

https://dspace.org/

Photonic circuits with light patterns

Koni, Mwezi

2025-06
University of the Witwatersrand, Johannesburg
https://hdl.handle.net/10539/47674
Koni, Mwezi. (2025). Photonic circuits with light patterns. [University of the Witwatersrand, Johannesburg]. WIReDSpace.

Downloaded from a DSpace repository. https://dspace.org/

UNIVERSITY OF THE WITWATERSRAND

MASTER DISSERTATION

Photonic circuits with light patterns

Author:
Mwezi KONI

Supervisors:
Dr Isaac NAPE
Prof. Andrew FORBES



WITS
UNIVERSITY

A thesis submitted in fulfillment of the requirements
for the degree of Masters of Science

in the

School of Physics

June 4, 2025

Declaration of Authorship

I, Mwezi KONI, declare that this thesis, "Photonic circuits with light patterns", is my own work. It was completed during my candidature for a research degree at this University. Any prior submissions for a degree are clearly stated, all sources are properly cited and acknowledged, and my contributions in collaborative work are explicitly specified.

Signed: 

Date: 05 June 2025

"More is different."

Philip W. Anderson

UNIVERSITY OF THE WITWATERSRAND

Abstract

Masters of Science

Photonic circuits with light patterns

by Mwezi KONI

Optical circuits leveraging structured light—the precise control of light's degrees of freedom (DOFs), including tranverse space (i.e amplitude and phase), frequency, and polarization—offer a promising platform for quantum information processing tasks. Information can be encoded in high dimenional states, and different degrees of freedom can be coupled creating hybrid structures for complex operations. Recent advancements in tunability techniques, particularly through spatial light modulators (SLMs), have enabled the precise encoding of light patterns as holograms. This capability facilitates complex, programmable operations with applications in imaging, cryptography, and communications, making SLMs indispensable tools for manipulating high-dimensional optical fields.

Building on these advancements, this dissertation explores how SLM-powered optical circuits can encode unitary operations by dynamically shaping light patterns to perform matrix multiplications—an essential component of quantum computation. The core contribution of this work is the demonstration of optical matrix multiplication as a means to emulate quantum computations, where quantum states are encoded as spatial light patterns and unitary operations are implemented via phase modulation. This approach exploits the inherent parallelism of the transverse spatial degree of freedom to efficiently simulate quantum algorithms within an optical framework.

By establishing a framework for optical circuits with structured light patterns, this dissertation provides a pathway for leveraging classical optical systems to simulate quantum computations. Inspired by Jozsa's insight that classical waves can efficiently mimic quantum algorithms without requiring entanglement, these findings contribute to the growing field of hybrid classical-quantum computing and highlight the potential of optical platforms in advancing scalable quantum technologies.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Prof. Andrew Forbes and Dr. Isaac Nape, for their invaluable guidance, mentorship, and critical feedback. Their support has refined my ideas and inspired me to become a more dedicated researcher.

I would also like to extend my sincere thanks to the lab seniors for their unwavering support. In particular, I am especially grateful to Dr. Angela Dudley for her invaluable assistance throughout my research journey; her kindness has made all the difference.

I gratefully acknowledge the financial support provided by the Council for Scientific and Industrial Research (CSIR) and Saquti through their scholarship programs, which have been instrumental in enabling this research.

My heartfelt thanks extend to all lab members for fostering a collaborative and inspiring environment, with special recognition to Hadrian Bezuidendhout, with whom I have worked extensively, and Pedro Ornelas for his insightful discussions.

Finally, I am deeply thankful to my friend, Light Mkhumbuza, and my family for their constant encouragement and support.

