A Review on Quantum Circuit Optimization using ZX-Calculus

Tobias Fischbach, Pierre Talbot, and Pascal Bouvry

Department of Computer Science, University of Luxembourg, Esch-sur-Alzette, Luxembourg

Quantum computing promises significant speed-ups for certain algorithms but the practical use of current noisy intermediate-scale quantum (NISQ)-era computers remains limited by resources constraints (e.g., noise, qubits, gates, and circuit depth). Quantum circuit optimization is a key mitigation strategy. In this context, ZX-calculus has emerged as an alternative framework that allows for semantics-preserving quantum circuit optimization.

We review ZX-based optimization of quantum circuits, categorizing them by optimization techniques, target metrics and intended quantum computing architecture. In addition, we outline critical challenges and future research directions, such as multi-objective optimization, scalable algorithms, and enhanced circuit extraction methods.

This survey is valuable for researchers in both combinatorial optimization and quantum computing. For researchers in combinatorial optimization, we provide the background to understand a new challenging combinatorial problem: ZX-based quantum circuit optimization. For researchers in quantum computing, we classify and explain existing circuit optimization techniques.

1 Introduction

Quantum computing belongs to the field of quantum information theory and uses quantum mechanical effects to process information. As a consequence, new applications emerge that are intractable for classical computers [38] (Section 3). Although quantum computing promises advances in drug discovery, material science, and climate modeling, its advantages are not universal across applications [43]. But as the quantum computing paradigm differs fundamentally from classical computing, quantum algorithms need to be carefully designed to exceed the performance of classical computers [64].

A *quantum circuit* [7] is the standard model to express quantum algorithms. Its basic building blocks are qubits and quantum gates. The *qubit* is the elementary unit of information [68]. A *Quantum gate* is an operator that acts on the state of qubits. We formally introduce quantum circuits in Section 3.

Quantum circuits provide a formal description of quantum algorithms, but their practical realization on existing hardware remains constrained. The current generation of quantum computers has limited practical use, as they are part of the *noisy intermediate scale quantum era (NISQ)* [64]. In particular, the duration in which a quantum device is stable and can reliably process information —called *coherence time*— is limited and restrict the available physical qubits. This is due to thermal noise, qubit error rate, gate fidelity, and measurement error [41]. Despite efforts to improve quantum hardware, mitigation strategies, such as quantum error correction and quantum circuit optimization, are required to allow the near-term usage of quantum computers [51, 14].

Quantum circuit optimization reduces the demand for resources of quantum algorithms to address these hardware limitations [51, 87]. We distinguish two classes of quantum circuit optimization: architecture-independent and architecture-dependent. *Architecture-independent* targets the reduction of noisy gates and circuit depth. The reduction of noisy gates is significant because they are challenging to implement on current architectures and introduce substantial error correction overhead [37]. Reducing the circuit depth

Tobias Fischbach: tobias.fischbach@uni.lu

is linked to speeding up the execution time of a quantum circuit. *Architecture-dependent* optimization takes hardware characteristics into account and maps a quantum circuit to a specific quantum device. In particular for superconducting architectures with a grid-based topology, architecture-aware optimization considers qubit connectivity and surface codes [87]. Qubit connectivity is the physical constraint on qubit interactions. Surface codes are the quantum error correction algorithms for different gates.

Conventional quantum circuit-based optimization techniques rely on gate cancellation and gate permutation rules [48]. However, it is necessary to prove that each simplification rule is semantics-preserving with respect to the chosen gate set. Furthermore, multiple rules are often required to capture the same underlying principle (e.g., two rules per single-qubit gate to commute through a CNOT gate).

ZX-calculus emerged as an alternative quantum circuit simplification framework beyond the traditional gate model [16] (Section 4). ZX-calculus expresses the computation as a gate set independent graph called ZX-diagram. ZX-diagrams are always composed of the same few elementary building blocks.

ZX-based optimization techniques take advantage of the small and semantics-preserving set of rewriting rules offered (Section 4.3). ZX-calculus enables reductions beyond the gate level for current and future quantum devices. Therefore, a review of ZX-based quantum circuit optimization techniques is required to evaluate their practical impact and identify opportunities for future research.

In sum, the contributions of this survey are as follows:

1. This survey addresses multiple research communities that are unfamiliar with quantum computing or ZX-calculus by establishing their respective core principles (Section 2).

2. Comprehensive literature overview of ZX-based quantum circuit optimization that is organized by optimization strategies, target metrics, and architectural dependence (Sections 7 and 6).

3. Identification of key challenges and future research directions that can benefit from the diverse background of the target communities (Section 12).

2 Background

Quantum computing Quantum computing leverages quantum mechanical principles, such as superposition and entanglement, to process information through the sequential application of unitary operators to quantum states (Section 3.1). Quantum gates are unitary operators that act on qubits (Section 3.2.1). A quantum circuit is a linear map of qubits that is composed of sequential quantum gates (Section 3.2.2). Quantum circuit optimization improves the resource requirements of quantum algorithms for current NISQ-era and future fault-tolerant quantum devices (Section 3.2.3).

ZX-calculus A ZX-diagram is tensor network composed by generators that can implement the linear map of qubits for every quantum circuit (Section 4.1). Phase gadgets, phase-polynomials, Pauli gadgets and spider nests are macroscopic structures commonly found in ZX-diagrams that permit efficient optimization (Section 4.2). The rewriting rules of ZX-calculus allow for semantics-preserving transformations of ZX-diagrams (Section 4.3). Circuit extraction from ZX-diagrams is computationally expensive, and the specified extraction algorithm determines the characteristics of the resulting quantum circuit (Section 4.4.3). The presence of causal flow, general flow, or Pauli flow [32, 6, 72]is sufficient to assert a deterministic behavior necessary for polynomial-time circuit extraction (Section 4.4.2).

3 Quantum Computing

This section introduces a concise and minimal summary of quantum computing necessary to understand the connection between quantum circuits and ZX-calculus. This short introduction follows the de facto reference textbook for quantum computing and information theory by Nielsen and Chuang [62]. Quantum mechanical descriptions and examples are taken from Griffiths [42].

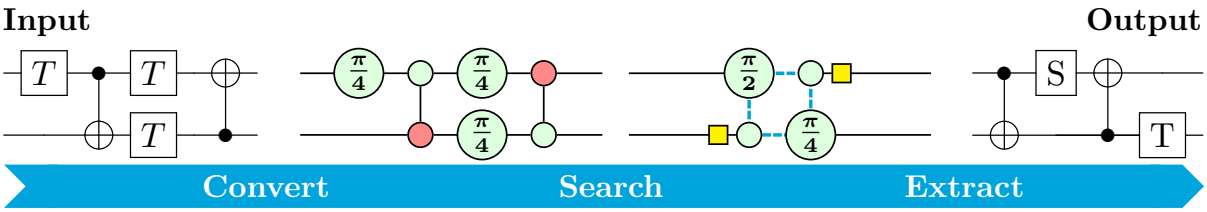


Figure 1: Pipeline for ZX-based quantum circuit optimization.

3.1 Fundamentals

3.1.1 Key Concepts

Quantum systems are described by a complex *Hilbert space*, an inner product space that is complete in the induced norm. Physical pure quantum states are represented in a complex Hilbert space by rays, an equivalence class of vectors that differ by the multiplication of a nonzero complex scalar. Vectors that only vary by a global phase characterize the same quantum state.

The *Dirac notation* is a widely used description for the linear algebra of complex vector spaces, such as Hilbert spaces, encountered in quantum mechanics [29]. In the Dirac notation, a quantum state is named ket $|\phi\rangle$ and defined up to a complex phase as a normalized Hilbert space vector. A bra $\langle\psi|$ is defined as the Hermitian conjugate of a ket $|\psi\rangle$, such that $\langle\psi| = (|\psi\rangle)^\dagger$. For Hilbert spaces of finite dimensions, this corresponds to the conjugate transpose of a ket vector. The inner product $\langle\phi\,|\,\psi\rangle$ gives the probability amplitude of a quantum system that originates in $|\psi\rangle$ can be found in state $|\phi\rangle$ upon measurement. The Born rule states that the probability to obtain outcome $|\phi\rangle$ when measuring state $|\psi\rangle$ is given by $|\langle\phi\,|\,\psi\rangle|^2$ [10]. All states throughout this section are assumed to be normalized vectors ($\langle\phi\,|\,\phi\rangle = 1$) to ensure that the total probability from measurement outcomes equals one.

A key concept in quantum mechanics is *superposition*. It signifies that a quantum state can be expressed as a linear combination of basis states with coefficients being complex valued probability amplitudes. If $|\phi\rangle$ and $|\psi\rangle$ form an orthonormal basis, then $|\alpha|^2$ and $|\beta|^2$ are the probabilities to obtain the states $|\phi\rangle$ and $|\psi\rangle$ when measuring $|\Psi\rangle$.

$$|\Psi\rangle = \alpha|\phi\rangle + \beta|\psi\rangle$$
$$1 = |\alpha|^2 + |\beta|^2$$

The tensor product composes the joint state between quantum systems. For two quantum systems with states $|\phi\rangle$ and $|\psi\rangle$, their joint state $|\Psi\rangle$ is given by the tensor product, such that $|\Psi\rangle = |\phi\rangle \otimes |\psi\rangle$.

Entanglement is another key concept in quantum mechanics. A composite state is entangled if it cannot be decomposed as a tensor product between states of the individual subsystems.

Operators $\hat{U}$ are linear maps that act on quantum states. The application of an operator $\hat{U}$ onto a state $|\phi\rangle$ result in a new state $|\phi_U\rangle$, such that $|\phi_U\rangle = \hat{U}|\phi\rangle$. Generally, operators do not commute $UV \neq VU$; hence the order in which the operators act on a state remains relevant.

Quantum operators are linear maps that act onto the Hilbert space. For two states $|\phi\rangle$, $|\psi\rangle$ with complex coefficients a, b, the operator satisfies $\hat{U}\left(a|\phi\rangle + b|\psi\rangle\right) = a\hat{U}|\phi\rangle + b\hat{U}|\psi\rangle$ holds. Furthermore, the time evolution of a quantum state in a closed system is described by *unitary operators*, such that $\hat{U}^\dagger\hat{U} = \hat{1}$. This restriction is required because only unitary operations preserve the norm of a quantum state vector and consequently the total probability under time evolution.

Hermitian operators remain unchanged under hermitian conjugation, such that $\hat{U}^\dagger = \hat{U}$. These operators represent physical observables and admit real eigenvalues. Unitary operators that are also Hermitian satisfy $\hat{U}^2 = \hat{1}$ and recover the original state after being applied twice.

3.1.2 Qubits

The basic unit of information in quantum computing is the *qubit* [68]. In contrast to a classical bit that can only represent states 0 and 1, qubits can be in a superposition of the computational basis states $|0\rangle$ and $|1\rangle$. A basis is a set of orthonormal states that can expressed any state of the Hilbert space as the linear combination of

Pauli Basis	*Matrix*	*Eigenstates*
Z-Basis (Computational)	$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
X-Basis (Hadamard)	$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\|+\rangle = \frac{\|0\rangle+\|1\rangle}{\sqrt{2}}, \quad \|-\rangle = \frac{\|0\rangle-\|1\rangle}{\sqrt{2}}$
Y-Basis	$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	$\|+i\rangle = \frac{\|0\rangle+i\|1\rangle}{\sqrt{2}}, \quad \|-i\rangle = \frac{\|0\rangle-i\|1\rangle}{\sqrt{2}}$

Table 1: Pauli Matrices and their Eigenstates as a superposition of the computational basis.

these state vectors.

Besides the computational basis, *Pauli bases* are frequently encountered in quantum computing (Table 1). An intuitive tool to visualize pure single-qubit states up to a global phase is the Bloch sphere [9]. Figure 2 shows a *Bloch sphere* with the state $|\Psi\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}} = |+\rangle$ indicated in red. The different axes correspond to the Pauli bases -X ($|-\rangle$, $|+\rangle$), -Y ($|-i\rangle$, $|+i\rangle$) and -Z ($|0\rangle$, $|1\rangle$) bases. The probability of a measurement outcome depends on the basis it is measured in as stated by the Born rule. A measurement of $|\Psi\rangle$ on the basis of Z would result in $|0\rangle$ and $|1\rangle$ with a probability of 50% for each outcome. If the same state $|\Psi\rangle$ is measured in the X-basis instead, the final state would always be $|+\rangle$.

Entanglement connects the state of multiple qubits such that the formed composite system can not be separated by as the tensor products of its individual systems [33]. For example, the composite state of two qubits in the computational basis is described by the tensor product of the computational basis states, thus $|\mathbf{00}\rangle = |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle$.

Other prominent examples of entanglement are the *Bell states* [8]. Bell states are maximally entangled two-qubit states. Maximally entangled qubits are perfectly correlated, and the measurement of one qubit will result in correlated outcomes regardless their distance. Highly entangled qubits are an essential resource for quantum error correction [55, 37].

Coherence time is the timespan in which a quantum system maintains its phase coherence. Over time, accumulated environmental noise, e.g. from heat, forces decoherence of the quantum states, limiting the duration in which quantum systems can reliably process information.

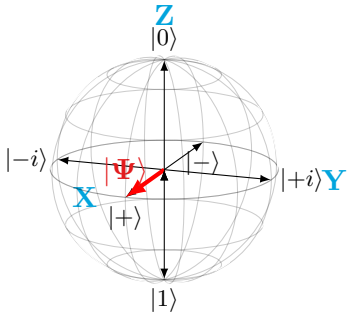


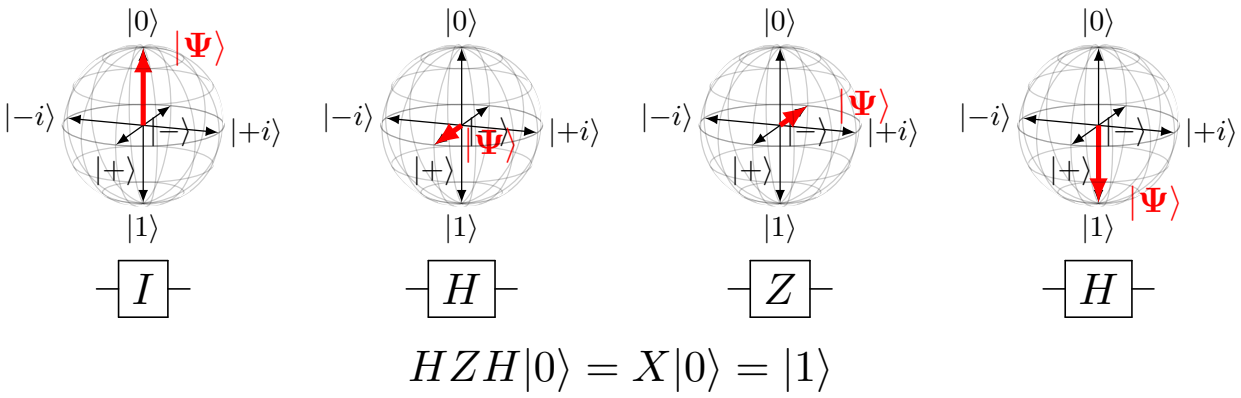
Figure 2: Bloch sphere.

