Seminar on Selected Topics in Quantum Computing

Xiao-Ting Michelle To
Florian Krötz
Korbinian Staudacher
Prof. Dr. Dieter Kranzlmüller

Winter term 2025/26

Contents

Quantum Computing vs. Firewalls: When Complexity Saves the Horizon

Adrian Mülthaler
LMU Munich
Munich, Germany
adrian.muelthaler@campus.lmu.de

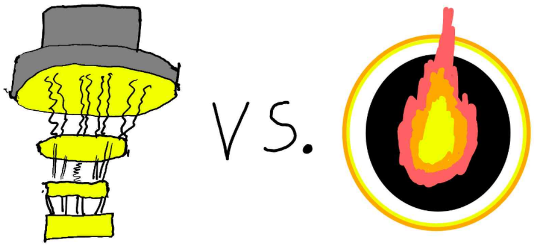
***Abstract*—The black hole firewall paradox, proposed by Almheiri, Marolf, Polchinski, and Sully, exposes a fundamental tension between the principles of quantum mechanics and the classical description of black hole event horizons. In response, Harlow and Hayden argued that limitations arising from quantum computational complexity render the firewall paradox thought experiment operationally unrealizable, thereby offering a potential resolution of the paradox. This seminar paper is a pedagogical review and synthesis of these ideas rather than a presentation of new results, providing a clear and accessible introduction to the firewall paradox and the Harlow–Hayden complexity argument. The paper is aimed at readers with a background in quantum computing but limited familiarity with black hole physics.**

***Index Terms*—Firewall Paradox, Quantum Computing, Black Holes**

I. Introduction

The firewall paradox, in black hole physics, emphasizes the profound complexities and unresolved questions surrounding the nature of black holes within the framework of quantum gravity. This paradox arises from a tension between fundamental principles of quantum mechanics and the classical understanding of black holes, which assumes that the event horizon, the boundary separating the interior of a black hole from the outside universe, is smooth and ordinary. In their 2012 paper by Almheiri, Marolf, Polchinski, and Sully [1], henceforth referred to as AMPS, showed that under certain reasonable assumptions about quantum theory, this smooth horizon cannot coexist with the requirement that information be preserved during black hole evaporation, leading to the controversial suggestion that an observer falling into a black hole would encounter a "firewall" of high-energy radiation, rather than experiencing a smooth passage through the event horizon.

Since its introduction, the AMPS argument has motivated numerous attempts within the physics community to question its conclusions or to find a way around them without abandoning empirically well-established principles of quantum theory, general relativity, or statistical thermodynamics. One such attempt was proposed by Harlow and Hayden [2], hereafter referred to as HH, who argue that limitations stemming from quantum computational complexity prevent the AMPS thought experiment from being operationally realizable, even in principle. While the HH argument is well motivated from the perspective of quantum computational complexity, the authors of AMPS responded with a counterargument in their subsequent apologia [3], suggesting that the required computation could instead be carried out in an auxiliary, decoupled quantum system, thereby circumventing the relevant time-complexity constraints.



This paper offers an exposition of the AMPS firewall paradox and the Harlow–Hayden complexity argument, tailored to an audience familiar with quantum computation but new to black hole theory. To accomplish this, the author has drawn schematics and developed a Jupyter Notebook[1] containing additional simulations and visualizations.

The paper begins with an introduction to black holes and their quantum-mechanical description in Section II, providing the necessary foundation for understanding the AMPS argument, which is presented in Section III. Section IV then introduces and examines the HH proposal in detail, followed by a discussion in Section V of the AMPS authors' response to the HH argument. Concluding remarks are presented in Section VI.

II. Description of a Black Hole

This section gives a brief introduction to the structure of black holes. It begins with general remarks, primarily based on Ref. [4], to which the interested reader is referred for a more detailed and formal description.

In general, a black hole is a region of the spacetime where gravity is so strong that nothing, not even light, can escape its gravitational pull. It is an extremely compact object that cannot emit radiation from its surface and is therefore perceived as black. The center of a black hole contains a *spacetime*

[1] The Jupyter Notebook can be found here: https://github.com/nairdanus/QCvsFirewalls/blob/main/main.ipynb and will be referred to as *the Notebook* for the remainder of this paper.

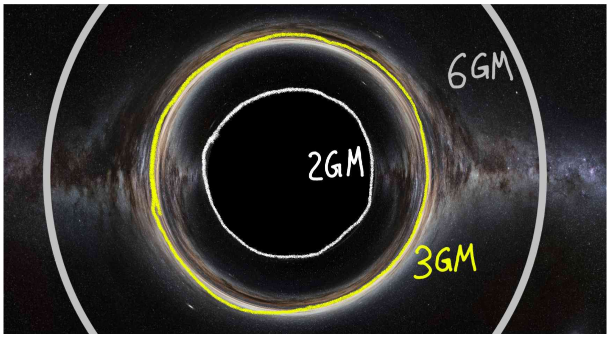


Fig. 1. Schematic illustration of a Schwarzschild black hole, indicating the *event horizon* at $r = 2GM$, the *photon sphere* at $r = 3GM$, and the *ISCO* at $r = 6GM$. The background image is based on a numerical simulation provided by the ESA Advanced Concepts Team. Additional annotations and illustrations were added by the author to improve clarity and accessibility.

singularity, where the classical description of spacetime breaks down [5].

For simplicity reasons, this paper, following HH, focuses on the simplest kind, the *Schwarzschild black hole*.[2] It describes a non-rotating and electrically uncharged black hole and can therefore be fully described by one single parameter, namely its mass M.

The escape velocity, defined as the minimal speed required for an object to escape the gravitational field of a celestial body, is given by

$$v_e = \sqrt{\frac{2GM}{r}}, \tag{1}$$

where G denotes Newton's gravitational constant and M the mass of the primary body. As light travels at a constant speed c, this implies the existence of a critical radius at which the escape velocity equals c. This radius is given by

$$r_H = \frac{2GM}{c^2}. \tag{2}$$

The surface defined by this radius is called the *event horizon*. As nothing can escape from within it, the interior is causally separated from the exterior region. Equation (2) applies specifically for Schwarzschild black holes. For rotating black holes, the event horizon has a smaller radius, depending on the angular momentum.

Since G and c in Equation (2) are constants, it is common to adopt natural units in which $c = 1$, yielding $r_H = 2GM$, or even $c = G = 1$, yielding $r_H = 2M$. The radial coordinate used here is the *Schwarzschild radial coordinate*, which measures the area of spheres centered on the singularity. We follow the convention of HH and set $c = 1$ throughout this work while keeping G explicit. Consequently, the radius of the event horizon will henceforth be written as $r_H = 2GM$.

At a radius $r = 3GM$, just outside the event horizon, lies the *photon sphere* of a Schwarzschild black hole [6]. At this distance, gravity is strong enough that light can orbit the black hole in circular paths. These orbits are unstable: small disturbances cause photons either to fall inward and eventually cross the event horizon, or to move outward and escape the gravitational pull of the black hole. While photons do not remain on these orbits indefinitely, the photon sphere strongly influences how a black hole appears to a distant observer and is responsible for the formation of a bright ring or halo around the black hole.

Similar to the photon sphere, the *innermost stable circular orbit* (ISCO) is the smallest radius at which a test particle with negligible mass can orbit a black hole in a stable circular trajectory. For a Schwarzschild black hole, the ISCO is located at a radius $r = 6GM$. At smaller radii, circular orbits become unstable, and particles inevitably spiral inward toward the event horizon. The ISCO therefore marks the inner edge at which matter can stably orbit the black hole.

The Heisenberg uncertainty principle implies that empty space must contain small, temporary fluctuations in energy.[3] These fluctuations can be described as the spontaneous creation and annihilation of entangled particle–antiparticle pairs occurring on extremely short timescales. This continuous process is known as *quantum vacuum fluctuation*.

Near the event horizon of a black hole, such quantum fluctuations occur as well. However, the gravitational pull on the particle closer to the event horizon is stronger than on its partner, which is slightly farther away. As a result, the pair can be separated before annihilation takes place. In this case, the particle closer to the horizon falls into the black hole, while the other escapes to infinity. The escaping particles become real and constitute what is known as *Hawking radiation*, a steady thermal radiation emitted by the black hole. [7]–[10]

Since this process effectively corresponds to the emission of particles, it must lead to a loss of energy from the black hole. Because the energy of a black hole is given by $E = Mc^2$, a decrease in energy implies a decrease in mass. Over very long timescales, this mechanism causes the black hole to gradually lose mass and eventually *evaporate*.

As Hawking radiation is thermal, the event horizon of a black hole can be treated as a thermodynamic system. This observation leads to the conclusion that the area of the event horizon is proportional to the entropy[4] of the black hole [8], [11]. This means that as the black hole shrinks, its entropy decreases, with the lost entropy being carried away by the radiation it emits. When a black hole has radiated away more than half of its initial entropy, it is referred to as an *old black hole*. The time at which this happens is known as the *Page time*.

III. The Firewall Paradox

The firewall paradox arises from a careful analysis of the quantum properties of Hawking radiation and their implications for the interior structure of an evaporating black

[2] Although HH formulate their argument primarily for Schwarzschild black holes, they also extend their claims to more general black holes in Section 5 of Ref. [2].

[3] The interested reader may find a detailed discussion of the underlying reasoning in Ref. [7].

[4] In thermodynamics, entropy is a measure of the number of microscopic configurations corresponding to a given macroscopic state, or equivalently, a measure of the information inaccessible to a macroscopic observer.

hole. Building on the semiclassical description of black hole evaporation reviewed in the previous section, the argument highlights a tension between three seemingly reasonable assumptions [3], [12]:

1) the *unitarity of quantum mechanics*, which also must hold for the evaporation of the black hole
2) the validity of *effective field theory*[5] outside the event horizon, which states that the immediate outside must be entangled with the immediate inside of the event horizon
3) the *equivalence principle*, and the resulting *No Drama* postulate, which predicts that an infalling observer experiences nothing unusual when crossing the horizon.

AMPS demonstrated that, for an old black hole these assumptions cannot all be simultaneously satisfied. If information is preserved in the Hawking radiation (1), late-time radiation must be strongly entangled with the early radiation. However, effective field theory near the horizon predicts that the same late-time modes are also entangled with interior degrees of freedom (2). This apparent violation of the monogamy of entanglement leads to the firewall paradox: the conclusion that the smooth horizon must be replaced by a region of high-energy excitations encountered by an infalling observer (3).

In the following, we outline the AMPS argument in detail, inspired by Ref. [2], starting with a quantum mechanical description of the interior and exterior of a black hole, each of which will be accompanied by a schematic we have drawn, visualizing the different modes (regions).

A. Outside the Black Hole

The evaporation process of a black hole can be described as a time evolution in which, at any given time, there is a state $|\Psi\rangle$ belonging to the Hilbert space $\mathcal{H}_{outside}$ associated with degrees of freedom accessible to an external observer. This Hilbert space can be factorized into three subfactors,

$$\mathcal{H}_{outside} = \mathcal{H}_H \otimes \mathcal{H}_B \otimes \mathcal{H}_R. \tag{3}$$

The distinction of these subspaces is to some extent arbitrary, but it is chosen by AMPS for convenience and physical relevance. $\mathcal{H}_R$ is the mode of the radiation field outside the black hole, roughly with Schwarzschild coordinates $r > 3GM$. In principle, this space would be infinitely large; however, in this work it is restricted to the subset of radiation degrees of freedom that still participate in the black hole dynamics. $\mathcal{H}_B$ contains the effective field theory modes just outside the event horizon, spanning approximately the region $2GM + \epsilon < r < 3GM$, with ϵ denoting an extremely small distance from the event horizon and serves as a *UV cutoff*[6]. $\mathcal{H}_H$ comprises the remaining degrees of freedom and can be thought of being the *stretched horizon* at $r = 2GM + \epsilon$.

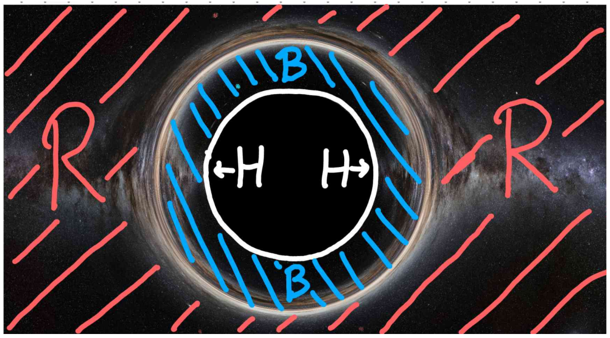


Fig. 2. Schematic illustration of the decomposition of a black hole into different quantum modes (Hilbert spaces) in the AMPS thought experiment, as seen from the perspective of an outside observer.

We call the dimensions of these Hilbert spaces $dim(\mathcal{H}_H) = |H|$, $dim(\mathcal{H}_B) = |B|$, and $dim(\mathcal{H}_R) = |R|$. At the time t at which the state $|\Psi\rangle$ is considered, both $\log_2 |H|$ and $\log_2 |B|$ are proportional to the entropy of the black hole and are therefore also proportional to the area of the event horizon.[7] As the black hole evaporates and loses mass, its entropy decreases, implying that $|H|$ and $|B|$ decrease over time, while $|R|$ increases as radiation accumulates. This observation leads to this natural distinction between young and old black holes:

$$\begin{aligned} \text{young black hole:} \quad & |R| < |H|\,|B|, \\ \textit{Page time}: \quad & |R| = |H|\,|B|, \\ \text{old black hole:} \quad & |R| > |H|\,|B|. \end{aligned} \tag{4}$$

At sufficiently late times, when the black hole is old, $|R| \gg |H|\,|B| \gg 1$ holds. In this regime, *Page's theorem* [13] applies, which states that for a generic pure state of a large quantum system, any sufficiently small subsystem is almost maximally entangled with the remainder of the system and is itself nearly maximally mixed. Applied to the description from Equation (3), this implies that the combined subsystem HB must be almost maximally entangled with a suitable subspace of the radiation R. Consequently, there is a time-dependent decomposition of $\mathcal{H}_R$ into the subspace being entangled with $\mathcal{H}_H$ and the one being entangled with $\mathcal{H}_B$:

$$\mathcal{H}_R = (\mathcal{H}_{R_H} \otimes \mathcal{H}_{R_B}) \oplus \mathcal{H}_{other}, \tag{5}$$

where $|R_H| = |H|$ and $|R_B| = |B|$. The full quantum state of the exterior degrees of freedom can then be written as

$$|\Psi\rangle = \left(\frac{1}{\sqrt{|H|}} \sum_h |h\rangle_H \, |h\rangle_{R_H} \right) \otimes \left(\frac{1}{\sqrt{|B|}} \sum_b |b\rangle_B \, |b\rangle_{R_B} \right), \tag{6}$$

where h and b label orthonormal bases of $\mathcal{H}_H$ and $\mathcal{H}_B$, respectively. The subsystems R_H and R_B are referred to as the *purifications* of H and B. This structure makes explicit that any measurement performed on B is perfectly correlated with a corresponding measurement on R_B.

[5] An effective field theory is a simplifying description of physics that works at chosen scales (e.g., those observable to humans) without the need to know the details of much smaller or higher-energy phenomena (e.g., quantum effects).

[6] A UV (ultraviolet) cutoff describes a short-distance / high-energy limit beyond which a theory is no longer trusted.

[7] The maximal entropy of a quantum system is given by the natural logarithm of the dimension of its Hilbert space. From a quantum-computation perspective, this is analogous to the fact that n qubits span a Hilbert space of dimension 2^n.

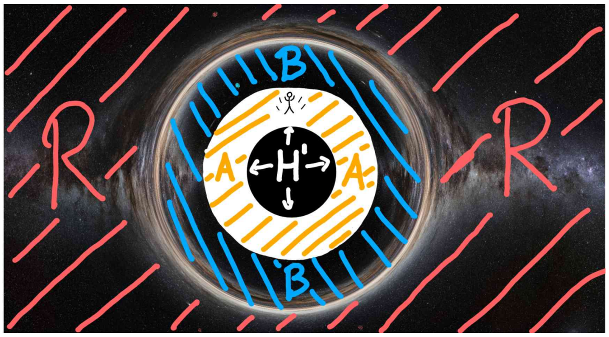


Fig. 3. Schematic illustration of the Hilbert-space decomposition of a black hole in the AMPS thought experiment from the viewpoint of an infalling observer.

B. Inside the Black Hole

If we accept the No Drama postulate, an infalling observer, Alice, can cross the event horizon without experiencing anything unusual. In this subsection, we examine the quantum description of the black hole from Alice's perspective. Although Alice will eventually reach the singularity, it should still be possible for her to get a quantum mechanical description of her immediate surroundings.

Once Alice has passed the horizon, she gains access to a new region of spacetime that is invisible to an outside observer. However, she still retains access to the exterior region, as the horizon is a one-way causal barrier. The Hilbert space from Alice's viewpoint $\mathcal{H}_{inside}$ can be factorized as:

$$\mathcal{H}_{inside} = \mathcal{H}_{H'} \otimes \mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_R. \tag{7}$$

Here, $\mathcal{H}_B$ and $\mathcal{H}_R$ correspond to modes shared with the outside observer, since both regions are causally accessible. $\mathcal{H}_A$ contains the effective field theory modes just inside the horizon, roughly over $GM < r < 2GM - \epsilon$. $\mathcal{H}_{H'}$ represents the remaining degrees of freedom associated with Alice's local horizon[8], which are not crucial for the AMPS argument. The modes of the global horizon $\mathcal{H}_H$ were omitted in this description, as Alice passes through the region $2GM - \epsilon < r < 2GM + \epsilon$ in an extremely short period of time, making it operationally irrelevant.

According to No Drama, Alice expects the regions A and B to form a continuous space across the horizon. In the vacuum of Minkowski space[9] two nearby regions are connected if the corresponding modes are nearly maximally entangled. Therefore, for Alice to experience a smooth horizon, $\mathcal{H}_A$ must be nearly maximally entangled with $\mathcal{H}_B$.

Since both Alice and an outside observer have access to $\mathcal{H}_B$ and $\mathcal{H}_R$, they must agree on the state in this combined subspace. As shown in Subsection III-A, $\mathcal{H}_B$ is maximally entangled with a subspace of the radiation, $\mathcal{H}_{R_B}$, which must hold from Alice's perspective as well.

In quantum mechanics, perfect correlations cannot be freely shared. If a quantum system A is perfectly correlated to system B, namely if they are maximally entangled, then it cannot be entangled to a third system C. [14] This restriction is known as the *monogamy of entanglement*.

Here lies the core of the paradox. The No Drama assumption requires A to be maximally entangled with B, while the outside description requires B to be maximally entangled with R_B. Both cannot be true simultaneously, due to monogamy of entanglement. To resolve this contradiction, AMPS proposed that the entanglement between A and B must not exist. This leads to a "firewall" of high-energy quanta at the horizon, which would destroy any infalling observer attempting to cross it.

C. The Firewall paradox as a Quantum Computation

We now reformulate the Hilbert-space structure of the black hole in terms of a finite number of qubits. This perspective is essential for the computational complexity arguments discussed in Section IV.

Following HH, we introduce a computational basis for the radiation field of the form

$$|bhr\rangle_R = |b_1 \dots b_k, h_1 \dots h_m, r_1 \dots r_{n-k-m}\rangle_R \,. \tag{8}$$

where $k = \log_2 |B|$ is the number of qubits corresponding to the Hilbert space $\mathcal{H}_B$, $m = \log_2 |H|$ is the number of qubits corresponding to $\mathcal{H}_H$, and $n = \log_2 |R|$ is the total number of qubits in $\mathcal{H}_R$. The qubits r_i are early Hawking radiation emitted before Page time and will play no significant role in the following discussion. The qubits b_i and h_i correspond to the subspaces $\mathcal{H}_{R_B}$ and $\mathcal{H}_{R_H}$ and are the purifications of the modes B and H respectively.

Including the degrees of freedom associated with the black hole, and using the fact that BH is entangled with a subspace of the radiation R, the full quantum state of the system can be written as

$$|\Psi\rangle = \frac{1}{\sqrt{|B||H|}} \sum_{b,h} |b\rangle_B \, |h\rangle_H \, U_R \, |bh0\rangle_R \,. \tag{9}$$

Here, the qubits r_i have been set to zero for simplicity, as they correspond to the irrelevant subspace $\mathcal{H}_{\text{other}}$ introduced in Equation (5). U_R denotes some complicated unitary transformation acting on $\mathcal{H}_R$, representing the scrambling of quanta carried by the Hawking radiation. Its precise form depends on the details of quantum gravity as well as on the initial state of the black hole.

In order to verify the entanglement between $\mathcal{H}_B$ and its purification $\mathcal{H}_{R_B}$, Alice would need to identify and apply the inverse unitary $U_R^\dagger$ to the Hawking radiation. Doing so would allow her to extract the subsystem $\mathcal{H}_{R_B}$ from the full Hawking radiation Hilbert space $\mathcal{H}_R$. This task can be seen as a *decoding* operation and can be performed on a quantum computer. This is visualized in the first part of the Notebook.

[8] Alice's horizon differs from the global event horizon because she has crossed the global causal boundary of the black hole, yet remains causally disconnected from regions closer to the singularity.

[9] Minkowski space is a four-dimensional mathematical spacetime that combines three dimensions of space and one of time into a single framework used to describe special relativity.

IV. QC vs. Firewalls

In this section, we present the HH argument, which analyzes the computational complexity of the quantum task that an infalling observer, Alice, would need to perform in order to test for the presence of a firewall. HH argue that the time required to carry out this quantum computation exceeds the remaining lifetime of the black hole. Consequently, although the relevant information may exist in principle within the Hawking radiation, it cannot be extracted in practice. The entanglement between the near-horizon modes and the radiation can never be operationally verified, thereby removing the necessity of introducing firewalls.

The Schwarzschild black hole with mass M has an entropy proportional to M^2 and evaporates in a time proportional to M^3. So M^3 would be the maximal amount of time Alice has to extract the needed information from the radiation.

Subsection IV-A introduces the complexity argument in a more formal way by discussing some fundamental limitations. Subsequently, Subsection IV-B presents a more standard and intuitive formulation of the argument, expressed in terms of quantum circuits and gate decompositions familiar to people with a background in quantum computing.

A. Computation via Natural Unitary Evolution

Suppose Alice has access to a quantum computer of arbitrarily large size, described by a Hilbert space $\mathcal{H}_C$. She may then couple this quantum computer to the Hawking radiation and allow the combined system $\mathcal{H}_R \otimes \mathcal{H}_C$ to evolve under its natural unitary time evolution, denoted by U_{comp}. The goal of this evolution is to decode the radiation and extract the subsystem $\mathcal{H}_{R_B}$.

To achieve this, Alice must carefully choose an initial state $|\Psi\rangle_C$ of her quantum computer such that the joint system evolves as

$$U_{comp} : U_R\,|bh0\rangle_R \otimes |\Psi\rangle_C \mapsto |\text{something}\rangle \otimes |b\rangle_{mem} \tag{10}$$

for all b and h. In this way, the information contained in the subsystem R_B is separated from the remaining degrees of freedom and stored in a designated memory register of the quantum computer.

The central challenge for Alice is therefore to determine an initial state $|\Psi\rangle_C$ that achieves this decoding. Since the group of unitary transformations is continuous, an exact realization of this is physically not achievable. Instead, it suffices for Alice to approximate the desired initial state within some small error ϵ, measured using the *trace norm*. For an operator A, the trace norm is defined as

$$||A||_1 = \text{Tr}(\sqrt{A^\dagger A}). \tag{11}$$

Accordingly, Alice only needs to find a state $|\Psi'\rangle$ such that

$$||(|\Psi\rangle\langle\Psi| - |\Psi'\rangle\langle\Psi'|)||_1 \;\leq\; \epsilon \tag{12}$$

To estimate the difficulty of this task, HH computed the probability that a randomly chosen initial state of the quantum computer lies within trace-norm distance ϵ of a suitable decoding state. They found this probability to be

$$P = \left(\frac{2}{\epsilon}\right)^{-2|C|(|R|2^m(2^k-1)-1)}. \tag{13}$$

The interested reader is referred to Section 3.1 of Ref. [2] for the detailed derivation. For large values of k, m, and $|C|$, this probability becomes astronomically small. Consequently, the expected time required for Alice to find an adequate approximation to $|\Psi\rangle_C$ by chance is extraordinarily large. Notably, increasing the size of the quantum computer only exacerbates this problem.

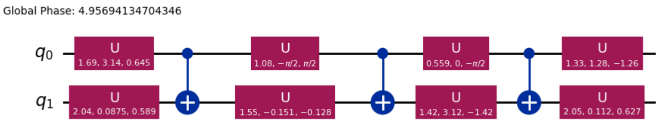


Fig. 4. Example of a gate decomposition of a random unitary transformation acting on two qubits.

This approach can be viewed as the most naive decoding strategy available to Alice. She effectively waits for the system to evolve close to the desired state through its natural dynamics. The time this takes can be interpreted as the *quantum recurrence time*, which characterizes how long it takes for a quantum system to return arbitrarily close to a given state. In the next subsection, we discuss how Alice improves upon this naive strategy by performing a structured quantum computation.

B. Computation via Gate Decomposition

We now discuss how Alice might attempt to decode the scrambled radiation by explicitly implementing the decoding operator $U_R^\dagger$, the inverse of U_R from Equation (9). For simplicity, we assume that $U_R^\dagger$ acts on the entire radiation Hilbert space $\mathcal{H}_R$. In Appendix A we show that it is sufficient for $U_R^\dagger$ to act only on a subspace of dimension $k+m$, and that this restriction does not affect the arguments that follow.

We take U_R to be a generic, highly scrambled unitary transformation without any special structure. Given the chaotic dynamics governing black hole evaporation, it would be surprising if the radiation scrambling resulted in a simple or efficiently describable unitary.

In the gate model of quantum computation, a quantum computer implements unitary transformations by applying a sequence of elementary *quantum gates*. A set of gates is called *universal* if it can implement any unitary transformation. A standard example of a universal gate set is $\{\text{CNOT}, H, T\}$. It is physically unrealistic to assume that a quantum computer can directly implement an arbitrary multi-qubit unitary in a single step. Instead, Alice must decompose $U_R^\dagger$ into a sequence of gates drawn from her universal gate set.

The time complexity of the decoding task can therefore be identified with the number of quantum gates that must be applied sequentially in this decomposition. We make the

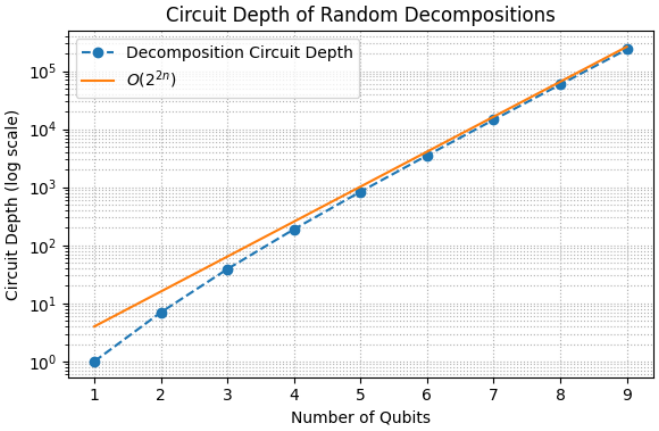


Fig. 5. Circuit depth (on a logarithmic scale) obtained when decomposing random unitary transformations using Quantum Shannon Decomposition, starting from three qubits. As a reference, the curve $O(2^{2n})$ is shown.

extremely optimistic assumption, that Alice possesses a quantum computer capable of implementing arbitrary single-qubit unitaries. There, any multi-qubit unitary can be decomposed into arbitrary one-qubit gates and two-qubit CNOT gates. This decomposition is schematically illustrated in Figure 4.

In the second part of the Notebook we analyzed the complexity of decomposing random unitaries using the Python library `qiskit` [15] (version 2.2.3). For systems of more than two qubits, `qiskit` implements unitary synthesis via the Quantum Shannon Decomposition (QSD) [16]. The QSD algorithm has an asymptotic gate complexity given by

$$f(n) = \frac{23}{48}4^n - \frac{3}{2}2^n + \frac{4}{3}. \qquad (14)$$

which clearly lies in the complexity class $O(4^n) = O(2^{2n})$.

Our numerical results confirm this scaling behavior. As shown in Figure 5, the circuit depth, defined as the minimum number of layers in which gates can be applied in parallel, approaches $O(2^{2n})$ for increasing numbers of qubits. This must be especially the case for black hole radiation, where the number of qubits is extremely large.

Since the number of qubits required to describe the radiation is proportional to the black hole entropy, which itself scales as $S \sim M^2$ (in Planck units), we may express the total decoding time as a function of the black hole mass:

$$t_{\text{decoding}} \sim 2^{2n} = 2^{2M^2}. \qquad (15)$$

As a result, even under the most optimistic assumptions, namely, that Alice possesses an extraordinarily powerful quantum computer capable of performing arbitrary single-qubit operations, and that each gate can be applied in the minimum physically allowed time (on the order of the Planck time), the total decoding time vastly exceeds the black hole evaporation time, which scales only polynomially as $t_{\text{evap}} \sim M^3$. Consequently, Alice cannot complete the decoding before the black hole evaporates, rendering any direct experimental verification of the firewall paradox impossible.

V. The AMPS Apologia

In their apologia [3], AMPS respond to several attempts to circumvent the firewall paradox, including the proposal by HH. Their counterargument addresses HH's claim that computational complexity fundamentally prevents an infalling observer from decoding the purification of a late Hawking mode. Importantly, AMPS do not dispute the decoding requirements or the exponential growth of complexity; instead, they argue that complexity alone does not make the experiment impossible in principle.

The central idea is that the infalling observer does not need to perform the decoding directly. One can introduce an auxiliary quantum system (AUX), entangled with the black hole and its early radiation, and execute the entire decoding unitary within AUX. The observer then receives only the purification of the late mode b, without having to carry out the computation themselves. This construction separates the computational cost, borne entirely by AUX, from the observer's experience, which is constrained by proper time.

AMPS assume that the Hamiltonian of AUX can, in principle, be rescaled arbitrarily:

$$H_{\text{AUX}} \to \Upsilon(t) H_{\text{AUX}}, \quad \Upsilon(t) \gg 1.$$

In standard quantum mechanics, rescaling the Hamiltonian proportionally rescales the rate of time evolution. Consequently, a computation that would ordinarily take time T can be completed in time T/Υ, allowing any finite quantum computation to be executed in arbitrarily short coordinate time. This shows that circuit complexity does not impose an in-principle lower bound on the time required for the decoding.

The disagreement between HH and AMPS is therefore conceptual rather than mathematical. HH argue that high circuit complexity renders the experiment physically impossible under reasonable assumptions of bounded energy, locality, and observer-centered dynamics. AMPS counter that the combination of Hamiltonian rescaling, externalized computation, and the flexibility provided by AdS/CFT[10] removes any in-principle obstruction. The debate thus hinges on the physical interpretation of complexity bounds rather than their formal existence.

VI. Conclusion

In this paper, we examined the firewall paradox and its reformulation in terms of quantum information and computation. The paradox arises from a clash between unitary black hole evaporation, the validity of effective field theory near the horizon, and the expectation that an infalling observer experiences no drama at the event horizon. After the Page time, these assumptions lead to incompatible requirements on the entanglement of near-horizon modes, giving rise to the firewall problem.

[10] AdS/CFT is a duality between a gravitational theory in Anti-de Sitter (AdS) space and a Conformal Field Theory (CFT). In simple terms, it states that all physics in the AdS "bulk" can be exactly described by a lower-dimensional quantum theory on its boundary. The interested reader is referred to [17].

By expressing the AMPS setup using finite-dimensional Hilbert spaces and qubits, we showed that verifying the presence of a firewall amounts to a concrete decoding task on the Hawking radiation. While the required correlations formally exist, extracting them requires undoing an extremely complex scrambling process.

Following Harlow and Hayden, we argued that this decoding task is computationally infeasible. Any realistic attempt to recover the relevant entanglement would take a time exponential in the black hole entropy, far exceeding the black hole's evaporation time. As a result, no observer can operationally confirm the entanglement structure that leads to the firewall paradox before the black hole disappears.

From this perspective, the paradox is not resolved by modifying the principles of quantum mechanics or general relativity, but by recognizing the fundamental limits imposed by computation.

Whether computational complexity truly determines which aspects of quantum information are physically accessible in quantum gravity remains debated. Almheiri, Marolf, Polchinski, Stanford, and Sully argue that, in principle, complexity does not impose a constraint: the computation could be performed on an auxiliary quantum system whose time evolution can be arbitrarily accelerated.