Contents

Declaration of Authorship iii

Abstract vii

1 Introduction 1

1.1 Quantum computing basics . . . 3

1.1.1 The quantum bit . . . 4

1.1.2 Multi-qubit systems . . . 5

1.1.3 Quantum gates and the circuit model of computation . . . 7

Boolean functions and the classical circuit Model . . . 8

Quantum gates and universality . . . 10

1.2 Degrees of freedom of light for information encoding and manipulation 14

1.2.1 Polarization as a two-level encoding system . . . 15

Jones calculus and SU(2) transformations . . . 16

Quarter-Wave Plate (QWP) . . . 17

Half-Wave Plate (HWP) . . . 17

Minimal SU(2) setup for polarization transformations . . . 18

1.2.2 Path encoding . . . 18

1.2.3 Continous temporal and spatial . . . 20

Temporal degrees of freedom: encoding in time bins . . . 21

Optical implementations for time-bin transformations . . . 22

Spatial basis . . . 22

1.3 Dissertation outline . . . 26

2 Experimental Techniques: Toolbox for optical matrix multiplication 29

2.1 Spatial Light Modulators: Principles and Applications . . . 29

2.1.1 Overview of SLM Operation . . . 29

2.1.2 Phase Modulation with SLMs . . . 31

2.1.3 Hologram Construction . . . 31

2.2 The Spatial Fourier Transforming Property of a Lens 35
2.3 Combining all tools for vector matrix multplication 38
2.3.1 Alignment verification 41

3 Learning to solve a useful problem on a quantum computer: reconstructing quantum states 43
3.1 Theory 44
3.1.1 State reconstruction as least square problem 44
3.1.2 VQE solution 48
3.1.3 Results and discussion 49

4 Emulating quantum computing with Matrix Multiplication 55
4.1 Introduction 55
4.2 Theory 56
4.2.1 Formulation using quantum states 56
4.3 Hadamard Gate 58
4.3.1 Application: Deutsch-Jozsa algorithm 61
4.4 Experiment 64
4.5 Results 65
4.5.1 Matrix-Multiplication 65
4.5.2 Deutsch-Jozsa Algorithm demonstration 66
4.6 Discussion and Conclusion 68

5 Conclusion and Future Work 71
5.1 Universality 72
5.2 Encoding Efficiency 73
5.3 Scalability 73
5.4 Query Complexity and Parallelization 74
5.4.1 Standard Query Model 74
5.4.2 OVMM as a Parallel Query Model 74

A Introduction to Quantum State Tomography 77
A.1 Formulating the Quantum State Estimation Problem 77
A.1.1 Measurement and Basis Selection Using Pauli Operators 78
Approaches to State Reconstruction 79

B Characterisation Based on the Hadamard Gate/Matrix **81**

C Variational Quantum Algorithm code **83**

D Exprimental results with fully labelled bitstring distribution **95**

List of Figures

1.1 The Bloch sphere representation of a qubit 5
1.2 Classical logical components of Disjunctive Normal Form (DNF) and circuit model of quantum computation 7
1.3 An illustration of light's different degrees of freedom for information encoding 14
1.4 Manipulation of information encoded across various DOFs 16
1.5 An illustration of discrete basis in the spatial transverse profile of light fields 23

2.1 A simplified diagram of the SLM 30
2.2 SLM hologram pattern construction for imparting orbital angular momentum (OAM) onto a Gaussian beam 31
2.3 Plot of the inverse of the first-order Bessel function 33
2.4 Experimentally generated spatial modes LG_{10}, HG_{10}, and HG_{11} with their holograms 34
2.5 Simulation of a lens performing a spatial Fourier transform. 35
2.6 Schematic illustration of the 1D Fourier transform by a cylindrical lens 37
2.7 Schematic of the optical matrix-vector multiplication setup. 39
2.8 Experimental verification of alignmen by modal decomposition 41

3.1 Concept figure on quantum state tomography. 44
3.2 VQE Algorithm Workflow 46
3.3 Training Data for Various Circuit Parameters 47
3.4 Variational circuit ansätze with different entanglement structures. . . . 50
3.5 Bitstring Quasi-Probability Distribution and Experimental Density Matrix Reconstruction. 51