3.2 Quantum Circuits

The quantum circuit model is one way to express quantum computation. Quantum circuits are graphical representations of the manipulation of quantum-mechanical states (qubits) by operators (quantum gates) [7]. Like classical electrical circuits, quantum circuits implement the semantics and control flow of a program. Individual operations are performed by quantum gates.

3.2.1 Quantum Gates

The elementary building blocks of quantum circuits are quantum gates. Like their classical counterparts, the logic gate in digital computers, quantum gates implement elementary operations onto a state. In contrast to classical logic gates, unitary quantum gates are *reversible*. The reversibility of computation mandates that the original states are always recoverable from the final states [69]. Certain quantum gates have classical analogs; the quantum X gate is the quantum equivalent of the classical NOT gate. Other gates are unique for the quantum computing paradigm without a classical counterpart; unique quantum



$$HZH|0\rangle = X|0\rangle = |1\rangle$$

Figure 3: Application sequence (left to right) of single-qubit gates starting at $|0\rangle$.

gates include the Hadamard H and Z gate.

From a quantum-mechanical perspective, all quantum gates are unitary operators that can be represented as unitary matrices. Quantum gates can operate on one or multiple qubits. Single-qubit gates can be visualized as rotations on the Bloch sphere. Figure 3 highlights the sequential application of the HZH operators to the $|0\rangle$ state, acting effectively as the X gate. The Z, X and H gates reappear later in the form of the elementary building blocks of ZX-calculus. If a quantum gate is both a unitary and a Hermitian operator, applying the same gate twice recovers the original state.

Quantum computers can implement universal gate sets to perform every computation. Although not universal, the Clifford gate set is a popular choice as a consequence of the Gottesman-Knill theorem, which states that Clifford computations can be efficiently simulated classically [40, 1]. Although there are an infinite number of universal gate sets, in practice the *Clifford+T* set is chosen due to its importance to fault-tolerant compilation where the cost of non-Clifford gates, such as the T-gate, are higher than Clifford gates [37]. Table 2 lists the gates that compose the Clifford + T-gate set with the corresponding unitary matrices and the quantum circuit notation.

As we saw before, quantum gates are unitary operators that do not necessarily commute. Therefore, different universal gate sets admit different gate commutation rules (e.g., Clifford+T or Toffoli). This is of particular importance, for the field of rule-based quantum circuit optimization [40, 48].

3.2.2 Quantum Circuits

To process information, quantum circuits combine an application sequence of quantum gates on qubits. Quantum circuits are a linear map of qubits.

They are composed of matrix and tensor products of operators. The matrix product is used for the sequential composition of gates along the program flow. The tensor product describes the composite system that arise from parallel gate application. As not all quantum gates commute, the exact sequence needs to be preserved upon composition of the full operator such that the semantics are preserved.

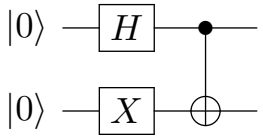


Figure 4: Generating circuit of the $|\Psi^+\rangle$ Bell state.

Figure 4 shows a quantum circuit that generates the $|\Psi^+\rangle$ Bell state. The control flow is from left to right. Starting from $|00\rangle$, an H gate is applied on the first qubit and a X gate on the second qubit, resulting in $\left(\frac{|01\rangle+|11\rangle}{\sqrt{2}}\right)$. Lastly, a CNOT gate is placed giving rise to the $|\Psi^+\rangle$ Bell state:

$$CX\dot{(}H \otimes X)|00\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = |\Psi^+\rangle$$

The familiar look of quantum circuits of classical computing should not deceive one from the fact that quantum computing is a fundamentally different paradigm that enforces different constraints not known by classical computation. The *no-cloning theorem* states that it is impossible to

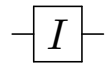
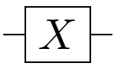
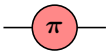
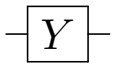
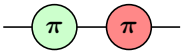
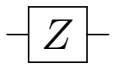
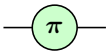
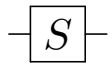
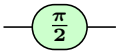
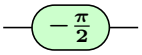
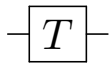
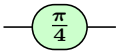
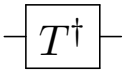
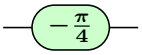
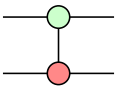
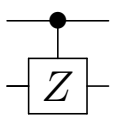
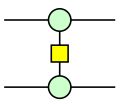
Gate	*Unitary Matrix*	*quantum circuit*	*ZX-Calculus*	*Hermitian*	*Single-Qubit*
Identity	$\mathbb{1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$			☑	☑
Pauli-X	$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$			☑	☑
Pauli-Y	$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$			☑	☑
Pauli-Z	$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$			☑	☑
Phase	$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$			☒	☑
S†	$S^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}$			☒	☑
T	$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$			☒	☑
T†	$T^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{bmatrix}$			☒	☑
Hadamard	$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$			☑	☑
CNOT	$CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$			☑	☒ (2-qubit)
Controlled Z	$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$			☑	☒ (2-qubit)

Table 2: Quantum gates that compose the universal Clifford+T gate set and the controlled Z (CZ) gate.

create a perfect and independent copy of an arbitrary unknown quantum state [86, 28]. As a direct consequence, quantum computing cannot use classical error correction codes that are based on the repetition of direct copying. Different quantum error correction schemes exist that encode information into entangled states (e.g., Shor [70], Steane [77], and surface codes [11]).

3.2.3 Quantum Circuit Optimization

Current generation quantum hardware possesses several resource restrictions, namely physical qubit availability and limited coherence time. Quantum circuit optimization addresses these limitations and can be broadly divided into two categories [51].

Architecture-independent optimization focuses on hardware-agnostic strategies at the logical qubit level without any consideration of their physical implementation. Logical qubits are error corrected qubits composed of several physical qubits. Physical qubits are the actual hardware elements. They are prone to noise and are not error corrected. With the aid of quantum error correction, multiple physical qubits form a logical qubit.

Typical optimization targets for NISQ-era quantum devices multi-qubit gates (e.g., CNOT, CZ and SWAP) due to their high implementation cost and connectivity constraints [64], while the cost of fault-tolerant architectures is driven by T-gates [41].

Decreasing the circuit depth reduces the execution time of a quantum circuit. Speeding up the execution time is especially important if the unoptimized circuit cannot be executed during the coherence time. Furthermore, quantum circuit synthesis aims to decompose arbitrary unitary operations into an optimal sequence of quantum gates [87].

Architecture-dependent optimization concentrates on hardware specific characteristics, primarily a circuit's qubit connectivity (which qubits can directly interact with each other), gate fidelity (accuracy of a gate's operation), circuit fidelity (accuracy of a full circuit's unitary operator), and error rate (probability of erroneous state changes) [88]. Multi-qubit gates can only operate on connected qubits. Extra operations, such as SWAP gates, facilitate connections between qubits. Routing optimizes maps a quantum circuit onto a quantum device and minimizes the overhead introduced by the inserted SWAP gates.

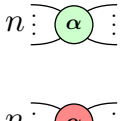
A quantum architecture encodes logical qubits using many physical qubits according to an quantum error correction scheme. Additionally, architecture-aware synthesis decomposes and aligns parts of the quantum circuit with respect to the connectivity and limitations of a given quantum architecture [87].

4 ZX-Calculus

ZX-calculus is a diagrammatic reasoning framework for quantum circuits. Both represent linear maps of qubits, but ZX-calculus offers a compact representation of complex quantum programs and admit a sound and complete set of rewriting rules. Originally introduced in 2008 by Coecke and Ross [16], ZX-calculus is gaining popularity in the field of quantum circuit optimization with further applications in quantum circuit verification. This section follows the book-sized introduction of Coecke and Ross [18] and the shorter introductory paper van de Wetering [80].

4.1 Fundamentals

The objects in ZX-calculus are referred to as ZX-diagrams, a type tensor networks that represent the linear map among qubits. The generators of ZX-diagrams are *spiders*. A spider represents a tensor that operates on qubits in the Z-basis $\{|0\rangle, |1\rangle\}$ (green) or X-basis $\{|-\rangle, |+\rangle\}$ (red). Spiders have n inputs, m outputs, and carry a phase of α. The following figure visualizes a Z- and X-spider with n inputs and m outputs and their respective unitary operator in Dirac notation.



$$= |0,...,0\rangle\langle 0,...,0| + e^{i\alpha}|1,...,1\rangle\langle 1,...,1|$$

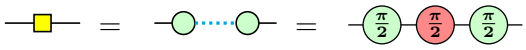
$$= |+,...,+\rangle\langle +,...,+| + e^{i\alpha}|-,...,-\rangle\langle -,...,-|$$

Wires connect the output of one spider with the input of another spider. They implement the identity linear map on a qubit and leave the quantum state unchanged:

$$= |0\rangle\langle 0| + |1\rangle\langle 1|$$

Phase-less spiders with single input and output wire, implement the identity linear map and therefore act like wires.

The Hadamard gate can be represented by either a yellow box connected by wires or a blue dotted line between spiders. The blue dotted line appears in the graph-formalism and is called a *Hadamard edge / wire*. Euler decomposition, known as the *Hadamard rule (hd)*, splits up a Hadamard wire into a sequence of Z and X spiders [31].



A typical ZX-diagram consists of many connected spiders. Matrix multiplication composes the linear map of sequentially connected spiders and the tensor product composes the linear map between parallel spiders. ZX-calculus is *universal*, therefore every quantum circuit can be expressed as a ZX-diagram, but the reverse is not trivial and is known as the circuit extraction problem (Section 4.4.3). Two ZX-diagrams are considered equal, if they implement the same linear map up to a global complex phase.

ZX-calculus is especially suited for quantum circuit optimization because it offers a rewriting system that is:

- *Sound:* all rules are semantic-preserving and provable correct.
- *Compact:* small number of elementary rewriting rules for different fragments, e.g. the scalar-free ZX-calculus with the Clifford fragment shown in Figure 6.
- *Complete:* all rewriting rules are derivable from first principles within the ZX-calculus up to arbitrary real phases [4, 5, 49, 50].

ZX-calculus demonstrates its effectiveness for quantum circuit optimization in particular, because rather than adhering to the rigid quantum circuit structure with local gate commutation and cancellation rules (e.g., Clifford gates [40] and rotation gates [48]), it works underlying symmetries and structures of the linear maps [32].

Although a spiders phase can be a real number in the general case, restricting the validly assignable phases can be useful to represent specific universal gate sets. Spiders with phases that are multiples of $\frac{\pi}{2}$ can implement all Clifford gates. Figure 6 shows the basic rewriting rules of the scalar-free Clifford ZX-calculus. A T-gate corresponds to a Z-spider with a phase of $\frac{\pi}{4}$. Clifford gates and the T-gate form a universal gate set together.

ZX-diagrams form a dagger compact product and permutation (PROP) where topological transformations, such as bending or moving wires without changing the connectivity, do not change the implemented linear map [58, 17].

In this survey, we focus on the graph-formalism of the ZX-calculus.. Every ZX-diagram can be translated into its graph-like representation (Definition 4.1) using the color changing, spider fusion

Definition 4.1. *Graph-like ZX-diagrams A ZX-diagram is graph-like if it admits the following characteristics [32]:*

1. *All spiders are Z-spiders.*
2. *No parallel Hadamard wires or self-loops.*
3. *Z-spiders are only connected by Hadamard wires.*
4. *All inputs and outputs are connected to a Z-spider.*
5. *Every Z-spider is connected to maximal one input or output.*

4.2 Macroscopic Structures

This section introduces macroscopic structures, larger scale patterns of generators, commonly found in ZX-diagrams that are used by some of the surveyed works. *Phase gadgets* are structures in ZX-diagrams that add a phase to a state [53, 21]. Legs are wires that connect spiders of the same color to a single spider of the opposite color, which subsequently connects to a state. Figure 5 visualizes a phase gadget with three legs on the left. Their unitary can be expressed as matrix exponentials $e^{-i\theta \oplus_i Z_i}$ or $e^{-i\theta \oplus_i X_i}$. Figure 5 shows the phase gadget and its decomposition into a CNOT ladder that implements the unitary $e^{-i\theta ZZZ}$. *Phase polynomials* are a class of circuits that are only composed of CNOT, and phase gates acting in the Z-basis. These circuits are interesting because they can be fully described by an unitary operator that takes the form of a matrix exponential $e^{-i\frac{\theta}{2}(1-2Px)}$ with P being the circuit's parity table. Phase polynomials are a

sequence of phase gadgets. While phase polynomials are based on the parity operator $\oplus$, *spider nests* are based on the and $\wedge$ operator [61]. *Pauli gadgets* extend the notion of phase gadgets by allowing each leg to connect to spiders that can be of type X, Y or Z.

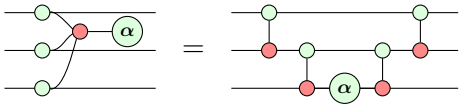


Figure 5: Example of a phase gadget and its decomposition into a CNOT ladder.

4.3 Rewriting Rules

This section introduces the basic rewriting rules of the ZX-calculus that are outlined in Figure 6 [18, 75].

