorithms, current quantum devices are provided via the cloud through interfaces with different execution semantics designed to address specific application requirements. The *Execution* patterns aim to facilitate the selection of a suitable quantum service offering meeting specific execution requirements [Georg et al. 2023]. For instance, the PRIORITIZED EXECUTION pattern aids in efficiently executing multiple quantum circuits sequentially, while the ORCHESTRATED EXECUTION assists in managing control and data flow for hybrid quantum applications comprising both quantum and classical components.

To incorporate classical data, e.g., numerical data, a hybrid algorithm must preprocess and encode it. The *Initialization & Data Encodings* patterns assist in encoding classical data for quantum computers. The STATE PREPARATION pattern provides a broad overview, which is refined by more specific approaches, such as the AMPLITUDE ENCODING and SCHMIDT DECOMPOSITION patterns [Weigold et al. 2021b]. The quantum states resulting from the state preparation can subsequently be manipulated using *Unitary Transformations* patterns. In the end, a measurement retrieves classical information from the quantum device. The *Measurement* category includes the POST-SELECTIVE MEASUREMENT pattern, which describes the selection of a subset of measurement outcomes based on a predetermined condition, deeming the measurement successful only when the quantum system collapses into one of the chosen outcomes. Since the whole execution and measurement process on NISQ devices is error-prone, the *Error Handling* patterns address this issue. For example, the GATE ERROR MITIGATION and READOUT ERROR MITIGATION patterns present strategies to reduce the impact of noise introduced during gate applications and measurements in an execution, thereby improving the accuracy of quantum computations.

In this work, we introduce new *Cutting* patterns, as shown in Figure 1, explaining established concepts for circumventing limitations of quantum devices, e.g., an insufficient number of qubits or high error rates. By decomposing the computation of a circuit into several smaller circuit executions, these patterns enable the utilization of quantum devices with fewer qubits. Moreover, the execution of these smaller circuits is more robust against errors. In contrast to *Error Handling* patterns, which seek to mitigate or correct occured errors, *Cutting* patterns can reduce errors by opting for circuits that are less prone to errors during execution. The CIRCUIT CUTTING pattern provides a comprehensive description of the overarching procedure, further refined by two distinct cutting techniques described by the GATE CUT and WIRE CUT patterns.

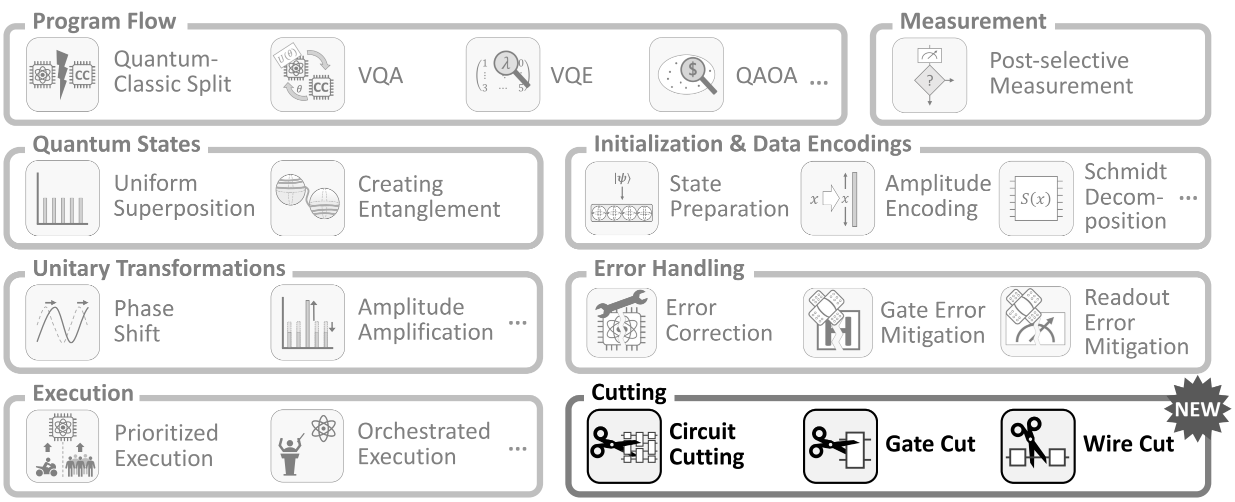
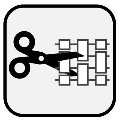


Fig. 1. Overview of the quantum computing pattern language including the new patterns based on [Georg et al. 2023].

3.1 Circuit Cutting Pattern



How to partition the computation of a quantum circuit into multiple smaller computations fitting the capabilities of available quantum devices?

Context: The maximum circuit width that a quantum device is able to execute is determined by its qubit count. As a result, devices with a limited number of qubits are restricted to executing circuits of small widths. Additionally, current NISQ devices face further limitations, such as high error rates and low coherence times, which impose restrictions on the number of gates that can be executed successfully in a circuit.