4.1 Conceptual illustration of our encoding scheme in the transverse spatial degree of freedom 56

4.2 Conceptual illustration of the implementation of Deutsch-Jozsa algorithm 59
4.3 Experimental setup for implementing optical vector matrix multiplication(OVMM) 62
4.4 Experimental setup characterization results 65
4.5 Deutsch-Jozsa algorithm results 67

D.1 Predicted bitstring distribution with all labelled measured bitstrings . 95

List of Tables

1.1 Conjunction terms for the majority function in DNF. 9

3.1 Summary of cost function convergence for different ansatz strategies at circuit depths 1 and 3. The no-entanglement ansatz achieves the lowest cost function, while entangled ansätze fail to reach zero. 52

List of Publications

Journal articles

1. **Koni, Mwezi**, Hadrian Bezuidenhout, and Isaac Nape. "Emulating quantum computing with optical matrix multiplication." APL Photonics 9.10 (2024). *M Koni built and perfomed the experiment. He also contributed to the data analysis and writing of the maniscript*

2. Bezuidenhout, Hadrian, **Mwezi Koni**, Jonathan Leach, Paola Concha Obando, Andrew Forbes, and Isaac Nape. "Variational approach to learning photonic unitary operators." Optics Express 32, no. 20 (2024): 35567-35578. *M Koni served as a supporting co-author assisting with planning, building and optimising the experiment.*

3. **Koni Mwezi**, Kassim Shawal, Obando Paola and Nape Isaac. Photonic state reconstruction on a quantum computer. **Manuscript in preparation** *M Koni contributed to the theory and performed the experiment. He also contributed to the writing of the manuscript*

Conference Proceedings

1. **Koni, Mwezi**, et al. "Harnessing OVMMs for quantum algorithms: an approach employing Gaussian modes." Liquid Crystals Optics and Photonic Devices. Vol. 13016. SPIE, 2024

2. Nape, I., **Koni, M.**, Bezuidenhout, H., Forbes, A. (2024, September). Optical matrix multiplication as an analogue for quantum computing. In Laser Science (pp. JTu5A-9). Optica Publishing Group.

Chapter 1

Introduction

Controlling various degrees of freedom of light, i.e. time, frequency, space and momentum, has become an emerging and promising tool for numerous information processing tasks in classical and quantum domains, ranging from novel imaging methods [1–7], communications [8–13] and computation [14–17], all harnessing the high dimensional nature of structured light fields [18–22]. This is because spatial modes enable for information encoding in qudit spaces for dimensions $d > 2$ per particle as opposed to the $d = 2$ (qubits) encoding levels offered by traditional qubit encoding such as with polarisation states. Because qubits are easy to control, they are a reliable resource for numerous protocols.

In traditional quantum computing, encoding basis states are typically formed from multiple qubits and the operations needed to construct arbitrary unitary gates in $d > 2$ dimensions often involve a mix of two-dimensional gate operations (such as the Hadamard gate and T-gate) along with a multi-qubit operations (like the Control-not gate) [23]. This allows for precise and efficient execution of arbitrary computations that are required to demonstrate quantum advantage [24]. Given that each qubit is restricted to two dimensions, the only way to enhance encoding capacity is by increasing the number of qubits and ensuring optimal connectivity among them. An alternative is to utilise qudits, which are particles that occupy $d > 2$ dimensions, because they offer larger encoding state spaces [25] and promise robustness against noise [26]. As a result, structured photon states emerge as a promising candidate for achieving this, as they occupy higher-dimensional Hilbert spaces. Furthermore, it has been demonstrated that vector-matrix multiplication, a fundamental operation in quantum mechanics, can be performed using transverse optical fields [27].