A rewriting rule transforms a ZX-diagram while preserving its underlying semantics under the following definition:

Definition 4.2. *Let* **LM** *be the set of linear maps among qubits, and let* **ZX** *be the set of ZX-diagrams. The function* $\gamma : \mathbf{ZX} \to \mathbf{LM}$ *is a function that maps a ZX-diagram to its linear map between qubits. Two ZX-diagrams that represent the same linear map between qubits* $g, h \in \mathbf{ZX}, \gamma(g) = \gamma(h)$ *have the same semantics. A rewriting rule is a function* $r : \mathbf{ZX} \to \mathbf{ZX}$ *that transforms a ZX-diagram while preserving its underlying semantics, such that* $g \in \mathbf{ZX}, \gamma(g) = (\gamma \circ r)(g)$.

All rules remain valid under color inversion. We give an example of the successive application of some rewriting rules on a simple ZX-diagram in Figure 7.

Spider fusion (f) Connected spiders of the same color fuse through modulo-2π addition of their phases. The reverse unfusing operation is always possible, because connecting additional spiders with a phase of $\alpha = 0$ will not change the modulo-2π addition. As a consequence, infinite spiders can be unfused. Figure 7 highlights the fusion of two green non-phase-carrying spiders with their neighboring phase-carrying spiders.

Color change (h) Adding Hadamard generators to each input and output inverts the color of a spider. In Figure 7, all red spiders turn green with the addition of Hadamard generators.

Identity removal (i1, i2) Phase-less spiders with a single input and output wire ($n = m = 1$) implement the identity linear map. Therefore, they act like wires and the phase-less spider can be removed. Similarly, two directly connected Hadamard generators cancel each other out and act like a wire.

Bialgebra (b) The bialgebra rule originates from the commutation relation between the *COPY* and *XOR* algebra. This rule permits connected spiders of opposite colors to move through each other at the cost of potentially adding spiders.

Copy (π) π copying moves an input spider that carries the phase $\alpha = \pi$ through an opposite colored spider to all connected wires while multiplying the phase by -1.

State copy (c) State copy is a special case of the copy rule when the input spider does not have any input wire ($n = 0$) and the phase is a multiple of π. It copies the computational basis through an opposite-colored spider and places the copied spider on every outgoing wire. The opposite-colored spider vanishes.

4.3.1 ZX-based quantum circuit Optimization

At the heart of ZX-based quantum circuit optimization is the successive application of rewriting rules. Figure 7 shows an example rewriting rule sequence of that applies the fusion (f) and color change (h) rules and results in a graph-like ZX-diagram (Definition 4.1).

Optimization Pipeline Figure 1 shows an idealized three-step optimization pipeline for ZX-based quantum circuit optimization.

1. *Convert:* The quantum circuit is converted to an equivalent ZX-diagram. ZX-calculus is universal; hence every quantum circuit can be converted.

2. *Search:* ZX-diagram optimization for different target metric(s) by applying semantic preserving rewriting rules (Figure 6)

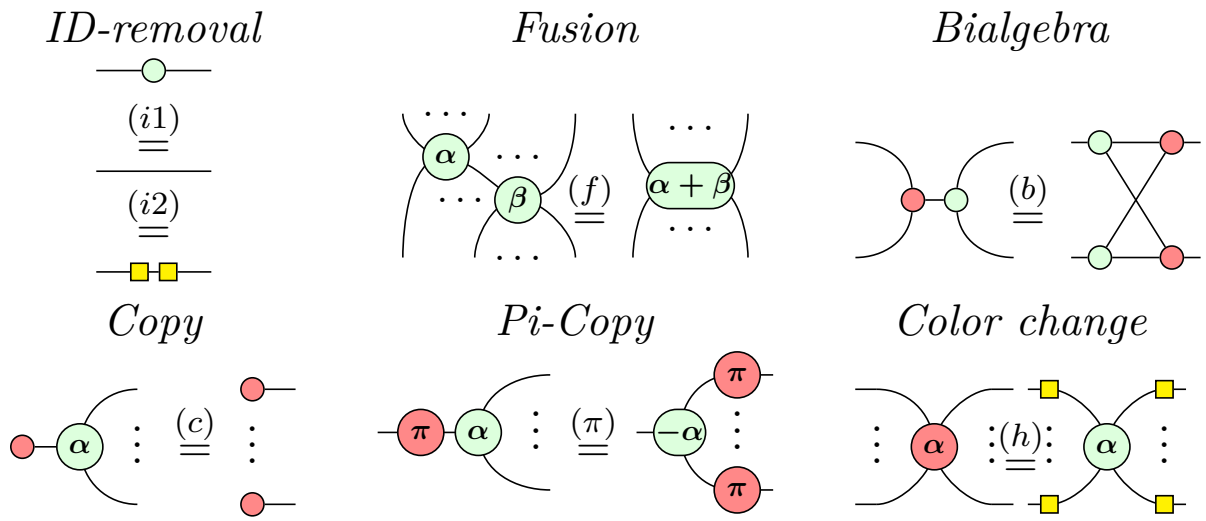


Figure 6: The basic rewriting rules of the scalar-free $\frac{\pi}{2}$ ZX-calculus.

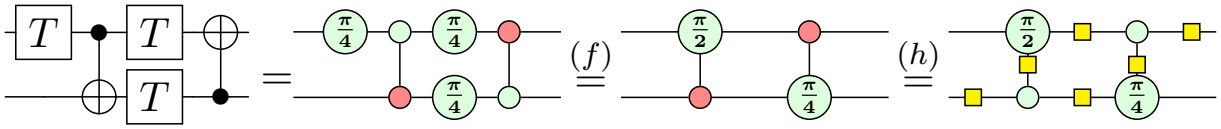


Figure 7: Successive applications of rewriting rules to a simple ZX diagram (to be read from left to right).

3. *Extraction:* Circuit extraction (Section 4.4.3) translates the ZX-diagram into an equivalent quantum circuit.

The two entry points to optimize ZX-diagrams are the search stage and the extraction stage. The search stage primarily targets ZX-diagram metrics that directly translate or approximate quantum circuit metrics (e.g., Non-Clifford spiders and Hadamard wires). The circuit extraction stage targets quantum circuit metrics (e.g., two-qubit gate count and circuit depth) that are possibly architecture-aware.

4.4 The Circuit Extraction Problem

The conversion of ZX-diagrams into quantum circuits necessitates computationally expensive circuit extraction algorithms. A ZX-diagram does not have one but many valid quantum circuit representations. As a consequence, the circuit characteristics and optimization results are heavily reliant on the chosen extraction method. This section introduces the principles of the circuit extraction algorithm proposed by Duncan et al. [32]. Its extension by Backens et al. [6] is included as the standard algorithm in many works, as it allows for the extraction of phase gadgets.

In principle, circuit extraction is #P-hard and poses an optimization problem with an upper bound of $\mathrm{NP}^{\mathrm{NP}^{\#\mathrm{P}}}$ [27, 60]. Therefore, it is computationally beneficial to reduce the need of circuit extraction as much as possible. Nevertheless, there are polynomial-time circuit extraction algorithms for ZX-diagrams in the MBQC form that maintain a graph-theoretic property known as general flow or the stricter causal flow [6].

These algorithms allow for the extraction of larger ZX-diagrams, but do so at the cost of introducing additional two-qubit gates to preserve the connectivity of spiders. As a result, these extraction algorithms can lead to circuits with an increased gate count and depth compared to the source circuit.

4.4.1 Measurement-based Quantum Computing

Measurement-based quantum computing (MBQC), also known as one-way quantum computing, is an alternative to the quantum circuit model [66, 12].

In this model, the computation is performed in two steps: (i) the preparation of a highly en-

tangled resource state, often a graph state, (ii) followed by sequential single-qubit measurements known as measurement patterns. Quantum measurements are inherently non-deterministic, and the measurement outcomes might introduce unwanted behavior in the remaining unmeasured computation. To ensure deterministic behavior, future qubits are only corrected based on previous measurement outcomes without affecting already measured qubits. The measurements itself can be carried out in different measurement planes (XZ-, XY-, YZ-planes). Determinism in the MBQC model requires sequence of measurement patterns where all qubits are measured, while only correcting future qubits at each measurement. For a detailed introduction into MBQC, we refer to the paper by Wei [83]. A major application of MBQC is photonic quantum computing [19].

Following the work of Backens et al. [6], ZX-diagrams can be interpreted as graph states (Definition 4.3) in the framework of MBQC.

Definition 4.3. Graph state diagram [6]

1. *All spiders are Z-spiders.*
2. *Only Hadamard edges between spiders.*
3. *Each spider has a single incident output wire.*

A ZX-diagram is in the *MBQC form* if it can be interpreted as a graph state *and* each spider is allowed to be connected to an input and a measurement effect instead of the output. If a ZX-diagram is in the MBQC form and the placement of single-qubit Clifford gates on the inputs and outputs is allowed, it is named *MBQC+LC (local Clifford) form.*

ZX-diagrams in the MBQC form are labelled open graphs (Definition 4.4)

Definition 4.4. Open graph [30] *Let* $G(\mathbf{V}, \mathbf{E}, \mathbf{I}, \mathbf{O}, \lambda)$ *be a finite undirected graph formed by the set of vertices* $\mathbf{V}$*, edges* $\mathbf{E}$*, inputs* $\mathbf{I} \subseteq \mathbf{V}$ *and outputs* $\mathbf{O} \subseteq \mathbf{V}$*. Let deg* $: \mathbf{V} \to \mathbb{N}$ *be a function that retrieves the degree, the number of connected edges, of a vertex. The sets of inputs* $\mathbf{I}$ *and outputs* $\mathbf{O}$ *only consist of single degree vertices, such that* $v \in \mathbf{I} \cup \mathbf{O}, deg(v) = 1$*. Together, the set of input and output vertices* $\mathbf{I} \cup \mathbf{O}$ *form the boundary of* G*. Let* $\tilde{\mathbf{O}} = \mathbf{V} \setminus (\mathbf{I} \cup \mathbf{O})$ *be the set of interior vertices. Let* λ *be a labeling function. The function* $\lambda : \tilde{\mathbf{O}} \to \{XY, XZ, YZ\}$ *assigns a measurement plane. We denote by* $\mathbf{ZX}$ *the set of all labeled open graphs.*

A ZX-diagram is deterministic when its underlying open graph admits a sequence of valid measurement patterns until all qubits are measured.

ZX-diagrams do not have a temporal or spatial order of spiders beyond its connectivity; hence no partial order can be inferred trivially. In contrast, the quantum circuit model places gates that act son specific qubits. This representation implicitly defines partial order on the gates since they are arranged by time.

When interpreting a ZX-diagram in the MBQC form, a sequence of deterministic measurement patterns induces a partial execution order on the vertices. In general, multiple partial orders can be consistent with a ZX-diagram's underlying open graph.

The presence of causal, general or Pauli flow inside a ZX-diagram's open graph are sufficient conditions for deterministic behavior [32, 6, 72]. For MBQC diagrams that admit one of these flows, polynomial-time circuit extraction algorithms exist. Circuit extraction is not a mere translation between representations, instead it asserts deterministic behavior and induces a valid execution order.

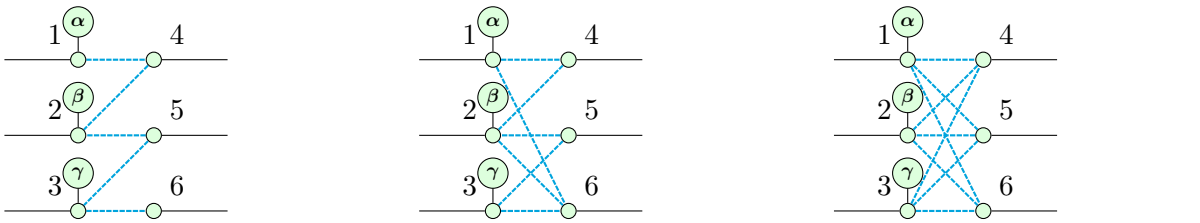
A graph-like ZX-diagram (Definition 4.1) corresponds to a ZX-diagram in the MBQC form where all measurements are performed in the XY-plane. For graph-like ZX-diagrams, the existence of *causal flow* suffices to guarantee a deterministic measurement pattern [23]. In the general case, ZX-diagrams in the MBQC form contain measurements in the XZ-, XY-, YZ- planes. The existence of *general flow* [13] is a sufficient condition for deterministic measurement patterns in the general case.

For the rest of this review paper, we assume that ZX-diagrams are in the MBQC form.

4.4.2 Causal and General Flow

Causal flow Causal flow is a sufficient condition for the existence of a deterministic measurement pattern for ZX-diagrams that are restricted to measurements in the XY-plane. Deterministic measurement patterns that always correct a single unmeasured qubit admit causal flow.

Definition 4.5 (Causal Flow [23])**.** *Let ZX be an open graph with the set of vertices* $\mathbf{V}$*, inputs* $\mathbf{I}$*,*



(a) ZX-Diagram with causal flow. (b) ZX-diagram with general flow. (c) ZX-diagram without flow.

Figure 8: The different flow states of ZX-diagrams.

and outputs $\mathbf{O}$*. The function* $g : \mathbf{V} \setminus \mathbf{O} \to \mathbf{V} \setminus \mathbf{I}$ *maps measured qubits to unmeasured correction qubits. The neighborhood* $N(v)$ *of a vertex* $v \in \mathbf{V}$ *is the set of all adjacent vertices. causal flow is a strict partial order* $\prec$ *on* $\mathbf{V}$*, such that* $\forall v \in \mathbf{V} \setminus \mathbf{O}$*:*

1. $v \prec g(v)$
2. *if* $w \in N(g(v)) \implies (w = v) \vee (v \prec w)$
3. $v \in N(g(v))$

Figure 8a shows a ZX-diagram that admits causal flow. Following the causal flow Definition 4.5, suppose $v = 1$ is the first qubit to be corrected with the correction vertex $g(1) = 4$. This implies the partial order of $1 \prec 4$. Let the $v = 2$ be the second qubit to be corrected by the correction vertex $g(2) = 5$, implying an ordering of $2 \prec 5$. The final qubit $v = 3$ is corrected with correction vertex $g(3) = 6$, implying an ordering of $3 \prec 6$. Combining all partial orders lead to a causal flow with a measurement order of $1 \prec 2 \prec 3$.