Forces: For smaller quantum devices, limited by their number of qubits and successfully executable gates, to contribute to the computation of a larger quantum circuit, it is necessary to divide the larger circuit into several smaller computations. Each of these smaller computations must require fewer resources, while collectively they must preserve the original circuit's result. However, several factors hinder the partitioning of a quantum circuit into multiple smaller computations. First, multi-qubit gates in a quantum circuit can create entanglement between the qubits on which they operate, resulting in the interdependence of their states. As a result, a quantum circuit with all qubits interconnected by multi-qubit gates cannot be partitioned into smaller, disconnected circuits that can be independently computed and then combined to produce the same result as the original circuit. Furthermore, removing multi-qubit gates to partition the circuit alters the computation's result. Moreover, the absence of shared entanglement or other means of quantum communication between quantum devices renders the distribution of a quantum circuit computation over multiple devices impossible.

Solution: Employ circuit cutting to divide a quantum circuit's computation into the computation of smaller circuits [Peng et al. 2019; Mitarai and Fujii 2021a] as shown in Figure 2. In the first step, the circuit is divided into multiple different variations of it, known as *subcircuits*. Each of these subcircuits can be partitioned along a cutting line, allowing the individual execution of its disconnected parts in the second step. In the final third step, these separate results can be combined for each subcircuit, and their outcomes are then merged to obtain the original circuit's output using classical post-processing.

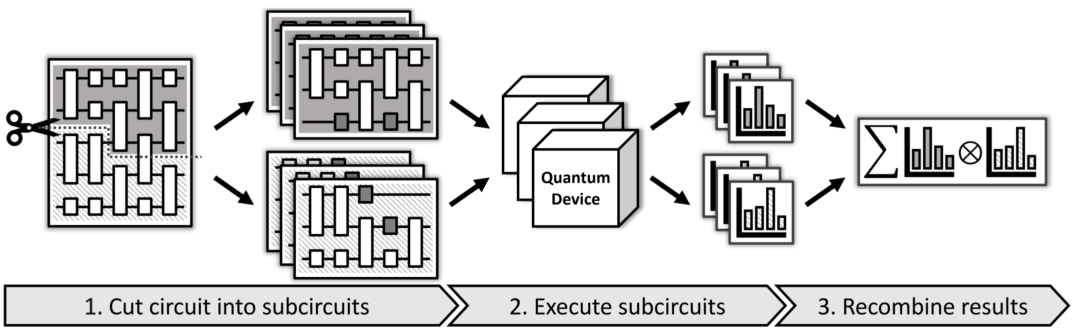
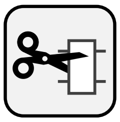


Fig. 2. Solution sketch for the CIRCUIT CUTTING pattern.

Result: Each individual subcircuit execution requires fewer qubits and gates, consequently reducing the hardware requirements of the quantum devices and enhancing the overall computation's robustness against errors and decoherence. Replacing the original circuit with a linear combination of subcircuits resulting from the cut enables replicating the effect of entanglement and, consequently, also the result. Each partitioned subcircuit can be executed successively on one or concurrently on multiple quantum devices to decrease the overall runtime of the computation [Bravyi et al. 2022]. However, compared to directly executing the original circuit, more shots are needed due to the multiplicative factor each cut introduces for estimating the expectation value with desired

statistical accuracy [Piveteau and Sutter 2023]. To minimize the resulting additional overhead and error propagation, the placement of cuts, e.g. WIRE CUTS and GATE CUTS, and the structure of the subcircuits should be optimized [Casciola et al. 2022]. Achieving this optimization can be automated through the utilization of mixed-integer programming techniques [Tang et al. 2021].
Related Patterns: Circuit cutting uses the GATE CUT and WIRE CUT patterns to decompose quantum circuits. The ORCHESTRATED EXECUTION can manage the control and data flow for circuit cutting, while PRIORITIZED EXECUTION can speed up subcircuit execution [Georg et al. 2023]. Circuit cutting can be applied in VQAS, e.g., VQE or QAOA [Weigold et al. 2021c]. It can also be used in combination with quantum error handling techniques, such as READOUT ERROR MITIGATION or GATE ERROR MITIGATION [Beisel et al. 2022].
Known Uses: Circuit cutting has been utilized in several works to extend the width of executable circuits beyond a device's qubit limit [Tang et al. 2021; Ying et al. 2023] and improve results [Ayral et al. 2021; Bechtold et al. 2023]. Implementations are integrated into Pennylane [Xanadu 2023] and Qiskit's Circuit Knitting Toolbox [IBM 2023a]. Additionally, a workflow modeling extension has been introduced for orchestrating circuit cutting [Beisel et al. 2023].

3.2 Gate Cut Pattern



How to partition a multi-qubit gate into independent gates while preserving the computation's result?