There have been numerous impressive proposals and implementations of quantum computing algorithms that utilize the higher dimensional nature of the internal degrees of freedom of photons [28–30]. This field has developed tremendously, aided by the advent of multiplane light conversion technology [31], where digital holography and free-space diffraction have been demonstrated to realize high dimensional, $d = 5$, single photon unitary gate operations such as the high-dimensional pauli-X, hadamard-H and controlled not $C - NOT$ gates [32]. Inspired by these advances, diffraction-based deep neural network architectures have also been proposed [33]. Additionally, harnessing mode mixing in complex media such as multimodal optical fibers [34], as opposed to diffractive MPLCs (Multi-Plane Light Converters), has emerged as a potential avenue (see review on this topic [35]) to achieve similar operations but with discrete pixel-like states [34]. To realise the full potential of qudits for quantum computing, efficient single photon sources and the tools for entangling and controlling them are necessary.

On the other hand, harnessing classical optical fields for quantum computing has also been an active area of research [14, 36–40]. The drive in this field has been inspired by the observation that some of the essential resources of quantum computing can be found in classical coherent fields. Like quantum states, classical waves can be prepared in superpositions and can be interfered, allowing for parallel information processing. Notably, Jozsa also argued that classical waves can provide an efficient simulation platform since not all quantum algorithms require entanglement as a resource [41, 42]. Recent advances have demonstrated that optical methods can implement unitary and nonunitary transformations efficiently, utilising programmable holographic techniques and spatial phase modulation to enable high-dimensional operations [43]. Similarly, parallel processing for computational tasks like multiplication modulo, a crucial component of Shor's algorithm, using phase modulation has been explored, offering promising avenues for structured light applications in computational tasks [44]. This opens up new possibilities for studying or implementing quantum computational tasks without relying on complex quantum hardware, by merely exploiting coherent fields and performing the required matrix-vector operations on them.

An intriguing and unexplored approach to utilizing coherent fields in quantum computing is Tamura's optical matrix-vector multiplication from the 1970s [45] ,

which was long forgotten but has recently resurfaced. This method harnesses pointwise multiplication (Hadamard product) of light and the Fourier transform capabilities of lenses. With this approach, operations have been applied in dimensions reaching up to $d = 56$ [27]. Hidden in this method, is the capability to encode information in Hilbert spaces formed from transverse modes of coherent laser fields.

Building on these insights, the primary goal of this dissertation is to revisit and extend the concept of optical matrix multiplication, establishing it as a viable platform for simulating quantum computational tasks. I present a comprehensive framework that leverages structured light fields—not only to perform complex matrix operations fundamental to quantum algorithms but also to encode high-dimensional quantum information. In doing so, we bridge the gap between classical optical processing and quantum computation, opening new avenues in the search for scalable, hybrid computing architectures.

In the sections that follow, I first introduce the language of quantum computing, laying the theoretical groundwork necessary to appreciate how quantum operations are defined and executed. We then explore the various degrees of freedom of light—such as polarization, path, time and space—and discuss how these can be harnessed to encode and process information. By connecting these foundational concepts, we set the stage for the subsequent chapters, which detail the experimental and theoretical developments that underpin our optical matrix multiplication framework.

1.1 Quantum computing basics

This section introduces the fundamentals of quantum computing, including qubits, multi-qubit systems, and quantum gates. These concepts serve as the foundation for understanding quantum information processing and provide a point of comparison for later discussions on how structured light can enable similar operation. Quantum computing presents a new way of thinking about computation, harnessing the principles of quantum mechanics to perform tasks that would be traditionally infeasible for classical computers, such as Shor's factoring algorithm and Grover's search algorithm [46, 47]. The fundamental unit of information in quantum computing is the quantum bit, or qubit, which serves as a quantum analogue to the classical bit. In classical computing, a bit is a binary unit of information that exists in one of two

possible states: 0 or 1. These bits are manipulated by logic gates, and each bit is in a definite state at any given moment. In contrast, qubits can exist in a superposition of both $|0\rangle$ and $|1\rangle$. We start by introducing the mathematical framework for describing qubits, extend this to a multi-qubit system and then introduce the circuit model of quantum computations.