General flow General flow is a necessary condition for the existence of a deterministic measurement pattern of general ZX-diagrams, including other measurement planes. In contrast to causal flow, general flow allows for the correction of multiple unmeasured qubits as a result of a single measurement.

Definition 4.6 (General Flow [13, 24])**.** *Let* ZX *be an open graph with the set of vertices* $\mathbf{V}$*, inputs* $\mathbf{I}$*, and outputs* $\mathbf{O}$*. The function* $g : \mathbf{V} \setminus \mathbf{O} \to \mathcal{P}(\mathbf{V} \setminus \mathbf{I})$ *maps measured qubits to a correction set of unmeasured qubits. Let* $\lambda : \mathbf{V} \to \{XY, XZ, YZ\}$ *be a function that assigns a measurement plane to each vertex. Let* $\tilde{N}(\mathbf{S})$ *be a function that maps the correction set* $\mathbf{S} \subseteq \mathbf{V}$ *to the set of vertices that have an odd number of neighbors in* S*, such* $\tilde{N}(\mathbf{S}) = \{\forall u \in \mathbf{V}, |N(u) \cap \mathbf{S}| = 1 \mod 2\}$ *where* $N(u)$ *denotes the neighborhood of vertex* u*. general flow is a partial order* $\prec$ *on* $\mathbf{V}$*, such that* $\forall v \in \mathbf{V} \setminus \mathbf{O}$*:*

1. *if* $w \in g(v) \wedge v \neq w \implies v \prec w$
2. *if* $w \in \tilde{N}(g(v)) \wedge v \neq w \implies v \prec w$
3. *if* $\lambda(v) = XY \implies v \notin g(v) \wedge v \in \tilde{N}(g(v))$
4. *if* $\lambda(v) = XZ \implies v \in g(v) \wedge v \in \tilde{N}(g(v))$
5. *if* $\lambda(v) = YZ \implies v \in g(v) \wedge v \notin \tilde{N}(g(v))$

The ZX-diagram visualized in Figure 8b admits general flow. Enumeration over all correction sets and subsequent filtering by the conditions of the general flow (Definition 4.6), it can be shown that the ZX-diagram admits 25 different general flows. In the following, one general flow is considered in more detail. Let qubit $v = 2$ be the first qubit to be corrected with the correction set $g(2) = \{4\}$. The resulting odd neighborhood $\tilde{N}(g(2)) = \{1, 2\}$ implies an ordering of $2 \prec 1$. Next, qubit $v = 1$ is to be corrected. With the choice of $g(2) = \{4\}$, $g(1)$ can take two different non-contradictory values, namely $\{5, 6\}$ and $\{4, 5\}$. Let's correct qubit $v = 1$ with the correction set $g(1) = \{4, 5\}$, such that the odd neighborhood takes the form of $\tilde{N}(g(1)) = \{1, 3\}$, implying $1 \prec 3$. Consequently, the last remaining qubit $v = 3$ can only be corrected by $g(3) = \{4, 6\}$ without effecting previously measured qubits with the odd neighborhood $\tilde{N}(g(3)) = \{3\}$. The transitive closure of the implied constraints lead to a general flow with a measurement order of $2 \prec 1 \prec 3$.

No flow ZX-diagrams are non-deterministic in the absence of causal flow, general flow or Pauli flow [32, 6, 72]. Figure 8c shows a ZX-diagram that does not admit any flow. Intuitively, this

can be seen from the fact that all input qubits are connected to all outputs. It is impossible to find a measuring pattern without correcting on previously measured qubits.

To illustrate the absence of flow, consider the following example. Let $v = 1$ be the first qubit to be corrected by the correction set $g(1) = \{4\}$ with the corresponding odd neighborhood $\tilde{N}(g(1)) = \{1, 2, 3\}$. Following the general flow Definition 4.6, an ordering of $1 \prec 2$ and $1 \prec 3$ is implied. Let $v = 2$ be the second qubit to be corrected by the correction set $g(2) = 5$, resulting in the odd neighborhood of $\tilde{N}(g(2)) = \{1, 2, 3\}$. This neighborhood implies an ordering of $2 \prec 1$ and $2 \prec 3$. Both orderings are contradictory, as they demand $1 \prec 2$ and $2 \prec 1$. Even when enumerating all possible combinations of correction sets, it is impossible to find an ordering that does not contradict the general flow Definition 4.6. Therefore, the ZX-diagram visualized in Figure 8c is non-deterministic and not extractable.

4.4.3 Circuit Extraction in Polynomial-Time

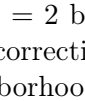
Polynomial-time circuit extraction algorithm exists for ZX-diagrams that preserve causal flow, general flow or Pauli flow [32, 6, 72]. Duncan et al. [32] introduce the first circuit extraction algorithm for ZX-diagrams with causal flow and focused general flow. Backens et al. [6] extend the previous work of Kissinger et al. [52] to consider general flow.

This section introduces the principles of the circuit extraction algorithm for graph-like (Definition 4.1) ZX-diagrams by Duncan et al. [32]. These techniques reemerge in various works other works, including the standard extraction algorithm by Backens et al. [6] and architecture-aware extraction (e.g., Villoria et al. [82], Staudacher et al.[76], and Kissinger et al. [52]).

Figure 9 illustrates the main steps during circuit extraction. The ZX-diagram is divided into an extracted and unextracted part separated by a frontier $\epsilon_1, \ldots, \epsilon_3$. The key to circuit extraction is that the addition of a CNOT gate changes the adjacency of the frontier. Specifically, it adds or removes a Hadamard wire between a frontier and a neighboring spider depending on the placement of the control and target qubit. This transformation does not change the underlying linear map as shown by Duncan et al. [32].

Starting from right to left (Figure 10), circuit extraction is an iterative process until no unextracted part remains:

1. Spiders that have a single input and output wire can be extracted directly by single qubit gates. Figure 9a shows the extraction of a T-gate in the top row.

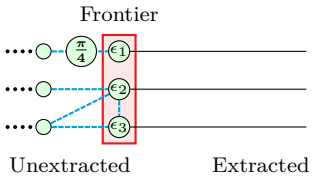
2. The wire between two directly connected frontier spiders is extracted by a CZ gate  . Figure 9b shows the CZ gate that results from the extraction of the directly connected frontier spiders ϵ_2 and ϵ_3.

3. CNOT  gates are used for frontier spiders that are connected to multiple unextracted spiders. Figure 9c shows how a CNOT gate is added to extract the diagonal connection of ϵ_2.

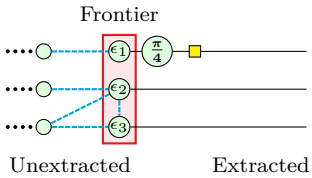
Frontier spiders with multiple connected neighbors are extracted based on their binary adjacency matrix. A spider is directly extractable iff it connects to a single frontier spider, indicated by rows contain a single non-zero entry (rule 1). The frontier can be manipulated by row additions of its adjacency matrix, with the goal to eliminate edges until rule 1 is applicable. Duncan et al. [32] showed that the addition of two rows corresponds to their bitwise XOR, with each row addition adding a CNOT gate and removing or adding a Hadamard wire.

Let's consider the example shown in Figure 10 that admits the adjacency matrix $\mathbf{A}$ for frontier spiders $\epsilon_1, \ldots, \epsilon_3$ and neighbors $v_1, \ldots, v_3$ (Figure 9c), where $\mathbf{A}_{\epsilon_i v_j} = 1$ iff ϵ_i connects v_j, and 0 otherwise. In this example, the addition of the last two rows adds a Gaussian elimination gate with control ϵ_2 and target ϵ_3 qubit (Figure 9c).

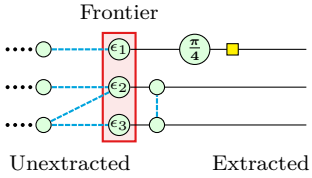
The state-of-the-art extraction algorithm uses Gaussian elimination to solve the adjacency matrix restricted to row additions and row swaps [32, 6]. Anton [3] offers a detailed introduction to the Gaussian elimination algorithm. A key limitation is that the Gaussian elimination algorithm does not solve the adjacency matrix with the minimal number of row additions [78, 57]. Consequently, reducing the number of row additions decreases the overall CNOT gate count of the extracted circuit.



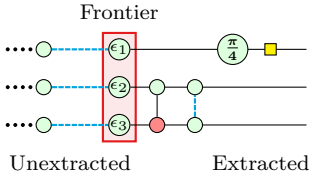
(a) Unextracted ZX-diagram.



(b) Spiders with a single input and output can be extracted directly.

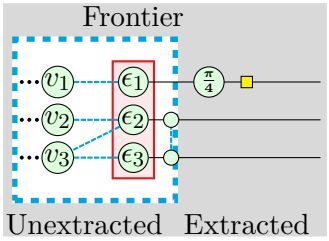


(c) Connections between frontier spiders are extracted by CZ gates.



(d) CNOT gates extract wires between frontier and unextracted spiders.

Figure 9: The different steps of circuit extraction.



$$\mathbf{A} = \begin{array}{c} \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

Figure 10: Example adjacency matrix between the frontier spiders $\epsilon_1, \ldots, \epsilon_3$ and nearest unextracted spiders $v_1, \ldots, v_3$.

5 Target Metrics

ZX-calculus allows for architecture-independent and architecture-aware optimization strategies. The objective of quantum circuit optimization is to reduce the resource demand of quantum programs and enable their execution on real quantum devices.

5.1 Noisy Intermediate-Scale Quantum Computing

Current NISQ-era devices are resource restricted and error-prone, primarily due to noise forcing decoherence [64]. They are limited in total gate count, circuit depth, two-qubit gate count, and qubit connectivity. In order to produce an executable quantum circuit, architectural characteristics need to be incorporated. For many NISQ-era quantum architectures, two-qubit gates contribute significantly more noise than typical single-qubit Clifford gates [64]. These noisy gates often require sophisticated error correction that further increases the demand for resources (e.g., Shor [70], Steane [77], and topological codes [11]).

Circuit Depth Circuit depth is directly linked to the execution time and noise exposure of the circuit. Shallower circuits can be prepared and executed faster. A quantum circuit can only be executed reliably on a quantum device if its execution time is shorter than the coherence time (see Section 3.1). The coherence time and gate error rates vary across architectures.

Two-Qubit Gate Count ZX-diagrams consist of generators and not quantum gates. The extraction of a quantum circuit is not trivial and might negate optimization success at the ZX-diagram level (see Section 4.4). Some works approximate the number of two-qubit gates from the number of Hadamard wires contained in ZX-diagrams as proposed by Staudacher et al. [75], Holker [46]. Architecture-aware strategies need to take hardware-dependent factors into account, such as qubit connectivity [52].

5.2 Fault-Tolerant Quantum Computing

Fault-tolerant quantum computing is the key to large-scale quantum computing. It implements logical qubits with many physical qubits and error-correction, such that errors can be corrected faster than they accumulate [71, 41].

Qubit Count Quantum circuits are composed of quantum gates that act on logical qubits. In the context of ZX-diagrams, the number of logical qubits is the maximum number of wires in-

tersected by any vertical cut through the diagram [81].

T-Gate Count The T-gate is a particularly noisy single-qubit gate that forms a universal gate set when combined with the Clifford gate set. Unlike typical Clifford gates, the T-gate significantly adds quantum error correction overhead on certain fault-tolerant quantum computing architectures [37]. Consequently, a prime objective of architecture-independent optimization is to reduce the T-gate count.

6 Survey

This survey categorizes ZX-based quantum circuit optimization algorithms by their underlying strategy and target metric. We selected 26 works with optimization strategies that focus on the semantic-preserving rewriting rules or improve the circuit extraction step.

Each approach is classified by its main optimization algorithm If an entry contains "Ad-hoc", the respective work introduces a novel procedure that reduces the indicated metric. If a heuristic is indicated, the optimization is guided by characteristics of the ZX-diagram. Other entries specify the algorithms used such as simulated annealing (SA), reinforcement learning (RL), genetic algorithms (GA), look-ahead (LA), integer linear programming (ILP), directed vertex feedback solver (DVFS) and template matching (TM).

Figure 11 shows the focus of the surveyed works. It counts the number of works that target a specific metric for each optimization strategy. An optimizer contributes a count of one to every target metric in the figure. In case a method approximates quantum circuit metrics at the ZX-diagram level, both are indicated. For example, this is the case when the number of two-qubit gates is approximated by the number of Hadamard wires.

Table 3 provides a high-level overview of our survey. We differentiate between architecture-agnostic in blue and architecture-aware strategies in red.

The following sections discuss the individual approaches and aim to highlight connections and identify future research directions.

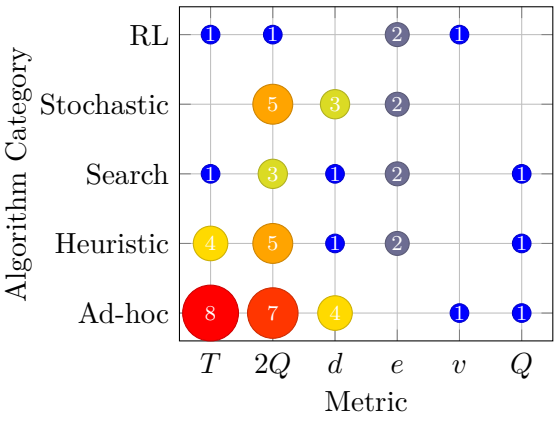


Figure 11: Number of works classified by their optimization strategy and target metric.

7 Target Metrics

ZX-calculus allows for architecture-independent and architecture-aware optimization strategies. The objective of quantum circuit optimization is to reduce the resource demand of quantum programs and enable their execution on real quantum devices.

7.1 Noisy Intermediate-Scale Quantum Computing

Current NISQ-era devices are resource restricted and error-prone, primarily due to noise forcing decoherence [64]. They are limited in total gate count, circuit depth, two-qubit gate count, and qubit connectivity. In order to produce an executable quantum circuit, architectural characteristics need to be incorporated. For many NISQ-era quantum architectures, two-qubit gates contribute significantly more noise than typical single-qubit Clifford gates [64]. These noisy gates often require sophisticated error correction that further increases the demand for resources (e.g., Shor [70], Steane [77], and topological codes [11]).