Context: A quantum circuit contains a multi-qubit gate operating between two partitions of qubits and cannot be expressed as a single product of local gates, i.e., gates that operate exclusively on qubits within a single partition. This gate can create entanglement between the partitions, and its execution requires direct physical connections between the involved qubits. The focus is on the expectation value across multiple shots, not single-shot experiments.
Forces: Limited connectivity between qubits of the multi-qubit gate requires compensating with SWAP operations that exchange the states of two qubits to establish required connections, thereby introducing additional errors [Leymann and Barzen 2020]. The complete absence of connectivity between the qubit partitions, e.g., multiple devices without quantum communication, renders the execution of gates between partitions impossible.
Solution: Apply a gate cut to decompose a circuit's multi-qubit gate into a weighted sum of subcircuits [Mitarai and Fujii 2021a]. As shown in Figure 3, in each subcircuit, the original multi-qubit gate U is replaced by local operations F_i^A and F_i^B, which act on two disjoint qubit partitions labeled A and B. The sum of the expectation values of these subcircuits, each weighted by real coefficient a_i, must replicate the expectation value of the original circuit with gate U. The local operations F_i^A and F_i^B can be either unitary transformations or projective measurements. Execute these subcircuits and combine their results using the real coefficients a_i.

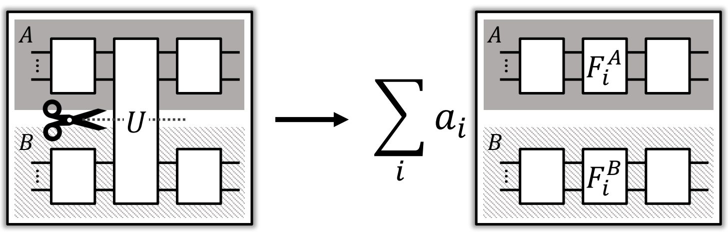
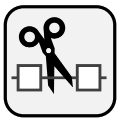


Fig. 3. Solution sketch for the GATE CUT pattern.

Result: The GATE CUT enables implementing a multi-qubit gate as a linear combination of local operations independent of the qubit connectivity. This eliminates the need for additional SWAP gates when applied on a single device and allows a multi-qubit gate across two quantum devices without requiring quantum communication.

However, replacing multi-qubit gates with local gates and projections through executing multiple subcircuits increases the computational overhead in terms of required shots to achieve the same statistical accuracy in the expectation value as the original circuit. To reduce computational overhead when cutting multiple gates, consider incorporating classical communication between the partitions [Piveteau and Sutter 2023]. Executing subcircuits in parallel on multiple devices reduces the overall computation runtime [Bravyi et al. 2022]. Moreover, this approach reproduces only the expectation value, making it unsuitable for single-shot experiments [Piveteau and Sutter 2023].
Related Patterns: The GATE CUT pattern is used in CIRCUIT CUTTING. PRIORITIZED EXECUTION can speed up subcircuit execution [Georg et al. 2023]. The POST-SELECTIVE MEASUREMENT pattern can handle local projective measurements [Weigold et al. 2021b]. Additionally, a GATE CUT can be employed in VQAS, such as VQE or QAOA, to reduce ansatz connectivity without creating disconnected subcircuits [Weigold et al. 2021c].
Known Uses: Gate cuts are used in various circuit cutting works, including two-qubit gate cutting [Mitarai and Fujii 2021b; Bechtold et al. 2023] and many-qubit gate cutting [Ufrecht et al. 2023]. Moreover, gate cuts are applied to reduce errors by minimizing the number of two-qubit gates [Yamamoto and Ohira 2023]. An implementation is provided in Qiskit's Circuit Knitting Toolbox [IBM 2023a].

3.3 Wire Cut Pattern



How to interrupt a wire in a quantum circuit classically such that no quantum information is transmitted while preserving the computation's result?

Context: In a quantum circuit, a wire transfers a non-trivial quantum state between two gates, that is, a state that exhibits superposition and has the potential of entanglement with other qubits. The precise state conveyed by the wire is unknown. Furthermore, the quantum state experiences decay over time. Moreover, the interest of the experiment lies in the expectation value over multiple shots rather than the outcome of a single-shot execution.
Forces: Given the non-trivial nature of the state, it cannot be measured and reinitialized based on the measurement outcome, as the qubit's full state is not revealed through the measurement process. Instead, the measurement collapses the qubit probabilistically to a state associated with the measurement basis, destroying its superposition and entanglement. Furthermore, since the state of the wire is unknown, it cannot be restored independently of the previous measurement either.
Solution: Apply a wire cut to decompose a circuit's wire into a weighted sum of subcircuits [Peng et al. 2019]. As shown in Figure 4, the subcircuits interrupt the wire with different measurements described by observables O_i and subsequent qubit initializations of the states $|\psi_i\rangle$. The sum of the expectation values of these subcircuits, each weighted by real coefficient a_i, must replicate the expectation value of the original circuit. To cut the wire, run the subcircuits and combine their results based on the coefficients a_i and the measurement outcomes.