1.1.1 The quantum bit

Mathematically, the state of a qubit is described as a linear combination of the classical states $|0\rangle$ and $|1\rangle$ [48]:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \alpha,\beta \in \mathbb{C}, \tag{1.1}$$

where α and β are complex probability amplitudes, and the normalization condition,

$$|\alpha|^2 + |\beta|^2 = 1,$$

ensures that the total probability of measuring the qubit in one of the two basis states is always 1. Qubits are unit vectors in a two-dimensional Hilbert space, denoted $\mathcal{H}_2$ with the computational basis states $|0\rangle$ and $|1\rangle$ forming an orthonormal basis of $\mathcal{H}_2$: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ the state in eq 1.1 can be expressed in this space as vector

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \tag{1.2}$$

The superposition principle provides quantum computers with parallelism, enabling them to perform computations on multiple states simultaneously.

To visualise the qubit state, we use the Bloch sphere (see Fig. 1.1), a geometric model in which the states $|0\rangle$ and $|1\rangle$ lie at the poles, and any point on the surface represents a potential qubit state. The general state of a qubit on the Bloch sphere is parameterized by two angles, θ and ϕ, which define the qubit's state on the surface of the sphere:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle. \tag{1.3}$$

Here, $\theta \in [0,\pi]$ represents the angle between the qubit state and the $|0\rangle$ state (north pole), while $\phi \in [0,2\pi]$ defines the phase of the qubit around the z-axis of the sphere.

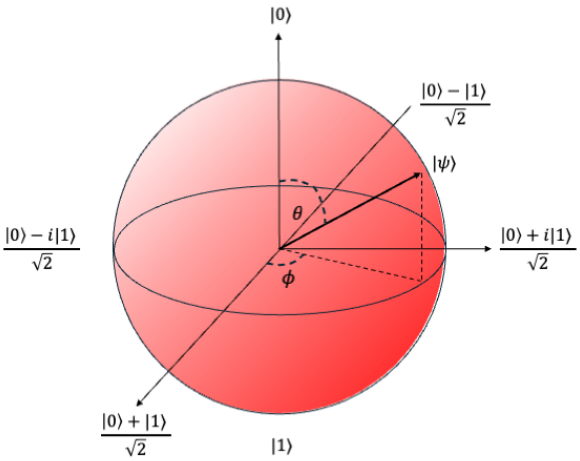


FIGURE 1.1: The Bloch sphere representation of a qubit. The states $|0\rangle$ and $|1\rangle$ are located at the north and south poles, respectively. Any point on the surface of the sphere corresponds to a possible qubit state, parameterized by angles θ and ϕ. The angle θ describes the position of the qubit state between the poles, while ϕ represents the phase around the z-axis, allowing for a full geometric visualization of the qubit's state.

This parameterization captures the full range of possible qubit states.

1.1.2 Multi-qubit systems

The power of quantum computing becomes apparent when we consider multiple qubits. With two classical bits, there are four possible states—00, 01, 10, and 11—but with each bit in a definite state. In quantum computing, two qubits exist in a superposition of all four possible states simultaneously. Mathematically, the state space for two qubits is the tensor product of two single-qubit Hilbert spaces $\mathcal{H}_2 \otimes \mathcal{H}_2$, and the general state of two qubits is given by:

$$|\psi\rangle = (\alpha_1 |0\rangle + \beta_1 |1\rangle) \otimes (\alpha_2 |0\rangle + \beta_2 |1\rangle) = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle, \tag{1.4}$$

with complex coefficients $\alpha_{00} = \alpha_1\alpha_2$, $\alpha_{01} = \alpha_1\beta_2$, $\alpha_{10} = \beta_1\alpha_2$, $\alpha_{11} = \beta_1\beta_2$, in vector notation the state can be expressed as

$$|\psi\rangle = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}, \tag{1.5}$$

highlighting an increase in elements defining the vector space, with the normalization condition

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1.$$

For a system of n qubits, the state space expands exponentially, as each qubit doubles the number of possible states. Specifically, while n classical bits can only represent one of 2^n possible states at a time, n qubits can exist in a superposition of all 2^n states simultaneously, for example a 2 and 3 qubit can be described by 4 and 8 dimensional vectors. Mathematically, the general state of an n -qubit system is given by:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle, \tag{1.6}$$

where each $|i\rangle$ represents one of the 2^n basis states and the coefficients $\alpha_i \in \mathbb{C}$ satisfy the normalization condition, $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$. This expression highlights the exponential growth in the number of states with each added qubit, providing quantum computers with state space that scales as 2^n.