Circuit Depth Circuit depth is directly linked to the execution time and noise exposure of the circuit. Shallower circuits can be prepared and executed faster. A quantum circuit can only be executed reliably on a quantum device if its execution time is shorter than the coherence time (see Section 3.1). The coherence time and gate error rates vary across architectures.

Author(s)	*Year*	*Algorithm*	*T*	*2Q*	*Depth*	*Qubits*	*Edges*	*Vertices*
Fagan and Duncan [35]	2019	Ad-hoc		✓	✓			
Kissinger and van de Griend [52]	2019	Ad-hoc		✓				
de Beaudrap et al. [26]	2020	Ad-hoc + Heuristic	✓					
de Beaudrap et al. [25]	2020	Ad-hoc + Heuristic	✓					
Duncan et al. [32]	2020	Ad-hoc	✓					
Kissinger and van de Wetering [53]	2020	Ad-hoc	✓					
Cowtan et al. [21]	2020	Ad-hoc + Greedy		✓				
Cowtan et al. [22]	2020	Ad-hoc		✓				
Munson et al. [61]	2021	Ad-hoc + Heuristic	✓					
Zilk et al. [89]	2022	Ad-hoc	✓					✓
Gogioso and Yeung [39]	2023	SA		✓				
Staudacher et al. [75]	2023	Rand. + Heuristic		✓			✓	
Winderl et al. [84]	2023	Ad-hoc + Heuristic		✓				
Riu et al. [67]	2023	RL		✓			✓	
van de Griend and Duncan [79]	2023	Ad-hoc + Heuristic			✓			
Vandaele [81]	2024	Ad-hoc + DVFS + Greedy				✓		
Holker [46]	2024	Rand. + Heuristic		✓			✓	
Nägele and Marquardt [63]	2024	RL						✓
Staudacher et al. [76]	2024	Ad-hoc			✓			
Ewen et al. [34]	2025	GA		✓	✓			
Huang et al. [47]	2025	Ad-hoc + Heuristic	✓	✓				
Mattick et al. [59]	2025	RL + Tree search		✓			✓	
Fischbach et al. [36]	2025	Tree search	✓				✓	
Liu et al. [56]	2024	TM		✓	✓			
Chen et al. [15]	2025	SA + LA		✓	✓			
Villoria et al. [82]	2025	ILP + Peephole		✓				

Table 3: Overview of the different quantum circuit optimization algorithms, the optimization procedure and the main target metrics. Architecture-independent approaches are highlighted in blue and architecture-aware approaches are highlighted in red.

Two-Qubit Gate Count ZX-diagrams consist of generators and not quantum gates. The extraction of a quantum circuit is not trivial and might negate optimization success at the ZX-diagram level (see Section 4.4). Some works approximate the number of two-qubit gates from the number of Hadamard wires contained in ZX-diagrams as proposed by Staudacher et al. [75], Holker [46]. Architecture-aware strategies need to take hardware-dependent factors into account, such as qubit connectivity [52].

7.2 Fault-Tolerant Quantum Computing

fault-tolerant quantum computing is the key to large-scale quantum computing. It implements logical qubits with many physical qubits and error-correction, such that errors can be corrected faster than they accumulate [71, 41].

Qubit Count quantum circuits are composed of quantum gates that act on logical qubits. In the context of ZX-diagrams, the number of logical qubits is the maximum number of wires intersected by any vertical cut through the diagram [81].

T-Gate Count The T-gate is a particularly noisy single-qubit gate that forms a universal gate set when combined with the Clifford gate set. Unlike typical Clifford gates, the T-gate significantly adds quantum error correction overhead on certain fault-tolerant quantum computing architectures [37]. Consequently, a prime objective of architecture-independent optimization is to reduce the T-gate count.

8 Heuristic

8.1 Circuit Depth

Fagan and Duncan [35] introduced the first ZX-based quantum circuit optimization algorithm. Their optimization strategy moves Pauli gadgets towards the inputs and group CNOT and TONC gates (CNOT gates with switched control and target qubits) that act on the same qubits. They demonstrate a significant reduction of the CNOT gate count by $\approx$ 16% and circuit depth up to $\approx$ 30% for pure Clifford circuits.

Architecture-Aware Staudacher et al. [76] add a step to the circuit extraction algorithm Backens et al. [6] between the CNOT and single-qubit gate extraction to target neutral-atom architectures. Instead of using the Clifford + T gate set, this work targets a gate set that consists of Z gates, multi-controlled phase gates and global single qubit rotations in the XY-plane. Phase gadgets are the native representation of multi-controlled phase gates. For identified frontier phase gadgets, the fusion rule and reverse gadget fusion rules are applied. Potentially missing phase gadgets are added to extract a full multi-controlled phase gate. Neutral-atom quantum computers can natively execute multi-controlled phase gates and are limited by execution time and the total gate count. Especially the global single-qubit rotation is an order of magnitude slower than the multi-controlled phase gate and the Z phase gates. The circuit's execution time is computed along the circuit as the sum of individual gates execution time, which is assumed to increase with the rotation angle. Their approach outperforms the execution time of Qiskit compiled circuits from 26% up to 40%, primarily due to the reduction in the number of global phase gates.

8.2 T-Gate Count

Architecture-Independent Kissinger and van de Wetering [53] use the theoretical framework Duncan et al. [32] to allow semantic preserving optimization of circuits composed of the entire Clifford + T-gate set. Their procedure aims to minimize the T-gate count. Their simplification strategies apply local complementation and pivoting to remove Pauli spiders, spiders with a phase that is a multiple of π, at the expense of adding phase gadgets. Applying the identity-removal and gadget fusion rules efficiently remove the phase gadgets. By rerunning the previous algorithm symbolically, it is possible to identify non-local phase recombination that further reduce the T-gate count. This step is known as *phase teleportation*. The resulting optimization algorithm is implemented as the standard simplification routine in the PyZX library [54] under the name "full reduce". It is found in many optimization pipelines as a pre- or post-processing step for T-gate minimization. The evaluation of a benchmark set that consists

of 36 structured standard quantum circuits shows that "full reduce" improves the state-of-the-art T-count on 17% and matches on 72% of the instances. Further improvements were achieved when full reduce is paired with the dedicated T-gate optimizer TODD [45]. Combining both methods improve the T-count in 56 % of the benchmark circuits.

The following approaches develop spider nest identities for $\frac{\pi}{4}$-parity-phase operations to eliminate T-gates from quantum circuits [26]. A spider nest is a ZX-diagram that consists only of $\frac{\pi}{4}$ phase gadgets. Instead of rules that modify a single or few generators, spider nest identities take the full nest into account. Munson et al. [61] independently derived the same spider nest identities and generalized them by connecting ZX-calculus and logical AND gates.

de Beaudrap et al. [26, 25] introduce the phase gadget elimination (PHAGE) strategy, a systematic method that takes advantage of spider nest identities to reduce the T-gate count in ZX-diagrams. The core idea of PHAGE is to decompose larger and more complex phase gadgets into simpler and smaller ones. This decomposition allows for the effective application of spider nest identities to eliminate T-gates. The evaluation of PHAGE on a benchmark set of 35 parameterized circuits reveals that the state-of-the-art T-gate counts are improved for 21 instances under runtime constraints.

Architecture-Aware Zilk et al. [89] use the phase teleportation algorithm Kissinger and van de Wetering [53] to reduce the T-gate count and the Clifford elimination algorithm of Fagan and Duncan [35] to target photonic quantum computers. The first step in their procedure is to express a quantum circuit only as a sequence of Hadamard, CZ, and arbitrary Z-phase gates. The resulting circuit is converted to a measurement graph and interpreted as a graph-like (Definition 4.1) ZX-diagram. This ZX-diagram is optimized with the phase teleportation and Clifford elimination algorithms. In the MBQC framework, every spider directly corresponds to a qubit; hence a reduction of spiders (e.g. Clifford and T-spiders) decreases the number of qubits. Measurement-graphs can be directly converted into hardware instructions. The limiting factor of photonic architectures is the number of available photons and optical instruments. Compared to the standard photonic compiler Perceval [44], the ZX-based approach successfully compiles all benchmarked circuits and mostly outperforms it with respect to the photon count and optical instrument count.

8.3 Qubit Count

Vandaele [81] introduced an approach to minimize the logical qubit count in ZX-diagrams by systematically reordering and (un)bending spiders. The core idea is to use the rewrite rules to change the structure of the ZX-diagram such that it is possible to find a vertical cut that reduces the maximum number of wires, hence minimizing the logical qubit count. The NP-hard problem of minimizing the number of wires through vertical cuts is formerly known as the "fixed-endbags path width problem". Consequently, finding an optimal solution is computationally expensive for large diagrams.

8.4 Two-Qubit Gate Count

Architecture-Independent Staudacher et al. [75] demonstrate that an average reduction of the edge count by 23% translates to a two-qubit gate reduction of 16% by applying a set of general flow-preserving rewrite rules (identity removal, spider fusion, local complementation, and pivoting). They introduce a new heuristic based on the expected change of the Hadamard wire count by the local complementation and pivoting rules. A reduction of the Hadamard wire count correlates with the expected reduction of two-qubit gates. Slightly worsening of the cost function was allowed to permit further improvements later on. Stochastic and greedy algorithms are used to select the rewriting rules. Further integration with the NAM framework allowed for greater edge count reductions of 29% that translated into two-qubit gate reductions of approximately 21%.

In a related work, Holker [46] extends these ideas by focusing on diagrams that preserve the stricter causal flow property. If causal flow is preserved, the effect of a rewriting rule on the two-qubit gate count can be exactly quantified from change in the wire count, completely bypassing circuit extraction for all intermediate optimization steps. By maintaining causal flow, Holker

[46] achieves an average two-qubit gate count reduction of 20%. ZX-diagrams with causal flow have a circuit-like structure, that allows for a straightforward and efficient extraction. Moreover, verifying causal flow is computationally less expensive than verifying general flow. However, it is important to note that only a limited set of rewriting rules, namely identity removal and spider fusion, are known to preserve causal flow. This limitation severely restricts the possible diagram transformations but offers a trade-off between the solution quality and verification complexity.

The following two optimization strategies use the tket [73] compiler. Cowtan et al. [21] demonstrated that the efficient pairwise synthesis of Pauli gadgets using tket decreased the average CNOT gate count by $\approx 55\%$ and improved the two-qubit depth by $\approx 58\%$. In a subsequent work, Cowtan et al. [22] showed that a three-step optimization routine improves the CNOT gate count and depth of the unitary coupled cluster (UCC) Ansatz, a subroutine of variation quantum eigensolver (VQE), by 69% and 75%: (i) the partition of the ZX-diagram of the UCC Ansatz into commuting sets is treated as a graph coloring problem, (ii) the resulting Pauli gadgets are converted to Phase gadgets, and (iii) the phase gadgets are efficiently synthesized using Matroid partitioning [2].

Another promising approach to quantum circuit optimization is the aggregation of multiple subcircuits that can be reordered and replaced by optimized template circuits. Template circuits are pre-optimized subcircuits that implement the same program flow as the subcircuit they are meant to substitute. Liu et al. [56] introduce a string-based intermediate representation of subcircuits and a template matching algorithm that improves the CNOT gate count. Although not explicitly relying on rewriting rules for optimization, Liu et al. [56] employ ZX-calculus to verify the correctness of the intermediate representation and their associated templates.

Architecture-Aware Kissinger and van de Griend [52] introduce the first architecture-aware optimization algorithm and outperform the existing compiler frameworks QuilC [74] and tket [20]. During circuit extraction, they restrict the Gaussian elimination algorithm of the parity map to only include nearest-neighbor rows using the Steiner-Gauss algorithm. Consequently, CNOT gates can only act between neighboring qubits.

Villoria et al. [82] modified the circuit extraction algorithm of Backens et al. [6] to target trap-ion computers that are highly dependent on global gates. Instead of Gaussian elimination algorithm, the frontier is extracted solving a linear program that ensures only vertices are extracted which can be included in the same global gate. Afterward peephole optimization merges the extracted two-qubit gates into global GMS gates. This algorithm outperforms the Qiskit implementation on most quantum circuits.

van de Griend and Duncan [79] develop a two-step recursive optimization strategy that considers qubit connectivity restrictions of the underlying quantum architecture, thus circumventing the need for an additional routing step, by using the notion of phase polynomials. The algorithm introduces a biadjacency matrix $\mathbf{P}$ between the phase gadgets legs and the attached qubits. For the base recursion step, the phase gadget with the lowest connectivity that is not needed to synthesize other phase gadgets is removed. The selected phase gadget must be a non-cutting vertex, meaning the column in $\mathbf{P}$ with the most 0 or 1 entries. Afterward, the phase gadgets are synthesized by decomposition into a CNOT ladder as shown in Figure 5. The second recursion step removes rows from the remaining phase-gadget biadjacency matrix by row addition. Every row operation adds CNOT gates to the ladder. To eliminate excess CNOT gates, the Steiner-Gauss algorithm is run on the resulting parity map.

Winderl et al. [85] adapt the architecture-aware approach of Gogioso and Yeung [39] by replacing simulated annealing with a heuristic search in combination with a divide-and-conquer strategy. The first step in their strategy is the simplification of the ZX polynomial by removing, merging, and moving phase gadgets. Afterward, the ZX-diagram is split into a left parity, a right parity, and a ZX polynomial region. A Gaussian elimination optimization algorithm minimizes the CNOT count based on the combined cost of the region. The cost of removing a phase gadgets legs is computed by the minimal spanning tree of the architectural-dependent connectivity. Their novel heuristic based on the shortest path between the control and target qubit of CNOT gates

is used in conjunction with the Steiner-Gauss algorithm to synthesize the phase gadgets in both parity regions. The remaining ZX polynomial is regrouped from a leg-based score and, following a divide-and-conquer strategy, split into subregions again. These steps are recursively repeated until no ZX polynomial remains. Both methods are outperformed by other state-of-the-art algorithms, such as tket [73], for structured circuits. However, the heuristic approach of Winderl et al. [85] exhibits better scaling in the qubit count and CNOT tree depth compared to the stochastic approach by Gogioso and Yeung [39].