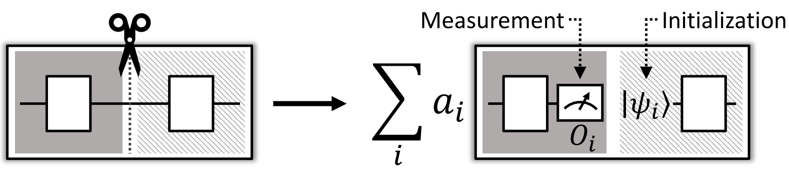


Fig. 4. Solution sketch for the WIRE CUT pattern.

Result: Applying a WIRE CUT interrupts the wire classically without disturbing the computation's expectation value. The behavior of superposition and entanglement is simulated by replacing the wire with a linear combination of subcircuits that perform a measurement and then reinitialize the qubit. However, the trade-off involves executing multiple subcircuits, which raises the computational overhead in terms of shots needed to maintain the same

statistical accuracy in the computed expectation value as the original circuit. To reduce this computational overhead, it is advantageous to incorporate classical communication in the cut, enabling each state preparation to depend on its immediate prior measurement outcome [Brenner et al. 2023; Pednault 2023; Harada et al. 2023]. This approach also lowers the cost of cutting multiple wires together compared to cutting each wire individually. Furthermore, leveraging multiple devices for parallel execution of subcircuits decreases the total computation runtime [Bravyi et al. 2022]. Additionally, employing maximum-likelihood fragment tomography can enhance subcircuit results and thereby improve the overall output quality [Perlin et al. 2021]. However, a wire cut only reproduces the result regarding the expectation value, and therefore, it is not suited for single-shot experiments [Brenner et al. 2023].
Related Patterns: The WIRE CUT pattern, used in CIRCUIT CUTTING, applies STATE PREPARATION for qubit initialization after measurement in the cut [Leymann 2019]. Additionally, it can be employed in VQAS, e.g, VQE or QAOA [Weigold et al. 2021c]. PRIORITIZED EXECUTION can speed up subcircuit execution [Georg et al. 2023].
Known Uses: Wire cutting is applied in several circuit cutting works [Tang et al. 2021; Lowe et al. 2023]. Moreover, their implementations are part of Pennylane [Xanadu 2023] and in Qiskit's Circuit Knitting Toolbox [IBM 2023a].

4. EXEMPLARY APPLICATION OF CIRCUIT CUTTING PATTERNS

This section presents an illustrative example showcasing the application of CIRCUIT CUTTING to reduce circuit width, employing the three introduced cutting patterns. Following the process outlined in the CIRCUIT CUTTING pattern, the initial step involves dividing the circuit into subcircuits, as illustrated in the center of Figure 5. The depicted five-qubit circuit on the left comprises one-qubit gates and two-qubit controlled-Z gates. A cutting line, annotated with the CIRCUIT CUTTING icon on the left, is chosen. It aims to minimize the number of cuts while dividing the circuit into approximately equal-sized parts. This entails a WIRE CUT on the third qubit and a GATE CUT separating a controlled-Z gate between the second and third qubits. Both the wire and gate are labeled with their respective cutting pattern icons. For a detailed comprehension, Figure 5 provides an example for a cut of the controlled-Z gate at the top [Peng et al. 2019] and for the wire at the bottom [Harada et al. 2023].

The GATE CUT of the controlled-Z gate uses $R_Z(\alpha)$ gates, performing rotations about the Z-axis by an angle α, and projections on the $|0\rangle$ and $|1\rangle$ states, represented as $(I+Z)/2$ and $(I-Z)/2$, respectively. Although these