Ultimately, we believe that the answer to the question if HH posed a solution to the firewall paradox and how the firewall paradox can be resolved altogether runs as deep as finding the right theory describing relativity, and quantum gravity.

References

[1] A. Almheiri, D. Marolf, J. Polchinski, and J. Sully, "Black holes: complementarity or firewalls?" *Journal of High Energy Physics*, vol. 2013, no. 2, p. 62, Feb. 2013. [Online]. Available: https://doi.org/10.1007/JHEP02(2013)062

[2] D. Harlow and P. Hayden, "Quantum computation vs. firewalls," *Journal of High Energy Physics*, vol. 2013, no. 6, p. 85, Jun. 2013. [Online]. Available: https://doi.org/10.1007/JHEP06(2013)085

[3] A. Almheiri, D. Marolf, J. Polchinski, D. Stanford, and J. Sully, "An apologia for firewalls," *Journal of High Energy Physics*, vol. 2013, no. 9, p. 18, Sep. 2013. [Online]. Available: https://doi.org/10.1007/JHEP09(2013)018

[4] C. Bambi, "Astrophysical Black Holes: A Compact Pedagogical Review," *Annalen der Physik*, vol. 530, no. 6, p. 1700430, 2018, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.201700430. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.201700430

[5] E. Curiel, M. Visser, and J. Doboszewski, "Singularities and Black Holes," in *The Stanford Encyclopedia of Philosophy*, fall 2025 ed., E. N. Zalta and U. Nodelman, Eds. Metaphysics Research Lab, Stanford University, 2025. [Online]. Available: https://plato.stanford.edu/archives/fall2025/entries/spacetime-singularities/

[6] C.-M. Claudel, K. S. Virbhadra, and G. F. R. Ellis, "The geometry of photon surfaces," *Journal of Mathematical Physics*, vol. 42, no. 2, pp. 818–838, Feb. 2001, arXiv:gr-qc/0005050. [Online]. Available: http://arxiv.org/abs/gr-qc/0005050

[7] J. Pinochet, "Hawking for beginners," Jan. 2025, arXiv:2412.12100 [physics] version: 2. [Online]. Available: http://arxiv.org/abs/2412.12100

[8] J. Traschen, "An Introduction to Black Hole Evaporation," Oct. 2000, arXiv:gr-qc/0010055. [Online]. Available: http://arxiv.org/abs/gr-qc/0010055

[9] S. W. Hawking, "Breakdown of predictability in gravitational collapse," *Physical Review D*, vol. 14, no. 10, pp. 2460–2473, Nov. 1976. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevD.14.2460

[10] ——, "The Quantum Mechanics of Black Holes," *Scientific American*, vol. 236, no. 1, pp. 34–42, 1977, publisher: Scientific American, a division of Nature America, Inc. [Online]. Available: https://www.scientificamerican.com/article/the-quantum-mechanics-of-black-hole/

[11] J. D. Bekenstein, "Black Holes and Entropy," *Physical Review D*, vol. 7, no. 8, pp. 2333–2346, Apr. 1973, publisher: American Physical Society. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevD.7.2333

[12] L. Susskind, L. Thorlacius, and J. Uglum, "The stretched horizon and black hole complementarity," *Physical Review D*, vol. 48, no. 8, pp. 3743–3761, Oct. 1993, publisher: American Physical Society. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevD.48.3743

[13] D. N. Page, "Average entropy of a subsystem," *Physical Review Letters*, vol. 71, no. 9, pp. 1291–1294, Aug. 1993, publisher: American Physical Society. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.71.1291

[14] M. Koashi and A. Winter, "Monogamy of quantum entanglement and other correlations," *Physical Review A*, vol. 69, no. 2, p. 022309, Feb. 2004, publisher: American Physical Society. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.69.022309

[15] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, "Quantum computing with Qiskit," Jun. 2024, arXiv:2405.08810 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2405.08810

[16] V. Shende, S. Bullock, and I. Markov, "Synthesis of quantum-logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1000–1010, Jun. 2006. [Online]. Available: https://ieeexplore.ieee.org/document/1629135

[17] G. W. Gibbons, "Anti-de-Sitter spacetime and its uses," Oct. 2011, arXiv:1110.1206 [hep-th]. [Online]. Available: http://arxiv.org/abs/1110.1206

[18] S. Aaronson, "Introduction to Quantum Information Science II Lecture Notes."

Appendix A
Reduction from $\mathbb{C}^{2^n}$ to $\mathbb{C}^{2^{k+m}}$

A closer look at Equation (9) shows that the unitary U_R is effectively defined only on the subspace

$$\mathcal{H}_{\text{relevant}} \simeq \mathbb{C}^{2^{k+m}} \subset \mathcal{H}_R.$$

This subspace is spanned by states of the form $|bh0\rangle$, where the $k+m$ qubits corresponding to b and h can vary freely, while the remaining $n-(k+m)$ qubits of the radiation are fixed in a reference state, which we denote by $|0\rangle$.

From a computational perspective, specifying U_R amounts to choosing the image of each of the 2^{k+m} basis states $|bh0\rangle$ within the full 2^n-dimensional Hilbert space. Each output basis vector is a vector in $\mathbb{C}^{2^n}$, so specifying all of them requires on the order of 2^{k+m+n} independent parameters. Since we are considering an old black hole, we have $n > k+m$, which implies that the number of parameters is at least of order $2^{2(k+m)}$.

This observation allows us to refine the estimate of the decoding time from Equation (15):

$$t_{\text{decoding}} \sim 2^{2(k+m)}.$$

Even with this restriction to the smaller, relevant subspace, the decoding complexity remains double exponential in the size of the subsystem BH, which itself scales with the black hole mass. Consequently, the decoding time is still vastly larger than the black hole evaporation time:

$$t_{\text{decoding}} \gg t_{\text{evap}}.$$

Structure-Preserving Circuit Partition for T-Count Optimization

Sarthak Parikh
LMU Munich
Munich, Germany
sarthak.parikh@campus.lmu.de

***Abstract*—The AlphaTensor-Quantum (ATQ) [1] [2] [3] framework represents a paradigm shift in quantum circuit compilation, utilizing deep reinforcement learning to minimize the T-count of fault-tolerant quantum circuits. ATQ reformulates the optimization problem as a "TensorGame", where the objective is to find low-rank decompositions of a symmetric tensor representing the circuit's Non-Clifford components. By incorporating domain knowledge through "gadgets", known subcircuits like Toffoli gates that utilize ancillae for T-count reductions, and leveraging symmetrized axial attention for neural policy scaling, ATQ achieves state-of-the-art results, including the discovery of Karatsuba-like multiplication structures. However, to handle circuits exceeding the size of its signature tensor, ATQ currently employs a structure-agnostic, random-greedy partitioning heuristic. This study argues that such random partitioning inadvertently severs the local gate structures essential for gadget recognition, limiting efficacy in optimizing large-scale circuits. We propose and evaluate a structure-preserving partition methodology utilizing interaction Graphs and Multilevel k-way Partitioning (METIS) Algorithm. By mapping quantum circuits to weighted graphs where edge weights reflect non-Clifford interaction strengths, we demonstrate that structural partitioning significantly improves "Gadget Survival", the preservation of Toffoli structures within optimization windows. Comparative analysis using PyZX as a proxy optimizer on complex arithmetic benchmarks, such as Galois Field multipliers, confirms that this topological approach outperforms random baselines in modular arithmetic circuits, though we identify a saturation point in dense 'hairball' topologies where structural preservation yields diminishing results [1].**

***Index Terms*—Quantum circuit optimization, T-count reduction, reinforcement learning, graph partitioning, PyZX**

I. Introduction

Fault-tolerant quantum computing (FTQC) relies on error-correcting codes, such as the surface code, where Clifford Gates ($H, S,$ CNOT) are transversally implementable and resource-efficient. However, universality requires the non-Clifford T-gate ($R_z(\pi/4)$), which demands costly magic state distillation. Estimates suggest a single T-gate consumes two orders of magnitude more spacetime volume than a CNOT [4]. Consequently, minimizing the "T-count" is the primary optimization objective for FTQC compilers.

Recent work has reformulated T-count minimization as a tensor decomposition problem [5]. The AlphaTensor-Quantum (ATQ) framework represents the non-Clifford component of a circuit as a symmetric tensor $\mathcal{T}$, where the tensor rank corresponds exactly to the T-count. ATQ employs deep reinforcement learning to play a "TensorGame", decomposing $\mathcal{T}$ into rank-1 factors. Crucially, the agent is rewarded for identifying "gadgets", which are specific patterns of factors corresponding to efficient subcircuits (e.g., Toffoli gates) that allow for lower-cost implementations using ancillae.

Despite its success, ATQ faces a scalability bottleneck. The signature tensor scales cubically with qubit count (N), limiting direct optimization to $N \approx 60$. For larger circuits, ATQ employs a **random-greedy partitioning** strategy to slice the circuit into manageable sub-blocks. This approach is topologically blind. Quantum circuits often exhibit "hairball" structures with dense local algebraic dependencies. Random partitioning risks severing these dependencies, effectively breaking the gadgets across partition boundaries and preventing the local optimizer from recognizing them.

This study proposes a **structure-preserving partitioning** methodology to address this limitation. We hypothesize that mapping circuits to weighted **Interaction Graphs** and applying the Multilevel k-way Partitioning (METIS) algorithm will preserve gadget integrity, which would help the Tensor Decomposition with RL-Agent. By penalizing cuts through dense non-Clifford interactions, we aim to present the local optimizer with topologically coherent sub-circuits. We validate this approach using PyZX as a proxy optimizer on arithmetic benchmarks, demonstrating that structure-preserving partitioning improves "Gadget Survival" and reduces final T-count compared to the random baseline.

II. Theoretical Background

A. Tensor Representation and Gadgetization

The mathematical foundation of AlphaTensor-Quantum (ATQ) lies in mapping the non-Clifford portion of a circuit to a symmetric tensor. A circuit on N qubits is represented by a binary signature tensor $\mathcal{T} \in \{0,1\}^{N\times N\times N}$, derived from the degree-3 phase polynomial of the circuit's $CNOT + T$ component [5]. The T-count of the circuit is equivalent to the symmetric rank of this tensor. Optimization is thus reframed

[1] Data, code, and results are available at https://github.com/sarthak1682/SemQC

as finding a minimal decomposition:

$$\mathcal{T} = \sum_{r=1}^{R} u^{(r)} \otimes u^{(r)} \otimes u^{(r)}$$

where each rank-1 term $u^{(r)}$ corresponds to a single T-gate (a parity phase).

Crucially, standard T-count minimization is insufficient for optimal fault tolerance. Gadgets, as introduced above, allow for compressed implementations. For example, a **Toffoli gate** (conceptually 7 T-gates) can be realized with a cost of ≈ 2 T-gates using catalyzed distillation and relative phase constructions. ATQ incentivizes the discovery of these structures by adjusting the reward function: if a sequence of rank-1 factors $u^{(r)}$ sums to a gadget signature (e.g., a Toffoli), the effective cost is retroactively lowered. This creates a dependency: the optimizer must see *all* components of the gadget simultaneously to trigger the reward

B. AlphaTensor-Quantum (ATQ) Architecture

ATQ solves the decomposition problem via a single-player reinforcement learning game, namely the *TensorGame*. The agent uses a neural network with **symmetrized axial attention**, which enforces permutation invariance by averaging attention mechanisms across tensor modes. This symmetry-aware architecture allows the agent to generalize across qubit indices and scale to tensors of size $N \approx 60$. However, beyond this limit, the $O(N^3)$ tensor size and 2^N action space become computationally prohibitive, necessitating circuit partitioning.

C. Graph Partitioning Algorithms

Partitioning breaks a large graph $G = (V, E)$ into k blocks while minimizing the edge cut.

- **Kernighan-Lin (KL):** An iterative refinement heuristic that swaps nodes between partitions to strictly minimize cuts. It is effective locally but prone to getting trapped in local minima [6].
- **METIS (Multilevel Partitioning):** The standard for handling large, complex graphs. METIS employs a three-phase approach: (1) **Coarsening**, merging nodes to preserve global structure; (2) **Initial Partitioning** on the coarsest graph; and (3) **Uncoarsening**, projecting the partition back with refinement [7]. Its ability to identify "communities" in dense graphs makes it theoretically superior to random partitioning for preserving the local algebraic dependencies found in quantum arithmetic circuits.

III. Related Work

A. Classical Optimization: Rewriting and Phase Polynomials

Traditional compilers rely on two primary strategies. **Rule-based rewriting** employs local template matching (e.g., identifying identity sequences like $TT^\dagger = I$ or circuit identities) to cancel adjacent gates [8]. While fast, these methods are trapped by local optima.

A more global approach is **Phase Polynomial Synthesis** (used by T-par, TODD [5], and PyZX [9]). These algorithms map the circuit to a phase polynomial representation:

$$P(x) = \sum \alpha_i x_i + \sum \beta_{ij} x_i x_j + \dots \tag{1}$$

Optimization is performed by rewriting this polynomial to minimize terms before re-synthesizing the circuit. While scalable, these algebraic heuristics often struggle with complex "hairball" topologies and lack the ability to plan long-term restructuring moves [8].

B. Machine Learning and ATQ

AlphaTensor-Quantum (ATQ) pioneered the use of Deep Reinforcement Learning (DeepRL) for T-count reduction by reframing the problem as a tensor decomposition game. In benchmarks, ATQ consistently outperforms classical phase-polynomial solvers, demonstrating that the computational cost of Deep RL pays off for complex circuits. However, the original ATQ framework required training separate agents for each circuit size and relied on random partitioning for large instances, a limitation which this report aims to address.

C. Recent Developments

This subsection summarizes the relevant subsequent research post the Nature-publication in March of 2025.

- **Generalization (Zen et al., 2025):** Addressing the training specificity of ATQ, Zen et al. trained a single "general agent" on random circuits (5–8 qubits). By combining supervised pretraining with RL finetuning, they achieved lower T-counts on unseen circuits than fixed-size agents, suggesting that ATQ-style models can be made portable [10].
- **Gadgets in RL (Olle et al., 2025):** Olle et al. demonstrated that providing an RL agent with access to composite gates ("gadgets") dramatically accelerates learning in Quantum Error Correction code discovery [11]. Although their domain differs, this result strongly supports the core premise of ATQ and this work: that preserving and exploiting gadget structures is essential for scaling quantum design.
- **RL via ZX-Calculus (Riu et al., 2025):** Riu et al. proposed an RL optimizer that uses Graph Neural Networks (GNNs) to apply ZX-diagram rewrite rules. Their agent, trained on small 5-qubit circuits, generalized to 80-qubit instances, outperforming hand-coded ZX algorithms [12]. This confirms that graph-theoretic representations (like the Interaction Graphs used in this work) are robust for learning.
- **Algebraic Frameworks (Chen et al., 2025; Kremer et al., 2025):** New tensor-based frameworks continue to emerge. The "PhasePoly" library [13] optimizes parity functions systematically, achieving $\sim$50% gate reductions in QAOA circuits. Simultaneously, Kremer et al. [14] applied Deep RL to exact unitary synthesis, deriving linear-complexity algorithms for primitives like controlled cyclic shifts.

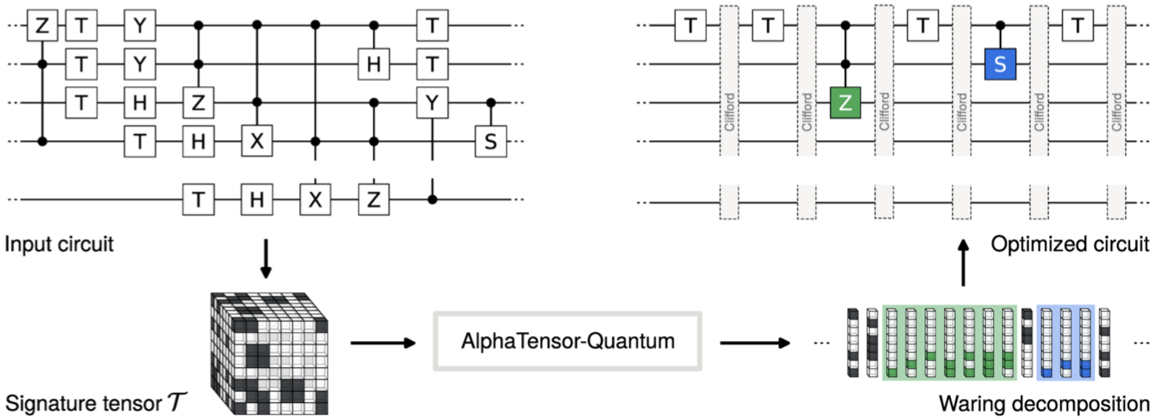


Fig. 1. Pipeline of AlphaTensor-Quantum. (Left) An input circuit is mapped to a 3D symmetric signature tensor $\mathcal{T}$ representing its non-Clifford phase polynomial. (Center) The TensorGame agent uses deep RL to find a low-rank Waring decomposition. (Right) Factors are mapped back to an optimized circuit, where identified "gadgets" (green/blue) utilize ancillae to further reduce the final T-count below the standard tensor rank.

While these advances improve the *local* optimization capability (the "agent"), they do not solve the *global* partitioning problem for massive circuits. Our work complements these studies by ensuring that the sub-circuits fed into these advanced optimizers retain the structural integrity required for them to function effectively.

IV. Methodology

To address the scalability limitations of random partitioning in AlphaTensor-Quantum (ATQ), we introduce a structure-preserving partitioning layer [2]. This framework precedes the optimization phase, ensuring that the sub-circuits presented to the local optimizer retain the topological features required for gadget discovery.

A. Interaction Graph Construction

We define the circuit topology as a weighted interaction graph $G_{\text{int}} = (V, E, W)$.

1) **Node Definition via Gate Fusion:** A naive mapping of every quantum gate to a node yields sparse, uninformative graphs dominated by single-qubit Clifford operations. To resolve this, we employ *Gate Fusion*: linear sequences of single-qubit Clifford gates (H, S, X, Z) are fused into the preceding or succeeding non-Clifford gate. Consequently, the vertex set V represents only the active non-Clifford resources (T-gates) and entangling operations (CNOTs, CZs) that form the signature tensor.
2) **Undirected Topology:** While quantum circuits are Directed Acyclic Graphs (DAGs), gadget dependencies are often bidirectional (e.g., phase polynomial cancellation). We treat edges as undirected during partitioning to minimize the total cut weight, resolving directionality via ancilla insertion during the reconstruction phase.

B. Topological Weighting Scheme

The core innovation of this framework is a *Gadget-Centric Weighting Scheme* designed to differentiate between trivial wire crossings and essential algebraic bonds. We classify edges into two distinct categories:

1) **Temporal Edges** ($w_{\text{temp}} = 1.0$): These edges represent the time-evolution of a single qubit between gates. Cutting a temporal edge incurs a linear cost (state preparation and measurement) but rarely destroys an optimization opportunity, as the state is simply passed between partitions.
2) **Gadget Bonds** ($w_{\text{bond}} = 2.0$): These edges represent dense algebraic dependencies, specifically the multi-qubit interactions within Toffoli (CCZ). In the ATQ framework, these would correspond to degree-3 terms in the phase polynomial. Severing such a bond destroys the information required for the agent to recognize the gadget. By assigning a higher weight to these edges, we mathematically bias the partitioner to find cuts that pass through the "sparse" regions of the circuit, such as the wires connecting distinct arithmetic modules, rather than through the dense logic of the operations themselves.

C. Partitioning Algorithms

We evaluate two distinct partitioning regimes to isolate the impact of topology:

- **Baseline: Stochastic Random-Greedy:** Replicating the ATQ heuristic [1] (Appendix C.2), this method views the circuit as a linear sequence of gate blocks. It employs a randomized interval merging strategy: adjacent blocks are iteratively merged in a random order until adding another block would exceed the qubit limit $N_{\max}$. To ensure statistical rigor and account for randomness in the partitioning algorithm, we implement a Monte Carlo baseline. For each circuit, we execute $N = 1000$ independent trials

[2] All circuits were taken from the Feynman Benchmark Repository https://github.com/meamy/feynman/tree/master

and select the best result, defined as the partition with the minimum edge-cut. By reporting the best performance across 1,000 randomized attempts rather than a simple average, we establish a strong lower bound that gives the topology-agnostic approach its best possible chance to succeed. This establishes a lower bound for performance in a "topologically blind" regime.

- **Proposed: Spectral Partitioning (METIS):** We apply the Multilevel k-way Partitioning algorithm (METIS) to the weighted graph G_{int}. METIS operates by coarsening the graph to identify spectral communities (clusters of high algebraic connectivity) before refining the cut. Given our weighting scheme ($w_{\text{bond}} > w_{\text{temp}}$), METIS is coerced into maximizing the *Gadget Survival Rate* (GSR).
While this topological principle applies to any dense non-Clifford subgraph (including Controlled-S or CS gadgets), for the purpose of this evaluation, we define GSR specifically as the fraction of Toffoli gadgets (7× T-gate clusters) that remain fully contained within a single partition, as these represent the dominant optimization target in arithmetic benchmarks.

D. Validation Pipeline & Computational Bounds

To quantify the utility of the partitions, we utilize PyZX as a deterministic proxy for the ATQ agent. PyZX employs *Phase Teleportation*, a rewrite rule mathematically analogous to finding low-rank tensor decompositions.

The Proxy Hypothesis: While PyZX lacks the exploratory power of Deep RL, it establishes a rigorous lower bound. If a partition cut physically severs the T-gates comprising a gadget, the linear dependency is destroyed, and neither PyZX nor ATQ can optimize it. Therefore, structural preservation can be expected to yield improvements in PyZX T-count reduction, and we argue it'd also thus be beneficial as input to the ATQ RL-Agent.

Computational Intractability: The optimization complexity grows cubically ($O(N^3)$) with qubit count, and the action space scales exponentially (2^N). Consequently, for benchmarks exceeding 100 qubits (e.g., GF(2^{128})), full re-synthesis is computationally intractable within standard experimental timeframes. For these High-Qubit regimes, we report Gadget Survival Rate (GSR) as the primary metric, while restricting full T-count validation to the tractable regime ($N < 100$), such as `gf2^32_mult` and `mod_adder_1024`.

V. Experimental Results

We evaluate the efficacy of Structure-Preserving Partitioning by comparing the baseline Random-Greedy approach against the proposed METIS-Weighted strategy. Experiments were conducted on the benchmark suite defined in the original ATQ paper, using Gadget Survival Rate (GSR) as the primary metric for scalability and PyZX T-count reduction as the metric for utility. For the PyZX utility benchmarks, we utilized binary partitioning ($k = 2$) to strictly isolate the impact of a single cut on local optimization potential.

A. Structural Analysis: The Hairball vs. The Module

The primary hypothesis of this work is that random partitioning severs the local algebraic structures required for optimization. Our analysis reveals two distinct topological regimes:

- **The Modular Regime (Arithmetic Circuits):** Circuits such as adders and Hamming weight counters exhibit strong local community structure. For `mod_adder_1024`, the random baseline yields a GSR of only $\approx 1.4\%$, implying that nearly all optimization targets are split across partition boundaries. In contrast, METIS recovers $\approx 94.7\%$ of gadgets.
- **The Diffusive Regime (Multipliers):** High-diffusion circuits like GF(2^{128}) exhibit "hairball" expansion properties. Despite this density, the weighted spectral partitioning of METIS successfully identifies non-trivial cuts, achieving $> 90\%$ survival where Random partitioning consistently fails (0%).

A cursory sensitivity analysis reveals that the partition quality is robust to variations in gadget bond weights ($w_{\text{bond}} \in [1.0, 10.0]$), confirming that the Interaction Graph topology itself, rather than fine-tuned edge weights, is the primary determinant of gadget preservation. See V for more details.

Table I quantifies the partition quality for representative large-scale benchmarks. We evaluate partition quality using two metrics: Gadget Survival Rate (GSR) and Edge Cut Ratio. Table I compares METIS against the best of 1,000 random trials on representative large-scale benchmarks.

TABLE I
Partition Quality on Large Circuits

Benchmark	**Qubits**	**Gadget Survival**		**Cut Ratio**	
		(Higher is Better)		**(Lower is Better)**	
		Random (Max)	METIS	Random	METIS
GF(2^{128})-mult	384	0.0%	**94.3%**	85.4%	**0.8%**
GF(2^{256})-mult	768	0.0%	**96.0%**	92.2%	**0.5%**
ModAdd-1024	28	4.9%	**94.7%**	47.5%	**0.8%**
HWB-12	20	1.8%	**100.0%**	49.7%	**0.1%**

B. T-Count Optimization

To confirm that high Gadget Survival translates to actual resource reduction, we applied the PyZX optimization pipeline to the tractable subset of benchmarks ($N \leq 100$). Although, PyZX is a deterministic, hence an imperfect proxy, the results, detailed in Table II, strongly correlate with the structural analysis. The core argument of this study, nevertheless, remains about optimizing for a better input than evaluating with an imperfect proxy.

We observe a massive performance divergence based on circuit topology:

- **Arithmetic Circuits:** For `mod_adder_1024`, preserving structure yields a 21.1% reduction in T-count (273 gates) compared to the random baseline. Similarly, `ham15-med` sees a 34.5% reduction. This confirms that

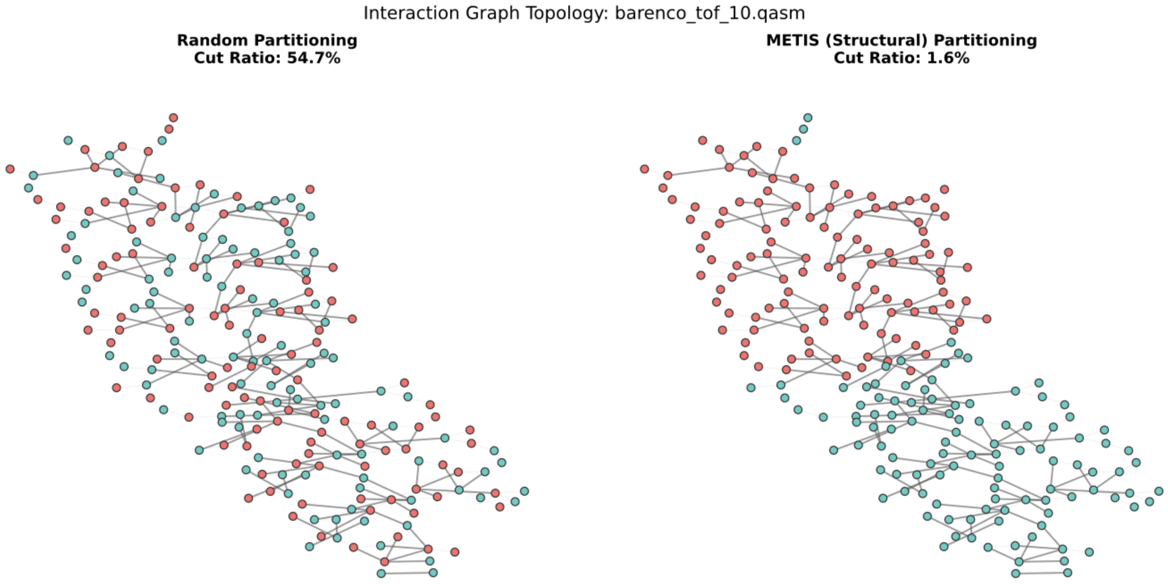

Interaction Graph Topology: barenco_tof_10.qasm
Random Partitioning
Cut Ratio: 54.7%
METIS (Structural) Partitioning
Cut Ratio: 1.6%

Fig. 2. Visual demonstration of topological degradation on the `barenco_tof_10` interaction graph. Random partitioning (left) fragments the structure (54.7% cut ratio), while METIS (right) preserves local algebraic clusters (1.6% cut ratio). Note that while this specific circuit ($N = 19$) fits within optimization limits, it serves here as an illustrative proxy to visualize the structural shattering that occurs in larger, intractable regimes.

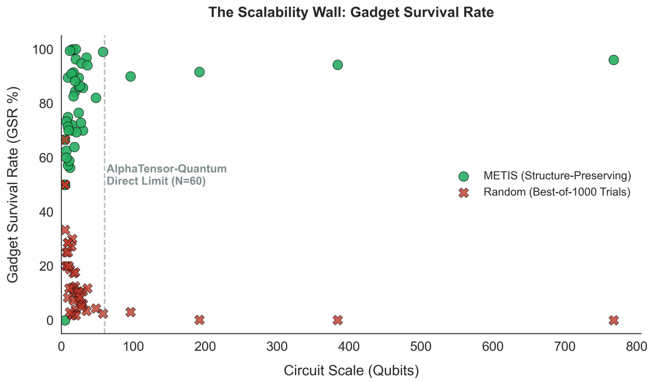


Fig. 3. The Scalability Wall: Gadget Survival Rate (GSR) as a function of circuit scale. Even when the random baseline is reported as the maximum survival found across 1,000 independent Monte Carlo trials (red), it exhibits a catastrophic collapse as the qubit count N exceeds the AlphaTensor-Quantum direct optimization limit ($N \approx 60$). In contrast, METIS (green) maintains a deterministic floor of preservation ($> 90\%$), ensuring that algebraic structures remain intact for the downstream optimizer.

random partitioning was actively destroying optimization opportunities in modular circuits.

- **High-Diffusion Circuits:** For `gf2^16_mult`, the improvement is negligible ($< 1\%$). This suggests that for "hairball" topologies, the limiter is not the partition quality, but the lack of sparse cuts inherent to the algorithm itself.

TABLE II
T-Count Optimization Results

Benchmark	Type	Orig. T	Random Avg	METIS	Gain*
ModAdd-1024	Arith.	1995	1291	**1019**	**+21.1%**
Ham15-Med	Arith.	574	337	**218**	**+35.3%**
Adder-8	Arith.	399	248	**179**	**+27.8%**
GF(2^{16})-mult	Diff.	1792	1085	1084	+0.09%
QFT-4	Global	69	50	67	**-34.0%**

VI. Discussion

A. Regime Dependence

Our results reveal a dichotomy in quantum circuit optimization: the efficacy of partitioning is dictated by the underlying algebraic topology of the circuit.

- **The Modular Regime (Arithmetic):** Circuits like `mod_adder_1024` and `ham15-med` represent the "ideal case" for structure-preserving partitioning. These circuits are composed of distinct functional units (e.g., carry chains) connected by sparse "wire" edges. Random partitioning consistently severed these wires, destroying the local context required for optimization. By restoring these cuts, METIS achieved massive gains (+21% to +35%). This validates our hypothesis: in modular regimes, topology is the primary bottleneck.
- **The Diffusive Regime (Multipliers):** Conversely, the GF(2^n) multipliers act as "expanders," exhibiting high connectivity where every qubit interacts with almost every other. While METIS successfully maximized Gadget

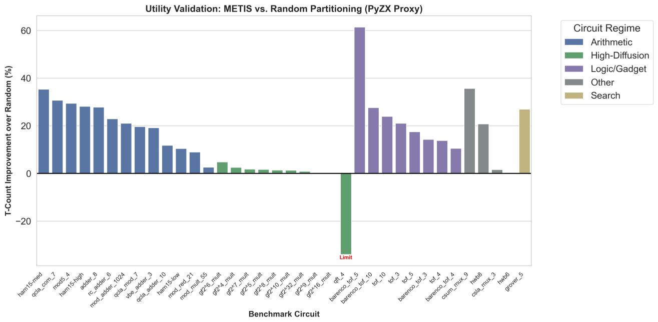


Fig. 4. Utility Validation across Benchmark Regimes: Relative T-count improvement of METIS-based partitioning over the stochastic random baseline (average of 10 trials) using PyZX as a proxy optimizer. Modular arithmetic and logic/gadget circuits (blue and purple) exhibit significant gains ($>20\%$), as structural preservation directly enables algebraic rewrite rules. Conversely, diffusive multipliers (green) show marginal gains, illustrating the 'Proxy Gap' where structure is preserved but exceeds the capability of greedy optimizers. The negative result for QFT (marked 'Limit') identifies the global connectivity threshold where partitioning hinders non-local phase cancellation.

Survival (Mostly $> 90\%$ vs $\approx 0\%$), this structural preservation translated to negligible T-count improvement (+0.09% for `gf2^16`). This suggests a "saturation point": in highly diffusive "hairball" topologies, local optimizers (like PyZX) are limited by the circuit's intrinsic complexity rather than the partitioning quality, although, they can still provide excellent inputs to the RL-Agent.