Huang et al. [47] develop a novel approach for architecture-aware synthesis of Trotter operators. Trotterized time evolution operators can be expressed in terms of Pauli gadgets. Pauli gadgets can be described by exponentiation of Pauli strings. Each letter of the Pauli string corresponds to a Pauli gate. The core idea of their approach is to lexicographically reorder the Pauli strings that compose the Pauli gadgets of the Trotter operator. As a result, phase gadget legs with the same letter are grouped on the same qubit. Reordering of non-commuting Pauli gadgets is possible because the introduced error (Trotter error) is outweighed by the error that originates from noisy gates. Their algorithm iteratively diagonalizes and disconnects qubits. The highest entangled qubit is chosen for the current iteration. The diagonalization step places single qubit Clifford gates on the selected qubit until the gadget has either no leg or a Z leg. Based on the Pauli gadget leg, the disconnection step introduces two CNOT gates and up to two single-qubit Clifford gates. This disconnection step reduces the entanglement of the selected qubit. The selection of the qubit is determined by an entanglement heuristic. Qubit entanglement is calculated from the occurrences of the I-gate in the Pauli string and the length difference between the largest and smallest substrings that exclude the I-gate. This approach outperforms the state-of-the-art CNOT count for random circuits and larger couple-cluster unitaries.

9 Metaheuristic

9.1 Two-Qubit Gate Count

Architecture-Independent Chen et al. [15] improve the two-qubit gate count by 2% compared to the heuristic approach of Staudacher et al. [75] using simulated annealing to partition a quantum circuit into subcircuits that are optimized. In a first step, the quantum circuit is divided into sequential layers of gates. Per layer, one single-qubit gate, one target and control qubit of multi-qubit gates can act at most on each qubit. Subcircuits are groups of sequential layers. Starting from a random configuration of subcircuits, each circuit is converted into a ZX-diagram, optimized, and extracted. The resulting circuits are iteratively merged. Following a delayed gate approach, known gate commutation and substitution rules are used to further improve the circuit depth and gate count. The delayed gate approach is implemented in PyZX [54] under the name *basic optimization*. For each iteration of the simulated annealing algorithm, the starting configuration of the subcircuits is changed.

Architecture-Aware Gogioso and Yeung [39] reduce the CNOT count for mixed phase gadgets ZX-diagrams by 27% on a grid-shaped topology. Their technique is grounded in the decomposition of phase gadgets and is paired with simulated annealing and CNOT conjugation rules. The ZX-diagram is split up into three layers: (i) a left nearest-neighbor CNOT layer, (ii) a right nearest-neighbor CNOT layer, and (iii) a mixed phase gadget ZX-diagram. The cost of implementing a phase gadget is computed from the distance between distinct legs that are mapped onto the topology. This corresponds to the nearest-neighbor CNOT count. The total cost of a circuit is the sum of all its CNOT gates. Simulated annealing is used to explore different phase gadget mappings. To exploit symmetries, the phase gadgets are converted to CNOT ladders (Figure 5), so that gate conjugation rules can be applied.

Ewen et al. [34] introduce a genetic programming approach for synthesizing shallow quantum circuits with fewer two-qubit gates from ZX-diagrams. The original quantum circuit is converted into a ZX-diagram that will be evolved via a set of mutation operations. In their work, they implement two different categories of mutations. Semantics-preserving mutations are formed by the rewriting rules of ZX-calculus. Semantics-breaking mutations, such as edge flip-

ping and edge addition, introduce new connectivity patterns in the ZX-diagram. Although these semantics-breaking mutations violate the correctness of the circuit, they permit the exploration of a larger state-space, at the cost of circuit fidelity. Experimental results demonstrate that ZX-based genetic programming can produce well-balanced circuit solutions, achieving results close to the state-of-the-art for circuit depth, circuit fidelity, and two-qubit gate count.

10 Reinforcement Learning

10.1 Metric Agnostic

Nägele and Marquardt [63] implement a scalable and general reinforcement learning framework for the optimization of ZX-diagrams that can be adapted for different metrics. Actions are grouped by node and edge impact, irrespective of the preservation of general flow or causal flow. As a consequence, quantum circuit metrics are not considered because of the lack of circuit extraction. In contrast to the reinforcement learning-approaches of Riu et al. [67] and Mattick et al. [59], the reversibility of rewriting rules is considered beyond spider un/fusion by including the bialgebra rule. Furthermore, the agent can mask actions to improve the efficiency of training.

10.2 Two-Qubit Gate Count

Riu et al. [67] introduce a reinforcement learning-based approach that uses graph neural networks to minimize the two-qubit gate count. They restrict the set of rewriting rules to general flow-preserving transformations. Circuit extraction is treated as a black-box, simplifying the optimization pipeline. The reinforcement learning agent is trained to either select and apply a rewriting rule or to terminate the optimization process. The agent can apply multiple rules until it decides to terminate. A reward function guides the decision based on metrics such as the two-qubit gate count. Moreover, the reinforcement learning framework is highly flexible and can incorporate other reward functions such as T-gate count or circuit depth. A key advantage of this reinforcement learning-based strategy is the scalability for larger ZX-diagrams. Although the initial training phase can be computationally demanding, the resulting agent can generalize its learned strategies to new diagrams independent of their size. Among the approaches surveyed, the combination of this reinforcement learning-based optimization approach with the causal flow-preserving framework of Holker [46] represents one of the most effective strategies to reduce the two-qubit gate counts in ZX-based circuit optimization.

11 Tree Search

11.1 Metric Agnostic

Fischbach et al. [36] propose an tree search strategy combined with pruning conditions to optimize ZX-diagrams. The objective of the search is to find a sequence of rewriting rules that optimizes a given metric. Their method explores the state-space using depth-first search (DFS) and iterative-deepening depth-first search (IDDFS) by applying all possible rule combinations. Pruning conditions terminate branches of the search tree, e.g., if circuit extraction is impossible. Despite being a complete search for a rewriting rule sequence that minimizes a given metric, the main drawback of their approach is the computational cost. A comparison of their IDDFS-based approach and the T-gate count obtained by the full reduce algorithm Duncan et al. [32] was performed. The analysis demonstrated equivalent results in 89% of the circuits within a benchmark set of 100 structured circuits. Nevertheless, one strength of their approach is the metric-agnosticism; it is not tied to the optimization of a single metric such as the T-gate count. They demonstrated its flexibility by targeting the edge count, a proxy for two-qubit gates, with the same strategy. While the exhaustive nature of the search makes the approach computationally inefficient for larger circuits, it allows for optimal solutions for small sized circuits.

Mattick et al. [59] propose a hybrid approach that combines reinforcement learning with a tree search algorithm that uses the full set of standard ZX-calculus rules. Their method slightly outperforms the stochastic and general flow preserving techniques Staudacher et al. [75] for random circuits. In this framework, the graph neural network replaces the heuristics by learning which and where a rewriting rule should be applied or if the agent needs to stop. The tree search allows backtracking if not beneficial transformations are encountered. Similarly to Riu et al.

[67], the agent learns where to apply which rule in the ZX-diagram based on a reward function that represents a metric. However, at each node of the tree, the agent is allowed to perform only one diagram transformation. The use of the complete set of ZX-calculus rewrite rules permits the exploration of a larger state-space at the cost of post-processing to ensure circuit extraction. The key advantage of this approach is its generality. The framework aims to learn optimal rewrite sequences for any chosen metric and is not limited to minimizing two-qubit gate counts.

12 Challenges

Comparability This survey lacks a comprehensive benchmark table that compares the various optimization strategies on the same set of quantum circuits. Most works are based on the PyZX [54] library. As the result of PyZX's fast development cycle and substantial rewrites of the rewriting rule system and the graph data structure, most optimizers are only compatible with the specific PyZX version they were originally implemented in.

Scalability A major challenge in the optimization of ZX-diagrams lies in the scalability of rule-based rewriting approaches, especially as small real-world quantum circuits result in large ZX-diagrams. With an increase in size of ZX-diagrams, the computational complexity grows rapidly, limiting the practical application of existing techniques to small-scale quantum circuits.

Current heuristic-based optimization strategies typically target a single metric, such as reducing the number of non-Clifford spiders (e.g., Duncan et al. [32], Kissinger and van de Wetering [53], and de Beaudrap et al. [25]) or minimizing the number of two-qubit gates (e.g., Staudacher et al. [75], Holker [46], and Fagan and Duncan [35]). However, while heuristics improve computational performance, focusing on a single objective fails to capture the complex features of quantum circuits and how to balance them. We identify the clear need for improved heuristics that can balance multiple metrics to improve the computational performance of existing strategies.

Recent works have explored reinforcement learning approaches for diagram rewriting (Nägele and Marquardt [63], Riu et al. [67], Mattick et al. [59]), demonstrating promising results on small-scale ZX-diagrams. Although reinforcement learning is still computationally expensive, the training stage can be seen as an upfront cost, as the learned strategies appear to be at least partially generalizable. Further enhancement of RL-based methods on small ZX-diagrams could allow better optimization results for large-scale circuits. Moreover, the lack of interpretability and the theoretical background as to why some rewriting sequences are more beneficial than others poses additional challenges. Incorporating explainable RL could form the basis for new and computationally efficient heuristics that are applicable for large-scale ZX-diagrams.

Another promising research direction is the introduction of intermediate representations that can aggregate subdiagrams and enable more efficient state-space exploration, similar to that of Liu et al. [56]. The use of an intermediate representation allows template matching of pre-optimized subcircuits. Chen et al. [15] dynamically group layers into subcircuits. Future work should focus on the efficient partition and resynthesis of quantum circuits for several reasons. Dividing larger quantum circuits into smaller subcircuits is beneficial because the resulting subdiagrams are smaller and can be optimized efficiently. As the different subdiagrams are independent of each other, each instance can be solved in parallel. This flexibility could lead the way for dynamic selection of the optimization algorithm based on the characteristics of each subdiagram. Furthermore, a subdiagram only needs to be optimized once and can be substituted for further instances. Another possible research direction is to allow for the two-dimensional partition of quantum circuits for improved optimization results using the intermediate representation of Liu et al. [56]. Replacing the SA approach that changes the subcircuit partition of Chen et al. [15] with an tree search could further improve the results.

Architecture-Awareness Quantum computing architectures fundamentally differ from each other and offer different advantages and limitations that quantum circuit optimization needs to take into account. There are four dominant quantum architectures currently considered by ZX-based quantum circuit optimization: (i) super-

conducting, (ii) trapped-ion, (iii) neutral-atom, and (iv) photonic quantum architectures.

To create executable quantum circuits for superconducting quantum computers spatial information (topology), qubit connectivity, noise, and error correction. Some work focuses on architecture-aware synthesis of phase gadgets and polynomials (e.g., van de Griend and Duncan [79], Gogioso and Yeung [39], and Winderl et al. [85]) that takes an architecture's topology into account, others (e.g., Kissinger and van de Griend [52] aim to restrict qubit connectivity). However, there is no unified framework that combines the different methods. Transpilation would be more efficient if ZX-diagram optimization could target a specific architecture without the need for additional optimization steps to consider decoherence, gate error, routing, and error correction. Phase gadgets are the native representation of multi-qubit gates in ZX-calculus, therefore improving the architecture and topology aware synthesis is a promising field for future research.

ZX-calculus is a natural candidate for photonic quantum computing because the architecture's measurement graph corresponds to a graph-like (Definition 4.1) ZX-diagram. This equivalence permits the modification of the measurement graph using rewriting rules. As the measurement graph can be directly converted into hardware instructions, additional optimization steps are not required. The initial work of Zilk et al. [89] demonstrates the effectiveness of ZX-calculus to target photonic quantum computers. Upcoming work could extend the approach of Zilke et al. to use different optimization algorithms for T-gate and Clifford gate elimination. Especially the RL-approach of Mattick et al. [59] seems to be a promising candidate for photonic architectures, as it balances the quality and exploration of the solution. As photonic architectures do not require circuit extraction, the quality of the solution is not impacted by post-processing and circuit extraction.

The standout features of trapped-ion quantum computers are the all-to-all qubit connectivity and the use of global gates. Villoria et al. [82] modify the circuit extraction algorithm of Backens et al. [6] to only extract vertices that take part in the same global gate. Phase gadgets are the ZX-calculus equivalent of multi-qubit gates. In the future, work could use the notion of phase gadgets during circuit extraction to improve global gate count and grouping. Furthermore, there is no dedicated ZX-based optimization strategy that targets trap-ion architectures outside of circuit extraction.

Circuit Extraction A significant limitation of ZX-based quantum circuit optimization is the computational cost of circuit extraction. In the general case, circuit extraction is #P-hard [27, 60]. Although polynomial-time algorithms exist for ZX-diagrams that preserve the graph-theoretic conditions of general flow and causal flow, only a small subset of rules has been proven to preserve these flow properties. In addition, verifying the presence of a flow is computationally expensive, with causal flow being less demanding than general flow. Especially noteworthy are the extraction algorithms of Duncan et al. [32] and its extension by Backens et al. [6] that form the basis for various architecture-aware synthesis algorithms (e.g., Kissinger and van de Griend [52], Villoria et al. [82], and Staudacher et al. [76]).

We established that many approaches disregard intermediate ZX-diagrams without the presence of causal flow, general flow or Pauli flow, effectively ignoring large parts of the state-space that might contain the optimal solution. Future research should focus on methods that explore intermediate ZX-diagrams without preserving flow properties while keeping the overhead introduced by circuit extraction at a minimum. This can be achieved by improving or avoiding circuit extraction as much as possible.

Based on Quanz et al. [65] subsequent work should aim to improve parallel circuit extraction to speed up other state-space exploration algorithms.

A promising research direction is to replace the Gaussian elimination algorithm of the biadjacency matrix during circuit extraction by a LP. Similarly to Villoria et al. [82], future LP formulation could include architectural constraints. Circuit extraction is an iterative process, and the biadjacency matrix only captures the connectivity of the current frontier. Therefore, the LP only encodes current information, and it is not guaranteed that the optimal solution of the current LP results in the best global solution. Upcoming work should combine LP-based circuit extraction with a backtracking algorithm that allows

to prune LP solutions that result in unfavorable quantum circuits. Another way of providing context for LP-based circuit extraction, is to provide information of the closest already extracted frontier gates for each qubit.