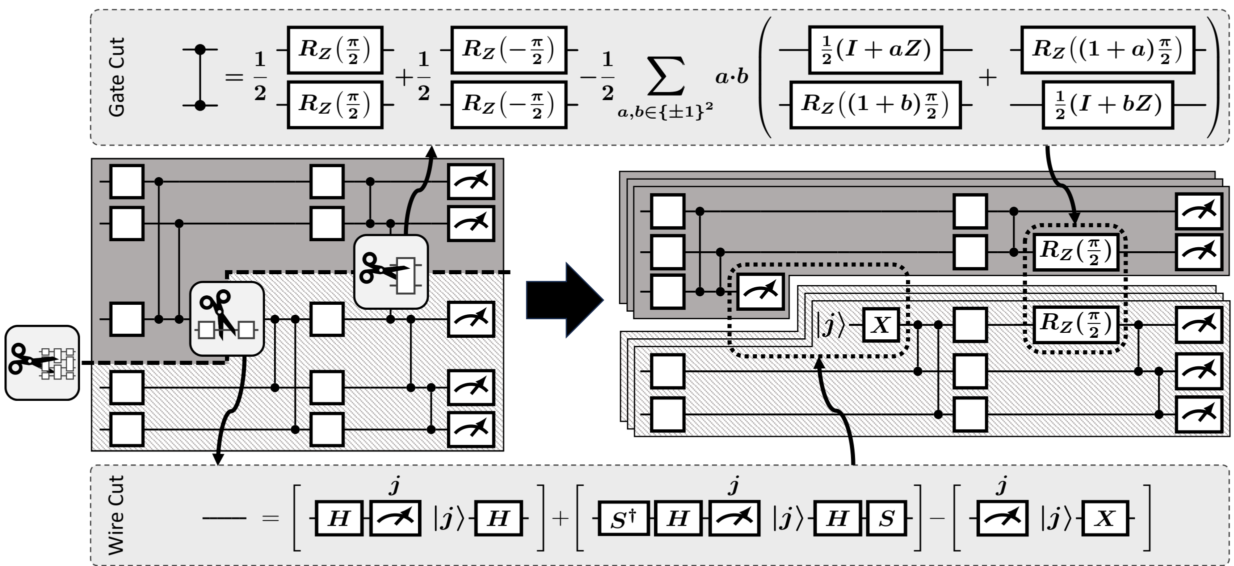


Fig. 5. Cutting a circuit into subcircuits by using a WIRE CUT and a GATE CUT.

projections are non-unitary and, therefore, not directly implementable as gates, they can be achieved through post-selective measurements [Mitarai and Fujii 2021a], as detailed in the associated pattern [Weigold et al. 2021a]. The employed WIRE CUT involves three different measurements based on preceding gates, and following qubit initializations are altered by subsequent gates. Specifically, the Hadamard gate H, the phase gate S, and the Pauli X gate are used in this process. Classical communication is utilized in this cut, as indicated by the measurement result j, which is used to initialize the following qubit state $|j\rangle$.

Leveraging these decompositions, the subsequent step involves generating subcircuits by replacing the wire with the circuit fragments from the wire cut and analogously replacing the controlled-Z gate with the circuit fragments from the gate cut. This is depicted on the right side of Figure 5, where the stacked circuits indicate the different subcircuits containing the replacements from the decompositions. As the subcircuits consist of two disconnected three-qubit circuits, they can be executed separately, aligning with the second step of the process outlined in the CIRCUIT CUTTING pattern. The utilization of classical communication by the wire cut leads to a dependency between the subcircuits. Hence, the upper subcircuits must be executed before the lower subcircuits, enabling the initialization of the wire in the lower subcircuits using the previously measured results.

As described by the final step of the CIRCUIT CUTTING pattern, the computed expectation values for each subcircuit, consisting of the product of its two disconnected parts, must be weighted with the product of the corresponding factors from the decomposition of the gate and the wire. In this example, all products of factors are either $\frac{1}{2}$ or $-\frac{1}{2}$. The sum of the weighted expectation values from the subcircuits reconstructs the expectation value of the original five-qubit circuit.

5. DISCUSSION

This section discusses circuit cutting in the context of upcoming, more advanced devices that are expected to have an increased qubit count and reduced error rates. However, scaling up the number of qubits in a single quantum device faces considerable technical challenges [Preskill 2018]. Consequently, qubits may remain scarce in the near term, with gradual improvements unable to satisfy the demand for larger circuits required to solve more complex problems. This scenario sustains the need for executing circuits beyond a single device's qubit capacity. As a result, the next stage of development is expected to involve modular quantum systems composed of multiple smaller devices, with circuit cutting serving as a potential enabling technology for their capabilities [Bravyi et al. 2022]. In addition, more mature devices with limited qubit numbers may be integrated at the edge in the existing computing infrastructure [Furutanpey et al. 2023], further highlighting the importance of circuit cutting. Moreover, applying error correction techniques will combine multiple physical qubits into a single error-corrected logical qubit, effectively reducing the number of available qubits from the newly added ones [Beisel et al. 2022]. Furthermore, recent research has demonstrated the usefulness of circuit cutting techniques even when sufficient qubits are available for direct execution on a single device [Ayral et al. 2021; Perlin et al. 2021; Bechtold et al. 2023].

In the long term, distributed quantum computation via quantum teleportation, which transfers a qubit's quantum state to a remote qubit without physical transmission, may challenge the need for circuit cutting [Cuomo et al. 2020]. However, this method necessitates shared entanglement between different quantum devices, which can only be established when these devices can exchange quantum information. Thus, while the future may present distributed quantum computation as an alternative to circuit cutting, circuit cutting will likely remain viable in environments where exchanging quantum information is infeasible.

Furthermore, circuit cutting techniques are expected to be incorporated by compiler and hardware providers, as circuit cutting is listed in IBM's quantum software stack [Bravyi et al. 2022], and initial implementations are already integrated into well-established quantum computing libraries such as Pennylane [Xanadu 2023] and Qiskit's Circuit Knitting Toolbox [IBM 2023a]. Consequently, the presented catalog of quantum circuit cutting patterns serves as a structured guide, promoting a comprehensive understanding of the topic and facilitating the use of circuit cutting. Furthermore, it can be expanded to include new patterns documenting alternative partitioning strategies for quantum computation, e.g., divide-and-conquer algorithms or the above-mentioned quantum teleportation.