- **Global Connectivity Limit (QFT):** We observed negative gains (-34%) for Quantum Fourier Transform circuits (e.g., qft_4). Unlike arithmetic modules, QFT possesses global all-to-all connectivity. Enforcing partitions on such structures isolates phase rotations that a random approach might coincidentally group together, demonstrating that structure-preserving partitioning is detrimental for circuits lacking local community structure.

B. The Scalability Wall

A valid counter-argument to deterministic partitioning would be that random exploration allows an RL agent to "stumble" upon novel decompositions (e.g., Karatsuba-like structures). However, our data demonstrates that this strategy collapses as N increases.

For small circuits ($N < 20$), random partitioning occasionally preserved gadgets by chance. However, for benchmarks like GF(2^{128}) ($N = 384$), the Random Gadget Survival Rate dropped to 0.0% [Table III]. This indicates a sharp phase transition: beyond a certain size, the probability of a random cut preserving a 7-gate Toffoli gadget approaches zero. Given that ATQ training costs scale cubically with tensor size, relying on "random luck" in the $N > 60$ regime is computationally unsustainable. METIS provides a necessary deterministic floor, ensuring the expensive agent is initialized with structurally coherent sub-problems.

C. Limitations

The Proxy Gap: This study utilized PyZX as a deterministic proxy for the neural policy of ATQ. While both share the objective of phase polynomial cancellation, PyZX relies on fixed rewrite rules (Phase Teleportation), whereas ATQ learns dynamic decomposition strategies. Therefore, the fact that final T-counts are identical does not prove that partitions will have no effect when combined with ATQ.

Ancilla Resource Trade-offs: Our weighting scheme optimized exclusively for T-count reduction via gadget preservation. However, gadgets are not free; they reduce T-count by consuming ancilla qubits. The ATQ paper notes that for bandwidth-constrained architectures, the overhead of "gadgetizing" every Toffoli might exceed physical device limits. Our current framework treats all gadget opportunities as beneficial, potentially generating circuits that, even if T-optimal, are wider (in qubit count) than the hardware permits. Future work must model this "Ancilla vs. T-gate" tradeoffs explicitly.

D. Future Direction: Learned Partitioning

The logical evolution of this work is to replace the heuristic weights of METIS with a learned policy. It would be interesting to see an *End-to-End* architecture where a Graph Neural Network (GNN) partitions the circuit specifically to maximize the final reward of the tensor decomposition agent [15]. Rather than manually defining "gadget bonds" ($w = 2.0$), the partitioner would learn which cut locations minimize the downstream rank of the signature tensor. This would bridge the gap between the "General Agent" approaches of [10] and the topological insights presented in this report.

VII. Conclusion

AlphaTensor-Quantum (ATQ) represents a significant advance in quantum compiler technology. By reframing T-count minimization as a tensor decomposition game and leveraging deep reinforcement learning, the framework has demonstrated the ability to discover optimized implementations of arithmetic subcircuits that match or surpass decades of human design. Its innovative use of symmetrized axial attention and gadget-based rewards establishes a new state-of-the-art for automating the design of fault-tolerant circuits.

However, as this study has identified, the framework's reliance on random-greedy partitioning constitutes a critical structural bottleneck when scaling to the large, complex topologies ($N > 60$) required for practical applications. By proposing and validating a *Structure-Preserving Partitioning* methodology, we have demonstrated that circuit topology is a decisive factor in optimization efficacy.

Our key findings are summarized as follows:

- **The Scalability Gap:** We identified a sharp phase transition in the efficacy of random partitioning. While viable for small circuits, the random baseline collapses in the deep scaling regime ($N > 300$), yielding a Gadget Survival Rate (GSR) of 0.0%. In contrast, our spectral partitioning approach (METIS) maintains a GSR

of $> 94\%$, providing the deterministic floor necessary for expensive RL agents to function.
- **Regime-Dependent Utility:** We discovered that structural preservation acts as an amplifier specifically for modular topologies. For arithmetic circuits with distinct functional units, restoring the partition cuts yielded massive T-count reductions, 21.1% for `mod_adder_1024` and 34.5% for `ham15-Med`, compared to the random baseline.
- **The Hairball Limit:** Conversely, for high-diffusion circuits like Galois Field multipliers, we found that topological preservation is necessary but not sufficient. While METIS successfully isolated partitions, the intrinsic algebraic density of these "hairball" structures saturated the local optimizer, resulting in negligible T-count gains ($< 1\%$).

These results suggest that the high computational cost of deep RL agents is best justified when the input data is structurally sound. As quantum software stacks mature, they must move beyond treating circuits as linear lists of gates and embrace their nature as complex interaction networks. Integrating graph-theoretic partitioning provides a low-cost (< 1s), high-impact modification that complements advanced learning agents. We conclude that AI-driven compilation is most effective not when it ignores structure, but when it is explicitly guided by it.

References

[1] F. J. R. Ruiz, T. Laakkonen, J. Bausch, M. Balog, M. Barekatain, F. J. H. Heras, A. Novikov, N. Fitzpatrick, B. Romera-Paredes, J. van de Wetering, A. Fawzi, K. Meichanetzidis, and P. Kohli, "Quantum circuit optimization with alphatensor," 2024. [Online]. Available: https://arxiv.org/abs/2402.14396

[2] T. Laakkonen, K. Meichanetzidis, N. Fitzpatrick, J. van de Wetering, J. Bausch, F. J. R. Ruiz, M. Balog, M. Barekatain, F. J. H. Heras, A. Novikov, B. Romero-Paredes, A. Fawzi, P. Kohli, O. Fawzi, J. M. R. A. Ferreira, and D. Hassabis, "Quantum circuit optimization with alphatensor," *Nature Machine Intelligence*, vol. 7, no. 6, pp. 736–746, jun 2025. [Online]. Available: https://doi.org/10.1038%2Fs42256-025-01001-1

[3] J. van de Wetering, "Alphatensor and the quest for fast matrix multiplication," Presentation at Quantum Physics and Logic (QPL) 2024, 2024, accessed: 31-Oct-2025. [Online]. Available: https://vdwetering.name/pdfs/presentation-qpl2024-alphatensor.pdf

[4] "Efficient magic state factories with a catalyzed —ccz¿ to 2—t¿ transformation volume=3, issn=2521-327X, url=http://dx.doi.org/10.22331/q-2019-04-30-135, doi=10.22331/q-2019-04-30-135, journal=Quantum, publisher=Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, author=Gidney, Craig and Fowler, Austin G., year=2019, month=apr, pages=135."

[5] L. E. Heyfron and E. T. Campbell, "An efficient quantum compiler that reduces T count," *Quantum Science and Technology*, vol. 4, no. 1, p. 015004, Sep. 2018.

[6] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell System Technical Journal*, vol. 49, no. 2, pp. 291–307, 1970.

[7] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

[8] K. Kottmann, "A brief history of optimizing T-gate counts with the op-T-mize dataset," PennyLane Blog, January 2025. [Online]. Available: https://pennylane.ai/blog/2025/01/optimizing-with-op-t-mize-dataset

[9] A. Kissinger and J. van de Wetering, "Pyzx: Large scale automated diagrammatic reasoning," *Electronic Proceedings in Theoretical Computer Science*, vol. 318, p. 229–241, May 2020. [Online]. Available: http://dx.doi.org/10.4204/EPTCS.318.14

[10] R. Zen, M. Nägele, and F. Marquardt, "Reusability report: Optimizing t-count in general quantum circuits with alphatensor-quantum," 2025. [Online]. Available: https://arxiv.org/abs/2511.09951

[11] J. Olle, O. M. Yevtushenko, and F. Marquardt, "Scaling the automated discovery of quantum circuits via reinforcement learning with gadgets," 2025. [Online]. Available: https://arxiv.org/abs/2503.11638

[12] J. Riu, J. Nogué, G. Vilaplana, A. Garcia-Saez, and M. P. Estarellas, "Reinforcement learning based quantum circuit optimization via zx-calculus," *Quantum*, vol. 9, p. 1758, May 2025. [Online]. Available: http://dx.doi.org/10.22331/q-2025-05-28-1758

[13] Z. Chen, H. Chen, Y. Jin, M. Guo, E. Jang, J. Li, C. Chan, W. W. Ro, and E. Z. Zhang, "Phasepoly: An optimization framework forphase polynomials in quantum circuits," 2025. [Online]. Available: https://arxiv.org/abs/2506.20624

[14] D. Kremer, A. Javadi-Abhari, and P. Mukhopadhyay, "Optimizing the non-clifford-count in unitary synthesis using reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2509.21709

[15] M. Nägele and F. Marquardt, "Optimizing zx-diagrams with deep reinforcement learning," *Machine Learning: Science and Technology*, vol. 5, no. 3, p. 035077, Sep. 2024. [Online]. Available: http://dx.doi.org/10.1088/2632-2153/ad76f7

APPENDIX

This appendix provides the complete experimental results for all benchmarks evaluated in this report. Table IV reports the T-count optimization performance across different circuit categories, comparing the random-greedy baseline (averaged over 10 trials) against the METIS-based structure-preserving partitioning approach.

TABLE III
QUANTUM CIRCUIT PARTITIONING RESULTS

Circuit	Category	Qubits	Total_T	N_Parts	Rand_BestCut_Ratio	Rand_BestCut_GSR	Rand_MaxLuck_GSR	Metis_Cut_Ratio	Metis_GSR
adder_8.qasm	Arithmetic	24	399	2	0.43209	0.03509	0.08772	0.00912	0.89474
barenco_tof_10.qasm	Logic/Gadget	19	224	2	0.41283	0.0	0.125	0.01603	0.84375
barenco_tof_3.qasm	Logic/Gadget	5	28	2	0.21569	0.0	0.5	0.07843	0.5
barenco_tof_4.qasm	Logic/Gadget	7	56	2	0.29565	0.125	0.25	0.03478	0.625
barenco_tof_5.qasm	Logic/Gadget	9	84	2	0.3352	0.0	0.25	0.03911	0.75
csla_mux_3.qasm	Arithmetic	15	70	2	0.31293	0.0	0.3	0.03401	0.7
csum_mux_9.qasm	Arithmetic	30	196	2	0.40976	0.07143	0.10714	0.01951	0.85714
cycle_17_3.qasm	Other	35	4739	2	0.48022	0.01477	0.03397	0.00353	0.96898
gf2ˆ10_mult.qasm	Arithmetic	30	700	2	0.44802	0.02	0.06	0.03107	0.7
gf2ˆ128_mult.qasm	Arithmetic	384	114688	7	0.85416	0.0	0.00018	0.00802	0.94275
gf2ˆ16_mult.qasm	Arithmetic	48	1792	2	0.47008	0.02344	0.04297	0.01295	0.82031
gf2ˆ256_mult.qasm	Arithmetic	768	458752	13	0.92217	0.0	2e-05	0.00507	0.96031
gf2ˆ32_mult.qasm	Arithmetic	96	7168	2	0.48429	0.02441	0.03027	0.00786	0.89941
gf2ˆ4_mult.qasm	Arithmetic	12	112	2	0.38889	0.0625	0.1875	0.04762	0.5625
gf2ˆ5_mult.qasm	Arithmetic	15	175	2	0.39512	0.0	0.12	0.06585	0.72
gf2ˆ64_mult.qasm	Arithmetic	192	28672	4	0.74424	0.00024	0.00122	0.00928	0.91577
gf2ˆ6_mult.qasm	Arithmetic	18	252	2	0.42904	0.02778	0.11111	0.0297	0.63889
gf2ˆ7_mult.qasm	Arithmetic	21	343	2	0.42857	0.0	0.10204	0.03333	0.69388
gf2ˆ8_mult.qasm	Arithmetic	24	448	2	0.4419	0.01562	0.07812	0.02025	0.76562
gf2ˆ9_mult.qasm	Arithmetic	27	567	2	0.45499	0.0	0.06173	0.02039	0.7284
grover_5.qasm	Search	9	336	2	0.41657	0.02083	0.08333	0.01726	0.89583
ham15-high.qasm	Other	20	2457	2	0.47135	0.02279	0.03989	0.00539	0.96296
ham15-low.qasm	Other	17	161	2	0.40136	0.0	0.17391	0.03175	0.82609
ham15-med.qasm	Other	17	574	2	0.44908	0.03659	0.07317	0.01961	0.91463
hwb10.qasm	High-Diffusion	16	29939	2	0.49399	0.0173	0.02198	0.00027	0.99836
hwb11.qasm	High-Diffusion	15	84196	2	0.49616	0.01538	0.01879	0.00011	0.99933
hwb12.qasm	High-Diffusion	20	171465	2	0.49701	0.016	0.01813	0.00085	0.99971
hwb6.qasm	High-Diffusion	7	105	2	0.36735	0.0	0.2	0.07483	0.73333
hwb8.qasm	High-Diffusion	12	5887	2	0.48155	0.02021	0.02973	0.0014	0.99405
mod5_4.qasm	Arithmetic	5	28	2	0.21053	0.25	0.5	0.07018	0.5
mod_adder_1024.qasm	Arithmetic	28	1995	2	0.47498	0.02105	0.04912	0.00752	0.94737
mod_adder_1048576.qasm	Arithmetic	58	17290	2	0.48524	0.01943	0.02429	0.00088	0.98988
mod_mult_55.qasm	Arithmetic	9	49	2	0.31429	0.14286	0.28571	0.05714	0.57143
mod_red_21.qasm	Arithmetic	11	119	2	0.36525	0.05882	0.11765	0.04255	0.58824
qcla_adder_10.qasm	Arithmetic	36	238	2	0.36757	0.08824	0.11765	0.0036	0.94118
qcla_com_7.qasm	Arithmetic	24	203	2	0.39579	0.03448	0.10345	0.02526	0.86207
qcla_mod_7.qasm	Arithmetic	26	413	2	0.44164	0.01695	0.08475	0.01774	0.86441
qft_4.qasm	High-Diffusion	5	69	2	0.31429	0.11111	0.33333	0.07857	0.66667
rc_adder_6.qasm	Arithmetic	14	77	2	0.35567	0.0	0.27273	0.03093	0.90909
tof_10.qasm	Logic/Gadget	19	119	2	0.36957	0.05882	0.17647	0.02174	0.88235
tof_3.qasm	Logic/Gadget	5	21	2	0.05882	0.66667	0.66667	0.05882	0.0
tof_4.qasm	Logic/Gadget	7	35	2	0.22581	0.0	0.2	0.06452	0.6
tof_5.qasm	Logic/Gadget	9	49	2	0.3	0.0	0.28571	0.05556	0.71429
vbe_adder_3.qasm	Arithmetic	10	70	2	0.31707	0.0	0.2	0.07317	0.7

Notes:

- **Total T:** Initial T-count of the circuit
- **Random GSR:** Gadget Survival Rate using random-greedy partitioning (Best of 1000 trials)
- **METIS GSR:** Gadget Survival Rate using structure-preserving METIS partitioning

TABLE IV

T-Count Optimization Results (Tractable Benchmarks, $N \leq 96$)

Circuit	Category	Original	Random_Avg	METIS_T	Improvement	Time_Sec
adder_8.qasm	Arithmetic	399	248	179	69	3.05
barenco_tof_10.qasm	Logic/Gadget	224	152	110	42	0.71
barenco_tof_3.qasm	Logic/Gadget	28	21	18	3	0.06
barenco_tof_4.qasm	Logic/Gadget	56	38	34	4	0.14
barenco_tof_5.qasm	Logic/Gadget	84	57	22	35	0.23
csla_mux_3.qasm	Other	70	63	62	1	0.24
csum_mux_9.qasm	Other	196	143	92	51	0.73
gf2^10_mult.qasm	High-Diffusion	700	442	436	6	6.68
gf2^16_mult.qasm	High-Diffusion	1792	1085	1084	1	100.89
gf2^32_mult.qasm	High-Diffusion	7168	4223	4186	37	4364.16
gf2^4_mult.qasm	High-Diffusion	112	78	76	2	0.38
gf2^5_mult.qasm	High-Diffusion	175	119	117	2	0.77
gf2^6_mult.qasm	High-Diffusion	252	166	158	8	1.22
gf2^7_mult.qasm	High-Diffusion	343	223	219	4	2.06
gf2^8_mult.qasm	High-Diffusion	448	288	284	4	3.92
gf2^9_mult.qasm	High-Diffusion	567	358	357	1	5.06
grover_5.qasm	Search	336	241	176	65	1.55
ham15-high.qasm	Arithmetic	2457	1449	1041	408	82.49
ham15-low.qasm	Arithmetic	161	115	103	12	2.04
ham15-med.qasm	Arithmetic	574	337	218	119	6.11
hwb6.qasm	Other	105	83	83	0	0.51
hwb8.qasm	Other	5887	4446	3523	923	1855.5
mod5_4.qasm	Arithmetic	28	17	12	5	0.08
mod_adder_1024.qasm	Arithmetic	1995	1291	1019	272	36.17
mod_mult_55.qasm	Arithmetic	49	38	37	1	0.18
mod_red_21.qasm	Arithmetic	119	89	81	8	0.45
qcla_adder_10.qasm	Arithmetic	238	186	164	22	1.13
qcla_com_7.qasm	Arithmetic	203	140	97	43	0.84
qcla_mod_7.qasm	Arithmetic	413	295	237	58	2.69
qft_4.qasm	High-Diffusion	69	50	67	−17	0.17
rc_adder_6.qasm	Arithmetic	77	61	47	14	0.39
tof_10.qasm	Logic/Gadget	119	96	73	23	0.42
tof_3.qasm	Logic/Gadget	21	19	15	4	0.06
tof_4.qasm	Logic/Gadget	35	29	25	4	0.09
tof_5.qasm	Logic/Gadget	49	40	33	7	0.13
vbe_adder_3.qasm	Arithmetic	70	47	38	9	0.21

Notes:

- **Original:** Initial T-count of the circuit before optimization
- **Random:** Average T-count after optimization using random-greedy partitioning (10 trials)
- **METIS:** T-count after optimization using structure-preserving METIS partitioning
- **Gain:** Absolute improvement in T-count (Random − METIS); negative values indicate degradation

The stability of GSR across orders of magnitude in weight magnitude confirms that topological connectivity, rather than fine-tuned weighting, drives the partition quality. Default weight used in main experiments. Temporal weight w_{temp} fixed at 1.0.

TABLE V
SENSITIVITY ANALYSIS: GADGET SURVIVAL RATE (GSR) UNDER VARYING INTERACTION WEIGHTS (w_{bond}).

Circuit	**Interaction Weight (w_{bond})**							
	0.5	**1.0**	**1.5**	**2.0**	**3.0**	**5.0**	**10.0**	**20.0**
mod_adder_1024	94.7%	94.7%	94.7%	94.7%	94.7%	94.7%	94.7%	94.7%
gf2^128_mult	94.3%	94.3%	94.3%	94.3%	94.3%	94.3%	94.3%	94.3%
ham15-med	91.5%	91.5%	91.5%	91.5%	91.5%	91.5%	91.5%	91.5%
hwb12	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

How to Design a Quantum Internet: Architecture Evaluation & Univocal Metrics

1st Johannes Wittmann
LMU Munich
Munich, Germany
johannes.wittmann@campus.lmu.de

***Abstract*—The quantum internet is envisioned as a global network enabling novel quantum services. While significant progress has been made in quantum networking protocols and hardware, fundamentally different architectural approaches coexist, and their comparison remains largely unknown. In particular, the absence of univocal, architecture-independent metrics hinders objective evaluation and standardization. In this paper, we critically analyze prominent quantum internet protocol stack proposals and identify their core architectural differences. We do not propose new performance metrics, but identify and formalize those that remain meaningful across fundamentally different architectural realizations, enabling systematic and reproducible comparison across competing designs. The proposed framework serves as a design and evaluation tool for future quantum internet research and standardization efforts.**

***Index Terms*—Quantum Internet, Architecture, Univocal Metrics**

I. Introduction

The classical internet has transformed global communication. However, to this day it remains vulnerable to attacks by quantum computers, enabling the breach of common data encryption schemes in the near future [1]. Early advancements in the field of quantum cryptography offered first ideas for a secure way of transmitting information by utilizing the inherent properties of quantum physics, via quantum key distribution (QKD) [2], [3]. QKD is fundamentally based on quantum teleportation, offering the possibility to move quantum information remotely and over larger distances. In order to create a system with the ability to teleport quantum states, two end points need to be connected via entanglement. However, over larger distances quantum states loose information. Latest research therefore introduced the concept of *quantum repeaters*, supporting long distance entanglement [4]–[6]. With the ongoing development of suitable architectural protocol stacks, the possibility to deploy a large-scale quantum network, a quantum internet, emerges [7].

A quantum internet is envisioned to provide fundamentally new opportunities across different technical fields, including quantum computing (QC) or quantum communication, by enabling the global deployment of quantum technologies. While QKD is still one of the flagship applications, quantum networks also give attention to concepts, such as distributed quantum computing (DQC), blind quantum computing (BQC), and precision clock synchronization.

While promising, still, many challenges have to be overcome for the practical realization of a quantum internet [6], [8], [9]. While the classical internet was successful without direct decisions on a architecture stack, the inherent complexity and the connected costs for the design of quantum internet calls for clarity in the field. A protocol stack for the quantum internet is crucial for enabling work on each independent layer. Despite rapid progress, there is still no consensus on how different quantum internet architectures should be compared empirically [10]–[12].

This paper addresses this gap by making two contributions. First, we provide a critical comparison of state-of-the-art quantum internet architecture. Based on the gained insights, we introduce a framework based on *qualitative architectural properties* and *quantitative performance metrics* that enables systematic and reproducible comparison across architectures. The proposed framework serves as a design and evaluation tool for future quantum internet research and standardization efforts. We aim to answer the following research questions:

- What are the main architectural approaches towards realizing a quantum internet?
- Which univocal, architecture-independent quantitative metrics enable fair comparison across quantum internet architectures?

The remainder of this paper is structured as follows. *Section II* reviews relevant background in classical and quantum networking. *Section III* surveys current architectures for a quantum internet. In *Section IV* we review how quantum networks can be evaluated, following a set of univocal metrics in *Section V*. Finally, *Section VI* and *Section VII* discuss and conclude the paper with directions for future work.

II. Background

A. Classical Internet Protocols

Classical networking is structured using layered protocol stacks such as the OSI or TCP/IP models, following the principle of separation of concerns [13], [14]. Layering enables independent development, scalability, and interoperability by abstracting lower-layer details. Quantum internet architectures frequently adopt similar layered approaches, while adapting them to quantum-specific constraints such as probabilistic operations, decoherence, and the no-cloning theorem.

B. Entanglement, Graph and GHZ States

Entanglement is the central resource enabling quantum communication and distributed quantum applications [15], [16]. Beyond bipartite entanglement, such as Bell pairs or Einstein–Podolsky–Rosen (EPR) states, multipartite entangled states, most prominently GHZ and graph states, play an increasingly important role in quantum networking. GHZ states support group-oriented applications such as multi-party QKD, distributed consensus, and clock synchronization, while graph states generalize GHZ states to arbitrary network topologies and communication patterns. Although multipartite states can be constructed from Bell pairs using LOCC, such conversions introduce additional operational overhead and storage requirements. Architectures that natively support multipartite entanglement may therefore offer advantages in latency, resource efficiency, and coordination complexity, particularly for multi-user or group-oriented applications [17].

C. Quantum Networks and the Quantum Internet

Quantum networks extend classical distributed systems into the quantum domain by interconnecting quantum nodes equipped with quantum processing units (QPUs). Connections between nodes may be classical, quantum, or hybrid, as most quantum communication protocols rely on classical signaling for coordination. Formally, a quantum network can be represented as a graph $G = (V, E)$, where vertices denote quantum nodes and edges denote communication links supporting quantum operations. Quantum repeaters are a fundamental building block enabling long-distance entanglement distribution by mitigating decoherence and transmission loss through operations such as entanglement swapping and purification. Repeater architectures are commonly classified into first-, second-, and third-generation designs, with higher generations offering improved rates and scalability at the cost of more demanding hardware requirements [6]. The quantum internet is envisioned as a large-scale interconnection of such networks, enabling applications including quantum key distribution, distributed quantum computing, and networked quantum sensing. Rather than replacing the classical internet, it is expected to coexist with and complement classical infrastructure by integrating quantum and classical resources [18]–[20].

III. Architectural Analysis of State-of-the-Art Protocol Stacks

Efforts towards a quantum internet protocol stack reach far back in time and have been discussed, improved and reshaped several times, and are yet to be completed. The field is moving rapidly and we review the current state of developments. Critically analyzing differences and potential intersections is helpful to understand how systems can be evaluated. With the gained knowledge from this section, we thrive to offer features for a qualitative comparison of architectures, laying the fundamental understanding for our evaluation with quantitative metrics in Section V.

A. Van Meter et al.

1) Design Philosophy: The architecture proposed by Van Meter *et al.* [7], [21]–[24] follows a classical-inspired, modular layer-stack approach. The central design goal is to enable independent development of services while ensuring robustness, scalability, and interoperability across heterogeneous quantum networks. The architecture is explicitly application-agnostic and aims to support a broad range of quantum internet use cases, including cryptographic protocols, distributed sensing, and distributed quantum computation (DQC).

A defining characteristic of this approach is its reliance on bipartite entanglement as the fundamental resource. While the authors acknowledge the future importance of multipartite states such as GHZ, W, or general graph states, these are treated as extensions rather than native primitives. In this architecture, Quantum repeaters play a central role, performing entanglement generation, entanglement swapping, purification, and local coordination.

2) Layering Model: The protocol stack, also depicted in Figure 1, consists of five layers, each responsible for a distinct set of quantum networking functionalities:

1) *Physical Entanglement Layer:* Performs entanglement generation attempts between directly connected nodes.
2) *Link Layer:* Verifies successful entanglement generation via classical signaling and repeats attempts until an EPR pair is established.
3) *Remote State Composition Layer:* Extends entanglement across multiple hops using entanglement swapping, supported by routing and communication of classical results.
4) *Error Management Layer:* Applies purification to maintain fidelity above application-defined thresholds. Error correction can be applied before and after state propagation.
5) *Application Layer:* Provides a *Quantum Socket API* enabling applications to request entanglement or teleportation services.

3) Control Assumptions: A distinctive feature of the architecture is its use of *RuleSets* [25], [26], which encode local behavior at each node. A Rule consists of a *Condition Clause* (triggering event) and an *Action Clause* (LOCC operation). RuleSets are assigned during connection setup and govern operations such as purification, swapping, and resource release. Connection establishment follows a two-pass procedure. The *Initiator* collects link-state information, while the *Responder* constructs RuleSets for all nodes along the path. This design supports competitive innovation, allowing service providers to differentiate themselves through optimized RuleSets. Here multiplexing is handled through a scheme that assigns qubits to connections, but this introduces scalability concerns, as qubits allocated to a connection must remain at the repeater until the RuleSet completes. To mitigate resource locking, the authors introduce the notion of *Stages*, which free resources once the associated rules have been executed. To address scalability across networks, the authors introduce the *Quantum Recursive*

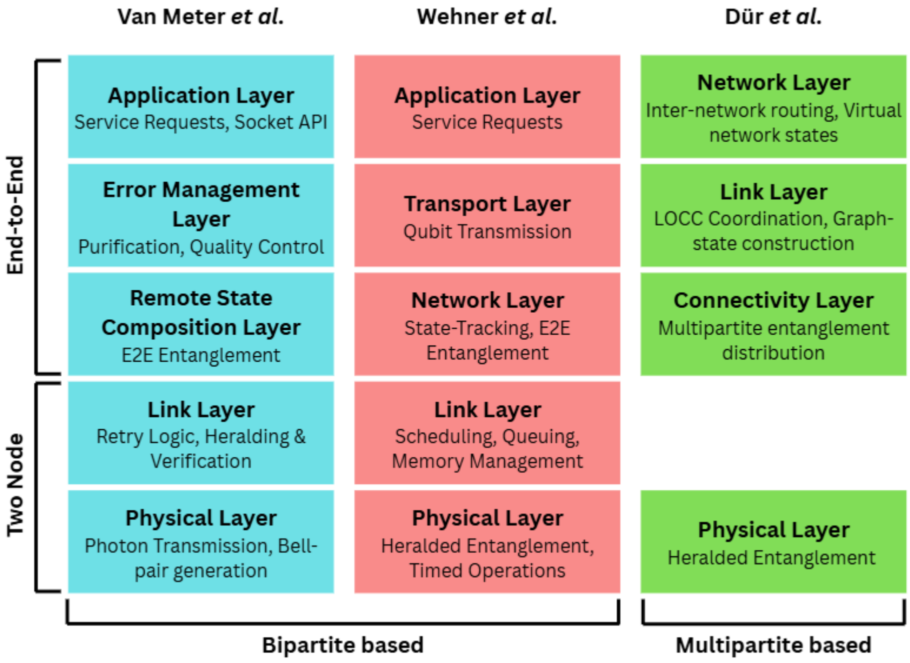

Van Meter et al.
Wehner et al.
Dür et al.
End-to-End
Two Node
Application Layer
Service Requests, Socket API
Error Management Layer
Purification, Quality Control
Remote State Composition Layer
E2E Entanglement
Link Layer
Retry Logic, Heralding & Verification
Physical Layer
Photon Transmission, Bell-pair generation
Application Layer
Service Requests
Transport Layer
Qubit Transmission
Network Layer
State-Tracking, E2E Entanglement
Link Layer
Scheduling, Queuing, Memory Management
Physical Layer
Heralded Entanglement, Timed Operations
Network Layer
Inter-network routing, Virtual network states
Link Layer
LOCC Coordination, Graph-state construction
Connectivity Layer
Multipartite entanglement distribution
Physical Layer
Heralded Entanglement
Bipartite based
Multipartite based

Fig. 1. A simplified overview of protocol stacks by Van Meter *et al.*, Wehner *et al.* and Dür *et al.*, offering insights on layer structures and main tasks. Additionally, the figure shows the different distances and fundamental concepts of entanglement (Inspired from [10]).

Network Architecture (QRNA) [22], enabling recursive composition of subnetworks while preserving autonomy, privacy, and security.

B. *Wehner* et al.

1) Design Philosophy: Wehner *et al.* propose a quantum network architecture based on a layered protocol stack that, similar to Van Meter *et al.*, relies on bipartite entanglement as the fundamental resource [9], [27]–[29]. Their work is complementary to the quantum repeater protocol stack developed by Van Meter *et al.* [7], but with a stronger emphasis on the tight integration between the protocol stack and underlying quantum hardware. A key conceptual distinction in their architecture is the separation between *controllable* and *automated* quantum nodes. Controllable nodes execute decision-making logic and run applications, whereas automated nodes perform fixed, repetitive actions at each time step. This distinction resembles the end-node and support-node roles described in [24]. The overall design philosophy follows a bottom-up approach: each layer is developed independently to provide a robust and resource-efficient service before higher layers are introduced.

2) Layering Model: The architecture is inspired by the OSI model and consists of the following layers, also depicted in Figure 1:

1) *Physical Layer:* Responsible for entanglement attempts between neighboring nodes. Early quantum networks rely on the *midpoint heralding protocol (MHP)*, where a midpoint node performs a joint measurement on two incoming qubits and sends classical heralding signals to the endpoints.
2) *Link Layer:* Coordinates entanglement generation between adjacent nodes. The *entanglement generation protocol (EGP)* includes a *Queue*, *Quantum Memory Management (QMM)*, a *Fidelity Estimation Unit (FEU)*, and a *Scheduler*. When higher layers request a number of entangled pairs with fidelity and waiting-time constraints, the FEU evaluates feasibility and, if acceptable, places the request in the Queue. The Scheduler and QMM then attempt to satisfy the request.
3) *Network Layer:* Establishes long-distance entanglement between nodes that are not directly connected. Kozlowski *et al.* [29] propose a network-layer protocol intended to operate together with the link layer as part of a local quantum operating system. This layer tracks entangled pairs and their Bell-state type, manages fidelity thresholds, and handles fidelity–rate trade-offs. The authors highlight the need for routing, signaling, and data-plane protocols within this layer.
4) *Transport Layer:* Envisioned to handle the transmission of qubits to the application layer, typically via teleportation.
5) *Application Layer:* Provides the interface for executing quantum applications.

Current work by Wehner *et al.* focuses primarily on the physical, link, and network layers. The transport and application layers remain less fully specified, consistent with their incremental bottom-up development strategy.