It is important to recognize that the circuit properties strongly depend on the circuit extraction algorithm itself. Current circuit extraction algorithms replicate spider connectivity by two-qubit gates, potentially increasing the circuit depth and two-qubit gate count. Improvements of circuit extraction algorithms, both in computational efficiency and in circuit quality, should allow for better optimization results. In particular, only the work of Villoria et al. [82] treats circuit extraction as a combinatorial problem. Consequently, efficient formulations beyond LP, constraints, and solvers for the circuit extraction problem provide a vast field for future endeavors.

Many non-ad-hoc approaches require regular circuit extraction (e.g., Fischbach et al. [36], Ewen et al. [34], and Riu et al. [67]). A critical challenge to avoid circuit extraction is that many characteristics of quantum circuits, such as the two-qubit gate count or circuit depth, are not native concepts of ZX-diagrams. Investigating other approximations of quantum circuit metrics at the ZX-diagram level, such as Hadamard wires serving as a proxy for two-qubit gates [75], might reduce the amount of circuit extraction required. A promising first step could be the construction of a surrogate model for circuit extraction that can be quickly evaluated by different approaches.

Multi-Objective Optimization The approaches presented in this survey are designed to primarily target one metric. Nevertheless, it does not suffice to optimize one metric alone to capture the complexity of current quantum computing architectures. Some works follow a lexicographic approach in which one metric after another is improved. A typical example is to first run the elimination of the T-gate by Duncan et al. [32], Kissinger and van de Wetering [53] and then optimize the two-qubit gate count on the resulting ZX-diagram (e.g. Staudacher et al. [75], Holker [46], Fagan and Duncan [35]).

The work of Ewen et al. [34] suggests that the best ZX-diagram results in a circuit close to its best known quantum circuit counterpart for circuit depth and two-qubit gate count. However, on the non-target metrics, the best circuits perform worse than their ZX-based counterparts.

So far, there exist no deliberate multi-objective optimization approaches applied to ZX-based quantum circuit optimization that aim to find a trade-off between fundamentally different metrics. Multi-objective optimization seems to be a promising candidate for architecture-aware optimization where a tradeoff between independent metrics and different architectural specifications, such as qubit connectivity and spatial dimensions, is required. Bridging the gap towards multi-objective optimization would greatly simplify the transpilation pipeline, resulting into an integrated and potentially less computationally demanding framework.

ZX-Diagram Feature Encoding Despite many advances in ZX-diagram optimization and circuit extraction techniques, a fundamental question remains: what features of a ZX-diagram accurately describe the quality of the resulting quantum circuit? While some connections between the ZX-diagram and the quantum circuit are trivial, e.g., the number of $\frac{\pi}{4}$-phase spiders directly translates to the number of T-gates, other features are only an approximation or not translated at all. The example we saw before was that the number of Hadamard edges and the overall spider connectivity serve as a proxy to estimate the two-qubit gate count [75].

Further research on the mapping between ZX-diagram characteristics and quantum circuit properties could enable the development of better optimization methods that explicitly account for architectural constraints. Especially the inclusion cumulative gate error rates, qubit connectivity limitations, and coherence time would alleviate the need of a full transpilation pipeline. Such mappings could serve as a surrogate model that allows quicker heuristic or learning-based optimization without repeatedly requiring circuit extraction and transpilation to assess the quality of the solution.

Furthermore, we propose the systematic addition of characteristics that form a composite metric that aggregates information from features derived from the ZX-diagram, properties of the logical quantum circuit, and characteristics of the final transpiled and executable circuit. Investi-

gating such composite metrics allows to dynamically evaluate characteristics based on the current solution quality and computational cost. Such approaches would allow identifying the relative importance of individual features in determining overall circuit quality. For example, it is unnecessary to take into account accumulated gate error and routing if the initial features already indicate an unfeasible solution.

This adaptive composite metric would allow excluding or adjust certain optimization steps based on the balance between the expected quality of the solution and the computational cost of fully evaluating that metric. By quantifying this trade-off, the composite metric could be added to many methods that require a cost function while also championing a full compilation workflow without being explicitly designed for it.

13 Summary

We provided a survey of quantum circuit optimization using ZX-calculus, with an emphasis on optimization techniques and target metrics. The surveyed works demonstrate that ZX-calculus offers a powerful, compact, and universal framework that enables optimizations beyond traditional circuit-level techniques. Furthermore, we identified that the adoption of ZX-based methods is impeded by scalability issues, a reliance on single-objective heuristics, and the computational cost of circuit extraction.

Several promising research directions emerge: Future work must expand beyond single-metric optimization and adopt multi-objective optimization that jointly consider architecture-independent and architecture-aware metrics. Additionally, there is a clear need to link ZX-diagram and quantum circuit characteristics, potentially through surrogate models or composite metrics that alleviate the computational cost of circuit extraction. Furthermore, circuit extraction could be improved by leveraging combinatorial optimization with the potential inclusion of architectural constraints. Finally, more approaches should take into account the underlying quantum computing architecture. Especially trapped-ion, neutral-atom, and photonic devices are underrepresented.

Overall, ZX-calculus is positioned as an intermediate representation for circuit optimization, that is capable of bridging diagrammatic reasoning with architectural constraints of current and future quantum hardware. Improvements of heuristics, scalability, and explainable learning-based methods are necessary to design algorithms that handle efficiently the non-terminating nature of ZX-calculus.

In summary, ZX-calculus is a candidate for an integrated framework that allows architecture-independent and architecture-aware quantum circuit optimization for current and future quantum computing devices.

References

[1] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, November 2004. DOI: 10.1103/PhysRevA.70.052328.

[2] Matthew Amy, Dmitri Maslov, and Michele Mosca. Polynomial-Time T-Depth Optimization of Clifford+T Circuits Via Matroid Partitioning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(10):1476–1489, October 2014. ISSN 1937-4151. DOI: 10.1109/TCAD.2014.2341953.

[3] Howard Anton. Elementary linear algebra, October 2013.

[4] Miriam Backens. The ZX-calculus is complete for stabilizer quantum mechanics. *New Journal of Physics*, 16(9):093021, September 2014. ISSN 1367-2630. DOI: 10.1088/1367-2630/16/9/093021.

[5] Miriam Backens. Making the stabilizer ZX-calculus complete for scalars. *Electronic Proceedings in Theoretical Computer Science*, 195:17–32, November 2015. ISSN 2075-2180. DOI: 10.4204/EPTCS.195.2.

[6] Miriam Backens, Hector Miller-Bakewell, Giovanni de Felice, Leo Lobski, and John van de Wetering. There and back again: A circuit extraction tale. *Quantum*, 5: 421, March 2021. ISSN 2521-327X. DOI: 10.22331/q-2021-03-25-421. URL `https://doi.org/10.22331/q-2021-03-25-421`.

[7] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computa-

tion. *Physical Review A*, 52(5):3457–3467, November 1995. DOI: 10.1103/PhysRevA.52.3457.

[8] J. S. Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1:195–200, Nov 1964. DOI: 10.1103/PhysicsPhysiqueFizika.1.195. URL https://link.aps.org/doi/10.1103/PhysicsPhysiqueFizika.1.195.

[9] F. Bloch. Nuclear induction. *Phys. Rev.*, 70:460–474, Oct 1946. DOI: 10.1103/PhysRev.70.460. URL https://link.aps.org/doi/10.1103/PhysRev.70.460.

[10] Max Born. Quantenmechanik der stoßvorgänge. *Zeitschrift für Physik*, 38(11):803–827, 1926. DOI: 10.1007/BF01397184.

[11] Sergey B Bravyi and A Yu Kitaev. Quantum codes on a lattice with boundary. *arXiv preprint quant-ph/9811052*, 1998. DOI: 10.48550/arXiv.quant-ph/9811052.

[12] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, January 2009. ISSN 1745-2481. DOI: 10.1038/nphys1157.

[13] Daniel E Browne, Elham Kashefi, Mehdi Mhalla, and Simon Perdrix. Generalized flow and determinism in measurement-based quantum computation. *New Journal of Physics*, 9(8):250, aug 2007. DOI: 10.1088/1367-2630/9/8/250. URL https://dx.doi.org/10.1088/1367-2630/9/8/250.

[14] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. Roads towards fault-tolerant universal quantum computation. *Nature*, 549(7671):172–179, September 2017. ISSN 1476-4687. DOI: 10.1038/nature23460.

[15] Kai Chen, Wen Liu, GuoSheng Xu, Yangzhi Li, Maoduo Li, and Shouli He. Quantum circuit optimization based on dynamic grouping and zx-calculus for reducing 2-qubit gate count, 2025. URL https://arxiv.org/abs/2507.14434.

[16] Bob Coecke and Ross Duncan. Interacting Quantum Observables. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 298–310, Berlin, Heidelberg, 2008. Springer. ISBN 978-3-540-70583-3. DOI: 10.1007/978-3-540-70583-3_25.

[17] Bob Coecke and Ross Duncan. Interacting quantum observables: Categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, April 2011. ISSN 1367-2630. DOI: 10.1088/1367-2630/13/4/043016.

[18] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, Cambridge, 2017. ISBN 978-1-107-10422-8. DOI: 10.1017/9781316219317.

[19] Christophe Couteau. Quantum computing using photons. *Eur. Phys. J. A*, 61(4), April 2025. DOI: 10.1140/epja/s10050-025-01517-5.

[20] Alexander Cowtan, Silas Dilkes, Ross Duncan, Alexandre Krajenbrink, Will Simmons, and Seyon Sivarajah. On the Qubit Routing Problem. In Wim van Dam and Laura Mančinska, editors, *14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2019)*, volume 135 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:32, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-112-2. DOI: 10.4230/LIPIcs.TQC.2019.5. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.TQC.2019.5.

[21] Alexander Cowtan, Silas Dilkes, Ross Duncan, Will Simmons, and Seyon Sivarajah. Phase Gadget Synthesis for Shallow Circuits. *Electronic Proceedings in Theoretical Computer Science*, 318:213–228, May 2020. ISSN 2075-2180. DOI: 10.4204/EPTCS.318.13.

[22] Alexander Cowtan, Will Simmons, and Ross Duncan. A generic compilation strategy for the unitary coupled cluster ansatz, 2020. URL https://arxiv.org/abs/2007.10515.

[23] Vincent Danos and Elham Kashefi. Determinism in the one-way model. *Phys. Rev. A*, 74:052310, Nov 2006. DOI: 10.1103/PhysRevA.74.052310. URL https://link.aps.org/doi/10.1103/PhysRevA.74.052310.

[24] Vincent Danos, Elham Kashefi, Prakash Panangaden, and Simon Perdrix. Extended measurement calculus, 2008.

[25] Niel de Beaudrap, Xiaoning Bian, and Quanlong Wang. Fast and Effective Techniques for T-Count Reduction via Spider Nest Identities. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*, volume 158 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:23, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-146-7. DOI: 10.4230/LIPIcs.TQC.2020.11. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.TQC.2020.11.

[26] Niel de Beaudrap, Xiaoning Bian, and Quanlong Wang. Techniques to reduce $\pi/4$-parity-phase circuits, motivated by the zx calculus. *Electronic Proceedings in Theoretical Computer Science*, 318:131–149, May 2020. ISSN 2075-2180. DOI: 10.4204/eptcs.318.9. URL http://dx.doi.org/10.4204/EPTCS.318.9.

[27] Niel de Beaudrap, Aleks Kissinger, and John van de Wetering. Circuit Extraction for ZX-Diagrams Can Be #P-Hard. In Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 119:1–119:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-235-8. DOI: 10.4230/LIPIcs.ICALP.2022.119. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2022.119.

[28] D. Dieks. Communication by epr devices. *Physics Letters A*, 92(6): 271–272, 1982. ISSN 0375-9601. DOI: https://doi.org/10.1016/0375-9601(82)90084-6. URL https://www.sciencedirect.com/science/article/pii/0375960182900846.

[29] P. a. M. Dirac. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35(3):416–418, July 1939. ISSN 1469-8064, 0305-0041. DOI: 10.1017/S0305004100021162.

[30] Lucas Dixon and Aleks Kissinger. Open-graphs and monoidal theories. *Mathematical Structures in Computer Science*, 23(2):308–359, 2013. DOI: 10.1017/S0960129512000138.

[31] Ross Duncan and Simon Perdrix. Graph States and the Necessity of Euler Decomposition. In Klaus Ambos-Spies, Benedikt Löwe, and Wolfgang Merkle, editors, *Mathematical Theory and Computational Practice*, Lecture Notes in Computer Science, pages 167–177, Berlin, Heidelberg, 2009. Springer. ISBN 978-3-642-03073-4. DOI: 10.1007/978-3-642-03073-4_18.

[32] Ross Duncan, Aleks Kissinger, Simon Perdrix, and John van de Wetering. Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus. *Quantum*, 4:279, June 2020. DOI: 10.22331/q-2020-06-04-279.

[33] A. Einstein, B. Podolsky, and N. Rosen. Can Quantum-Mechanical Description of Physical Reality Be Considered Complete? *Physical Review*, 47(10):777–780, May 1935. DOI: 10.1103/PhysRev.47.777.

[34] Tom Ewen, Ivica Turkalj, Patrick Holzer, and Mark-Oliver Wolf. Application of zx-calculus to quantum architecture search. *Quantum Machine Intelligence*, 7(1), March 2025. ISSN 2524-4914. DOI: 10.1007/s42484-025-00264-6. URL http://dx.doi.org/10.1007/s42484-025-00264-6.

[35] Andrew Fagan and Ross Duncan. Optimising Clifford Circuits with Quantomatic. *Electronic Proceedings in Theoretical Computer Science*, 287:85–105, January 2019. ISSN 2075-2180. DOI: 10.4204/EPTCS.287.5.

[36] Tobias M. Fischbach, Pierre Talbot, and Pascal Bouvry. Exhaustive search for quantum circuit optimization using zx calculus. In Bernabe Dorronsoro, El-Ghazali Talbi, Dua Weraikat, and Sadok Bouamama, editors, *Optimization and Learning*, pages 239–253, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-13589-6. DOI: https://doi.org/10.1007/978-3-032-13589-6_18.

[37] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale

quantum computation. *arXiv.org*, August 2012. DOI: 10.1103/PhysRevA.86.032324.