6. RELATED WORK

In this work, we present patterns for quantum circuit cutting, extending the existing and continuously growing quantum computing pattern language [Leymann 2019; Weigold et al. 2021b; Weigold et al. 2021c; Beisel et al. 2022; Georg et al. 2023]. Although other publications define terms and summarize concepts in the quantum computing domain [Gilliam et al. 2019; Huang and Martonosi 2019; Perdrix 2007], they do not adhere to the pattern format introduced by Alexander et al. [1977]. Examples of pattern languages in information technology encompass object-oriented software design patterns software [Gamma et al. 1994], enterprise integration patterns [Hohpe and Woolf 2004], and cloud computing patterns [Fehling et al. 2014]. They contain patterns, such as the FORK-JOIN pattern in parallel computing [McCool et al. 2012] and the COMPOSED MESSAGE PROCESSOR of the enterprise integration patterns [Hohpe and Woolf 2004], that exhibit a similar processing flow to the CIRCUIT CUTTING pattern, where an entity is broken down into smaller entities, processed, and their results combined. To the best of our knowledge, no other patterns in the quantum computing domain have been published that specifically tackle quantum circuit cutting.

Existing research in the field of quantum circuit cutting primarily focuses on providing theoretical insights [Piveteau and Sutter 2023; Brenner et al. 2023; Marshall et al. 2023] and enhancing existing methods [Lowe et al. 2023; Harada et al. 2023; Pednault 2023; Ufrecht et al. 2023]. While newly introduced methods are compared with existing ones, there is a lack of comprehensive surveys comparing current techniques, let alone providing abstract guidance for users in applying circuit cutting. Furthermore, the detailed method descriptions in existing resources require a profound background in quantum computing, making them challenging and time-consuming for new entrants to comprehend, particularly when seeking a general overview of quantum circuit cutting techniques. In contrast, our work presents easily accessible, well-structured, and concise knowledge artifacts that facilitate a rapid comprehension of the subject. By providing such resources, we aim to foster a common understanding and facilitate educating new entrants in the field.

To further support the practical application of abstract patterns, *solution languages* offer concrete solutions for specific patterns, e.g., usable implementations [Falkenthal et al. 2014]. These concrete solution artifacts are linked to their respective patterns and associated with other solutions according to the relations within the pattern language [Falkenthal et al. 2015]. This approach enables the expansion of our patterns with practical solutions via solution languages, further increasing their value for users. Additionally, the *Pattern Atlas* visualizes connections between patterns and promotes interconnectivity across domains [Leymann and Barzen 2021]. It enables the creation of connections between pattern languages and *pattern views*, i.e., subsets of patterns and connections from different languages [Weigold et al. 2020]. Therefore, it enhances understanding of pattern relationships and fosters innovative solutions by combining patterns and connections from various languages.

7. SUMMARY AND OUTLOOK

This paper investigates quantum circuit cutting as an approach for decomposing computations of quantum circuits into smaller circuits more suitable for execution on current NISQ devices. By introducing three novel patterns, namely CIRCUIT CUTTING, GATE CUT, and WIRE CUT, to the existing quantum computing pattern language, we have made the knowledge of circuit cutting and involved cutting techniques more accessible to a broader audience.

For future work, we intend to integrate the quantum circuit cutting patterns in the Pattern Atlas [Leymann and Barzen 2021] on the PlanQK platform [PlanQK 2023], a platform for sharing knowledge about quantum computing. This integration fosters continuous evolution and re-evaluation of the pattern language, enabling the refinement of circuit cutting patterns based on community feedback and ensuring ongoing improvement. Additionally, we plan to develop a solution language facilitating the application of the quantum computing patterns and integrate it into PlanQK to enable users to contribute code snippets linked to corresponding patterns.

ACKNOWLEDGMENT

This work was partially funded by the BMWK projects *PlanQK* (01MK20005N), *EniQmA* (01MQ22007B), and *SeQuenC* (01MQ22009B).

REFERENCES

Christopher Alexander, Sara Ishikawa, and Murray Silverstein. 1977. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press.

Thomas Ayral, François-Marie Le Régent, Zain Saleem, Yuri Alexeev, and Martin Suchara. 2021. Quantum Divide and Compute: Exploring the Effect of Different Noise Sources. *SN Computer Science* 2, 3 (2021), 132. Issue 3.

Marvin Bechtold, Johanna Barzen, Frank Leymann, Alexander Mandl, Julian Obst, Felix Truger, and Benjamin Weder. 2023. Investigating the effect of circuit cutting in QAOA for the MaxCut problem on NISQ devices. *Quantum Science and Technology* 8, 4, Article 045022 (2023).