3) Control Assumptions: Control in this architecture is distributed across controllable nodes, while automated nodes execute fixed routines. The link layer plays a central role in coordinating entanglement generation, with the EGP components (Queue, QMM, FEU, Scheduler) jointly determining when and how entanglement attempts occur. The network layer assumes local tracking of Bell-state types and fidelity information, enabling routing and resource allocation decisions. Overall, the architecture assumes that each node maintains local state information and participates in coordinated scheduling, but without the explicit rule-based control model used by Van Meter *et al.*

C. Dür et al.

1) Design Philosophy: In contrast to the architectures by Van Meter *et al.* and Wehner *et al.*, which rely primarily on bipartite Bell-pair entanglement, the architecture proposed by Dür *et al.* [17], [30] treats multipartite entanglement as a native and fundamental resource. While other approaches acknowledge the future importance of multipartite states, Dür *et al.* build their entire architecture around the generation, distribution, and manipulation of multipartite graph states. Their design follows a bottom-up philosophy in which entanglement is generated prior to application requests. This proactive strategy is motivated by the observation that pre-distributed multipartite states can significantly reduce operational overhead for many quantum networking tasks. The architecture is structured around three operational phases, dynamic, static, and adaptive, which together define how entanglement is created, stored, and transformed.

2) Layering Model: The three operational phases motivate a layered protocol stack with the following components, see Figure 1:

1) *Physical Layer:* Establishes quantum channels between neighboring nodes and generates entanglement using a heralding scheme, enabling repeated attempts until success.
2) *Connectivity Layer:* Creates long-distance entanglement across multiple links using quantum repeaters. Both bipartite and multipartite states may be generated. Entanglement generation and routing protocols reside here and can be exchanged without affecting upper layers.
3) *Link Layer:* Coordinates the lower layers to construct the global *network state* and transform it into the specific graph states requested by applications.
4) *Network Layer:* Handles inter-network routing and enables requests across administrative boundaries. It establishes a *virtual network state* via quantum routers connected through multipartite entanglement. Combined with intra-network states, this forms an end-to-end network state between devices. The routing protocol itself is not specified and is typically implemented by network administrators. This layer is inspired by the Quantum Recursive Network Architecture (QRNA) [22].

Each layer may employ auxiliary protocols such as entanglement distillation, reachability checks, entanglement swapping and merging, error correction, and network monitoring. These are considered layer-independent building blocks.

3) Control Assumptions: The architecture is structured around three operational phases. First, in the *dynamic phase* Multipartite entangled states are proactively distributed across the network, forming a shared *network state*. This pre-distributed entanglement is essential for efficient graph-state generation. Then the generated entangled states are stored in quantum memories until needed, referred to as the *static phase*. Lastly, in the *adaptive phase*, upon receiving a request, or when a device failure occurs, the stored states are manipulated using LOCC operations to produce the desired graph state. Unlike earlier approaches relying on a central master node, Dür *et al.* emphasize distributed generation. Control is therefore distributed and heavily dependent on local LOCC operations, with the network state serving as a shared resource that can be adapted to different application needs.

This concept is the least experimentally tested. It misses detailed protocol specifications, is limited by current state of hardware and not developed for multi user handling. While lacking real-world applicability, the concept of multipartite entanglement is crucial for a future quantum internet. Additionally, the concept of pre-distributed entanglement can have advantages to on-demand schemes, as proposed by Dür and Van Meter.

D. Cross-Architecture Comparison

The three architectures analyzed above share the common goal of enabling scalable quantum networking, yet they differ fundamentally in their design philosophy, entanglement model, and control assumptions. The field remains highly exploratory, but also drifts towards more collaboration in terms of fundamental principles and goals [31]. In this section we offer a qualitative comparison of the models (also see I), which will lay the foundation the identify suitable univocal metrics for performance comparisons.

1) Entanglement Model: The most fundamental distinction lies in the choice of entanglement resource. Van Meter *et al.* and Wehner *et al.* rely on bipartite Bell-pair entanglement generated on demand and extended via entanglement swapping. This model aligns well with current experimental capabilities and supports incremental deployment. In contrast, Dür *et al.* treat multipartite entanglement as a native resource. Their architecture emphasizes pre-distributed network states, enabling efficient multi-party operations but imposing stricter requirements on quantum memory coherence and coordination. This distinction has direct implications for scalability, and application suitability.

2) Control Structure: Van Meter *et al.* adopt a hierarchical control model in which end-to-end connection setup is orchestrated through RuleSets constructed by the responder. This allows nodes to operate autonomously and asynchronously, reacting to events rather than waiting for centralized instructions. Wehner *et al.* pursue a more decentralized approach, with clearly defined responsibilities at each layer and a strong emphasis on local scheduling, fairness, and hardware abstraction, handling service requests, without global end-to-end orchestration. Their model is tightly coupled to re-

Dimension	**Van Meter *et al.***	**Wehner *et al.***	**Dür *et al.***
Entanglement Model	Bipartite, on-demand	Bipartite, on-demand	Multipartite, pre-distributed
Control Structure	Hierarchical, RuleSet-based	Decentralized, scheduler-driven	Distributed LOCC, phase-based
Hardware Assumptions	Moderate, repeater-centric	Strong hardware abstraction	Advanced memories, multipartite sources
Application Suitability	Teleportation, QKD, point-to-point DQC	Multi-user QKD, shared network services	Multi-party tasks, distributed computing, graph-state protocols
Deployment Readiness	Near-term feasible	Near-term feasible	Long-term, forward-looking

TABLE I
HIGH-LEVEL COMPARISON OF THE THREE QUANTUM INTERNET ARCHITECTURES.

alistic device capabilities. In the architecture by Dür *et al.*, requests are handled in the adaptive phase by selecting and transforming pre-distributed multipartite network states via distributed LOCC operations, rather than triggering real-time entanglement generation or end-to-end scheduling.

3) Hardware Assumptions: Van Meter *et al.* focus more on architectural formulation, while Wehner *et al.* explicitly target heterogeneous hardware platforms and integrate protocol design with realistic device constraints. In comparison, Dür *et al.* assumes advanced capabilities for multipartite entanglement generation and long-lived quantum memories, making their approach more forward-looking but less immediately deployable.

4) Application Suitability: All three architectures support point-to-point applications such as QKD, DQC and teleportation-based communication. However, multipartite-native architectures offer clear advantages for group-oriented tasks, including distributed consensus, clock synchronization, and certain distributed quantum computing models. Bipartite architectures, by contrast, are better suited for early-stage networks where reliability, incremental deployment, and compatibility with existing experimental platforms are essential.

Eventually, architectures should and can not be viewed as mutually exclusive alternatives but rather as complementary design philosophies. No architecture is complete or mature enough to be declared a definitive solution. Therefore, in the next section, we analyze metrics, which could provide a framework for making these trade-offs explicit and comparable across architectures eventually.

IV. Evaluating Quantum Networks

Proposed architectures for a quantum internet are still largely in a theoretical phase. Nevertheless, early and systematic comparison is crucial to guide ongoing research and design choices. This need intensifies as experimental platforms mature and the prospect of real-world quantum networks draws closer. However, evaluating and comparing quantum internet architectures remains challenging due to the lack of unified, architecture-independent evaluation criteria.

A. Metrics and Evaluation Frameworks

A wide range of metrics has been proposed for evaluating quantum networks, but these efforts remain fragmented and are often tied to specific protocols or assumptions. Existing work focuses on isolated layers of the quantum networking stack, making cross-architecture comparison difficult. This section structures evaluation metrics into different categories, highlighting the absence of a unified, architecture-aware metric suite.

1) Physical-Layer Metrics: Physical-layer metrics quantify the quality and stability of quantum states and the capabilities of underlying hardware components. The most widely used metric is *fidelity*, which measures the closeness between an actual shared quantum state and an ideal Bell or GHZ state [32]. Fidelity thresholds often determine whether entanglement is suitable for applications. Closely related is the *quantum bit error rate (QBER)*, an experimentally accessible proxy for fidelity, particularly in QKD-oriented evaluations [32]. Additional physical-layer metrics include *decoherence time* and *quantum memory lifetime*, which determine how long entanglement can be stored before degrading [33], *gate and measurement error rates* [12], or *channel loss* and *photon arrival probability* [34].

2) Link- and Path-Level Metrics: At the link level, the most common metric is the *entanglement generation rate / throughput*, typically expressed as the number of EPR pairs generated per second [35]. This rate is typically constraint to a fidelity threshold of pairs. Other commonly used link- and path-level metrics include *latency* or *processing overhead*, measuring the time from request initiation to the availability of entanglement at the destination [32], *success probability* of entanglement generation or entanglement swapping operations [34], *hop count*, which correlates with fidelity degradation and classical signaling delays [36], [37].

3) Application- and Network-Level Metrics: Continuous entanglement distribution capabilities of a network can be measured with *virtual neighborhood size*, number of other nodes with which a node can establish usable entanglement, and *virtual node degree*, number simultaneous possible entanglements with nodes [38]. Some metrics evaluate quantum networks from the perspective of end applications rather than underlying network behavior, including secret key rate in QKD [6]. A commonly used metric is the classically inspired *fairness*, calculated with *Jain's fairness index* [39], ensuring that performance does not depend disproportionately on request origin or network position [9]. Beyond physical and network-centric metrics, several works propose utility-based evaluation frameworks. *Quantum Network Utility Maximization (QNUM)* evaluates routing protocols by assigning utility values to entanglement distribution outcomes [40]. Similarly, Lee *et al.* introduce a capacity benchmarking framework that incorporates social and commercial value into a unified utility metric [41]. Such approaches can provide valuable high-level perspectives, however often are far from real world applicability.

B. Simulation-Centric Evaluation

As of now, large field studies are not feasible, therefore simulation tools and test bed experiments are the state-of-the-art evaluation methods. While test bed experiments are highly relevant, they are still limited to just a few nodes, and thus more relevant for specific use case studies, like employing a QKD protocol. Hence, for the evaluation of large scale quantum networks and protocol stack proposals, simulation is the state-of-the-art option for evaluation [12].

Simulators vary concerning assumptions, flexibility, and physical modeling. Most prominent simulation tools are *NetSquid* [42], SeQUeNCe [43], *QuNetSim* [44], *SimQN* [45] or *QUISP* [46]. Studies have shown that, under equal assumptions, such can support reproducible architectural comparison. For example, analytical performance comparisons of alternative node-level architectural designs have been validated in NetSquid by evaluating entanglement fidelity, throughput, and waiting-time distributions under realistic hardware assumptions [47]. Regarding, comparisons across different simulators, recent cross-validation studies show significant discrepancies between NetSquid and SeQUeNCe, driven by differences in timing models, noise assumptions, and architectural abstractions, while fidelity values remain consistent [48]. Such inconsistencies highlight a critical concern: the conclusions drawn from simulation-based evaluations need to be valid, posing the urgent need for standardized benchmarking and cross-platform validation to ensure the reliability of future quantum internet research.

V. Univocal Evaluation Metrics Across Architectures

As emphasized by Illiano *et al.* [10], the identification of univocal evaluation metrics for quantum networks is a central but unresolved challenge. While quantum internet architectures differ fundamentally, many metrics used in the literature are still tightly coupled to specific architectures or protocols and therefore do not support fair cross-architecture comparison. A metric is univocal if it is externally observable at the service boundary and retains its meaning across architectural realizations. In this section, we identify a minimal set of *univocal quantitative metrics*, enabling objective comparison without prescribing a particular architectural model. Additionally, we try to identify missing metrics that could be subject to future efforts.

A. Design Principles

We base this section on the architectural principles outlined in [31], providing design guidelines intended to remain valid across technologies, architectures, and further stages of deployment. The quantum internet is primarily a service for creating, managing, and delivering entanglement between remote nodes. Metrics should therefore view *entanglement as a fundamental service*, quantifying the delivery, quality, and availability of entangled states rather than internal protocol operations. As architecture must accommodate *heterogeneous hardware platforms*, link technologies, and repeater generations, metrics should not rely on technology-specific assumptions and must remain interpretable across diverse implementations. Due to finite coherence times and limited quantum memory lifetimes, *time as a critical resource* in quantum networking. Metrics must therefore capture latency, throughput, and time-dependent resource consumption. Eventually, architectures must tolerate imperfect devices, lossy channels, component failures and multiple users or applications. Evaluation metrics should reflect *robustness and availability* of services. Given the early stage of quantum networking technology, metrics should remain valid under future evolution, avoiding tight coupling to current design choices.

B. Entanglement Metrics

These metrics quantify whether and how effectively entanglement reaches the application layer.

- *Throughput (T):* The average rate at which usable entanglement resources are delivered to the application layer, measured in entangled states per unit time [6], [49]–[52]. Throughput can be defined as

$$T = \frac{N_{\text{delivered}}}{\Delta t} \tag{1}$$

  where $N_{\text{delivered}}$ denotes the number of entangled states successfully delivered to the application layer within the observation interval Δt.
- *End-to-End Fidelity (F_{e2e}):* Fidelity of the delivered entangled state as observed by the application, independent of the generation mechanism [48]–[51], [53]. For a delivered quantum state ρ_{del} and an ideal target state $|\psi_{\text{ideal}}\rangle$, the end-to-end fidelity is defined as

$$F_{e2e} = \langle \psi_{\text{ideal}} | \, \rho_{\text{del}} \, | \psi_{\text{ideal}} \rangle \tag{2}$$

C. Temporal Metrics

Temporal performance metrics capture the time-related aspects of entanglement delivery [51], [53], [54].

- *Request Latency (L_r):* The elapsed time between the initiation of an entanglement service request and its completion at the requesting node. Formally,

$$L_r = t_{\text{complete}} - t_{\text{request}} \tag{3}$$

  where t_{request} denotes the time at which the request is issued, and t_{complete} denotes the time at which all requested entanglement resources have been successfully delivered to the application layer, including all contributing delays, such as queuing, entanglement generation attempts, entanglement swapping, and classical signaling [27].

- *Unit Latency (L_u):* The average time required to generate and deliver a single entanglement unit. It is defined as

$$L_u = \mathbb{E}\left[\frac{L_r}{N_u}\right] \tag{4}$$

where N_u denotes the number of entanglement units requested. The entanglement unit is kept abstract to support both bipartite and multipartite entangled resources [27].
- *Scaled Latency (L_s):* The request latency normalized by the number of entanglement units requested, capturing the effective latency per delivered unit for a given request [27]:

$$L_s = \frac{L_r}{N_u} \tag{5}$$

This metric reflects the impact of scheduling, contention, and congestion. In the absence of concurrent requests, L_s reduces to the unit latency L_u.

D. Robustness & Fairness Metrics

We note *Fairness (J_i)* as the sole currently utilized metric in this category. It calculates the balance of resource allocation across request origins or network positions (Jain's index). The metric is resource type independent and can be defined for several measures i (e.g., T or L_r) [45], [52], [55].

$$J_i = \frac{\left(\sum_{k=1}^{K} x_k^{(i)}\right)^2}{K\sum_{k=1}^{K}\left(x_k^{(i)}\right)^2} \tag{6}$$

where $x_k^{(i)}$ denotes the value of performance measure i observed for request origin or network position k, and K is the total number of considered origins or positions. The fairness index satisfies $0 < J_i \leq 1$, with $J_i = 1$ indicating perfectly fair allocation.

Classical networking metrics often include *availability* [56], describing the percentage of time a user is able to request a service over one or multiple paths. A inspired metric could be interesting for a quantum network metric, however is out of scope for this work.

E. Metric Applicability and Taxonomy Validation

As current quantum network protocol stacks do not yet expose a fully specified application layer, we validate our metrics by instantiating the metrics at the highest available service boundary provided by an existing research prototype. Our study is based on the NetSquid implementation outlined in [29], including physical-, link-, and network-layer functionality. Metrics are therefore measured at the network service interface. We consider a minimal three-node repeater topology (RA-$R0$–RB) with symmetric entanglement requests between end nodes. To demonstrate that the taxonomy remains applicable across different operational regimes, we evaluate two entanglement-generation models, a *MHP* scheme as implemented in the reference protocol, and an idealized generation model, referred to as *Magic*. In addition, baseline and degraded link conditions are considered to assess metric behavior under controlled noise. As expected, idealized entanglement generation results in higher values compared to the heralded scheme, while degradation leads to a pronounced reduction in end-to-end fidelity, caused by introduced noise. Under symmetric demand, fairness remains close to one in all cases (see Appendix Table II and III). These results do not aim to provide architectural insights, instead, they demonstrate that the proposed univocal metrics can be instantiated and measured consistently in an existing protocol stack, thereby validating the practical applicability of the taxonomy.

F. Discussion

The comparison of quantum internet architectures showed, that concepts fundamentally differ in terms of real world applicability. We noticed, the development of a quantum internet will be a joined effort, and will require collaboration across different fields. Based on the state of current architectures, we tried to identify a set of representative metrics, offering a fair comparison between models across fundamentally different architectural designs. We acknowledge, that a comprehensive empirical validation across simulators, topologies, or hardware platforms is beyond the scope of this work and left for future benchmarking studies. This also highlights the need for benchmarking of large scale quantum networks, with a set of representative tasks. Our metrics provide a foundation for future standardization efforts, however we note that a full implementation of large scale quantum network in any kind of form is crucial fur further progress as we are still missing relevant understanding of the purpose of a quantum internet. Without the full knowledge, the identification of univocal metrics is a hard tasks, needing iterative improvement over the next development stages.

VI. Conclusion and Future Work

This paper examined the progress regarding several quantum internet architectures and addressed the lack of univocal metrics for their evaluation. By critically analyzing state-of-the-art protocol stacks and proposing a taxonomy that separates qualitative architectural decisions from quantitative performance metrics, we provide a structured framework for evaluating competing designs. Our analysis highlights trade-offs between scalability, feasibility, entanglement models, and performance. The proposed taxonomy enables these trade-offs to be made explicit and comparable. Ultimately, we envision this framework contributing to the development of standardized reference models for the quantum internet.

Future work needs to further cross-check simulation tools and create benchmarks to compare different architectural approaches under unified assumptions. This would enable us to further validate our proposed metric stack. Regarding the development of a quantum internet architecture, one of the most critical open challenges is scalable multiplexing. From our work, no architecture could provide details on how concurrent users could access a quantum internet with equal efficiency and fairness.

Data and Code Availability

The code used to generate the results of this work and all data are publicly available at: https://github.com/JohannesWittmann9/QuantumNetworkMetrics

References

[1] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.

[2] C. H. Bennett, G. Brassard, and A. K. Ekert, “Quantum cryptography,” *Scientific American*, vol. 267, no. 4, pp. 50–57, 1992.

[3] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, “Quantum cryptography,” *Reviews of modern physics*, vol. 74, no. 1, p. 145, 2002.

[4] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller, “Quantum repeaters based on entanglement purification,” *Physical Review A*, vol. 59, no. 1, p. 169, 1999.

[5] M. Zwerger, W. Dür, and H. Briegel, “Measurement-based quantum repeaters,” *Physical Review A—Atomic, Molecular, and Optical Physics*, vol. 85, no. 6, p. 062326, 2012.

[6] K. Azuma, S. E. Economou, D. Elkouss, P. Hilaire, L. Jiang, H.-K. Lo, and I. Tzitrin, “Quantum repeaters: From quantum networks to the quantum internet,” *Reviews of Modern Physics*, vol. 95, no. 4, p. 045006, 2023.

[7] R. Van Meter, *Quantum networking*. John Wiley & Sons, 2014.

[8] H. J. Kimble, “The quantum internet,” *Nature*, vol. 453, no. 7198, pp. 1023–1030, 2008.

[9] S. Wehner, D. Elkouss, and R. Hanson, “Quantum internet: A vision for the road ahead,” *Science*, vol. 362, no. 6412, p. eaam9288, 2018.

[10] J. Illiano, M. Caleffi, A. Manzalini, and A. S. Cacciapuoti, “Quantum internet protocol stack: A comprehensive survey,” *Computer Networks*, vol. 213, p. 109092, 2022.

[11] Y. Li, H. Zhang, C. Zhang, T. Huang, and F. R. Yu, “A survey of quantum internet protocols from a layered perspective,” *IEEE Communications Surveys & Tutorials*, vol. 26, no. 3, pp. 1606–1634, 2024.

[12] V. Kumar, C. Cicconetti, M. Conti, and A. Passarella, “Quantum internet: Technologies, protocols, and research challenges,” *International Journal of Networked and Distributed Computing*, vol. 13, no. 2, p. 22, 2025.

[13] H. Zimmermann, “Osi reference model-the iso model of architecture for open systems interconnection,” *IEEE Transactions on communications*, vol. 28, no. 4, pp. 425–432, 1980.

[14] V. Cerf and R. Kahn, “A protocol for packet network intercommunication,” *IEEE Transactions on communications*, vol. 22, no. 5, pp. 637–648, 1974.

[15] A. Einstein, B. Podolsky, and N. Rosen, “Can quantum-mechanical description of physical reality be considered complete?” *Physical Review*, vol. 47, no. 10, pp. 777–780, 1935.

[16] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, “Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels,” *Physical Review Letters*, vol. 70, no. 13, pp. 1895–1899, 1993.

[17] A. Pirker, J. Wallnöfer, and W. Dür, “Modular architectures for quantum networks,” *New Journal of Physics*, vol. 20, no. 5, p. 053054, 2018.

[18] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi, “Quantum internet: Networking challenges in distributed quantum computing,” *IEEE Network*, vol. 34, no. 1, pp. 137–143, 2019.

[19] A. Singh, K. Dev, H. Siljak, H. D. Joshi, and M. Magarini, “Quantum internet—applications, functionalities, enabling technologies, challenges, and research directions,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2218–2247, 2021.

[20] W. J. Munro, N. Piparo, J. Dias, M. Hanks, and K. Nemoto, “Designing tomorrow’s quantum internet,” *AVS Quantum Science*, vol. 4, no. 2, 2022.

[21] R. Van Meter, T. D. Ladd, W. J. Munro, and K. Nemoto, “System design for a long-line quantum repeater,” *IEEE/ACM Transactions On Networking*, vol. 17, no. 3, pp. 1002–1013, 2008.

[22] R. Van Meter, J. Touch, and D. Horsman, “Recursive quantum repeater networks,” *arXiv preprint arXiv:1105.1238*, 2011.

[23] R. Van Meter and J. Touch, “Designing quantum repeater networks,” *IEEE Communications Magazine*, vol. 51, no. 8, pp. 64–71, 2013.

[24] R. Van Meter, R. Satoh, N. Benchasattabuse, K. Teramoto, T. Matsuo, M. Hajdušek, T. Satoh, S. Nagayama, and S. Suzuki, “A quantum internet architecture,” in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2022, pp. 341–352.

[25] T. Matsuo, C. Durand, and R. Van Meter, “Quantum link bootstrapping using a ruleset-based communication protocol,” *Physical Review A*, vol. 100, no. 5, p. 052320, 2019.

[26] T. Matsuo, “Simulation of a dynamic, ruleset-based quantum network,” *arXiv preprint arXiv:1908.10758*, 2019.

[27] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpedek, M. Pompili, A. Stolk, P. Pawełczak, R. Knegjens, J. de Oliveira Filho *et al.*, “A link layer protocol for quantum networks,” in *Proceedings of the ACM special interest group on data communication*, 2019, pp. 159–173.

[28] W. Kozlowski and S. Wehner, “Towards large-scale quantum networks,” in *Proceedings of the sixth annual ACM international conference on nanoscale computing and communication*, 2019, pp. 1–7.

[29] W. Kozlowski, A. Dahlberg, and S. Wehner, “Designing a quantum network protocol,” in *Proceedings of the 16th international conference on emerging networking experiments and technologies*, 2020, pp. 1–16.

[30] A. Pirker and W. Dür, “A quantum network stack and protocols for reliable entanglement-based networks,” *New Journal of Physics*, vol. 21, no. 3, p. 033003, 2019.

[31] W. Kozlowski, S. Wehner, R. Van Meter, B. Rijsman, A. S. Cacciapuoti, M. Caleffi, and S. Nagayama, “Rfc 9340: Architectural principles for a quantum internet,” 2023.

[32] D. P. Castro, J. Fernández-Herrerín, A. Fernández-Vilas, M. Fernández-Veigaa, and R. P. Díaz-Redondo, “Simulation of entanglement based quantum networks for performance characterization,” *arXiv preprint arXiv:2501.03210*, 2025.

[33] P. C. Humphreys, N. Kalb, J. P. Morits, R. N. Schouten, R. F. Vermeulen, D. J. Twitchen, M. Markham, and R. Hanson, “Deterministic delivery of remote entanglement on a quantum network,” *Nature*, vol. 558, no. 7709, pp. 268–273, 2018.

[34] J. Halder, A. Rajabov, R. Bassoli, F. H. Fitzek, and G. P. Fettweis, “Optimal routing and end-to-end entanglement distribution in quantum networks,” *Scientific Reports*, vol. 14, no. 1, p. 19262, 2024.

[35] Y. Lee, E. Bersin, A. Dahlberg, S. Wehner, and D. Englund, “A quantum router architecture for high-fidelity entanglement flows in quantum networks,” *npj Quantum Information*, vol. 8, no. 1, p. 75, 2022.

[36] S. Khatri, C. T. Matyas, A. U. Siddiqui, and J. P. Dowling, “Practical figures of merit and thresholds for entanglement distribution in quantum networks,” *Physical Review Research*, vol. 1, no. 2, p. 023032, 2019.

[37] M. Caleffi, M. Amoretti, D. Ferrari, J. Illiano, A. Manzalini, and A. S. Cacciapuoti, “Distributed quantum computing: A survey,” *Computer Networks*, vol. 254, p. 110672, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128624005048

[38] Á. G. Iñesta and S. Wehner, “Performance metrics for the continuous distribution of entanglement in multi-user quantum networks,” *arXiv preprint arXiv:2307.01406*, 2023.

[39] R. K. Jain, D.-M. W. Chiu, W. R. Hawe *et al.*, “A quantitative measure of fairness and discrimination,” *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, vol. 21, no. 1, pp. 2022–2023, 1984.

[40] G. Vardoyan and S. Wehner, “Quantum network utility maximization,” in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 1. IEEE, 2023, pp. 1238–1248.

[41] Y. Lee, W. Dai, D. Towsley, and D. Englund, “Quantum network utility: A framework for benchmarking quantum networks,” *Proceedings of the National Academy of Sciences*, vol. 121, no. 17, p. e2314103121, 2024.

[42] T. Coopmans, R. Knegjens, A. Dahlberg, D. Maier, L. Nijsten, J. de Oliveira Filho, M. Papendrecht, J. Rabbie, F. Rozpedek, M. Skrzypczyk *et al.*, “Netsquid, a network simulator for quantum information using discrete events,” *Communications Physics*, vol. 4, no. 1, p. 164, 2021.

[43] X. Wu, A. Kolar, J. Chung, D. Jin, T. Zhong, R. Kettimuthu, and M. Suchara, “Sequence: a customizable discrete-event simulator of quantum networks,” *Quantum Science and Technology*, vol. 6, no. 4, p. 045027, 2021.

[44] S. DiAdamo, J. Nötzel, B. Zanger, and M. M. Beşe, “Qunetsim: A software framework for quantum networks,” *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–12, 2021.

[45] L. Chen, K. Xue, J. Li, N. Yu, R. Li, Q. Sun, and J. Lu, "Simqn: A network-layer simulator for the quantum network investigation," *IEEE Network*, vol. 37, no. 5, pp. 182–189, 2023.

[46] R. Satoh, M. Hajdušek, N. Benchasattabuse, S. Nagayama, K. Teramoto, T. Matsuo, S. A. Metwalli, P. Pathumsoot, T. Satoh, S. Suzuki *et al.*, "Quisp: a quantum internet simulation package," in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2022, pp. 353–364.

[47] G. Vardoyan, M. Skrzypczyk, and S. Wehner, "On the quantum performance evaluation of two distributed quantum architectures," *ACM SIGMETRICS Performance Evaluation Review*, vol. 49, no. 3, pp. 30–31, 2022.

[48] J. Chung, M. Hajdušek, N. Benchasattabuse, A. Kolar, A. Singal, K. S. Soon, K. Teramoto, A. Zang, R. Kettimuthu, and R. Van Meter, "Cross-validating quantum network simulators," in *IEEE INFOCOM 2025-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2025, pp. 1–6.

[49] C. H. Chan, C. Jain, E. Kissel, W. Wu, E. Barnes, S. E. Economou, and I. Monga, "Theoretical analysis and simulations of memory-based and all-photonic quantum repeaters and networks," *arXiv preprint arXiv:2512.23111*, 2025.

[50] H. Miller, C. Zhan, M. Bishof, J. Chung, H. Xu, P. Kumar, and R. Kettimuthu, "Simulation of a heterogeneous quantum network," *arXiv preprint arXiv:2512.04211*, 2025.

[51] M. Ghaderibaneh, H. Gupta, and C. Ramakrishnan, "Generation and distribution of ghz states in quantum networks," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 1. IEEE, 2023, pp. 1120–1131.

[52] C. Li, T. Li, Y.-X. Liu, and P. Cappellaro, "Effective routing design for remote entanglement generation on quantum networks," *npj Quantum Information*, vol. 7, no. 1, p. 10, 2021.

[53] E. Shchukin, F. Schmidt, and P. van Loock, "Waiting time in quantum repeaters with probabilistic entanglement swapping," *Physical Review A*, vol. 100, no. 3, p. 032322, 2019.

[54] G. Ni, L. Ho, and H. Claussen, "Adaptive optimization of latency and throughput with fidelity constraints in quantum networks using deep neural networks," *arXiv preprint arXiv:2505.12459*, 2025.

[55] C. Cicconetti, M. Conti, and A. Passarella, "Resource allocation in quantum networks for distributed quantum computing," in *2022 IEEE International conference on smart computing (SMARTCOMP)*. IEEE, 2022, pp. 124–132.

[56] W. Zou, M. Janic, R. Kooij, and F. Kuipers, "On the availability of networks," 2007.

Appendix

Metrics Experiments

We offer detailed results of our metrics implementation from Section V-E in Table II and III. Note that the unit and scaled latency metrics (L_u, L_s) in Table II are always equal. This is caused by missing concurrency of request, as elaborated in Section V-C.

TABLE II
Performance metrics comparison: Magic vs MHP Entanglement Generation across ideal and degraded network conditions. We compare request from both end nodes RA and RB.

Mode	**Scenario**	T **(states/s)**	L_r **(ms)**	L_u **(ms)**	L_s **(ms)**	F_{e2e}	J_T
Magic EG	Ideal (RA)	1420.2	1.43	0.72	0.72	0.970	0.983
	Ideal (RB)	1408.3	1.45	0.73	0.73	0.970	
	Degraded (RA)	1460.5	1.39	0.69	0.69	0.659	0.991
	Degraded (RB)	1498.8	1.34	0.67	0.67	0.659	
MHP EG	Ideal (RA)	1215.1	1.68	0.84	0.84	0.970	0.983
	Ideal (RB)	1251.4	1.63	0.81	0.81	0.970	
	Degraded (RA)	1102.8	1.84	0.92	0.92	0.659	0.990
	Degraded (RB)	1091.8	1.84	0.92	0.92	0.659	

TABLE III
Fairness metrics for resource allocation

Configuration	J_T **(Throughput)**	J_{Lr} **(Request Latency)**	J_{Fe2e} **(Fidelity)**
Magic EG, Ideal	0.983	0.978	1.000
Magic EG, Degraded	0.991	0.990	1.000
MHP EG, Ideal	0.983	0.979	1.000
MHP EG, Degraded	0.990	0.990	1.000

HHL: A quantum algorithm for solving linear systems of equations

Tom Naftali Körner
LMU Munich
Munich, Germany
T.Koerner@campus.lmu.de
Supervisor: Karl Fürlinger

***Abstract*—Quantum algorithms aim to achieve an advantage over classical methods, offering the potential for exponential speedups in specific computational tasks. One such fundamental challenge is the solution of Linear System Problems (LSP). This paper analyzes the Harrow Hassidim Lloyd (HHL) algorithm, which is an existing, efficient quantum method developed to find the solution to such systems. The HHL algorithm is notable for offering an exponential speedup in comparison to the best-known classical method, positioning it as a key tool for implementing scalable quantum machine learning models where linear algebraic operations are crucial. However, this speed advantage is constrained, as it does not apply to the full readout of the solution vector, thus restricting its applicability to scenarios where only statistical information or further quantum processing is required. Nevertheless, the principles demonstrated by HHL hold significant promise for future applications, particularly in complex simulations like tensor network simulation for many-body simulations and in accelerating high-performance computing through advanced matrix operations.**

***Index Terms*—Quantum Algorithms, HHL, LSP, Classiq, Cirq**

I. Introduction

Over four decades ago, Richard Feynman, one of the most influential physicists of the 20th century and a Nobel Prize winner, imagined a computer that could exploit the very principles of quantum mechanics. In his 1981 talk “Simulating physics with computers,” Feynman argued that classical computers are inherently limited in their ability to model quantum systems and thus was the first to suggest a “quantum computer” that could efficiently simulate the behavior of nature itself [1].