[38] Sukhpal Singh Gill, Adarsh Kumar, Harvinder Singh, Manmeet Singh, Kamalpreet Kaur, Muhammad Usman, and Rajkumar Buyya. Quantum computing: A taxonomy, systematic review and future directions. *Software: Practice and Experience*, 52(1):66–114, 2022. DOI: https://doi.org/10.1002/spe.3039. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.3039.

[39] Stefano Gogioso and Richie Yeung. Annealing optimisation of mixed zx phase circuits. In Stefano Gogioso and Matty Hoban, editors, Proceedings 19th International Conference on *Quantum Physics and Logic,* Wolfson College, Oxford, UK, 27 June - 1 July 2022, volume 394 of *Electronic Proceedings in Theoretical Computer Science*, pages 415–431. Open Publishing Association, 2023. DOI: 10.4204/EPTCS.394.20.

[40] Daniel Gottesman. The heisenberg representation of quantum computers, 1998. URL https://arxiv.org/abs/quant-ph/9807006.

[41] Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation, 2009. URL https://arxiv.org/abs/0904.2557.

[42] David J. Griffiths. *Introduction to Quantum Mechanics*. Pearson International Edition. Pearson Prentice Hall, Upper Saddle River, NJ London, 2. ed edition, 2005. ISBN 978-0-13-111892-8 978-0-13-191175-8. DOI: https://doi.org/10.1017/9781316995433.

[43] Vikas Hassija, Vinay Chamola, Adit Goyal, Salil S. Kanhere, and Nadra Guizani. Forthcoming applications of quantum computing: Peeking into the future. *IET Quantum Communication*, 1(2):35–41, 2020. ISSN 2632-8925. DOI: 10.1049/iet-qtc.2020.0026.

[44] Nicolas Heurtel, Andreas Fyrillas, Grégoire de Gliniasty, Raphaël Le Bihan, Sébastien Malherbe, Marceau Pailhas, Eric Bertasi, Boris Bourdoncle, Pierre-Emmanuel Emeriau, Rawad Mezher, Luka Music, Nadia Belabas, Benoît Valiron, Pascale Senellart, Shane Mansfield, and Jean Senellart. Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing. *Quantum*, 7:931, February 2023. ISSN 2521-327X. DOI: 10.22331/q-2023-02-21-931. URL https://doi.org/10.22331/q-2023-02-21-931.

[45] Luke E Heyfron and Earl T Campbell. An efficient quantum compiler that reduces t count. *Quantum Science and Technology*, 4 (1):015004, sep 2018. DOI: 10.1088/2058-9565/aad604. URL https://dx.doi.org/10.1088/2058-9565/aad604.

[46] Calum Holker. Causal flow preserving optimisation of quantum circuits in the ZX-calculus, January 2024.

[47] Qunsheng Huang, David Winderl, Arianne Meijer-van de Griend, and Richie Yeung. Redefining lexicographical ordering: Optimizing pauli string decompositions for quantum compiling. In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 01, pages 885–896, 2024. DOI: 10.1109/QCE60285.2024.00108.

[48] Toshinari Itoko, Rudy Raymond, Takashi Imamichi, and Atsushi Matsuo. Optimization of quantum circuit mapping using gate transformation and commutation. *Integration*, 70:43–50, 2020. ISSN 0167-9260. DOI: https://doi.org/10.1016/j.vlsi.2019.10.004. URL https://www.sciencedirect.com/science/article/pii/S0167926019302755.

[49] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A Complete Axiomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 559–568, New York, NY, USA, July 2018. Association for Computing Machinery. ISBN 978-1-4503-5583-4. DOI: 10.1145/3209108.3209131.

[50] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Diagrammatic Reasoning beyond Clifford+T Quantum Mechanics. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 569–578, New York, NY, USA, July 2018. Association for Computing Machinery. ISBN 978-1-4503-5583-4. DOI: 10.1145/3209108.3209139.

[51] Krishnageetha Karuppasamy, Varun Puram, Stevens Johnson, and Johnson P. Thomas. A comprehensive review of quantum circuit

optimization: Current trends and future directions. *Quantum Reports*, 7(1):2, January 2025. ISSN 2624-960X. DOI: 10.3390/quantum7010002. URL http://dx.doi.org/10.3390/quantum7010002.

[52] Aleks Kissinger and Arianne Meijer van de Griend. Cnot circuit extraction for topologically-constrained quantum memories, 2019. URL https://arxiv.org/abs/1904.00633.

[53] Aleks Kissinger and John van de Wetering. Reducing the number of non-clifford gates in quantum circuits. *Phys. Rev. A*, 102:022406, Aug 2020. DOI: 10.1103/PhysRevA.102.022406. URL https://link.aps.org/doi/10.1103/PhysRevA.102.022406.

[54] Aleks Kissinger and John van de Wetering. PyZX: Large Scale Automated Diagrammatic Reasoning. *Electronic Proceedings in Theoretical Computer Science*, 318: 229–241, May 2020. ISSN 2075-2180. DOI: 10.4204/EPTCS.318.14.

[55] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Phys. Rev. A*, 55:900–911, Feb 1997. DOI: 10.1103/PhysRevA.55.900. URL https://link.aps.org/doi/10.1103/PhysRevA.55.900.

[56] Yuxiang Liu, Zaichen Zhang, Yi Hu, Fanxu Meng, Tian Luan, Xianchao Zhang, and Xutao Yu. Practical circuit optimization algorithm for quantum simulation based on template matching. *Quantum Inf. Process.*, 23(2), February 2024. DOI: https://doi.org/10.1007/s11128-023-04252-2.

[57] Hugo Daniel Macedo. Gaussian elimination is not optimal, revisited. *Journal of Logical and Algebraic Methods in Programming*, 85(5, Part 2):999–1010, 2016. ISSN 2352-2208. DOI: https://doi.org/10.1016/j.jlamp.2016.06.003. URL https://www.sciencedirect.com/science/article/pii/S2352220816300529. Articles dedicated to Prof. J. N. Oliveira on the occasion of his 60th birthday.

[58] Saunders MacLane. Categorical algebra. *Bull. New Ser. Am. Math. Soc.*, 71(1):40–106, 1965. DOI: 10.1090/S0002-9904-1965-11234-4.

[59] Alexander Mattick, Maniraman Periyasamy, Christian Ufrecht, Abhishek Y. Dubey, Christopher Mutschler, Axel Plinge, and Daniel D. Scherer. Optimizing quantum circuits via zx diagrams using reinforcement learning and graph neural networks, 2025. URL https://arxiv.org/abs/2504.03429.

[60] Piotr Mitosek. Constructing $NP^{\#P}$-complete problems and #P-hardness of circuit extraction in phase-free ZH. *arXiv preprint arXiv:2404.10913*, 2024. DOI: 10.48550/arXiv.2404.10913.

[61] Anthony Munson, Bob Coecke, and Quanlong Wang. And-gates in zx-calculus: Spider nest identities and qbc-completeness. *Electronic Proceedings in Theoretical Computer Science*, 340:230–255, September 2021. ISSN 2075-2180. DOI: 10.4204/eptcs.340.12. URL http://dx.doi.org/10.4204/EPTCS.340.12.

[62] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, December 2010. ISBN 978-1-139-49548-6. DOI: 10.1017/CBO9780511976667.

[63] Maximilian Nägele and Florian Marquardt. Optimizing zx-diagrams with deep reinforcement learning. *Machine Learning: Science and Technology*, 5(3):035077, sep 2024. DOI: 10.1088/2632-2153/ad76f7. URL https://dx.doi.org/10.1088/2632-2153/ad76f7.

[64] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2: 79, August 2018. ISSN 2521-327X. DOI: 10.22331/q-2018-08-06-79. URL https://doi.org/10.22331/q-2018-08-06-79.

[65] Marcel Quanz, Korbinian Staudacher, and Karl Furlinger. Parallel Quantum Circuit Extraction from MBQC-Patterns . In *2024 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 1078–1087, Los Alamitos, CA, USA, May 2024. IEEE Computer Society. DOI: 10.1109/IPDPSW63119.2024.00179. URL https://doi.ieeecomputersociety.org/10.1109/IPDPSW63119.2024.00179.

[66] Robert Raussendorf and Hans J. Briegel. A One-Way Quantum Computer. *Physical Review Letters*, 86(22):5188–5191, May 2001. DOI: 10.1103/PhysRevLett.86.5188.

[67] Jordi Riu, Jan Nogué, Gerard Vilaplana, Artur Garcia-Saez, and Marta P. Estarellas. Reinforcement Learning Based Quantum Circuit Optimization via ZX-Calculus. *Quantum*, 9:1758, May 2025. ISSN 2521-327X. DOI: 10.22331/q-2025-05-28-1758. URL https://doi.org/10.22331/q-2025-05-28-1758.

[68] Benjamin Schumacher. Quantum coding. *Physical Review A*, 51(4):2738–2747, April 1995. DOI: 10.1103/PhysRevA.51.2738.

[69] Vivek V. Shende, Aditya K. Prasad, Igor L. Markov, and John P. Hayes. Reversible Logic Circuit Synthesis, February 2003.

[70] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995. DOI: 10.1103/PhysRevA.52.R2493. URL https://link.aps.org/doi/10.1103/PhysRevA.52.R2493.

[71] P.W. Shor. Fault-tolerant quantum computation. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 56–65, 1996. DOI: 10.1109/SFCS.1996.548464.

[72] Will Simmons. Relating Measurement Patterns to Circuits via Pauli Flow. In Chris Heunen and Miriam Backens, editors, *Proceedings 18th International Conference on Quantum Physics and Logic, Gdansk, Poland, and online, 7-11 June 2021*, volume 343 of *Electronic Proceedings in Theoretical Computer Science*, pages 50–101. Open Publishing Association, 2021. DOI: 10.4204/EPTCS.343.4.

[73] Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross Duncan. t|ket⟩: a retargetable compiler for nisq devices. *Quantum Science and Technology*, 6(1):014003, nov 2020. DOI: 10.1088/2058-9565/ab8e92. URL https://dx.doi.org/10.1088/2058-9565/ab8e92.

[74] Robert S. Smith, Michael J. Curtis, and William J. Zeng. A practical quantum instruction set architecture, 2017. URL https://arxiv.org/abs/1608.03355.

[75] Korbinian Staudacher, Tobias Guggemos, Sophia Grundner-Culemann, and Wolfgang Gehrke. Reducing 2-QuBit Gate Count for ZX-Calculus based Quantum Circuit Optimization. In *Electronic Proceedings in Theoretical Computer Science*, volume 394, pages 29–45, November 2023. DOI: 10.4204/EPTCS.394.3.

[76] Korbinian Staudacher, Ludwig Schmid, Johannes Zeiher, Robert Wille, and Dieter Kranzlmüller. Multi-controlled phase gate synthesis with zx-calculus applied to neutral atom hardware. *Electronic Proceedings in Theoretical Computer Science*, 406:96–116, August 2024. ISSN 2075-2180. DOI: 10.4204/eptcs.406.5. URL http://dx.doi.org/10.4204/EPTCS.406.5.

[77] Andrew Steane. Multiple-particle interference and quantum error correction. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 452 (1954):2551–2577, 11 1996. ISSN 1364-5021. DOI: 10.1098/rspa.1996.0136. URL https://doi.org/10.1098/rspa.1996.0136.

[78] Volker Strassen. Gaussian elimination is not optimal, August 1969.

[79] Arianne van de Griend and Ross Duncan. Architecture-aware synthesis of phase polynomials for nisq devices. *Electronic Proceedings in Theoretical Computer Science*, 394:116–140, November 2023. ISSN 2075-2180. DOI: 10.4204/eptcs.394.8. URL http://dx.doi.org/10.4204/EPTCS.394.8.

[80] John van de Wetering. ZX-calculus for the working quantum computer scientist, December 2020.

[81] Vivien Vandaele. Qubit-count optimization using zx-calculus, 2024. URL https://arxiv.org/abs/2407.10171.

[82] Alejandro Villoria, Henning Basold, and Alfons Laarman. Optimisation and synthesis of quantum circuits with global gates. *Quantum Science and Technology*, 11(1):015040, jan 2026. DOI: 10.1088/2058-9565/ae3029. URL https://doi.org/10.1088/2058-9565/ae3029.

[83] Tzu-Chieh Wei. Measurement-based quantum computation, 03 2021. URL https://oxfordre.com/physics/view/10.1093/acrefore/9780190871994.001.0001/acrefore-9780190871994-e-31.

[84] David Winderl, Qunsheng Huang, and Christian B. Mendl. A recursively parti-

tioned approach to architecture-aware zx polynomial synthesis and optimization. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, page 837–847. IEEE, September 2023. DOI: 10.1109/qce57702.2023.00098. URL http://dx.doi.org/10.1109/QCE57702.2023.00098.

[85] David Winderl, Qunsheng Huang, and Christian B. Mendl. A recursively partitioned approach to architecture-aware ZX Polynomial synthesis and optimization. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 837–847, September 2023. DOI: 10.1109/QCE57702.2023.00098.

[86] W K Wootters and W H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, October 1982. DOI: 10.1038/299802a0.

[87] Ge Yan, Wenjie Wu, Chen Yuheng, Kaisen Pan, Xudong Lu, Zhou Zixiang, Wang Yuhan, Ruocheng Wang, and Junchi Yan. Quantum circuit synthesis and compilation optimization: Overview and prospects. *CoRR*, abs/2407.00736, 2024. DOI: 10.48550/ARXIV.2407.00736. URL https://doi.org/10.48550/arXiv.2407.00736.

[88] Chenghong Zhu, Xian Wu, Zhaohui Yang, Jingbo Wang, Anbang Wu, Shenggen Zheng, and Xin Wang. Quantum compiler design for qubit mapping and routing: A cross-architectural survey of superconducting, trapped-ion, and neutral atom systems, 2025. URL https://arxiv.org/abs/2505.16891.

[89] Felix Zilk, Korbinian Staudacher, Tobias Guggemos, Karl Fürlinger, Dieter Kranzlmüller, and Philip Walther. A compiler for universal photonic quantum computers. In *2022 IEEE/ACM Third International Workshop on Quantum Computing Software (QCS)*, pages 57–67, 2022. DOI: 10.1109/QCS56647.2022.00012.