Martin Beisel, Johanna Barzen, Marvin Bechtold, Frank Leymann, Felix Truger, and Benjamin Weder. 2023. QuantME4VQA: Modeling and Executing Variational Quantum Algorithms Using Workflows. In *In Proceedings of the 13th International Conference on Cloud Computing and Services Science*. SciTePress, 306 – 315.

Martin Beisel, Johanna Barzen, Frank Leymann, Felix Truger, Benjamin Weder, and Vladimir Yussupov. 2022. Patterns for Quantum Error Handling. In *Proceedings of the 14th International Conference on Pervasive Patterns and Applications (PATTERNS 2022)*. Xpert Publishing Services (XPS), 22–30.

Sergey Bravyi, Oliver Dial, Jay M. Gambetta, Darío Gil, and Zaira Nazario. 2022. The future of quantum computing with superconducting qubits. *Journal of Applied Physics* 132, 16 (2022), 160902.

Lukas Brenner, Christophe Piveteau, and David Sutter. 2023. Optimal wire cutting with classical communication. (2023). arXiv:2302.03366.

Yudong Cao, Jhonathan Romero, and Alán Aspuru-Guzik. 2018. Potential of quantum computing for drug discovery. *IBM Journal of Research and Development* 62, 6 (2018), 6:1–6:20.

Nadir Casciola, Edoardo Giusto, Emanuele Dri, Daniel Oliveira, Paolo Rech, and Bartolomeo Montrucchio. 2022. Understanding the Impact of Cutting in Quantum Circuits Reliability to Transient Faults. In *2022 IEEE 28th International Symposium on On-Line Testing and Robust System Design (IOLTS)*. 1–7. ISSN: 1942-9401.

Daniele Cuomo, Marcello Caleffi, and Angela Sara Cacciapuoti. 2020. Towards a distributed quantum computing ecosystem. *IET Quantum Communication* 1, 1 (2020), 3–8.

Michael Falkenthal, Johanna Barzen, Uwe Breitenbücher, Christoph Fehling, and Frank Leymann. 2014. Efficient Pattern Application: Validating the Concept of Solution Implementations in Different Domains. *International Journal On Advances in Software* 7, 3&4 (2014), 710–726.

Michael Falkenthal, Johanna Barzen, Uwe Breitenbücher, Christoph Fehling, Frank Leymann, Aristotelis Hadjakos, and et al. 2015. Leveraging Pattern Application via Pattern Refinement. In *Proceedings of the International Conference on Pursuit of Pattern Languages for Societal Change (PURPLSOC 2015)*. epubli, 38–61.

Christoph Fehling, Frank Leymann, Ralph Retter, Walter Schupeck, and Peter Arbitter. 2014. *Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications*. Springer. 367 pages.

Alireza Furutanpey, Johanna Barzen, Marvin Bechtold, Schahram Dustdar, Frank Leymann, Philipp Raith, and Felix Truger. 2023. Architectural Vision for Quantum Computing in the Edge-Cloud Continuum. In *2023 IEEE International Conference on Quantum Software (QSW)*. IEEE, 88–103.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. 1994. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley.

Daniel Georg, Johanna Barzen, Martin Beisel, Frank Leymann, Julian Obst, Daniel Vietz, Benjamin Weder, and Vladimir Yussupov. 2023. Execution Patterns for Quantum Applications. In *Proceedings of the 18th International Conference on Software Technologies - ICSOFT*. SciTePress, 258–268.

Austin Gilliam, Charlene Venci, Sreraman Muralidharan, Vitaliy Dorum, Eric May, Rajesh Narasimhan, and Constantin Gonciulea. 2019. Foundational Patterns for Efficient Quantum Computing. (2019). arXiv:1907.11513.

Hiroyuki Harada, Kaito Wada, and Naoki Yamamoto. 2023. Optimal parallel wire cutting without ancilla qubits. (2023). arXiv:2303.07340.

Gregor Hohpe and Bobby Woolf. 2004. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley.

Yipeng Huang and Margaret Martonosi. 2019. Statistical assertions for validating patterns and finding bugs in quantum programs. In *Proceedings of the 46th International Symposium on Computer Architecture*. ACM.

IBM. 2023a. Qiskit - Circuit Knitting Toolbox. `https://qiskit-extensions.github.io/circuit-knitting-toolbox/`. (2023). [accessed: 2023.05.26].

IBM. 2023b. Qiskit - Documentation. `https://qiskit.org/documentation/`. (2023). [accessed: 2023.05.26].

Richard Jozsa and Noah Linden. 2003. On the role of entanglement in quantum-computational speed-up. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 459, 2036 (2003), 2011–2032.

Frank Leymann. 2019. Towards a Pattern Language for Quantum Algorithms. In *Quantum Technology and Optimization Problems (Lecture Notes in Computer Science (LNCS))*, Vol. 11413. Springer, 218–230.