The so-called *Quantum advantage* refers to the demonstrated cases where quantum computers can outperform classical computers in specific tasks. Proven examples include *quantum simulation*, which represents a rather trivial but fundamental case, as quantum systems can be inherently better mapped by a quantum computer. But also famous examples like the Quantum Key Distribution (QKD) protocols in *quantum cryptography*, which gained a lot of attention due to *Shor's algorithm*threatening classical RSA encryption schemes [2]. Another notable example is *Grover's algorithm* for searching through unsorted databases exponentially more efficiently than any classical counterpart so far [3]. Nevertheless, the development of practical and large-scale quantum computers remains an ongoing area of active research. Interestingly, the first programming language existed before the first computer, and history might now repeat itself with the development of quantum algorithms preceding fully realized quantum computers [4].

The seminar paper will present another very powerful quantum algorithm intuitively, the so-called Harrow Hassidim Lloyd (HHL) algorithm [5], which is an efficient method to solve Linear System Problems (LSP). This algorithm will be defined and explained in section II. Moreover, the HHL algorithm will be analyzed in detail in section III by explaining the principles as well as the limits of the method. Finally, a reference implementation is presented in section IV.

II. Background

A. Linear System Problem

A LSP describes the task of determining the unknown vector $\vec{x}$ that satisfies a given system of linear equations, as can be seen in eq. 1.

$$A \cdot \vec{x} = \vec{b} \iff \vec{x} = A^{-1} \cdot \vec{b} \tag{1}$$

Hereby, A is a hermitian[1], and thus quadratic matrix with dimension $N \times N$ and $\vec{x}$ and $\vec{b}$ are vectors with dimension N.

B. Classical Algorithms

Before introducing the quantum algorithms, it is important to name and explain some classical references to be able to classify the superiority of the quantum alternative.

1) Gaussian: The most widely known method to retrieve the solution of an LSP is the so-called Gauss Elimination Method (GEM), whereby the first step is to create a new matrix A' as in eq. 2 that is then possible to bring into a so-called row echelon form [6] with equivalent transformations as shown in eq. 2.

$$A' = \begin{pmatrix} A & b \end{pmatrix} \rightsquigarrow \begin{pmatrix} I & X \\ 0 & 0 \end{pmatrix} \tag{2}$$

[1] Any quadratic matrix can be converted to a hermitian matrix by creating a new matrix $A' = \begin{pmatrix} 0 & A \\ A^{\dagger} & 0 \end{pmatrix}$ with dimensions $2N \times 2N$. Then the equation $A'\vec{y} = \begin{pmatrix} \vec{b} \\ 0 \end{pmatrix}$ would be solved to obtain the solution $\begin{pmatrix} 0 \\ \vec{x} \end{pmatrix}$ [5].

Although this method is relatively simple to understand and straightforward to implement, it unfortunately exhibits cubic scaling with respect to the matrix dimension N [7].

2) Conjugate Gradient Method: A much more powerful method is the so-called Conjugate Gradient Method (CGM), the currently fastest known classical solving algorithm for the LSP. In comparison to the GEM, whose complexity is $\mathcal{O}(N^3)$, it only scales linearly with respect to the matrix dimension N. As the name suggests, the algorithm searches for the solution in the direction of the steepest descent after calculating it, depending on an initial guess. The loss function $l(\vec{x'})$ to be minimized is, hereby, simply given by the difference between the two sides of eq. 1:

$$l(\vec{x'}) = \vec{b} - A\vec{x'} \tag{3}$$

For the algorithm to be efficient, the matrix A needs to be sparse. Sparsity describes the ratio of the zero elements in comparison to the number of rows and columns of the matrix. Therefore, a sparse matrix contains only a very limited number of non-zero elements and is otherwise called a dense matrix [7].

C. Quantum Algorithms

Quantum algorithms aim to solve computational problems more efficiently than classical algorithms, often targeting an exponential speedup for specific tasks. They have found applications in areas such as optimization, simulation of quantum systems, and Quantum Machine Learning, where they can be used for clustering, classification, and pattern recognition in large datasets. In the context of quantum machine learning, one of the most widely used algorithms for this purpose is the HHL algorithm, introduced in 2008 by Aram Harrow, Avinatan Hassidim, and Seth Lloyd. The reason for this is that efficiently solving linear systems of equations is crucial, as many models rely on linear algebraic operations, making the HHL seemingly a key tool for implementing scalable quantum machine learning models [8].

D. Related work

There are a various amount of papers presenting the HHL Algorithm and others that show use cases. As this seminar paper aims to present the algorithm understandably, a focus is laid on explaining the basics of the algorithm, and thus fundamental papers, instead of applications.

Therefore, the paper by Harrow, Hassidim, and Lloyd [5] is presented, which explains the HHL algorithm in mathematical detail and provides a thorough analysis of its runtime, optimality, and error behavior, and is cited several thousand times.

Furthermore, [7] is a self-contained paper that provides a step-by-step walkthrough of the algorithm, covering both the quantum circuit and a simple numerical example to support understanding. It also includes the necessary background knowledge, such as Quantum Phase Estimation, the Quantum Fourier Transform, and fundamental concepts of Quantum Mechanics and Quantum Computing. Moreover, it presents an implementation on IBM Quantum and discusses the obtained results from the author's own experiments.

Finally, [8] presents a paper that thoroughly discusses the limitations of the HHL algorithm and seeks to clarify common misconceptions about its applicability and potential use cases.

III. HHL Algorithm

A. Description

The goal of the HHL algorithm is to find the solution quantum mechanically by introducing quantum states. Thus, the solution state $|x\rangle$ to the linear system problem $A|x\rangle = |b\rangle$ is to be found in the form $A^{-1}|b\rangle = \sum_{i=0}^{2^{n_b}-1} \lambda_i^{-1} b_i |u_i\rangle$. The algorithm starts by encoding the input vector $|b\rangle$ using amplitude encoding and simulating the matrix A by encoding it as the Hamiltonian of a unitary gate. Furthermore, Quantum Phase Estimation (QPE) is used to find the eigenphase of the unitary gate, which is proportional to the eigenvalue of A, and store this value using basis encoding. This stored eigenvalue is then used to perform a controlled rotation on a single ancilla qubit. This step is carefully designed so that, upon successful measurement of the ancilla qubit as $|1\rangle_a$, the desired factor $\lambda_i^{-1} b_i$ appears in the coefficients of the quantum state. Since the registers are entangled at this point, an uncomputation using inverse QPE (IQPE) is required to disentangle them and leave the final answer, $|x\rangle$, stored in the appropriate register [7].

B. Quantum Circuit

One way of implementing the HHL Algorithm is a quantum circuit composed of a total of $n_b + n + 1$ qubits, which are initialized to the $|0\rangle$ state and partitioned into three registers. The *b-register* consists of n_b qubits and is used to encode the components of the input vector $|b\rangle$ using amplitude encoding, as well as to store the final solution vector $|x\rangle$. The *c-register* contains n clock qubits. It is related to the time in the controlled rotation of the QPE part, and it stores the phase values of the eigenvalues of the matrix A using basis encoding. The final qubit is the single *ancilla qubit* $|\psi\rangle_a$ which is utilized to extract the solution before being discarded by performing the measurement [7]. In the following, the steps of the quantum circuit are discussed in detail.

1) State preparation: The total $n_b + n + 1$ qubits are initialized as $|\Psi_0\rangle = |0\cdots 0\rangle_b |0\cdots 0\rangle_c |0\rangle_a$. The objective of the State Preparation step is to rotate the $|0\cdots 0\rangle_b$ state in the b-register so that its amplitudes correspond to the coefficients of the input vector $\vec{b}$, resulting in the state $|b\rangle$. The state of the circuit after this operation is then $|\Psi_1\rangle = |b\rangle_b |0\cdots 0\rangle_c |0\rangle_a$. The specific state preparation operation is, thus, dependent on the actual values of $\vec{b}$ [7].

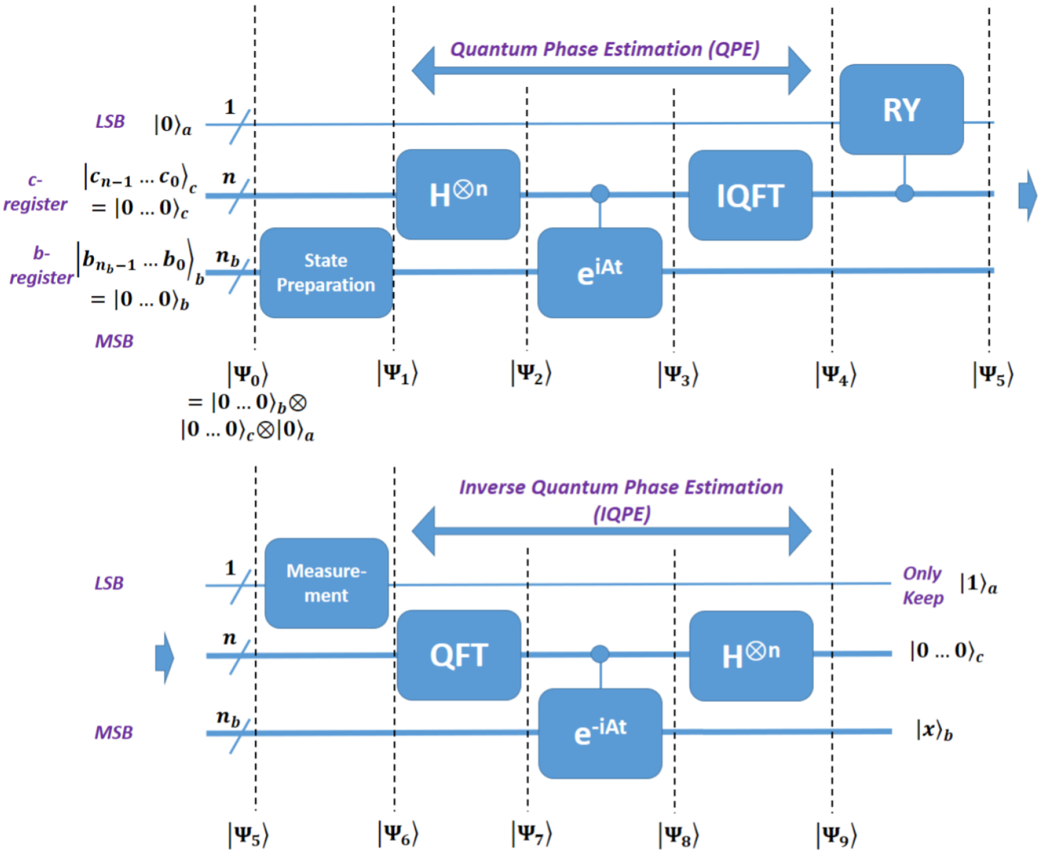


Fig. 1. Quantum Circuit Diagram illustrating the various steps and different registers of a potential implementation for the HHL Algorithm [7]. This includes the three registers b, c, and the register for the ancilla qubit, as well as the steps State Preparation, QPE, Controlled rotation (in the upper part), and the Measurement and IQPE (in the lower part).

2) Quantum Phase Estimation: QPE is an eigenvalue estimation subroutine that aims to estimate the phase ϕ of the eigenvalues of the unitary rotation matrix $U = \exp(iAt)$ in the controlled gate from the c-register to the b-register. This gate encodes the linear system matrix A as its Hamiltonian[2]. The QPE process has three components: superposition of the clock qubits through Hadamard gates, controlled rotation, and Inverse Quantum Fourier Transform (IQFT). The process begins by applying Hadamard gates to the n clock qubits, creating a superposition state, $|\Psi_2\rangle = |b\rangle \frac{1}{2^{n/2}}(|0\rangle + |1\rangle)^{\otimes n}|0\rangle$. Next, controlled-rotation gates, U^{2^r} with r being the index of the clock qubit in the c-register, are applied to the b-register, with the clock qubits acting as the controllers. Finally, IQFT is applied only to the clock qubits. This causes the clock qubits to encode the eigenvalues of A (or at least scaled versions $\tilde{\lambda}_j$) using basis encoding, resulting in the state $|\Psi_4\rangle = \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle |\tilde{\lambda}_j\rangle |0\rangle_a$ [7].

3) Controlled Rotation and Measurement: After the eigenvalues $\tilde{\lambda}_j$ have been encoded in the c-register, a controlled rotation is performed on the single ancilla qubit, which is also controlled by the c-register. This rotation is equivalent to an RY gate, and it transforms the ancilla qubit from $|0\rangle_a$ to the superposition:

$$\sqrt{1 - \frac{C^2}{\tilde{\lambda}_j^2}}|0\rangle_a + \frac{C}{\tilde{\lambda}_j}|1\rangle_a \tag{4}$$

The constant C is, thereby, chosen to maximize the probability of measuring the state $|1\rangle_a$. After that, the ancilla qubit is measured. If the measurement result is $|0\rangle_a$, the result is discarded, and the circuit must be rerun. If $|1\rangle_a$ is obtained, the state has been designed so that the desired component $\lambda_i^{-1} b_i$ appears in the coefficients of the quantum state of the circuit [7].

4) Inverse Quantum Phase Estimation (IQPE): Uncomputation is required because after the controlled rotation, the b-register and c-register are entangled, and this entanglement prevents obtaining the correct answer upon measurement. This disentangling is performed by using IQPE, a circuit composed of applying Quantum Fourier Transform (QFT) to the clock qubits, followed by applying the inverse controlled rotation with $U^\dagger = \exp(-iAt)$, and finally Hadamard gates. The combined effect of these operations disentangles the b-register from the clock qubits, returning the clock qubits to the $|0\rangle^{\otimes n}$

[2] In quantum mechanics, the time evolution of a state can be performed by applying the operation $U = \exp(iHt)$, H being the Hamiltonian of the system. In general, any unitary matrix U can be written as $U = \exp(iA)$.

state. The final state $|\Psi_9\rangle$ has the solution $|x\rangle$ stored in the b-register, unentangled from the now-reset clock and measured ancilla qubits [7].

C. Limits of the algorithm

Up until now, the HHL algorithm has been presented as a promising and ideal quantum algorithm for solving systems of linear equations. Unfortunately, there are some very limiting factors to the algorithm, which drastically reduce the range of possible use cases and leave only relatively few suitable applications for HHL. Thus, as argued in [8], defining HHL as a general solving algorithm for linear systems of equations is actually not entirely correct. In the following, these arguments, as well as some aspects of the error analysis of HHL, will be discussed in detail.

Whereby it is mentioned above that we have an exponential speedup over the classical alternative of the LSP solving algorithm, this actually does not hold for the readout of $\vec{x}$, which would be of order $\mathcal{O}(N)$. Thus, it is only possible to use the algorithm's result for limited statistical information about $\vec{x}$, or for further calculations such as the expectation value of an operator, the matrix A, or simply a scalar product with another state $\vec{x}'$.
The reason for this is that the result obtained after applying the algorithm is not the actual vector $\vec{x}$, but rather a quantum state that approximately represents the desired vector $\vec{x}$ [8]. Consequently, there is a not entirely negligible error that occurs, whereby the main error source is the phase estimation step [5].
Moreover, the aforementioned complexity of $\mathcal{O}(\log(N))$ firstly only accounts for the execution of the algorithm, and secondly, it is actually a simplified representation. In reality, it depends on the system's condition number κ, the sparsity s, as well as the desired (maximum) error ϵ, as can be seen in Eq. 5 [7].

$$\mathcal{O}\left(\frac{\kappa \cdot s \cdot \log(N)}{\epsilon}\right) \tag{5}$$

Thus, certain conditions for the matrix A must be fulfilled, such as sparsity, robust invertibility, or being well-conditioned, which also implies the aforementioned $\kappa = \left|\frac{\lambda_{\max}}{\lambda_{\min}}\right|$ being small defined by the ratio between the largest and smallest eigenvalue of A [8]. If, in addition to that, amplitude amplification is to be used, the overall runtime needs to be multiplied by an additional factor of $\mathcal{O}(\kappa)$ [5].

In addition to all that, according to [8], it is possible that state preparation might scale polynomially with the vector's dimension N. The same problem might occur for the unitary transformation, which does not necessarily scale logarithmically with the matrix's dimension N. This, however, would be a problem that all quantum algorithms need to address first, as state preparation is a fundamental requirement for any quantum algorithm.

For a numerical example and step-by-step walkthrough to understand the in closer detail, I again refer to [7].

IV. Reference Implementation and Results

In the following, two reference implementations are presented to show the applications of the HHL algorithm in a more realistic scenario: one provided by Classiq [9], an Israeli quantum software company founded in 2020, and one based on Cirq [10], an open-source quantum computing framework developed by Google for designing, simulating, and running quantum circuits on near-term quantum hardware. Both implement different additional methods for applying the HHL algorithm efficiently and compare them to the classical alternative and the naive quantum algorithm. This section aims to show that, firstly, additional research is possible in order to use the algorithm more efficiently. Secondly, it provides a deeper understanding of what is possible with this algorithm.

A. Classiq

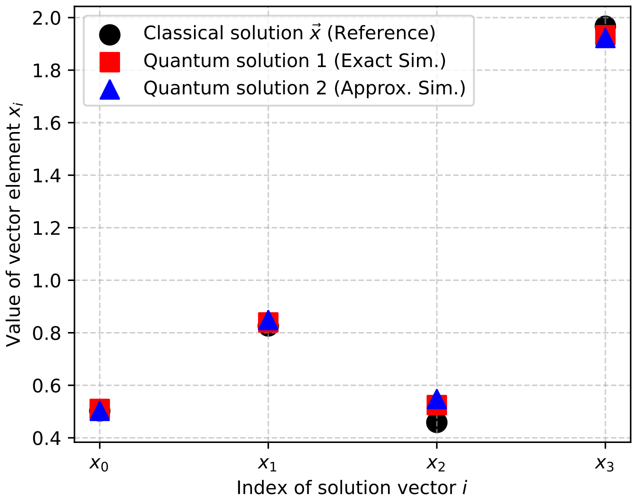


Fig. 2. Comparison of the Classical HHL Solution to the quantum solution with and without approximation of the Controlled Unitary gates of the QPE step.

The distinctive aspect of this implementation is the direct comparison of two quantum-mechanical approaches for realizing the QPE step of the HHL algorithm, in contrast to the classical solution of linear systems. In the first approach, the controlled unitary operators required for the QPE step from the c to the b register are simulated exactly. In particular, the time-evolution operator

$$U(t) = e^{-iHt} \tag{6}$$

is implemented without approximation. However, this exact simulation becomes increasingly demanding as the system size grows, since the Hamiltonian H generally acts on a Hilbert space whose dimension scales exponentially with the number of qubits. As a consequence, the construction and simulation of the corresponding unitary operators quickly become computationally intractable.

To mitigate this issue, the Suzuki–Trotter expansion [11] is employed as an approximation scheme and analyzed as an alternative realization of the time-evolution operator. The Hamiltonian is decomposed into a sum of non-commuting terms,

$$H = \sum_{j=1}^{m} H_j \tag{7}$$

which allows the unitary evolution to be approximated by a product formula. In the first-order (Lie–Trotter) decomposition, the time-evolution operator is approximated as

$$e^{-iHt} \approx \left(\prod_{j=1}^{m} e^{-iH_j\, t/r} \right)^{r} \tag{8}$$

where r denotes the number of Trotter steps.

The approximation error of the first-order Suzuki–Trotter expansion scales as $\mathcal{O}\left(\frac{t^2}{r}\right)$. Consequently, increasing the number of Trotter steps r systematically reduces the approximation error at the cost of an increased circuit depth.

Index	Classical	Quantum (exact)	Quantum (approx.)
0	0.50	0.51	0.50
1	0.83	0.84	0.85
2	0.46	0.52	0.55
3	1.97	1.93	1.92

TABLE I
COMPARISON OF THE CLASSICAL SOLUTION WITH CORRECTED QUANTUM SOLUTIONS ROUNDED TO THE SECOND NUMBER AFTER THE COMMA FOR CLARITY REASONS.

Fig. 2 compares the exact unitary implementation with the Suzuki–Trotter approximation in terms of numerical accuracy and computational complexity for increasing system dimensions. Tab. IV-A presents the rounded numbers of the resulting vector's components used in the fig. 2 for a more accurate analysis of the approximation's impact.

B. Cirq

The Cirq implementation follows a different approach and aims to improve the accuracy and success probability of the quantum algorithm by employing a technique known as amplitude amplification [12]. Rather than relying solely on a direct implementation of the HHL circuit, this approach enhances the probability of measuring the desired solution state by iteratively amplifying its amplitude in the quantum state space. Amplitude amplification can be understood as a generalization of Grover's search algorithm and operates by alternating between reflections about the target subspace and the initial state [12]. In the context of the HHL algorithm, this technique is used to increase the probability of successfully post-selecting the ancilla qubit in the desired state after the controlled rotation step, which encodes the inverse eigenvalues of the system matrix as can be seen from eq. 4. Within Cirq, this strategy is implemented by explicitly constructing the corresponding reflection operators as quantum circuits and applying them iteratively. This leads to a higher overall success probability compared to a single-shot HHL execution, thereby reducing the number of required circuit repetitions.

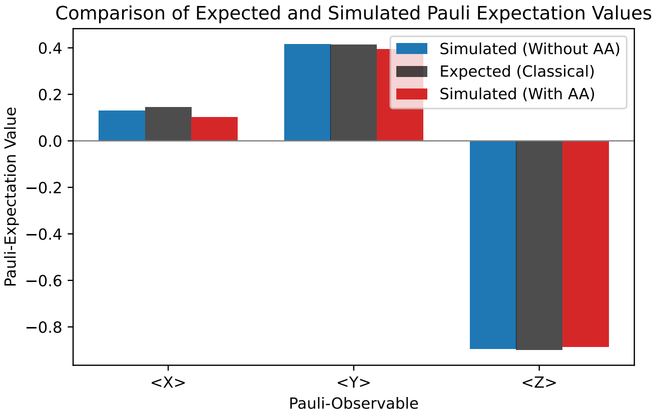


Fig. 3. Comparison of the expectation values of the Pauli observables X, Y, and Z obtained from the HHL algorithm. The expected (gray) values are shown alongside the numerically estimated values from simulations with (blue) and without (red) amplitude amplification.

Pauli Operator	Expected Value	No AA	With AA
X	0.14	0.13	0.10
Y	0.41	0.42	0.39
Z	-0.90	-0.90	-0.89

TABLE II
EXPECTATION VALUES OF PAULI OPERATORS WITH AND WITHOUT AMPLITUDE AMPLIFICATION (AA)

Fig. 3 shows that while the expectation values are not systematically altered by amplitude amplification, the increased success probability leads to improved statistical stability and closer agreement with the expected results. Tab. IV-B presents the rounded numbers of the resulting observable's values used in fig. 3 for a more accurate analysis of the presented method's impact.

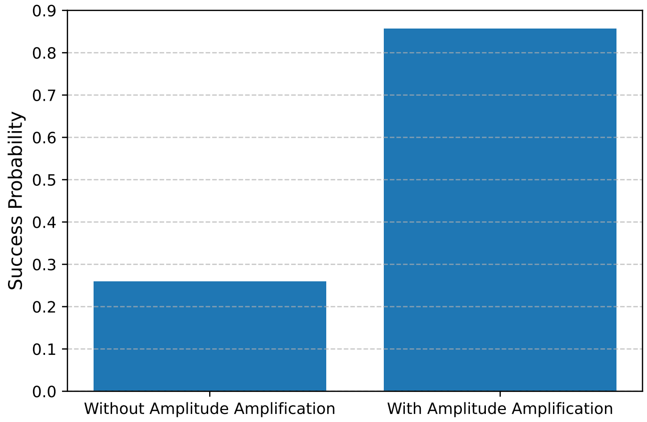


Fig. 4. Success probability per single execution of the HHL circuit with and without amplitude amplification.

Fig. 4 illustrates that amplitude amplification significantly increases the probability of successful post-selection, thereby reducing the number of required circuit repetitions.

V. Conclusion

Throughout this seminar paper, the powerful HHL algorithm has been analyzed. It is shown to represent an efficient quantum mechanical approach to solving LSP by applying quantum mechanical methods (i.e., gate operations) to exploit the quantum advantage. The HHL algorithm is, thus, known for promising an exponential speedup over the best-known classical method, establishing it as a key tool for implementing scalable quantum machine learning models, which frequently rely on linear algebraic operations.

However, it could be seen that the HHL algorithm is not a general solving algorithm for all LSP, as several limiting factors drastically reduce its range of possible use cases. The exponential speedup does not hold for reading out the full solution vector $|x\rangle$, which scales as $\mathcal{O}(N)$. Therefore, the result can only be used for limited statistical information or for further calculations. The $\mathcal{O}(\log(N))$ complexity is shown to be a simplification; in reality, the runtime is found to depend on several factors and requires the matrix A to fulfill the aforementioned conditions. Furthermore, fundamental requirements like state preparation and the unitary transformation may scale polynomially with the dimension N, which poses a general challenge for all quantum algorithms.

Despite these inherent limitations, the HHL algorithm and its underlying principles remain a powerful area of research for quantum computation. Potential applications are found in fields requiring high-speed linear algebra. For instance, the method is a promising candidate for application in *tensor network simulations*, e.g., for Many Body Simulation in computational physics, especially helpful for ground state search of Hamiltonians. In these complex physical systems, where the computational complexity of interacting quantum states is immense, the logarithmic scaling of HHL could drastically reduce calculation times for determining system properties. Furthermore, the principles of efficient matrix computation are highly relevant for *High-Performance Computing*. Given the role of HHL in rapidly solving linear systems, its generalization or adaptation could lead to quantum subroutines that accelerate the large-scale linear algebraic processes central to HPC tasks, thereby providing a significant advantage in diverse areas such as fluid dynamics, structural analysis, and financial modeling.

References

[1] J. Preskill, “Quantum computing 40 years later,” 2023. [Online]. Available: https://arxiv.org/abs/2106.10522

[2] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Journal on Computing*, vol. 26, no. 5, p. 1484–1509, Oct. 1997. [Online]. Available: http://dx.doi.org/10.1137/S0097539795293172

[3] L. K. Grover, “A fast quantum mechanical algorithm for database search,” 1996. [Online]. Available: https://arxiv.org/abs/quant-ph/9605043

[4] Wikipedia contributors, “History of programming languages — Wikipedia, the free encyclopedia,” https://en.wikipedia.org/w/index.php?title=History_of_programming_languages&oldid=1312815541, 2025, [Online; accessed 30-October-2025].

[5] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Physical Review Letters*, vol. 103, no. 15, Oct. 2009. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.103.150502

[6] S. Leon, *Linear Algebra with Applications*, ser. Featured Titles for Linear Algebra (Introductory) Series. Pearson/Prentice Hall, 2010. [Online]. Available: https://books.google.de/books?id=SSVdPgAACAAJ

[7] A. Zaman, H. J. Morrell, and H. Y. Wong, “A step-by-step hhl algorithm walkthrough to enhance understanding of critical quantum computing concepts,” *IEEE Access*, vol. 11, p. 77117–77131, 2023. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2023.3297658

[8] S. Aaronson, “Quantum machine learning algorithms : Read the fine print,” 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:12646410

[9] T. Goldfriend, I. Reichental, A. Naveh, L. Gazit, N. Yoran, R. Alon, S. Ur, S. Lahav, E. Cornfeld, A. Elazari, P. Emanuel, D. Harpaz, T. Michaeli, N. Erez, L. Preminger, R. Shapira, E. M. Garcell, O. Samimi, S. Kisch, G. Hallel, G. Kishony, V. v. Wingerden, N. A. Rosenbloom, O. Opher, M. Vax, A. Smoler, T. Danzig, E. Schirman, G. Sella, R. Cohen, R. Garfunkel, T. Cohn, H. Rosemarin, R. Hass, K. Jankiewicz, K. Gharra, O. Roth, B. Azar, S. Asban, N. Linkov, D. Segman, O. Sahar, N. Davidson, N. Minerbi, and Y. Naveh, “Design and synthesis of scalable quantum programs,” 2024.

[10] Cirq Developers, “Cirq,” Aug. 2025. [Online]. Available: https://github.com/quantumlib/Cirq

[11] N. Hatano and M. Suzuki, *Finding Exponential Product Formulas of Higher Orders*. Springer Berlin Heidelberg, Nov. 2005, p. 37–68. [Online]. Available: http://dx.doi.org/10.1007/11526216_2

[12] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, “Quantum amplitude amplification and estimation,” p. 53–74, 2002. [Online]. Available: http://dx.doi.org/10.1090/conm/305/05215

Solving Linear Differential Equations on a Quantum Computer

Federico Harjes Ruiloba
LMU Munich
Munich, Germany
f.harjes@campus.lmu.de

***Abstract*—Linear differential equations (LDEs) play a prominent role across a wide range of scientific disciplines. Consequently, understanding novel techniques for solving LDEs, such as methods relying on quantum algorithms, is a valuable endeavor. This work presents a pedagogical review of quantum LDE solvers based on two different paradigms: Quantum Linear-System-Solving (QLSS) and Time-Marching. The goals of this review are (a) to introduce quantum LDE solvers to a broader audience, (b) to explicitly analyze the sources of quantum speed-ups and limitations of these methods, and (c) to give practical decision guidance for choosing between QLSS and Time-Marching based on problem characteristics.**

***Index Terms*—Quantum Algorithms, Differential Equation Solvers, QLSS, Hamiltonian Simulation**

I. Introduction

Differential equations (DEs) are ubiquitous in disciplines ranging from plasma physics and aerodynamics to molecular biology and financial risk analysis [1]. DEs provide a concise manner for expressing the evolution of quantities through space or time, making them a valuable tool for describing evolution processes and scientific laws. However, only the simplest differential equations are analytically solvable. For this reason, methods to approximate the solutions to DEs efficiently are widely studied.

A recently explored category of such methods are DE solvers based on quantum algorithmic subroutines, promising potential speed-ups outside the reach of classical alternatives [2]–[8]. Although methods for solving non-linear differential equations have been proposed [9], [10], quantum methods are most intuitively employed to linear differential equations (LDEs). For this reason, the present review focuses on two promising quantum paradigms for solving LDEs: Quantum Linear-System-Solving (QLSS) and Time-Marching.

Understanding how these two different methods work requires bridging classical numerical methods, quantum encoding techniques, and primitive quantum algorithmic subroutines. The purpose of this paper is to concisely review and connect the primitives employed by quantum LDE solvers in order to form an intuition for both QLSS and Time-Marching approaches. Moreover, the speed-ups promised by QLSS and Time-Marching approaches will be explained, clarifying which mechanisms might enable such speed-ups and also which limitations might cause them to vanish. Afterwards, to support the practical application of the presented quantum solvers, a decision framework will be presented, showing under which conditions each of the two paradigms is better suited. Finally, open challenges and possible developments in the space will be discussed, culminating in possible avenues for future research.

II. Background

A. (Linear) Differential Equations

Before introducing the building blocks of the quantum algorithms discussed in this study, it is worthwhile to revisit DEs. In general, a DE is an equation that connects one or more unknown functions with their derivatives [11]. It is usual to distinguish between two classes: partial differential equations (PDEs) and ordinary differential equations (ODEs). PDEs contain unknown multi-variable functions related to their partial derivatives, while ODEs deal with single-variable functions related to their derivatives.

Both PDEs and ODEs can be subcategorized further as non-linear or linear differential equations. As the name suggests, LDEs can be written as a linear combination of derivatives. Linear PDEs can be compactly expressed as follows:

$$Lu = f(x_1, ..., x_d). \tag{1}$$

Here, u is an unknown function of interest, which could, for instance, model the evolution of a physical quantity through temporal or spatial dimensions, or, more generally, through d different dimensions denoted by $x_1, ..., x_d$. L is a linear differential operator mapping u to a linear combination of its partial derivatives with d variables. Finally, $f(x_1, \ldots, x_d)$ is a known function often called the source term or forcing function. To solve an LDE, a function u must be found, so that when L acts on this function, the forcing function f is obtained.

In the case where u is a single-variable function ($d = 1$), for a linear ODE containing an n-order derivative, we obtain a linear differential operator of the following form using $a_0 \ldots a_n$ functions of x:

$$L = a_0(x) + a_1(x)\frac{d}{dx} + a_2(x)\frac{d^2}{dx^2} + \ldots + a_n(x)\frac{d^n}{dx^n}. \tag{2}$$

Finally, to obtain a unique solution to an LDE, one must also specify appropriate boundary conditions, or initial conditions in the case of time-dependent problems. These conditions constrain the solution at specific points in the domain. For example, boundary conditions for a second-order ODE on a domain $[a, b]$ may take the form:

$$u(a) = \alpha, \quad u(b) = \beta, \tag{3}$$

where α and β are specified constants.

B. Solving Linear Equations Classically

While analytical solutions provide an exact description of the system modeled by the DE of interest, many DEs of interest, such as the Navier-Stokes and Schrödinger equations, do not have a closed-form solution in general. Approximate numerical methods have therefore become indispensable tools for solving DEs, resulting in a rich corpus of literature discussing these methods [12].

A fundamental element of many such numerical methods is the discretization of the continuous domain of the DE. The unknown function $u(x)$ can be discretized into a grid of N points with spacing h, producing a vector of the values of the function at the grid points, $\mathbf{u} \in \mathbb{R}^N$. Since the DE must be satisfied at each grid point, we obtain N equations with N unknowns. Concretely, the derivatives appearing in the DE can be approximated at each grid point using finitie differences. For example, the first derivative can be approximated for a small spacing h as $\frac{du}{dx} \approx \frac{u(x+h)-u(x)}{h}$, relating the derivative at each point to the function at neighboring points. Finite-difference methods can similarly be applied to approximate higher-order derivatives.

For an LDE, these finite-difference approximations are linear combinations of the unknown values of u at the grid points. The resulting system takes the form $\mathbf{Au} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{N \times N}$ encodes the discretized differential operator L and the boundary conditions, while $\mathbf{b}$ encodes the discretized forcing function f and the concrete values of the boundary conditions.

To find the solution vector $\mathbf{u}$ approximating the unknown function at the grid points, assuming $\mathbf{A}$ is an invertible matrix, the system can be inverted producing $\mathbf{u} = \mathbf{A^{-1}b}$ using Gaussian elimination or full-inversion methods [13]. As more accurate solutions require larger N, the computational cost of inverting $\mathbf{A}$ grows rapidly and becomes the computational bottleneck for many classical DE solvers. The first quantum method discussed later in Section IV-A of this review review, QLSS [6], [14], directly tackles this limitation.

The approach described above, discretizing the full domain and solving the resulting linear system, is sometimes referred to as direct approach for solving LDEs. A different popular classical approach, particularly suitable for time-dependent DEs, is Time-Marching. Again, the domain of the LDE of interest is discretized, in this case into a grid of time steps $0 = t_0 < t_1 < \ldots < t_L = T$. Starting from an initial condition at t_0, the solution at each subsequent time step is calculated based on previously obtained values.

Time-Marching variants include one-step methods, such as the Euler method [15] and Runge-Kutta methods [16], which compute the solution at t_{l+1} using only information from t_l, and multi-step methods, such as Adams methods, which use information from several previous time steps. The classical Time-Marching paradigm forms the basis of the second quantum method [8] discussed in Section IV-B of this review.

C. Quantum Primitives

1) HHL: Introduced by Harrow, Hassidim and Loyd, the HHL algorithm promises an exponential speed-up for solving linear systems of equations under certain conditions [14]. For a system $\mathbf{Au} = \mathbf{b}$, HHL is able to compute a quantum state $|u\rangle$ encoding the solution vector $\mathbf{u}$ with logarithmic complexity in the size of $\mathbf{u}$. However, this speed-up can vanish in practice, as quantum state tomography [17] must be employed to reconstruct the full classical solution $\mathbf{u}$ from $|u\rangle$, a step with exponential complexity with respect to the size of $\mathbf{u}$ in most cases [18].

In addition to this challenge, HHL requires the state $|b\rangle$ corresponding to the forcing function vector $\mathbf{b}$ to be prepared on a quantum computer, which is difficult in many cases. Furthermore, the effectiveness of HHL also depends on the sparsity and the condition number of $\mathbf{A}$[1], both of which are influenced by the order of the LDE of interest. The complexity of HHL with respect to sparsity and condition number of $\mathbf{A}$ has been greatly reduced through modifications to the original version of the algorithm [20], making it a powerful subroutine for tasks like the one discussed in the present work [6].

2) Hamiltonian Simulation: The first proposal of quantum computers is often attributed to Feynman, who recognized the intrinsic limitations of classical computers in simulating quantum dynamics [21]. Feynman coined the term of Hamiltonian simulation [21], which involves efficiently approximating the evolution of a quantum system. Concretely, the evolution of a quantum state $|\psi\rangle$ is described by the Schrödinger equation

$$\frac{d}{dt}|\psi(t)\rangle = -iH|\psi(t)\rangle, \tag{4}$$

where H is a Hermitian matrix (i.e. $H = H^\dagger$) governing the evolution, known as the Hamiltonian operator. To calculate the state of $|\psi\rangle$ at a time t, the unitary evolution

$$|\psi(t)\rangle = e^{-iHt}\,|\psi(0)\rangle\,, \tag{5}$$

can be implemented using a sequence of quantum gates acting on an n-qubit system initialized with the state $|\psi(0)\rangle$ [22]. A common approach to achieve this is the use of product formulas, also known as Trotter-Suzuki decompositions [23]. If the Hamiltonian can be decomposed as $H = H_1 + H_2 + \ldots H_k$ with $k \in N$, the evolution operator $U = e^{-iHT}$ can be approximated as:

$$U \approx \left(e^{-iH_1T/r}e^{-iH_2T/r}\ldots e^{-iH_kT/r}\right)^r. \tag{6}$$

[1] In numerical analysis, the condition number of a function measures how much the output value of the function can change for a small change in the input argument. The condition number associated with the linear equation $\mathbf{Au} = \mathbf{b}$ gives a bound on how inaccurate the solution $\mathbf{u}$ will be after approximation [19].

In this manner, U can be implemented as a sequence of evolution steps, where each evolution step $e^{-iH_kT/r}$ is a composition of quantum gates. The length of the sequence, given by r, controls the accuracy of the approximation.

The efficiency of Hamiltonian simulation and the parallels it has with classical Time-Marching make it a natural choice as a primitive for solving LDEs quantumly. When an LDE can be mapped to a Schrödinger-type equation, direct application of Hamiltonian simulation techniques becomes possible. However, general LDEs involve non-Hermitian A matrices, requiring additional techniques such as block encoding [24]. Furthermore, Hamiltonian simulation depends on long sequences of imperfect quantum gates, leading to vanishing success probabilities for large r. Solutions to these challenges, making this quantum primitive practically relevant for LDE solving, will be discussed in Section IV-B.

III. Related Work

The application of quantum algorithms to DEs and LDEs in particular has developed along two main directions building upon the two primitives introduced in the previous section. On the one hand, solvers exploiting the HHL algorithm and its variants, which fully invert the discretized LDE in a single step. On the other hand, solvers following the paradigm of Hamiltonian simulation, which propagate quantum states through time, analogously to classical Time-Marching.

A. QLSS Approaches

Following the introduction of the HHL algorithm [14], Berry et al. [3] first proposed a protocol to encode linear ODEs as sparse linear systems which could be efficiently tackled combining an HHL subroutine with high-order multistep methods [25]. Subsequent work [26]–[28] further improved the efficiency of QLSS approaches, achieving the current state of the art complexity for such algorithms. Applications of QLSS to specific LDEs have been extensive, including work on Poisson's equation [29], Schrödinger's equation [28], and more general second-order elliptic equations [4].

B. Time-Marching Approaches

Fang et al. [8] presented the first effective protocol for Time-Marching quantum LDE solvers. Making use of Singular Value Amplification [30], Fang et al. were able to overcome the problem of exponentially vanishing success probabilities [8], a fundamental obstacle for iterative quantum methods. A very recent addition close to this line of research is the Schrödingerisation technique [7], which makes it possible to tackle the equally challenging and practically relevant class of stochastic differential equations on a quantum device [31].

C. Alternative Approaches

In addition to QLSS and Time-Marching approaches, several other quantum methods have been explored for solving DEs. For instance, one of the first quantum methods for this task used Grover's algorithm [32], resulting in a quadratic speed-up over classical methods under certain conditions [2]. A similar method based on amplitude estimation was later proposed for the Navier-Stokes equations [33].

Alternatively, variational algorithms [34] have been proposed as a subroutine capable of solving DEs by mapping the solution to the ground state of a Hamiltonian describing the equation of interest [35]. For nonlinear DEs, methods based on the technique of Carleman linearization [5] have emerged as a promising technique.

D. Reviews and Comparative Analyzes

As is the case for the present review, different authors have previously explored quantum DE solvers from a pedagogical perspective. Broeckmann [6] provided a careful treatment of QLSS applied to Poisson's equation. Pesah [36] offered a comprehensive classification of quantum PDE solvers, with a special focus on the complexity of different steps of both QLSS approaches and Hamiltonian simulation approaches. Linden et al. [37] performed a rigorous comparison of quantum and classical methods for the heat equation, establishing precise conditions for quantum advantage.

This review provides a focused pedagogical comparison between QLSS and Time-Marching paradigms, explicitly analyzing quantum speedup sources and limitations while offering practical decision guidance based on problem characteristics.

IV. Quantum Paradigms for Solving LDEs

Having introduced relevant classical and quantum building blocks, this section will discuss two central quantum paradigms for solving LDEs in detail. Both share a representational strategy: the solution vector $\mathbf{u} = (u_1, \ldots, u_N)^T$, where N is the number of points in the discretized grid, is encoded in the amplitudes of a quantum state,

$$|u\rangle = \frac{1}{\|\mathbf{u}\|} \sum_{j=1}^{N} u_j |j\rangle, \tag{7}$$

compressing N values into $O(\log N)$ qubits. This compression underlies the potential for quantum speedup, but encoding the result in a quantum state implies the need for quantum tomography [17] in order to recover the full classical solution.

A. QLSS Paradigm

Conceptually, QLSS follows the direct paradigm discussed in Section II: *discretize the entire problem first, then solve the entire resulting linear system at once*. This section will concisely describe the main components of QLSS approaches for solving LDEs.

1) Encoding LDEs as Linear Systems: Consider a linear PDE such as Poisson's equation, $\Delta u = f$, defined on a continuous d-dimensional domain Ω. Here, Δ represents the Laplacian operator:

$$\Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \cdots + \frac{\partial^2 u}{\partial x_d^2}, \tag{8}$$

where $x_1, \ldots, x_d \in \Omega$. As discussed in Section II, classical numerical methods approximate Ω with a discrete grid of N

points. The unknown function u is represented by a vector $\mathbf{u} \in \mathbb{R}^N$ of its values at these grid points. Similarly, the forcing function f is represented by $\mathbf{b} \in \mathbb{R}^N$, while the differential operator Δ becomes a matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ acting on $\mathbf{u}$, resulting in the linear system $\mathbf{Au} = \mathbf{b}$.

To illustrate how $\mathbf{A}$ is constructed, consider the one-dimensional case $\frac{d^2u}{dx^2} = f(x)$ on $[0, 1]$ with boundary conditions $u(0) = \alpha$, $u(1) = \beta$. Discretizing the domain into N interior grid points with spacing $h = \frac{1}{N+1}$, the second derivative at each point x_i can be approximated as

$$\left.\frac{d^2u}{dx^2}\right|_{x_i} \approx \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}. \tag{9}$$

Since this expression relates each unknown u_i to its neighbors, applying it to all N grid points produces a tridiagonal matrix:

$$\mathbf{A} = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & \cdots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & 0 & \cdots & 1 & -2 \end{pmatrix}. \tag{10}$$

In order to store and manipulate the solution of the LDE on a quantum device, such a discretization is performed [3]. An important remark at this stage is that the discretization and the encoding on the quantum device are not always trivial steps, and their complexity can even negate the speed-up achieved by QLSS-based methods [36].

2) Applying HHL: Once the differential equation has been formulated as a linear system of equations, the HHL algorithm [14], or any of its variants [28], can be applied as a solver for the system. In order to do so, $\mathbf{b}$ and $\mathbf{A}$ must be encoded in a quantum circuit. First, the state $|b\rangle$ is encoded analogously to the final solution state $|u\rangle$ in Eq. (7). To encode $\mathbf{A}$, the Hamiltonian simulation method can be employed, analogously to the simulation of the Hamiltonian operator H in Eq. (4) through the evolution operator U in Eq. (5) and (6). To construct such an evolution operator, which can be implemented as a sequence of quantum gates performing e^{iAt}, $\mathbf{A}$ must be Hermitian (or embedded into a larger Hermitian matrix)

$$\tilde{\mathbf{A}} = \begin{pmatrix} 0 & \mathbf{A} \\ \mathbf{A}^\dagger & 0 \end{pmatrix}. \tag{11}$$

To efficiently solve the system of linear equations of interest, HHL exploits a fundamental property of Hermitian matrices: if $\mathbf{A}$ is Hermitian, it can be represented in its eigenbasis as a diagonal matrix, with its main diagonal consisting of its eigenvalues $\lambda_1, \ldots, \lambda_N$:

$$\mathbf{A} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{pmatrix} \tag{12}$$

Inverting the matrix therefore reduces to inverting its eigenvalues, which can be achieved by replacing each λ with $1/\lambda$. While classical algorithms must compute eigenvalues explicitly, HHL extracts the eigenvalues of $\mathbf{A}$ via Quantum Phase Estimation (QPE)[2] [39].

As visible on Fig. 1, the eigenvalues $\lambda_1, \ldots, \lambda_N$ are first encoded in an auxiliary "clock" register through a QPE block which contains the encoding of $\mathbf{A}$. To invert the eigenvalues, HHL then performs a rotation to an ancilla qubit, controlled by the clock register value. If the angle is chosen so that for each λ, $\sin(\theta(\lambda)/2) \approx 1/\lambda$, each amplitude of the state $|1\rangle$ in the ancilla qubit is proportional to each corresponding λ.

After this operation, the quantum state contains the inverted eigenvalue information. To disentangle the clock register from the input and the ancilla, the Hermitian QPE† is applied, reverting the first QPE operation. Finally, measurement and post-selection are applied on the ancilla qubit, keeping only the outcomes where $|1\rangle$ was observed, which correspond with the cases where the ancilla encoded $1/\lambda$. By doing so, the original input register collapses to the desired $|u\rangle$, which is proportional to the solution to the LDE.

As previously mentioned, the efficient application of HHL requires $\mathbf{A}$ to be sparse and well-conditioned, and the right-hand side $\mathbf{b}$ to be efficiently preparable as $|b\rangle$ on a quantum computer. For many LDEs, these requirements are naturally satisfied. The discrete Laplacian in Eq. (10), for instance, is sparse, as each grid point couples only to its neighbors. However, some LDEs, particularly those those with time dependent, non-smooth coefficient matrices $\mathbf{A}$, are challenging for QLSS approaches [8]. The following section introduces a paradigm which has proven more effective for such cases.

B. Time-Marching Paradigm

While QLSS methods solve for the entire solution at once, Time-Marching methods take a fundamentally different approach: *propagate the quantum state encoding the solution forward through time, step by step*. In this section, we highlight the central ideas of Time-Marching, drawing from [8].

1) Evolving a State through Time: Mirroring classical explicit time-stepping schemes, Time-Marching is particularly natural for initial value problems and time dependent LDEs. For instance, consider the initial value problem

$$\frac{d}{dt}|\psi(t)\rangle = \mathbf{A}(t)|\psi(t)\rangle, \quad |\psi(0)\rangle = |\psi_0\rangle, \tag{13}$$

which arises from the discretization of a time dependent LDE, given by a time dependent matrix $\mathbf{A}(t)$. This is a similar form to the one encountered in Eq. (4). Recall that the Hamiltonian governing the system was approximated by the evolution operator U by composing many short-time propagators of the form $e^{-iHT/r}$ in the Trotter-Suzuki formula (6). This is precisely a time-marching approach: the total evolution time T is divided into small steps $t = T/r$, and the solution is propagated forward through each step sequentially.

Concretely, adopting the notation from [8], the temporal domain of $\mathbf{A}$, denoted $[0, T]$, must be discretized into L time

[2] A more detailed analysis and explanation on QPE, as well as many other quantum algorithms including HHL can be found in [38].

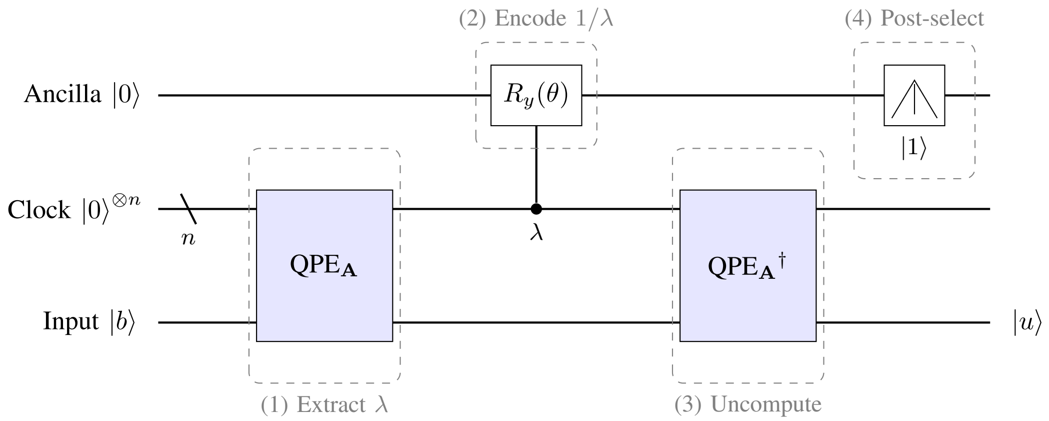


Fig. 1. Quantum circuit for the HHL algorithm. The input register is prepared in the state $|b\rangle$ encoding the source term of the LDE. (1) QPE encodes the eigenvalues λ of $\mathbf{A}$ into the clock register. (2) A rotation controlled by the clock register acts on the ancilla qubit, encoding $1/\lambda$ into the amplitude of its $|1\rangle$ component. (3) Inverse QPE "uncomputes" the clock register, disentangling it form the ancilla and input qubits. (4) Finally, measuring the ancilla and post-selecting on $|1\rangle$ produces the solution state $|u\rangle$.

steps $t_0 < t_1 < \cdots < t_L = T$. The solution can then be propagated through each time step via short-time propagators of the following form:

$$\Xi_l = \mathcal{T} e^{\int_{t_{l-1}}^{t_l} A(t)dt}, \tag{14}$$

where $\mathcal{T}$ denotes the time-ordering operator[3]. These propagators are the generalization of the evolution operator $U = e^{-iHt}$ from Hamiltonian simulation to non-Hermitian coefficient matrices. The final solution is obtained by composing all propagators:

$$|\psi(T)\rangle = \Xi_L \Xi_{L-1} \cdots \Xi_2 \Xi_1 |\psi_0\rangle. \tag{15}$$

For Hamiltonian simulation, each Ξ_l is unitary, resulting in a straightforward composition. However, many physical systems, such as Poisson's equation as discussed in the previous section, produce non-Hermitian coefficient matrices, resulting in non-unitary propagators.

2) Non-Unitary Propagation: Since quantum operations must be unitary, non-unitary propagators are implemented via block encoding, embedding Ξ_l into a larger unitary, as was the case for $\mathbf{A}$ in Eq. (11). In practice, block encoding introduces a probabilistic error into the application of the propagator, as the successful application of the block encoded propagator requires measuring an ancilla qubit in the correct state. Over L time steps, these probabilities with $p < 1$ compound multiplicatively, leading to an overall success probability that decays exponentially with the evolution time. This exponential decay was recognized early as a fundamental obstacle to quantum Time-Marching [3].

3) The Solution through Uniform Singular Value Amplification: A central contribution in [8] was to apply the technique of Uniform Singular Value Amplification (USVA) to restore favorable success probabilities. Intuitively, the root cause of the exponential decay is that standard block encoding introduces more normalization than necessary, i.e., the normalization factor is larger than the norm of the propagator $\|\Xi_l\|$. USVA uses the Quantum Singular Value Transformation primitive to construct a new block encoding whose normalization matches $\|\Xi_l\|$, thereby increasing the success probability at each step.

After applying USVA at each time step, the success probability no longer suffers from the artificial exponential decay introduced by block encoding. Instead, it reflects only how much the approximation intrinsically deviates from unitary evolution, quantified by the amplification ratio

$$Q = \frac{\prod_{l=1}^{L} \|\Xi_l\|}{\||\psi(T)\rangle\|}. \tag{16}$$

For unitary dynamics (e.g., Hamiltonian simulation), $Q = 1$ and the algorithm succeeds deterministically. For dissipative systems or solutions that decay over time, Q can be larger.

In great measure due to the adaption of the USVA subroutine, superficially discussed in this section, Time-Marching on quantum hardware has proven an extremely flexible and promising paradigm for solving a variety of DEs [8].

C. Selecting a Paradigm: Decision Guidance

Having presented both paradigms in detail, a natural question arises: given a specific DE of interest, which approach should one choose? The answer depends on several problem characteristics, which are systematized in this section. Concretely, five decision factors play an important role.

Time-dependence of coefficients. The most immediate distinguishing factor is whether the coefficient matrix $A(t)$ varies with time. QLSS methods are most naturally suited to time-independent problems, where the entire domain can

[3] The time-ordering operator $\mathcal{T}$ ensures that when the exponential is expanded as a series, operators at later times appear to the left. This is necessary when $\mathbf{A}(t)$ does not commute with itself at different times. For time-independent $\mathbf{A}$, or when $\mathbf{A}(t)$ commutes with itself at all times, the propagator simplifies to the matrix exponential $\Xi_l = e^{A(t_l - t_{l-1})}$.

be encoded in a single linear system. While time-dependent problems can be handled through a multi-step formulation, this increases the system size. Time-Marching, by contrast, handles time-dependent coefficients natively, updating the evolution operator at each step without reformulating the problem.

Smoothness requirements. QLSS methods based on spectral decomposition typically require the coefficient matrix to be diagonalizable and for higher accuracy, smooth. Time-Marching achieves high accuracy under weaker regularity conditions, requiring only bounded variation instead. For problems with discontinuous coefficients, such as those arising from abrupt material property changes, Time-Marching may be the only viable option.

Character of the evolution. The amplification ratio Q plays a central role in Time-Marching complexity. When the evolution is nearly unitary (e.g., Hamiltonian simulation or mildly dissipative systems), $Q \approx 1$ and Time-Marching is highly efficient. For strongly non-unitary evolution where the solution grows or decays significantly, Q can become large, favoring QLSS methods that depend on the condition number κ instead. However, it must ve noted that the relation between the amplification ratio and the condition number have not been studied comprehensively. Furthermore, the length of the evolution time can play a decisive role. Time-Marching scales less favorably in terms of the evolution time, and more propagation steps might pose a challenge due to the multiplicative success probabilities, even if the propagators are close to unitary.

Problem structure. QLSS methods can exploit special structure in the discretized differential operator. Kronecker sum decompositions, as in the Poisson equation, or sparsity patterns enable more efficient implementations as shown in [6].

Output requirements. Perhaps the most critical factor is what information is needed from the solution. Both paradigms produce a quantum state encoding the solution in its amplitudes. Extracting the full classical solution requires quantum state tomography with exponential complexity with respect to the dimension of the problem, which can eliminate any quantum advantage. However, if only partial information is needed, such as expectation values, samples, or the solution in a small region, the quantum speedup can survive. This consideration applies equally to both paradigms.

A summary of the criteria can be found in Tab. I. A dash indicates that the paradigm is not ideally suited for the given condition, a check mark indicates that the paradigm is well suited under the given criterion, and a check mark followed by a plus sign indicates that the paradigm is particularly effective the criterion.

TABLE I
CHOOSING QLSS VS. TIME-MARCHING FOR QUANTUM LDE SOLVING.

Criterion	QLSS	Time-Marching
Time-dependent problems	✓	✓+
Non-smooth $\mathbf{A}(t)$	–	✓
Long evolution time T	✓+	✓
Nearly unitary dynamics	✓	✓+
Highly non-unitary dynamics	✓	–
Full output required	–	–

V. Discussion and Conclusion

To conclude this review of quantum methods for solving LDEs, this section presents a summary of the key mechanisms, discussing sources and limitations of quantum advantage, and outlining directions for future research.

A. Summary of Mechanisms

The QLSS paradigm operates by discretizing the entire differential equation into a single linear system $\mathbf{Au} = \mathbf{b}$, subsequently applying the HHL algorithm to prepare a quantum state encoding the solution. The key primitive is QPE, which extracts eigenvalue information and enables matrix inversion through controlled rotations. However, in order for QPE to work, the system must first be encoded in the quantum hardware, requiring both amplitude and block encoding. Implicitly, Hamiltonian simulation is also required as part of the QPE primitive. Finally, a post-selection and measurement mechanism is required to produce the state encoding the solution.

The Time-Marching paradigm instead propagates the quantum state forward iteratively. The paradigm is closely related to Hamiltonian simulation, and can be implemented analogously if the coefficient matrix is Hermitian or anti-Hermitian. In cases where the coefficient matrix is not unitary, block encoding must be utilized, eventually introducing an excessive normalization of the propagators, resulting in vanishing success probabilities. However, this challenge has recently been overcome through USVA, which restores favorable success probabilities at each step.

B. Sources of Quantum Advantage

At a low level, both paradigms derive their potential speedups from the same fundamental source: the ability to encode N-dimensional solution vectors in $O(\log N)$ qubits. This logarithmic scaling in the representation is what enables potential polynomial or even exponential speedups over classical methods that must explicitly store N numbers.

For QLSS methods, the speedup manifests through the HHL algorithm's $O(\kappa^2 \log(N))$ complexity for well-conditioned, sparse systems. This ultimately comes from the ability to extract eigenvalues effectively through QPE. For Time-Marching, the speedup arises from efficient Hamiltonian simulation techniques that propagate quantum states polynomially in time T and the amplification ratio Q, rather than exponentially in the system dimension.

However, these speedups depend on several conditions: efficient state preparation, favorable problem structure (sparsity, condition number), and crucially, the ability to extract useful information from the quantum solution state without full reconstruction. Furthermore, it must be mentioned that subroutines employed by both algorithms, QPE and QSVT respectively, are extremely expensive in terms of required quantum resources, considerably above the current state-of-the-art hardware.

C. The Measurement Bottleneck

Perhaps the most significant intrinsic limitation shared by both paradigms, as well as many other quantum algorithms in general, is the measurement bottleneck. Both methods produce a quantum state $|u\rangle$ whose amplitudes encode the solution. Extracting the full classical solution vector requires quantum state tomography, with an exponential complexity in most cases [18], eliminating any quantum advantage gained during the solving phase.

This observation suggests that quantum LDE solvers are most promising in scenarios where the full solution is not required. Several such scenarios exist in practice. In general, if only specific expectation values of observables of intererest $\langle u|O|u\rangle$ are needed, these can be estimated efficiently through repeated measurements. Furthermore, for quantum-to-quantum pipelines, when the solution of a differential equation serves as input to another quantum computation, $|u\rangle$ can be used directly without classical readout. Such pipelines are particularly relevant in quantum chemistry and quantum simulation applications. Similarly, when only the solution in a small region is needed, amplitude amplification can be used to extract this information efficiently.

D. Future Directions

In spite of the limitations discussed, the methods presented in this review are of high scientific and economic relevance. The overview of the methods presented opens a variety of avenues for future work.

First, as discussed in Section I, DEs are present in a wide variety of scientific and technological domains. The decision framework presented in Section IV-C should be applied systematically to DEs across scientific domains. Reaction-diffusion systems in chemistry, the Boltzmann equation in kinetic theory and even the Black-Scholes equation in financial risk theory (although the latter might require more powerful methods due to its stochastic nature) present attractive test cases. For each, careful analysis is needed to determine which paradigm is better suited and whether end-to-end advantage is achievable.

Another insight from the present review is the sometimes ignored limitation that we referred to as ”the measurement bottleneck”. By encoding the solution in a quantum state, the methods discussed in this review are limited to specific questions. Mapping relevant questions to expectation values of observables, or exploring quantum-to-quantum workflows, remain promising and underexplored tasks.

Finally, it is worthwhile to insist on the possibility of using quantum DE solvers to get full solutions to problems of interest. In particular, a variety of efficient tomography schemes has been developed in the last decades, such as Matrix-Product-State [40] and Classical Shadow Tomography [41], greatly improving the efficiency of ”naive” full quantum state tomography. Combining modern tomography approaches with quantum DE solvers might enable the path towards a quantum advantage, even when seeking the full solutions to DEs of interest.

E. Conclusion

Solving DEs is an impactful application of quantum algorithms connecting numerical analysis, quantum information theory, and scientific computing. The QLSS paradigm is already a promising starting point for this goal, but recent fundamental developments in the field, such as [8], which enabled Time-Marching on quantum devices, hint at an even greater potential for quantum DE solving.

The path to practical quantum advantage requires not only algorithmic advances but also careful problem selection and reformulation. By identifying applications where partial solution information suffices, exploiting efficient tomography techniques for structured solutions, and designing quantum-native computational workflows, the promise of exponential speedups for DE solving may eventually be realized. Until then, the theoretical foundations laid by QLSS and Time-Marching methods provide relevant groundwork for future quantum scientific computing endeavors.

References

[1] M. Braun and M. Golubitsky, *Differential equations and their applications*. Springer, 1983, vol. 4.

[2] B. Kacewicz, “Randomized and quantum algorithms yield a speed-up for initial-value problems,” *Journal of Complexity*, vol. 20, no. 6, pp. 821–834, 2004.

[3] D. W. Berry, “High-order quantum algorithm for solving linear differential equations,” *Journal of Physics A: Mathematical and Theoretical*, vol. 47, no. 10, p. 105301, 2014.

[4] A. M. Childs, J.-P. Liu, and A. Ostrander, “High-precision quantum algorithms for partial differential equations,” *Quantum*, vol. 5, p. 574, Nov. 2021. [Online]. Available: https://doi.org/10.22331/q-2021-11-10-574

[5] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs, “Efficient quantum algorithm for dissipative nonlinear differential equations,” *Proceedings of the National Academy of Sciences*, vol. 118, no. 35, p. e2026805118, 2021.

[6] J. Broeckmann, “Solving linear differential equations quantumly: Quantum vs classical methods by example of Poisson’s equation,” August 2024, course material, RWTH Aachen University, Quantum Algorithms lecture.

[7] S. Jin, N. Liu, and Y. Yu, “Quantum simulation of partial differential equations via schrödingerization,” *Physical Review Letters*, vol. 133, no. 23, p. 230602, 2024.

[8] D. Fang, L. Lin, and Y. Tong, “Time-marching based quantum solvers for time-dependent linear differential equations,” *Quantum*, vol. 7, p. 955, 2023.

[9] H. Krovi, “Improved quantum algorithms for linear and nonlinear differential equations,” *Quantum*, vol. 7, p. 913, 2023.

[10] H.-Y. Wu, A. E. Paine, E. Philip, A. A. Gentile, and O. Kyriienko, “Quantum algorithm for solving nonlinear differential equations based on physics-informed effective hamiltonians,” 2025. [Online]. Available: https://arxiv.org/abs/2504.13174

[11] J. David Logan, *A first course in differential equations*. Springer, 2011.

[12] B. Denis, “An overview of numerical and analytical methods for solving ordinary differential equations,” *arXiv preprint arXiv:2012.07558*, 2020.

[13] A. Gupta, “Recent advances in direct methods for solving unsymmetric sparse systems of linear equations,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 28, no. 3, pp. 301–324, 2002.

[14] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Physical review letters*, vol. 103, no. 15, p. 150502, 2009.

[15] B. N. Biswas, S. Chatterjee, S. Mukherjee, and S. Pal, “A discussion on euler method: A review,” *Electronic Journal of Mathematical Analysis and Applications*, vol. 1, no. 2, pp. 2090–2792, 2013.

[16] J. Butcher, “Runge–kutta methods for ordinary differential equations,” *Numerical Analysis and Optimization: NAO-III, Muscat, Oman, January 2014*, pp. 37–58, 2015.

[17] A. Czerwinski, “Selected concepts of quantum state tomography,” *Optics*, vol. 3, no. 3, pp. 268–286, 2022.

[18] A. Anshu and S. Arunachalam, “A survey on the complexity of learning quantum states,” *Nature Reviews Physics*, vol. 6, no. 1, pp. 59–69, 2024.

[19] D. J. Higham, “Condition numbers and their condition numbers,” *Linear Algebra and its Applications*, vol. 214, pp. 193–213, 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0024379593000669

[20] X. Liu, H. Xie, Z. Liu, and C. Zhao, “Survey on the improvement and application of hhl algorithm,” in *Journal of Physics: Conference Series*, vol. 2333, no. 1. IOP Publishing, 2022, p. 012023.

[21] R. P. Feynman, “Simulating physics with computers,” *International journal of theoretical physics*, vol. 21, no. 6/7, pp. 467–488, 1982.