Frank Leymann and Johanna Barzen. 2020. The bitter truth about gate-based quantum algorithms in the NISQ era. *Quantum Science and Technology* (2020), 1–28.

Frank Leymann and Johanna Barzen. 2021. *Pattern Atlas*. Springer, 67–76.

Angus Lowe, Matija Medvidović, Anthony Hayes, Lee J. O'Riordan, Thomas R. Bromley, Juan Miguel Arrazola, and Nathan Killoran. 2023. Fast quantum circuit cutting with randomized measurements. *Quantum* 7 (2023), 934.

Simon C. Marshall, Jordi Tura, and Vedran Dunjko. 2023. All this for one qubit? Bounds on local circuit cutting schemes. (2023). arXiv:2303.13422.

Michael McCool, James Reinders, and Arch Robison. 2012. *Structured Parallel Programming Patterns for Efficient Computation*. Elsevier.

Gerard Meszaros and Jim Doble. 1997. Pattern Languages of Program Design 3. Addison-Wesley, Chapter A Pattern Language for Pattern Writing, 529–574.

Kosuke Mitarai and Keisuke Fujii. 2021a. Constructing a virtual two-qubit gate by sampling single-qubit operations. *New Journal of Physics* 23, 2 (2021), 023021. Issue 2.

Kosuke Mitarai and Keisuke Fujii. 2021b. Overhead for simulating a non-local channel with local channels by quasiprobability sampling. 5 (2021), 388.

Michael A. Nielsen and Isaac L. Chuang. 2009. *Quantum Computation and Quantum Information*. Cambridge University Press.

Edwin Pednault. 2023. An alternative approach to optimal wire cutting without ancilla qubits. (2023). arXiv:2303.08287.

Tianyi Peng, Aram Harrow, Maris Ozols, and Xiaodi Wu. 2019. Simulating Large Quantum Circuits on a Small Quantum Computer. *Physical Review Letters* 125 (2019), 150504. Issue 15.

Simon Perdrix. 2007. Quantum Patterns and Types for Entanglement and Separability. *Electronic Notes in Theoretical Computer Science* 170 (2007), 125–138.

Michael A. Perlin, Zain H. Saleem, Martin Suchara, and James C. Osborn. 2021. Quantum Circuit Cutting with Maximum Likelihood Tomography. *npj Quantum Information* 7, 1 (2021).

Christophe Piveteau and David Sutter. 2023. Circuit knitting with classical communication. *IEEE Transactions on Information Theory* (2023).

PlanQK. 2023. PlanQK - Pattern Atlas. `https://patterns.platform.planqk.de/pattern-languages`. (2023). [accessed: 2023.05.26].

John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum 2, 79 (2018)* 2 (2018), 79.

Wei Tang, Teague Tomesh, Martin Suchara, Jeffrey Larson, and Margaret Martonosi. 2021. CutQC: using small Quantum computers for large Quantum circuit evaluations. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM, 473–486.

Christian Ufrecht, Maniraman Periyasamy, Sebastian Rietsch, Daniel D. Scherer, Axel Plinge, and Christopher Mutschler. 2023. Cutting multi-control quantum gates with ZX calculus. *Quantum* 7 (2023), 1147.

Manuela Weigold, Johanna Barzen, Uwe Breitenbücher, Michael Falkenthal, Frank Leymann, and Karoline Wild. 2020. Pattern Views: Concept and Tooling of Interconnected Pattern Languages. In *Proceedings of the 14th Symposium and Summer School on Service-Oriented Computing (SummerSOC 2020)*. Springer, 86–103.

Manuela Weigold, Johanna Barzen, Frank Leymann, and Marie Salm. 2021a. Encoding patterns for quantum algorithms. *IET QuantumCommunication* 2, 4 (2021), 141–152.

Manuela Weigold, Johanna Barzen, Frank Leymann, and Marie Salm. 2021b. Expanding Data Encoding Patterns For Quantum Algorithms. In *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 95–101.

Manuela Weigold, Johanna Barzen, Frank Leymann, and Daniel Vietz. 2021c. Patterns for Hybrid Quantum Algorithms. In *Proceedings of the 15th Symposium and Summer School on Service-Oriented Computing (SummerSOC 2021)*. Springer, 34–51.

Xanadu. 2023. Pennylane - Documentation. `https://docs.pennylane.ai/en/stable/code/qml.html`. (2023). [accessed: 2023.05.26].

Takahiro Yamamoto and Ryutaro Ohira. 2023. Error suppression by a virtual two-qubit gate. *Journal of Applied Physics* 133, 17 (2023).

Chong Ying, Bin Cheng, Youwei Zhao, He-Liang Huang, Yu-Ning Zhang, Ming Gong, and et al. 2023. Experimental Simulation of Larger Quantum Circuits with Fewer Superconducting Qubits. *Physical Review Letters* 130, 11 (2023), 110601.