[22] K. L. Brown, W. J. Munro, and V. M. Kendon, “Using quantum computers for quantum simulation,” *Entropy*, vol. 12, no. 11, pp. 2268–2307, 2010.

[23] M. Suzuki, “General theory of fractal path integrals with applications to many-body theories and statistical physics,” *Journal of mathematical physics*, vol. 32, no. 2, pp. 400–407, 1991.

[24] C. Sünderhauf, E. Campbell, and J. Camps, “Block-encoding structured matrices for data input in quantum computing,” *Quantum*, vol. 8, p. 1226, 2024.

[25] L. Brugnano and D. Trigiante, “High-order multistep methods for boundary value problems,” *Applied Numerical Mathematics*, vol. 18, no. 1, pp. 79–94, 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/016892749500045V

[26] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, “Exponential improvement in precision for simulating sparse hamiltonians,” in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, ser. STOC ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 283–292. [Online]. Available: https://doi.org/10.1145/2591796.2591854

[27] A. M. Childs, R. Kothari, and R. D. Somma, “Quantum algorithm for systems of linear equations with exponentially improved dependence on precision,” *SIAM Journal on Computing*, vol. 46, no. 6, pp. 1920–1950, 2017.

[28] P. C. Costa, S. Jordan, and A. Ostrander, “Quantum algorithm for simulating the wave equation,” *Physical Review A*, vol. 99, no. 1, p. 012323, 2019.

[29] A. Montanaro and S. Pallister, “Quantum algorithms and the finite element method,” *Physical Review A*, vol. 93, no. 3, p. 032324, 2016.

[30] G. H. Low and I. L. Chuang, “Hamiltonian simulation by uniform spectral amplification,” *arXiv preprint arXiv:1707.05391*, 2017.

[31] S. Jin, N. Liu, and W. Wei, “Quantum algorithms for stochastic differential equations: A schrödingerisation approach,” *Journal of Scientific Computing*, vol. 104, no. 2, pp. 1–32, 2025.

[32] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.

[33] F. Gaitan, “Finding flows of a navier–stokes fluid through quantum computing,” *npj Quantum Information*, vol. 6, no. 1, p. 61, 2020.

[34] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’brien, “A variational eigenvalue solver on a photonic quantum processor,” *Nature communications*, vol. 5, no. 1, p. 4213, 2014.

[35] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, “Variational quantum algorithms for nonlinear problems,” *Physical Review A*, vol. 101, no. 1, p. 010301, 2020.

[36] A. Pesah, “Quantum algorithms for solving partial differential equations,” *University College London: London, UK*, 2020.

[37] N. Linden, A. Montanaro, and C. Shao, “Quantum vs. classical algorithms for solving the heat equation,” *Communications in Mathematical Physics*, vol. 395, no. 2, pp. 601–641, 2022.

[38] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press, 2010.

[39] A. Y. Kitaev, “Quantum measurements and the abelian stabilizer problem,” *arXiv preprint quant-ph/9511026*, 1995.

[40] M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu, “Efficient quantum state tomography,” *Nature Communications*, vol. 1, no. 1, Dec. 2010. [Online]. Available: http://dx.doi.org/10.1038/ncomms1147

[41] H.-Y. Huang, R. Kueng, and J. Preskill, “Predicting many properties of a quantum system from very few measurements,” *Nature Physics*, vol. 16, no. 10, p. 1050–1057, Jun. 2020. [Online]. Available: http://dx.doi.org/10.1038/s41567-020-0932-7

Quantum Singular Value Transformation: A unifying framework for Quantum Computing?

Patrick Paetsch
LMU Munich
Munich, Germany
P.Paetsch@campus.lmu.de

***Abstract*—For many years there have been well known and often used quantum algorithms in quantum computing, like Shor's Algorithm or Grover's Algorithm. In 2019 a new framework called Quantum Singular Value Transformation (QSVT) was introduced by Gilyén et al. The idea was a framework to simplify the construction of quantum algorithms. In this paper we investigate this framework and how it influenced the development of new algorithms. Based on ideas by Martyn et al. we answer the question why QSVT can be seen as a grand unification of quantum algorithms and therefore is a significant advancement in quantum computation.**

***Index Terms*—Quantum Computing, QSVT, quantum algorithms**

I. Introduction

Many problems have been solved using classical computing for decades, long before quantum computing emerged. Quantum computing introduces new methods for solving certain problems that can not be solved with or are inefficient for classical computers. Additionally, for some problems which can be solved using classical or quantum algorithms, the quantum algorithms sometimes solve them more efficiently in terms of runtime. One prominent example is the search problem, which can be solved classically in $\mathcal{O}(N)$ time. In 1996 Grover's algorithm [1] showed an improvement to $\mathcal{O}(\sqrt{N})$ time when using a quantum algorithm, which offers a quadratic speedup. This algorithm was specifically constructed to solve the search problem. One of the goals in developing new algorithms in quantum computing is to have significant speedups when compared to their classical counterparts. As [2] mentions, we can see the same result for other algorithms, like ones for quantum phase estimation [3] or Hamiltonian simulation [4]. Based on these ideas, [5] developed the Quantum Singular Values Transformation (QSVT) to describe some of these algorithms. They realized that these algorithms have similar structures to a certain degree and searched for a framework that could describe the algorithms. They built their framework on the technique of quantum signal processing (QSP) [6] as well as qubitization. [4]

In this paper, we closely follow the approach of [2] to explain why QSVT can be seen as a grand unification of quantum algorithms. For this, in section II we first want to take a short look at the general idea behind QSVT, as well as important terms and definitions used throughout the paper. In section IV we look at four different cases mentioned before: The Search Problem in IV-A, the Eigenvalue Threshold Problem in IV-B, the Phase Estimation in IV-C and the Hamiltonian Simulation in IV-D. For each of these cases we will give simple examples on how QSVT can construct an intuitive quantum algorithm and why these are an improvement over already existing algorithms. Following this, we can then answer the question, why QSVT can be seen as a grand unification and what influence it had on quantum computing in section V. In section VI, we will then conclude the paper by giving a short summary as well as an outlook on how we can further build on the QSVT and its techniques.

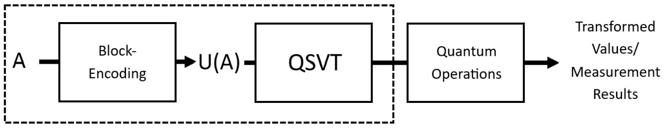


Fig. 1. Workflow of algorithm constructed with QSVT

II. Background and Important Terms

Before we look at the algorithms and answer the question, if QSVT can be seen as a grand unification, we first give a short overview over the functionality of QSVT as well as some important terms and definitions used throughout the paper.

A. QSVT

As [5] describes, QSVT is developed as a framework for quantum algorithms. It uses and combines the techniques of qubitization and QSP. It uses a matrix as input and transforms the eigenvalues or the singular values - depending on the functions we want to apply - to reach the desired result. QSVT is mainly used in two different ways. Since quantum algorithms require unitary matrices to apply quantum operations, we first take a given matrix A and encode it into a unitary matrix U. QSVT can then apply functions to the matrix without having to calculate the matrix first. In some algorithms - like the search problem mentioned in section IV-A - QSVT can also be used to directly encode the matrix, especially if handling both steps together is more efficient. This also helps for the second use: to construct a new quantum algorithm based on the framework to solve a specific problem. If we prepare a matrix using QSVT, we can then construct the algorithm by applying different functions that solve the problem. An overview of the workflow of an algorithm constructed with QSVT is given

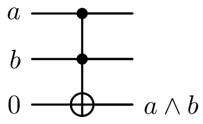


Fig. 2. quantum AND-operator

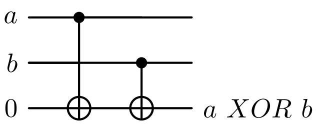


Fig. 3. Quantum XOR-operator with ancilla qubit

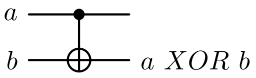


Fig. 4. Quantum XOR-operator without ancilla qubit

in figure 1. The dotted box marks the part that is needed in every quantum algorithm that is built with this framework, while the later part shows where the problem specific part is implemented.

B. Ancilla qubits

Ancilla qubits are an important part which is used in QSVT. These qubits are auxiliary qubits that are used to perform certain operations. For example if we have an operation that needs 3 qubits to be executed on, but we only have 2 qubits we can use, we can add an ancilla qubit. This qubit allows us to execute the operation without changing the context. Examples for this are the quantum AND-operator and the quantum XOR-operator. As we can see in figure 2 for the qubits a and b we need an ancilla qubit - predefined as 0 - to calculate $a \wedge b$. The same works for $a\ XOR\ b$ as can be seen in figure 3. In this case we could alternatively calculate it without the use of an ancilla qubit like in figure 4, but here b will now store the result of $a\ XOR\ b$ and we will therefore lose the state of the original qubit, while in the first case we can still use both a and b for further operations and work with $a\ XOR\ b$ as well. That way ancilla qubits can also be used as an aid to do measurements or apply operations without losing the state of the original qubits.

C. Unitary matrix

A unitary matrix U is a matrix that fulfills the following requirements. $U^\dagger U = UU^\dagger = I$ with $U^\dagger$ being the conjugate transpose of U and I being the identity matrix.

D. Block-encoding

We also work with block-encoding. The basic idea behind block-encoding of a given matrix A is to build a unitary, that has A in the upper-left corner.

$$U = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix} \tag{1}$$

The $\cdot$ are used to show that these values of the matrix are not important to know. When working with matrices, we also often work with eigenvectors and eigenvalues.

E. Gates

In quantum computing the gates are unitaries and they are applied to a quantum state. If we have a given matrix A we can not apply the unitary without block-encoding the A into an unitary. This can be seen as putting a box around the matrix to make it usable for applying the quantum operations. [7]

F. Matrix values

An eigenvector is a non-zero n-vector, which does not change its direction, even if a $n \times n$ matrix is applied to it. It fulfills the formula $Av = \lambda v$ with A being a $n \times n$ matrix, v being the eigenvector and λ being the eigenvalues. Eigenvalues are the factor by which the eigenvector stretches or shrinks, if the matrix is applied to it [8, p.68].

Singular values are non-negative real numbers that tell us how the matrix might stretch or shrink in any direction, not just one specific direction like with the eigenvector and eigenvalues. It is computed by taking the square roots of the eigenvalues of $A^\dagger A$. Since we will work with unitary matrices, which are a special case of normal matrices, we can also compute the singular values through the absolute values of the given matrix U after block-encoding A into it. We can write $U = X\Sigma Y^*$ with X and Y being unitary matrices and Σ being a rectangular diagonal matrix with the singular values on the diagonal [8, p.79].

G. Phase

In quantum computing, each state can be written with a possibility of a basis state, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ These possibilities α and β often given as complex numbers. Therefore we can rewrite $\alpha = |\alpha|e^{i\theta_\alpha}$ and $|\beta|e^{i\theta_\beta}$ in polar form. Then θ_α and θ_β are the respective global phases. For complex numbers, the global phase defines the angle of the number in the complex plane. The relative phase between α and β is defined as θ with $\alpha = e^{i\theta}\beta$ [8, p.93] This phase describes the interference behavior of the state.

H. Hamiltonians

Hamiltonians are also an important part when working with QSVT. As we will see later, some of the mentioned algorithms use Hamiltonians as input parameter. A Hamiltonian is a special operator, that describes the energy of a system. For quantum systems that means for any given time t we can understand how the quantum system behaves and also how it evolved into this state [8, p.83]. As we will see in section IV-D, we can use QSVT to simulate for a given Hamiltonian, how the system might look like at a time t, without having to use and measure a real quantum system.

III. Related work

As mentioned above, the development of QSVT was published in [5]. While analyzing quantum algorithms they realized, that there is an underlying concept these algorithms follow. Through extracting these concepts they built the framework QSVT. Building on this paper, [2] explained why QSVT might be seen as a grand unification. They followed the

```
Input: Oracle $U$, error tolerance $\delta$, precision $\Delta \leq 2/\sqrt{N}$
Output: flagged state $|m\rangle$
Runtime: $\mathcal{O}\Big(\sqrt{N}\log(1/\delta)\Big)$
Procedure:
1. Initialize quantum state
2. Apply polynomial approximation
3. Apply QSVT operator to amplify amplitude
4. Measure final state
```

Algorithm I
ALGORITHM FOR UNSTRUCTURED SEARCH

```
Input: Hamiltonian $H$, an $\alpha \geq \|H\|$
Output: H has at least one eigenvalue $\lambda \leq \lambda_{th} - \Delta_\lambda$ or all
        eigenvalues obey $\lambda \geq \lambda_{th} + \Delta_\lambda$
Runtime: $\mathcal{O}\Big(\frac{\alpha}{\Delta_\lambda \zeta^2}\log(1/\zeta)\log(1/\delta)\Big)$
Procedure:
1. Prepare unitary block encoding of $H/\alpha$
for $i = 1$ to $\xi$ do
    2. Use QSVT to apply polynomial to state, controlled by ancilla qubit
    3. Apply Hadamard gate to ancilla qubit and measure it in computational
       basis
end for
4. Measure probability of getting $|0\rangle$:
if closer to $\beta_1$ then
    $\exists$ an eigenvalue $\lambda \leq \lambda_{th} - \Delta_\lambda$ with high probability
else if closer to $\beta_2$ then
    all eigenvalues obey $\lambda \geq \lambda_{th} + \Delta_\lambda$ with high probability
end if
```

Algorithm II
ALGORITHM FOR EIGENVALUE THRESHOLD PROBLEM

development of QSVT given by [5] and constructed different algorithms using this framework. This way they gave a proof for their suggestion, that QSVT can be seen as grand unification of quantum algorithms. This is the main paper we follow. [7] published a short online article to give a mathematical overview on how QSVT works, supported by code parts, to show the functionality. In 2023 [9] gave another simplified summary of how QSVT works. Both of these references focus on the functionality of the QSVT, with [7] approaching it from a mathematical point of view and focusing on the basic idea, while [9] simplifies some of the pieces of [5] to give a easy to understand guide of QSVT. As we can see, there are many different approaches to explain the idea behind QSVT, but most of them focus on how the algorithm works. With this paper, we also give a short and easy to understand overview of how QSVT works, but mainly focus on why this framework is important and particularly why it can be seen as a grand unification by closely following the approach by [2]. For further reading on this topic and more detailed explanations, the references [2], [5], [7], [9] may be helpful.

IV. ALGORITHMS BUILT WITH QSVT

In this section we take a look at different problems in quantum computing and how QSVT can be used to built algorithms for them. By examining different applications of QSVT we address the question why QSVT can be seen as a grand unification. For each algorithm we give a short introduction of the problem it solves, followed by a summarization of the algorithm. In the next step we explain each of the steps of the algorithm in more detail. For all the following problems and constructed algorithms we follow the ideas of [2].

A. Search Problem

The first algorithm we look at is the unstructured search. Here we are searching for a specific element within an unstructured database. One approach in quantum computing is Grover's algorithm, which already has a significant improvement in runtime ($\mathcal{O}(\sqrt{N})$) compared to the classical counterpart ($\mathcal{O}(N)$), which is a quadratic speed up. But Grover's algorithm has some shortcomings. Although the algorithm first starts to increase the probability to find the correct solution, after an optimal number of iteration it starts to diverge from the solution again. [2] wanted to solve this problem with QSVT and hoped for a comparable runtime while at the same time dealing with the shortcomings. The basic idea behind how QSVT constructs an algorithm for the unstructured search is shown in algorithm I.

As input we use an oracle U that bit-flips an auxiliary qubit, if the target state $|m\rangle$ is given. We also have an error tolerance of δ, leading to a success probability of $1 - \delta$, as well as a precision $\Delta \leq 2/\sqrt{N}$. As output we want the flagged state $|m\rangle$, which is the state we are searching for. First we bring the quantum register into a superposition by applying Hadamard gates to it. Through polynomial approximation - [2] suggests we use the sign function - we map the amplitude of our marked state to a value close to 1. This would maximize the probability to find the correct state. In the third step we can apply the QSVT operator to amplify the amplitude - this is similar to Grover's algorithm. Through the amplification we can also increase the probability of successfully finding the correct state. As final step we measure the quantum state. This measurement gives the wanted result with a probability of $1 - \delta$. If the measurement fails, the QSVT application is repeated. As we saw, this algorithm uses a similar idea to Grover's algorithm. The runtime of $\mathcal{O}(\sqrt{N}\log(1/\delta))$ is close to the runtime of $\mathcal{O}(\sqrt{N})$ by Grover's algorithm. But since polynomials are designed to provide a monotonic increase, there is no optimal point after which the probability might decrease again. Therefore we can overcome the risk of divergence and ensure a higher probability of success.

B. Eigenvalue Threshold Problem

A second problem that can be solved by QSVT according to [2] is the eigenvalue threshold problem. For a predetermined threshold we want to know whether there exist eigenvalues that are below the threshold or if all eigenvalues are above the threshold. The goal is to create a measurable state, that gives us this information about the underlying quantum system. The algorithm to check for these eigenvalues can be seen in algorithm II.

We start with a Hamiltonian H. For $\|H\| \leq 1$ we can directly block-encode H into a unitary. To be able to deal with any kind of Hamiltonian, we instead use an $\alpha \geq \|H\|$ and can block-encode H/α instead. Using the Chernoff bound

```
Input: Oracle $U$, eigenvector $|u\rangle$, eigenvalue $e^{2\pi i\varphi}$, $n$ number of
       bits of the phase we want to approximate $n+1$ ancilla bits,
       error tolerance $\epsilon$, precision $\Delta$
Output: $\theta$ such that $|\theta - \varphi| \leq \frac{1}{2^n}$
Runtime: $\mathcal{O}(n\log(n/\delta)$
Procedure:
1. Initialize parameter $\theta = 0$
for j do
    2. Construct matrix $A_j(\theta) = \frac{1}{2}(I + e^{-2\pi i\theta U^{2j}})$
    3. Apply QSVT to extract bit of $\varphi$
    4. Apply Hadamard gate to ancilla qubit and measure it to get result of
    current bit of $\varphi$
    5. Update $\theta$
end for
6. Repeat for all bits $n$
```

Algorithm III
ALGORITHM FOR PHASE ESTIMATION

[10] as well two Bernoulli distributions β_1 and β_2, we define an upper bound ξ - we refer to [2] for the construction - for the number of repetitions needed. λ_{th} defines the threshold we want to set for our problem, with λ_{th}/α being used in the algorithm, since we normalized H with α as well. Δ_λ is the precision of the eigenvalue λ. For each repetition we first use QSVT to apply a polynomial to the quantum state $|\psi\rangle$, which is controlled by an ancilla qubit in state $|+\rangle$. We use this state, so when applying the polynomial in the next step, we can process the possibilities for both the state $|0\rangle$ as well as the state $|1\rangle$ simultaneously. The polynomial we choose approximates a function, that maps the eigenvalues to 1 and -1, depending whether they are smaller or larger than λ_{th}/α. After applying a Hadamard gate to the ancilla qubit, we can then measure it in the computational basis to collapse the superposition. The algorithm is constructed in such way, that the application of the polynomial sets the probabilities of the states $|0\rangle$ and $|1\rangle$ according to whether the eigenvalues are above or below the threshold. If we measure $|0\rangle$ the probability of existence of an eigenvalue that is below our threshold - it fulfills $\lambda \leq \lambda_{th} - \Delta_\lambda$ - is close to 100%. On the other hand, if the measurement result in $|1\rangle$, there is a high probability - close to 100%, that all eigenvalues are above our threshold. The runtime of $\mathcal{O}\Big(\frac{\alpha}{\Delta_\lambda \zeta^2}\log(1/\zeta)\log(1/\delta)\Big)$ is dependent on the error tolerance δ, the α we used to normalize the Hamiltonian, the precision Δ_λ as well as an overlap ζ between the state and the low-energy subspace of the Hamiltonian [8].

C. Phase Estimation

The third algorithm we look at is for the phase estimation. In phase estimation we are given an unitary operator U as well as an eigenvector $|u\rangle$ with $U|u\rangle = e^{2\pi i\varphi}|u\rangle$ and want to calculate the value of the phase φ. It uses a feedback procedure and is inspired by Kitaev's algorithm [11]. Algorithm III summarizes the structure.

The algorithm starts with an oracle U, its eigenvector $|u\rangle$ with eigenvalue $e^{2\pi i\varphi}$, $n+1$ ancilla bits for measuring the results in computational basis, an error tolerance of ϵ as well as a precision of Δ. First we initialize a parameter $\theta = 0$ which is used to approximate the phase step by step. We also prepare the ancilla qubits by applying a Hadamard gate to each of the qubits. Starting from the last bit of the phase we want to approximate ($j = n-1$) we repeat the following steps until we reach the first bit of the phase ($j = 0$): We construct a block-encoding of $A_j(\theta) := \frac{1}{2}(I + e^{-2\pi i\theta U^{2j}})$. On this matrix we can then use QSVT to apply an operator to extract the current bit j of our phase. By applying a Hadamard gate to the ancilla qubit, we can then measure the result in the computational basis. We then update our parameter θ as follows: Let a be the result of the measurement and $\theta = 0.\theta_1\theta_2...$, then the updated θ is $\theta = 0.a\theta_1\theta_2$. This means we always add the current result to the front of θ. By adding it to the front and since we started with the last bit of the phase, we built our approximation from back to front, given us the correct result after n steps. Since we have this θ as a parameter in the matrix construction in step 2, we have a cascading structure, using the current result in the next iteration. After we reached $j = 0$ - so after n iterations - θ then stores an approximation of φ with an error of $|\theta - \varphi| \leq \frac{1}{2^n}$. Therefore the more iterations we apply, the better the approximation..

Phase estimation is an important part of other algorithms as well, for example as part of Shor's algorithm [12]. Besides that it can also be used in solving factoring problems or as a generalization of the quantum Fourier transform [13]. As mentioned before, we have a cascading structure in the given algorithm for phase estimation. A similar structure can be seen in the binary search. If we look at the quantum Fourier transform, the inverse version can also be seen as a binary search, because similar to binary search it halves the search space by mapping the superpositions back to computational basis [8, p.222]. With this example we can see a connection between QSVT and quantum Fourier transform. If we dive deeper into this connection it leads us to the conclusion, that QSVT can use its advantage of polynomial approximations and therefore handle the operations more efficient as well as handle more complex operations, that quantum Fourier transform can not execute. Thus QSVT is a generalization of quantum Fourier transform.

D. Hamiltonian Simulation

Finally we are interested in an algorithm for so called function-evaluation problems. In these kind of problems we want to evaluate a function of a matrix. When working with QSVT we achieve this by applying a polynomial. As [2] mentions, these problems could be solved with quantum eigenvalue transformation, but we want to use QSVT to show that the framework is also applicable here. The Hamiltonian simulation is a prominent example for these kind of problems. Given a Hamiltonian, the algorithm calculates the time evolution and its operator e^{-iHt} for a Hamiltonian H as well as the time t. This means in Hamiltonian simulation we simulate how the quantum system evolves over time, given an energy operator (Hamiltonian) [8, p.83]. The algorithm IV shows the process of calculating or rather approximating the time evolution operator.

Input: Hamiltonian H, desired time t, an error tolerance ϵ, and $\alpha \geq \|H\|$
Output: block-encoded ϵ-approximation of e^{-iHt}
Runtime: $\mathcal{O}\Big(\alpha|t| + \frac{\log(1+\epsilon)}{\log(e+\frac{\log(1/\epsilon)}{\alpha|t|})}\Big)$
Procedure:
1. Prepare unitary block encoding of H/α
2. Apply QSVT twice, using polynomials
3. Use results to run circuit in Figure 5

Algorithm IV
ALGORITHM FOR HAMILTONIAN SIMULATION

We start with a given Hamiltonian as well as the time t at which we want to know how the system has changed from the beginning up to t. This will be measured through the time evolution operator e^{-iHt} of this Hamiltonian. To be able to apply QSVT later on, we need to block-encode the Hamiltonian. We have already seen in section IV-B how to block-encoded H depending whether $\|H\| \leq 1$ is fulfilled or not. As mentioned by [2] this leads to a small drawback, because we need to have some prior knowledge about H to be able to calculate the norm $\|H\| \leq 1$. But since there is a large class of Hamiltonians [4] with well known structures, for which block-encoding already exists, we can estimate the form of H with minor to no knowledge and apply a corresponding block-encoding, therefore making the drawback a minor problem. After we have a block-encoded unitary, we can then use QSVT to apply polynomial approximations to it. The operator we want to calculate is the exponential function, which does not have a definite parity. Therefore we can not just apply the Poly(a) from the QSVT, because for indefinite parities the theorem used to construct the polynomials for QSVT is not fulfilled. If we use polynomial approximations of $\cos(xt)$ as well as $\sin(xt)$ we can work around this problem, because these two approximations have definite parities and are also connected to the exponential function (because $e^{-ixt} = \cos(xt) - i\sin(xt)$). Therefore if we use these approximations, we can later rebuild our operator. This is done by inserting the results given by the approximations into the quantum circuit given in figure 5. Compared to other algorithms for Hamiltonian simulation, such as Hamiltonian simulation by qubitization as developed by [4], we can see that runtime improves significantly with this approach.

Another interesting example for function-evaluation problems is the matrix inversion. Given a Matrix A we want to approximate its inverse A^{-1}. For this example we refer to [2].

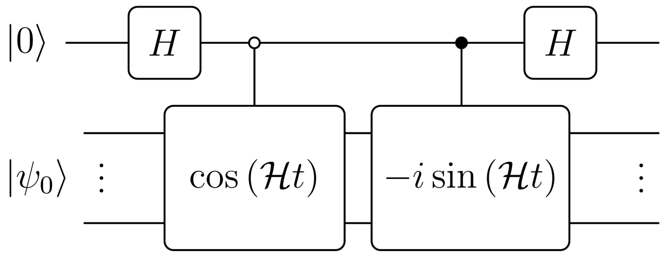


Fig. 5. Circuit created by Martyn et al. for use in Hamiltonian simulation

V. GRAND UNIFICATION

After exploring the construction of some algorithms for different problems with QSVT, we want to answer the question, if QSVT can be seen as grand unification. During the construction of the phase estimation algorithm we already heard of two different algorithms, which have a connection to this newly built algorithm: Shor's algorithm as well as the Quantum Fourier Transform. Since phase estimation is part of Shor's algorithm, we can therefore use the advantages of QSVT in this already existing and widely used algorithm as well. The quantum Fourier transform on the other hand shows how QSVT can also be used to generalize some algorithms. Phase estimation is also a great example to show how powerful an algorithm constructed with QSVT is. Besides the connections just mentioned, it also has a wide field of applications like factoring or robust phase estimation. The same goes for the idea behind Hamiltonian simulation. We already mentioned that Hamiltonian simulation is a function-evaluation problem. We can use the approach in constructing this algorithm for other function-evaluation problems as well, therefore being able to construct algorithms for a big set of problems. [5] also mentions algorithms for quantum linear systems as well as quantum walks. These are already strong indicators that QSVT can be seen as a grand unification. In the algorithms mentioned in this paper as well as other algorithms constructed by QSVT we can often see an improvement over already existing algorithms. It can construct algorithms that not only have the possibility to improve the runtime and therefore the efficiency, but also generalize existing ideas, making them applicable to wider problems. One example for this is the Harrow-Hassidim-Lloyd algorithm for linear systems [14], which could not be applied in general, for example if there was no access to the matrix's eigenvalues. QSVT can offer a more general algorithm through applying polynomial approximation. It is also important to mention once more, that [5] developed QSVT by searching for an underlying concept behind existing algorithms. Therefore QSVT being a grand unification was already a component of the framework from the start.

VI. CONCLUSION AND FUTURE WORK

We have seen different use cases, where QSVT can be useful to construct algorithms for solving specific problems. Although it can sometimes reach a slightly worse performance than already existing quantum computing approaches - like when compared to Grover's quantum algorithm for the unstructured search - it overall offers many advantages that make QSVT a powerful tool. The biggest advantage is its wide spread usability for all kinds of problems. By having one framework that can handle these problems, it is not only possible to improve already existing approaches but it can also make it easier to tackle problems, for which no algorithm exists yet. It also gives a deeper understanding on how these algorithms work and how future algorithms could look like.

With its widespread usability and the possibility to improve and generalize existing ideas, QSVT can indeed be seen as a grand unification of quantum algorithms. According to [2], QSVT also has some caveat in the use, so future work can also focus on how to improve QSVT to reduce these difficulties. One of the biggest caveats is the requirement of block-encoding. QSVT opened many ways to expand into the world of quantum computation and solving problems using quantum algorithms. Since quantum computation is a promising but also complex area, in future works we want to look at new algorithms constructed with the help of QSVT as well as developments in QSVT itself and simplify these ideas as well.

REFERENCES

[1] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, ser. STOC '96. New York, NY, USA: Association for Computing Machinery, 1996, pp. 212–219. [Online]. Available: https://doi.org/10.1145/237814.237866

[2] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, "Grand unification of quantum algorithms," *PRX Quantum*, vol. 2, p. 040203, Dec 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PRXQuantum.2.040203

[3] A. Y. Kitaev, "Quantum measurements and the abelian stabilizer problem," 1995. [Online]. Available: https://arxiv.org/abs/quant-ph/9511026

[4] G. H. Low and I. L. Chuang, "Hamiltonian simulation by qubitization," *Quantum 3*, vol. 3, p. 163, 2019. [Online]. Available: https://doi.org/10.22331/q-2019-07-12-163

[5] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2019. New York, NY, USA: Association for Computing Machinery, 2019, pp. 193–204. [Online]. Available: https://doi.org/10.1145/3313276.3316366

[6] G. H. Low, T. J. Yoder, and I. L. Chuang, "Methodology of resonant equiangular composite quantum gates," *Physical Review X*, vol. 6, 2016. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.6.041067

[7] J. M. Arrazola. (2023) Intro to qsvt. [Online]. Available: https://pennylane.ai/qml/demos/tutorial_intro_qsvt

[8] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge, UK: Cambridge University Press, 2010.

[9] E. Tang and K. Tian, "A cs guide to the quantum singular value transformation," 2023. [Online]. Available: https://arxiv.org/abs/2302.14324

[10] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *The Annals of mathematical statistics*, vol. 23, no. 4, pp. 493–507, 1952.

[11] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and quantum computation*, ser. Graduate Studies in Mathematics. American Mathematical Society, 2002, vol. 47.

[12] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124–134.

[13] D. Coppersmith, "An approximate fourier transform useful in quantom factoring," 2002. [Online]. Available: https://arxiv.org/abs/quant-ph/0201067

[14] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters*, vol. 103, no. 15, 2009. [Online]. Available: https://doi.org/10.1103/physrevlett.103.150502

Measurement-Based Quantum Computing: Deriving Universality via Cluster States

Jimmy Tan
Munich Network Management Team
LMU Munich
Munich, Germany
Jimmy.Tan@campus.lmu.de

***Abstract*—Quantum computing has traditionally been framed within the unitary circuit model, where information is processed through reversible logic gates. However, the Measurement-Based Quantum Computing (MBQC) paradigm, or the One-Way Quantum Computer, offers a fundamental shift: computation is driven by irreversible single-qubit measurements performed on a highly entangled resource known as a cluster state. This paper presents a didactic derivation of how universal quantum computation is achieved solely through cluster states and measurements. We demonstrate how operations in the quantum circuit model map to specific measurement patterns, with the aim of making the mechanisms of MBQC more accessible to beginners. Furthermore, we discuss the critical role of classical feed-forward mechanisms in correcting the inherent randomness of quantum measurements, thereby ensuring deterministic computation.**

***Index Terms*—Measurement-Based Quantum Computing (MBQC), Cluster States, One-Way Quantum Computer, Universal Quantum Computation.**

I. Introduction

Since Richard Feynman's initial proposal of quantum simulation in the 1980s [7], the primary mental model for quantum computation has been the "Quantum Circuit Model" [8]. In this framework, isolated qubits are initialized in a fiducial state (usually $|00...0\rangle$), manipulated by a temporal sequence of unitary logic gates $(U_1, U_2, \ldots, U_n)$, and finally measured to extract the result. This model is intuitive because it mirrors the architecture of classical logic gates: wires represent qubits, and time flows from left to right.

However, in 2001, Raussendorf and Briegel proposed a radically different approach: the "One-Way Quantum Computer" [1]. In this model, the difficulty of the computation is front-loaded into the creation of a massive, multi-particle entangled resource known as a Cluster State [9]. Once this state is prepared, the computation proceeds solely by performing single-qubit measurements.

While the theoretical underpinnings of this model were rigorously detailed in foundational works using the stabilizer formalism [2], these primary sources are often highly technical and dense. Conversely, Raussendorf has also produced pedagogical video tutorials [5] that sketch the intuition behind universality. However, while these resources introduce simplified measurement patterns, such as the alternative CNOT implementation discussed in this work, they often omit rigorous step-by-step derivations, delegating them as exercises for students.

Consequently, a gap remains for the intermediate learner. To address this, this paper provides a self-contained, pedagogical derivation of universality tailored for a broad academic audience. We detail derivations typically omitted in abbreviated presentations, like the ones by Raussendorf himself [5], to ensure a complete understanding. Specifically, we map the temporal logic of the circuit model to the spatial geometry of the cluster state, clarifying information propagation for the non-specialist. Prioritizing visual and geometric intuition over the stabilizer formalism, we demonstrate how 'wires' and 'gates' are carved into the entanglement substrate. By synthesizing rotations and CNOT gate derivations into a unified narrative, we connect standard quantum algorithms with measurement-based implementations.

To this end, we structure our derivation as follows: First, we introduce the Cluster State and the Stabilizer Formalism in Sec. II. In Sec. III, we rigorously derive the components of a universal gate set: beginning with state propagation (the 'Quantum Wire'), followed by arbitrary single-qubit rotations, the CNOT gate, and the protocol for removing qubits via Z-measurements. Throughout these derivations, we highlight the necessity of classical feed-forward mechanisms to ensure deterministic computation. Finally, in Sec. IV, we discuss the conceptual implications of this model, specifically the translation of temporal logic into spatial resources.

II. Background: The Substrate of Computation

To understand how a sequence of measurements can drive a deterministic computation, we must first characterize the physical resource being measured. Unlike the circuit model, where the resource is time, here the resource is the entanglement substrate itself.

A. Cluster States and Graph States

A Cluster State, a two-dimensional lattice, is a specific instance of a graph state, mathematically described by a graph $G = (V, E)$, where the vertices V represent qubits and the edges E represent Ising-type entanglement interactions (in this case, it is just CZ interactions between all adjacent qubits) [1].

The physical construction of a cluster state follows a precise, two-step protocol that separates the difficulty of entanglement generation from the logic of the computation:

1) **Initialization:** All qubits in the lattice are initialized in the superposition state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. This creates a uniform superposition across the entire register.
2) **Entanglement:** A Controlled-Phase (CZ) gate is applied between all pairs of qubits connected by an edge in the graph. The CZ gate applies a phase of -1 to the joint state only if both qubits are in the state $|1\rangle$.

Fig. 1 shows the cluster state. The blue circles are the initialized qubits in the $|+\rangle$ state and the line connecting those dots represents the CZ interactions. A crucial feature of this construction is that all CZ gates commute. Consequently, the order of their application is irrelevant, allowing for the massive, simultaneous generation of entanglement across the entire lattice in a single time step [5].

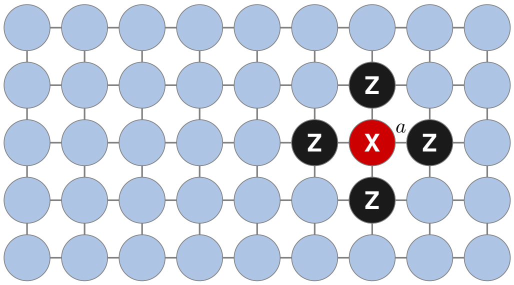


Fig. 1. The 2D cluster state lattice. Vertices represent qubits initialized in $|+\rangle$, and edges represent CZ entanglement interactions. The highlighted overlay illustrates the stabilizer structure (see Sec. II-B), where the operator K_a consists of a Pauli-X on the central qubit (red) and Pauli-Z operators on its neighbors (black) [5].

B. The Stabilizer Formalism

While the physical construction is intuitive, the computational properties of the cluster state $|\phi\rangle_C$ are best described using the stabilizer formalism [10]. Instead of writing out the full wavefunction, this formalism defines the state by a set of ”rules” or stable properties that it must satisfy. Specifically, the cluster state is defined as the unique state where measuring the following operators (illustrated by the colored overlay in Fig. 1) always yields the value $+1$:

$$K_a = X_a \bigotimes_{b \in N(a)} Z_b \tag{1}$$

Where a is a specific qubit, $N(a)$ denotes the set of its connected neighbors in the lattice, and X and Z are the standard Pauli matrices.

C. Teleportation as a Primitive

The fundamental engine of MBQC is a technique known as ”One-Bit Teleportation” [11]. Unlike the circuit model, where a logical state physically resides on a specific wire and evolves over time, in MBQC, the logical state is teleported from one physical qubit to its neighbor via measurement [1].

To illustrate this, consider the simplest possible cluster state: two qubits connected by a CZ interaction. Let qubit 1 carry an arbitrary logical state $|\psi\rangle$ and qubit 2 be initialized in $|+\rangle$. The total state is $|\Psi\rangle = CZ(|\psi\rangle_1 \otimes |+\rangle_2)$.

If we measure qubit 1 in the X-basis, it is ”consumed” and removed from the cluster. Through the phenomenon of gate teleportation, the quantum information $|\psi\rangle$ is not destroyed; rather, it is transferred to qubit 2, subjected to a Hadamard operation H.

D. The Necessity of Feed-Forward (Deterministic Correction)

However, because quantum measurements are inherently probabilistic, the transfer is not always identical. Referring back to the example in Sec. II-C, the measurement outcome s (where $s = 0$ for outcome $+1$ and $s = 1$ for outcome -1) introduces a specific Pauli byproduct: if $s = 0$, the state on qubit 2 is $H|\psi\rangle$; if $s = 1$, the state on qubit 2 is $XH|\psi\rangle$. We must record this outcome and ”feed it forward” to correct future qubits [2].

E. Gate Application via Basis Rotation

The teleportation mechanism serves as more than just a wire. By changing the measurement basis of qubit 1 from the standard X-basis to a rotated basis in the X-Y plane (defined by an angle ϕ), we modify the unitary transformation applied to the teleported state. Instead of a simple Hadamard, the state undergoes a rotation $R_z(\phi)$ followed by H. Thus, in the One-Way Computer, state transfer (teleportation) and gate application are the exact same physical process [2].

III. Universality Proof

To prove that the Measurement-Based Quantum Computer (MBQC) is a universal model, we must demonstrate its ability to simulate any arbitrary quantum circuit. Universality is achieved if a model can implement a set of gates capable of approximating any unitary operation with arbitrary precision [6]. Typically, this set consists of:

1) Arbitrary single-qubit rotations.
2) A single two-qubit entangling gate (the CNOT gate).

However, unlike the circuit model where these gates are applied to static qubits over time, MBQC “carves” these operations out of the cluster state. Therefore, to establish universality in the One-Way model, we must at least demonstrate how the cluster state supports the following five structural primitives:

1) **State Propagation (The Quantum Wire):** Unlike the circuit model, where a wire represents a qubit evolving in time, in MBQC the physical qubits are stationary resources. To create a logical “wire,” we must transport the quantum information from one physical site to another. This is achieved via the “One-Bit Teleportation” mechanism, effectively simulating a wire by propagating the state across the cluster.
2) **Arbitrary Single-Qubit Rotations:** We must show that by altering the basis of the measurement (specifically measuring in the X-Y plane at an angle ϕ), we can induce specific unitary rotations on the teleported state.

3) **Two-Qubit Entanglement (CNOT):** We must derive a geometric measurement pattern that interacts two parallel "logical wires" within the 2D cluster lattice to reproduce the logic of a CNOT gate.
4) **Removing Qubits from the Cluster:** A generic 2D cluster state acts as a blank canvas. We require a method to delete unnecessary qubits to isolate specific logical paths. This is achieved by measuring unwanted qubits in the Z-basis, effectively removing them from the graph entanglement.
5) **Deterministic Correction (Feed-Forward):** Measurements are inherently probabilistic. To ensure deterministic computation, we must prove that "unwanted" measurement outcomes (Pauli errors) can be corrected by adapting the measurement angles of future qubits (classical feed-forward).

In this section, most of our derivations for the Universality Proof stem from Raussendorf's YouTube lecture [5], unless stated otherwise.

A. The Quantum Wire (Identity Gates)

To simplify the proof of state propagation, we restrict our analysis to a one-dimensional three-qubit cluster state. We demonstrate the validity of the quantum wire by performing a series of circuit equivalence transformations step-by-step, mapping the physical realization in Fig. 2 to the simplified logical circuit in Fig. 5.

We begin with the canonical cluster implementation depicted in Fig. 2. Consistent with the entanglement protocol defined in Section II, CZ gates are applied sequentially to adjacent qubit pairs $(1,2)$ and $(2,3)$. We define the register such that qubit 1 represents the logical input initialized in an arbitrary state $|\psi\rangle$, while the subsequent qubits are initialized in the superposition state $|+\rangle$. The protocol concludes with X-basis measurements on qubits 1 and 2. The collective effect of this circuit is the teleportation of the input state $|\psi\rangle$ from the first to the third qubit. This also means that **measuring** qubit 1 & 2 in the X-basis **moves** the arbitrary state $|\psi\rangle$ from qubit 1 to qubit 3 (which is why it is called the quantum wire).

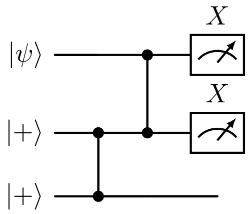


Fig. 2. The canonical cluster implementation (physical quantum wire) [5]

To verify this teleportation, we first transform Fig. 2 into the equivalent circuit shown in Fig. 3. This is achieved by inserting two consecutive Hadamard gates on the second qubit, positioned between the two CZ gates. Since the Hadamard gate is self-inverse ($H^2 = I$), this is essentially an identity operation which makes both circuits mathematically identical.

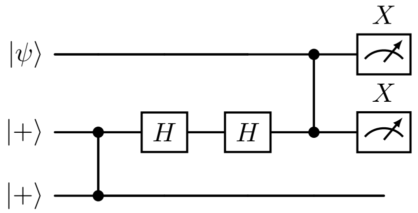


Fig. 3. Equivalent transformation from Fig. 2 by adding two Hadamards as an Identity gate [5]

The transformation from Fig. 3 to Fig. 4 requires propagating the final X-basis measurements on qubits 1 and 2 logically "forward" through the circuit. Specifically, the measurement on the first qubit must commute through the CZ gate, while the measurement on the second qubit must traverse both the CZ gate and the Hadamard gate. While the validity of these transformations can be rigorously proven via the stabilizer formalism as proposed by Raussendorf in his video [5], we omit the exhaustive derivation here. Instead, we verify these commutations using standard operator transformation tables, such as those provided in the documentation from the Stim [4] repository. This approach leverages the Heisenberg picture of quantum mechanics [12]. Unlike the more common Schrödinger picture which tracks the temporal evolution of the state vector (e.g., $|0\rangle \rightarrow |+\rangle$), the Heisenberg picture tracks how operators evolve as they pass through unitary gates. By applying the Pauli frame changes listed in Table I and Table II, we effectively simulate this operator evolution, confirming that the circuit in Fig. 3 is mathematically equivalent to Fig. 4.

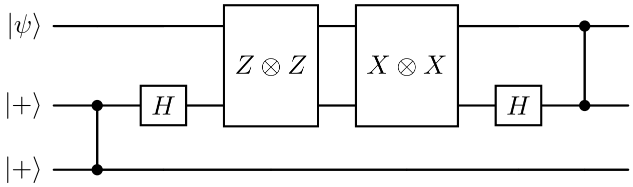


Fig. 4. Circuit with propagated measurements (equivalent transformation of Fig. 3) [5]

The bottom left part of Fig. 4 that includes both $|+\rangle$ of qubit 2 and 3, the CZ gate, as well as the Hadamard gate on qubit 2, makes up a Bell state, specifically the $|\Phi^+\rangle$ state. The subsequent parity measurements, $Z \otimes Z$ and $X \otimes X$, are mathematically equivalent to a complete Bell measurement. While Raussendorf's video [5] omits the mathematical derivation, this equivalence arises because the Pauli product operators $Z \otimes Z$ and $X \otimes X$ commute ($[Z \otimes Z, X \otimes X] = 0$) and share the four Bell states as their unique simultaneous eigenstates. Specifically, the $Z \otimes Z$ measurement extracts the "bit parity" (distinguishing the even-parity subspace $\{|\Phi^\pm\rangle\}$ from the odd-parity subspace $\{|\Psi^\pm\rangle\}$), while the $X \otimes X$ measurement extracts the "phase parity" (distinguishing the superpositions with positive phase $\{|\Phi^+\rangle, |\Psi^+\rangle\}$ from those with negative phase $\{|\Phi^-\rangle, |\Psi^-\rangle\}$). Together, the joint outcomes of these two observables yield the two classical bits necessary to uniquely identify the Bell state. Consequently, Fig. 4 is mathematically equivalent to Fig. 5, which depicts the standard quantum teleportation protocol [3]. The Hadamard gate on qubit 2 and CZ gate on qubit 1 & 2 can be ignored after the Bell measurement for this purpose.

With this teleportation protocol, we can easily see that this circuit teleports the arbitrary qubit state $|\psi\rangle$ from qubit 1 to qubit 3, thereby forming the aforementioned "Quantum Wire".

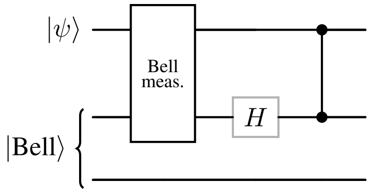


Fig. 5. Standard quantum teleportation protocol (equivalent transformation of Fig. 4). The H gate and the CZ gate after the Bell measurement can be ignored in this context. [5]

However, a more intuitive interpretation is that Fig. 13 corresponds to the top portion of Fig. 2. As derived in Sec. III-C, this component implements a Hadamard gate. Consequently, Fig. 2 effectively applies the Hadamard gate twice, resulting in an Identity gate.

Input Pauli Operator	Output Pauli Operator
X	Z
Z	X

TABLE I
STABILIZER GENERATORS FOR THE H-GATE [4]

Input Pauli Operator	Output Pauli Operator
X_	XZ
Z_	Z_
_X	ZX
_Z	_Z

TABLE II
STABILIZER GENERATORS FOR THE CZ-GATE [4]

B. Arbitrary Single-Qubit Rotations

We now progress to step two of the universality proof and discuss the implementation of single-qubit rotations in MBQC. Fig. 6 has a familiar structure of the previous protocol.

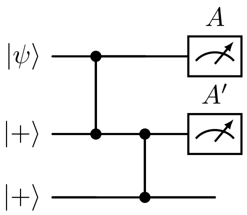


Fig. 6. [5]

This is like the identity gate but what is changed now are the measurements. Previously we are looking at measurement in the X-basis, but we are now looking at more general one-qubit measurement. In this structure, we can see familiar substructures, which are boxed in red in Fig. 7.

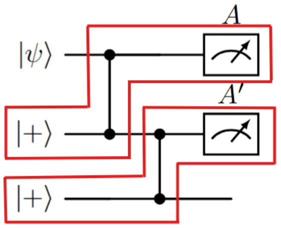


Fig. 7. [5]

Let's look at the top structure first, and now consider the observable $A = U_z^\dagger X U_z$, with $U_z = exp(i\alpha Z)$. Recall that arbitrary rotation gates are defined as:

$$R_P(\theta) = e^{-i\frac{\theta}{2}P} = \cos\left(\frac{\theta}{2}\right) I - i \cdot \sin\left(\frac{\theta}{2}\right) P; \quad P \in \{X, Y, Z\}$$

Therefore, U_z simply means rotation around the Z-axis. With these, the top structure will be as illustrated in Fig. 8, which is also mathematically equivalent to Fig. 9.

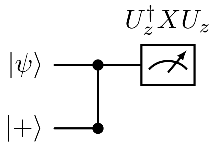


Fig. 8. [5]

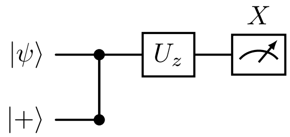


Fig. 9. Equivalent transformation of Fig. 8 [5]

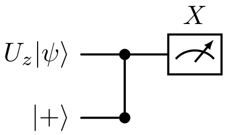


Fig. 10. Equivalent transformation of Fig. 9 [5]

Because the rotation gate U_z commutes with the CZ gate, it can then be re-expressed as the structure in Fig. 10. This configuration is termed 'half-teleportation' because it constitutes a portion of the full teleportation circuit shown in Fig. 2. The resulting state transformation is described by:

$$|\psi\rangle \to |\psi'\rangle = HZ^s U_z |\psi\rangle$$

Functionally, this circuit implements a Hadamard gate followed by a conditional Pauli-Z correction determined by the X-basis measurement outcome s. The correction is applied only when the negative eigenvalue (-1) is measured; it is omitted for the positive eigenvalue $(+1)$.

We can then do the same for the bottom structure of Fig. 7 by taking the entire state of $|\psi'\rangle = HZ^sU_z|\psi\rangle$ from above and treat it as the input for the bottom structure. With the same logic, combining both ”half-teleportation” yields:

$$|\psi\rangle \to |\psi''\rangle = (HZ^{s'}U'_z)(HZ^sU_z)|\psi\rangle = HZ^{s'}U'_zHZ^sU_z|\psi\rangle$$

Since Hadamard gates annihilate pair-wise (self-inverse), we can simplify it by bringing the first Hadamard forward in time to the Hadamard in the middle. By doing so, the $Z^{s'}$ and U'_z are then conjugated into $X^{s'}$ and U'_x (refer to Table I again for this conjugation). Therefore, the combination of two ”half-teleportation” then yields:

$$|\psi\rangle \to |\psi''\rangle = X^{s'}U'_xZ^sU_z|\psi\rangle$$

This derivation is significant, as it demonstrates the capability to perform rotations about both the Z and the X-axis. Intuitively, just as walking along lines of longitude and latitude allows one to reach any destination on Earth, alternating rotations around these orthogonal axes allows us to navigate the entire surface of the Bloch sphere. Formally, this creates a universal set for single-qubit control, as any arbitrary unitary gate can be synthesized by a sequence of these rotations (e.g., via Z-X-Z decomposition [13]).

C. CNOT Gate

As mentioned in Section III, the implementation of CNOT gate is inseparable from achieving universality. Raussendorf provided two alternatives for realising it on the cluster state. The first alternative, as illustrated in Fig. 11, is the original implementation provided by Raussendorf that involves measurement in the X- and the Y-basis [2]. The second alternative, as illustrated in Fig. 12, is the easier implementation of CNOT from Raussendorf's tutorial on YouTube [5]. Although Raussendorf has provided the proof for the first alternative in his paper [2], the proof for the second alternative is not shown, but rather it is left as an exercise for students. Here, we prove, step by step, that it implements a CNOT gate.

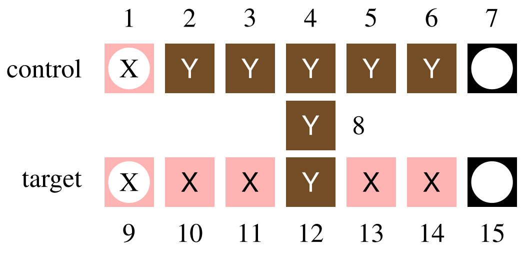


Fig. 11. Realisation of CNOT-gate from Raussendorf's paper [2]

Recall that in the implementation of the arbitrary single-qubit rotation protocol in subsection III-B, we can implement any qubit rotation using one of the sub-structures (boxed in red) as illustrated in Fig. 7.

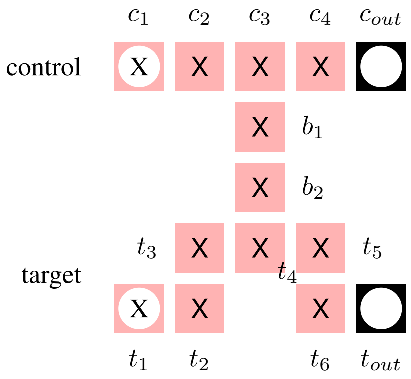


Fig. 12. Alternative realisation of CNOT-gate from Raussendorf's YouTube tutorial [5]

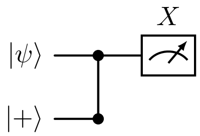


Fig. 13.

Let's extract one of the sub-structures and change the measurement basis to the X-basis, as illustrated in Fig. 13. The input qubit 1 has some arbitrary state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, and qubit 2 is prepared in the plus state $|+\rangle$. After the CZ gate, the total state will become the following:

$$|\Psi_{12}\rangle = \alpha|0\rangle_1|+\rangle_2 + \beta|1\rangle_1|-\rangle_2$$

Now, we measure qubit 1 in the X-basis. Since the eigenvectors of Pauli-X are $|+\rangle$ & $|-\rangle$, we project qubit 1 onto these vectors. In an ideal case, when the measurement outcome is $+1$, the projection is as follows:

$$|\Psi_{12}\rangle = \alpha\langle +|0\rangle|+\rangle_2 + \beta\langle +|1\rangle|-\rangle_2$$

(Recall that $\langle +|0\rangle = \frac{1}{\sqrt{2}}$ and $\langle +|1\rangle = \frac{1}{\sqrt{2}}$)

$$= \frac{1}{\sqrt{2}}(\alpha|+\rangle_2 + \beta|-\rangle_2)$$

Let's look at the resulting state on qubit 2:

$$\alpha|+\rangle + \beta|-\rangle$$

Substituting definitions of $|+\rangle$ and $|-\rangle$:

$$= \alpha\frac{|0\rangle + |1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$= \frac{1}{\sqrt{2}}[(\alpha+\beta)|0\rangle + (\alpha-\beta)|1\rangle]$$

Note that this is exactly $H|\psi\rangle$. So measuring in X-basis essentially performs a Hadamard gate if the measurement outcome is $+1$. This clearly demonstrates that the state propagation and computation is done simultaneously in MBQC (in this case, performing a Hadamard gate and transmitting the logical state to qubit 2 by a single measurement in the X-basis). What happens then, if the measurement result is -1?

By performing the calculation (analogous to the derivation above), one concludes that the state of the qubit 2 would be $\alpha|+\rangle - \beta|-\rangle$. We would have to do a Pauli-X correction on that state to correct the result, which falls under the **"Deterministic Correction (Feed-Forward)"** part of Section III but we will not perform rigorous proof for this in the scope of this paper.

Since two Hadamards implement an Identity gate, we can treat even numbers of X-measurements as Identity gates (quantum wire) and odd numbers as Hadamard gates. Note that by stacking the structure of Fig. 13, we get the exact same structure as depicted in Fig. 2 in subsection III-A (the quantum wire proof), which just acts as an Identity gate.

Using this logic, we can see that in Fig. 12, the control row of qubits $(c_1, ..., c_4)$ has even number of X-measurements that essentially implements an Identity gate. The two "bridge" qubits (b_1, b_2) that connects the control row to the target row also effectively implements an Identity gate.

Focusing now on the target row, from the qubit t_1 to t_3, these measurements essentially applies an initial Hadamard. Now the logical qubit is sitting at t_4 and the bridge connects t_4 to c_3. Since the connection of any adjacent qubits in the cluster states are defined as CZ, this essentially applies a CZ gate between the control and the target. Now, to move the state from t_4 to the output, we measure in X-basis for another 3 times, and this again applies another Hadamard. Recall that "sandwiching" a Z-gate between two Hadamards turns it into an X-gate, now we can see that the whole circuit is essentially just a CNOT-gate which can be seen as:

$$\text{Circuit} = (I \otimes H) \cdot \text{CZ} \cdot (I \otimes H) = CNOT$$

D. Removing Qubits from the Cluster

As mentioned previously, a generic 2D cluster state acts as a blank canvas. Therefore, We need a way to delete unnecessary qubits to isolate specific logical paths. This is accomplished by measuring the unnecessary qubits in the Z-basis, which effectively eliminates them from the graph entanglement.

Raussendorf has trivially proven how qubits can be removed from the cluster states with Z-measurement [5]. For this, let's look at the trivial 1D 3-qubit cluster states again, as illustrated in Fig. 14. The goal of this proof is to show that measuring the bottom-most qubit in the Z-basis (as shown in the LHS) yields a circuit 'equivalent' to the RHS, provided the measurement outcome is +1.

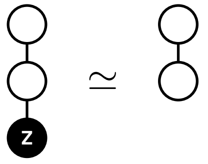


Fig. 14. [5]

Let's consider the circuit in Fig. 15. We assume that the outcome obtained in the Z-measurement on qubit 3 is $|0\rangle$, which is the eigenstate of Pauli-Z with positive eigenvalue $(+1)$.

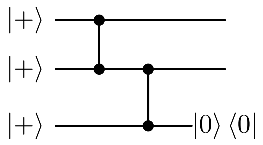


Fig. 15. [5]

Since the measurement commutes with the CZ gate on qubit pair $(2, 3)$, we can then propagate it backwards in time as shown in Fig. 16.

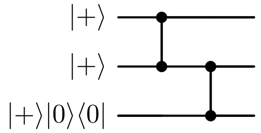


Fig. 16. [5]

It is then equivalent to Fig. 17, since the state $|+\rangle|0\rangle\langle 0|$ is nothing more than just the state $|0\rangle$.

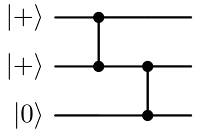


Fig. 17. [5]

Furthermore, we also know that if an input $|0\rangle$ is involved in a CZ gate, then the gate does nothing, because it will only flip the phase when both qubits have the state $|1\rangle$. Therefore, we can simply remove the CZ gate, and as you can see in Fig. 18, the third qubit is effectively removed from the cluster state.

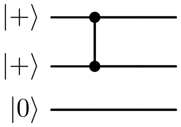


Fig. 18. [5]

However, this is only true when the measurement result in the Z-basis has positive eigenvalue ($|0\rangle$), which is why we put quotation marks between the word 'equivalent' earlier. What Raussendorf didn't show in his video [5], however, is what happens when the measurement result has negative eigenvalue ($|1\rangle$). We'll show it here step-by-step.

Referring to Fig. 15 again, our start state looks like the following:

$$|\Psi_{123}\rangle = |+\rangle_1 |+\rangle_2 |+\rangle_3$$

which is entangled by CZ_{12} and CZ_{23}. Let's write out the 8 terms **with the coefficients ignored**. A term only gets a sign flip if a pair of neighbors are both $|1\rangle$. Those 8 terms are:

$$+|000\rangle, +|001\rangle, +|010\rangle, -|011\rangle,$$

$$+\left|100\right\rangle, +\left|101\right\rangle, -\left|110\right\rangle, +\left|111\right\rangle$$

If the measurement outcome is $\left|1\right\rangle$ on qubit 3, then the whole state collapses into a state with only these 4 following terms:

$$+\left|001\right\rangle, -\left|011\right\rangle, +\left|101\right\rangle, +\left|111\right\rangle$$

And since every term ends in $\left|1\right\rangle_3$, we can pull it out:

$$\left|\Psi_{123}\right\rangle = (\left|00\right\rangle - \left|01\right\rangle + \left|10\right\rangle + \left|11\right\rangle)_{12} \otimes \left|1\right\rangle_3$$

We can clearly see that qubit 3 is now fully separated, therefore the wire between qubit 2 and 3 is cut. Now the question is: what happens to the neighbouring qubit 2? To answer this, let's look at the original cluster state with just qubit 1 & 2.

$$\left|+\right\rangle_1 \left|+\right\rangle_2 \xrightarrow{CZ} \left|00\right\rangle + \left|01\right\rangle + \left|10\right\rangle - \left|11\right\rangle$$

Now compare this with the previous result of $(\left|00\right\rangle - \left|01\right\rangle + \left|10\right\rangle + \left|11\right\rangle)_{12}$, we can see that qubit 2 has effectively received a Z-gate, which needs to be corrected as mentioned in the **"Deterministic Correction (Feed-Forward)"** part of Section III.

In short, if the measurement result is $\left|0\right\rangle$, then the qubit is effectively removed from the cluster state without any side effects. However, if the measurement results in $\left|1\right\rangle$, the qubit will still be removed from the cluster state, but all of its directly neighboring qubits will receive a Z-gate operation, which needs to be corrected.

IV. Discussion

A. Scope and Limitations

As this paper is intended to serve as a first point of entry for readers with basic knowledge of quantum computing, our primary focus has been establishing the physical realization of a universal gate set. By rigorously deriving the quantum wire, arbitrary single-qubit rotations, and the CNOT gate, we have confirmed that the cluster state possesses the necessary computational power to simulate any unitary evolution.

Consequently, we omit the formal derivation of gate composition and the global Pauli-tracking algorithm (feed-forward). However, the critical importance of deterministic correction was explicitly demonstrated throughout our derivations in Section III. We established that measurement outcomes are probabilistic and can introduce specific Pauli errors, such as the Z-correction required during "half-teleportation" or the corrections necessitated by removing qubits via Z-measurements. While the full classical control logic is outside the scope of this work, we have shown locally when these corrections must occur to ensure the computation remains deterministic.

B. Spatial vs. Temporal Resources

The derivations presented in Section III highlight a fundamental conceptual shift: the trade-off between time and space. In the standard circuit model, complex algorithms are executed by applying gates sequentially over time to a fixed number of qubits. In contrast, MBQC "front-loads" the computational difficulty into the creation of the cluster state. Complexity is no longer defined by circuit depth (time), but by the spatial dimensions of the cluster state resource. For instance, our derivation of the "Quantum Wire" reveals that simply maintaining a logical qubit in memory requires the continuous consumption of physical entanglement, effectively converting the "time" resource of the circuit model into the "space" resource of the cluster lattice.

V. Conclusion

In this work, we have presented a comprehensive, pedagogical derivation of universality in Measurement-Based Quantum Computing. By explicitly expanding upon the abbreviated presentations found in resources like Raussendorf's tutorials [5], we have supplied the rigorous step-by-step proofs necessary for a complete understanding. This detailed reconstruction demonstrated that the temporal logic of the quantum circuit model is not merely analogous, but mathematically equivalent to spatial measurement patterns on a generic 2D cluster state.

Technically, we established that "One-Bit Teleportation" serves as the primary engine of information transport, allowing logical qubits to propagate through the lattice. We further proved that by rotating the measurement basis, we can enact arbitrary unitary rotations simultaneously with this propagation. Finally, by adopting one of Raussendorf's proposed measurement structures and rigorously deriving its operation, we confirmed that a CNOT gate can indeed be implemented, completing the universal gate set.

The conceptual significance of this derivation lies in how these operations are realized. We observed that "gates" in this model are not external operations applied to the system, but are carved out of the entanglement substrate itself. This proves that the irreversible nature of measurement does not limit computational power; rather, it is the very mechanism that drives information processing.

References

[1] R. Raussendorf, H. Briegel. "A One-Way Quantum Computer," in Phys. Rev. Lett., vol. 86, pp. 5188–5191, 2001.

[2] R. Raussendorf, D. Browne, H. Briegel. "Measurement-based quantum computation on cluster states," in Phys. Rev. A, vol. 68, pp. 022312, 2003.

[3] Bennett, C., et al. "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels," in Phys. Rev. Lett., vol. 70, pp. 1895–1899, 1993.

[4] Craig, G. (2024) Stim v1.13 Gate reference, GitHub. Available at: https://github.com/quantumlib/Stim/wiki/Stim-v1.13-Gate-Reference (Accessed: 12 December 2025).

[5] Raussendorf, R. (2020) MBQC Intro Part 2, YouTube. Available at: https://youtu.be/g4NAqTHRYD4?si=l9weHDWT2NufJnTg (Accessed: 14 December 2025).

[6] A Yu Kitaev (1997). Quantum computations: algorithms and error correction. Russian Mathematical Surveys, 52(6), 1191.

[7] R. P. Feynman, "Simulating physics with computers," International Journal of Theoretical Physics, vol. 21, no. 6, pp. 467–488, Jun. 1982, doi: 10.1007/BF02650179.

[8] D. E. Deutsch, "Quantum computational networks," Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, vol. 425, no. 1868, pp. 73–90, Sep. 1989, doi: 10.1098/rspa.1989.0099.

[9] H. J. Briegel and R. Raussendorf, 'Persistent Entanglement in Arrays of Interacting Particles', Phys. Rev. Lett., vol. 86, pp. 910–913, Jan. 2001.

[10] D. Gottesman, Stabilizer codes and quantum error correction. California Institute of Technology, 1997.

[11] X. Zhou, D. Leung, I. Chuang. "Methodology for quantum logic gate construction," in Phys. Rev. A, vol. 62, pp. 052316, 2000.

[12] J. Sakurai, J. Napolitano, Modern quantum mechanics. Cambridge University Press, 2020.

[13] M. Nielsen, I. Chuang, Quantum computation and quantum information. Cambridge university press, 2010.