Not all quantum states can be expressed as separable states. A separable (or product) state, such as in eq(1.4):

$$|\psi_{\text{separable}}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle,$$

represents a system in which each qubit's state is independent of the others. In this case, the overall state can be decomposed into individual qubit states. However, quantum states can be non-separable and cannot be written in this form; instead, they exhibit entanglement. Entangled states are characterized by non-local correlations that link the measurement outcomes of one qubit to those of another. For example, the following Bell state is a maximally entangled:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

This non-separable nature of entangled states is essential for quantum protocols like quantum teleportation and computational algorithms such as Shor's algorithm.

1.1.3 Quantum gates and the circuit model of computation

Bernstein and Vazirani formalised a model of quantum computation using a quantum Turing machine, demonstrating how information on a quantum tape could be processed with unitary transformations, allowing symbols and tape head positions to be in superposition [49]. Independently, Andrew Yao developed an alternative model, quantum circuits, which are computationally equivalent to quantum Turing machines but offer a more intuitive, diagrammatic approach that has become standard in quantum computation today [50] . In a quantum circuit (shown in Fig1.2 b)), computation begins with a system of N qubits, initialized either in a known state or as an input to the problem. Some qubits may represent ancillary states (initialized to $|0\rangle$), providing workspace. The computation then proceeds by applying a sequence of quantum gates, each represented by unitary matrices acting on one or two qubits, until the desired operation is performed. Finally, measurements are made to retrieve the outcome of the computation.

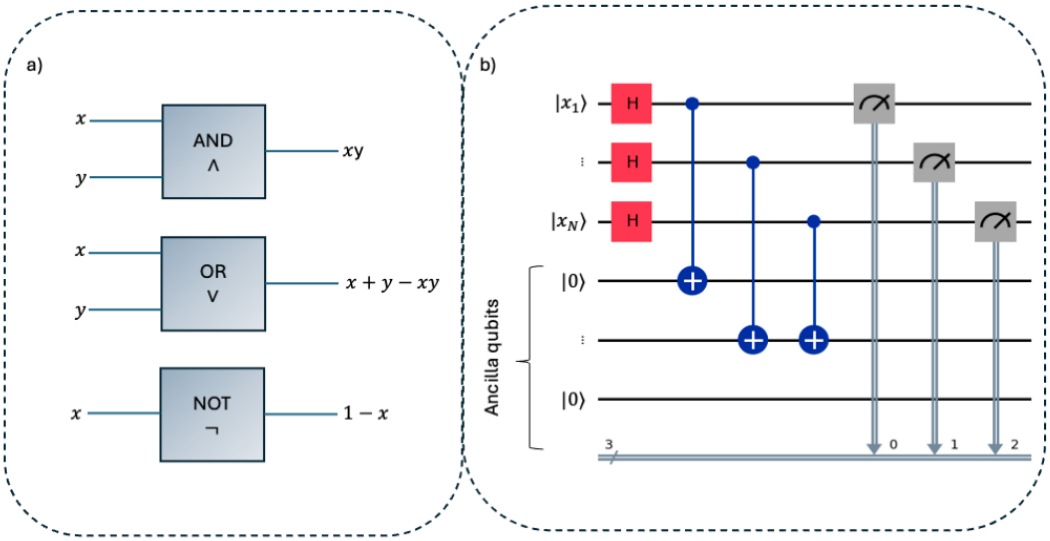


FIGURE 1.2: a) Logic gates forming compononents of the disjunctive normal form. This set is universal for classical logic circuits. b) A quantum circuit where computation starts with a system of N qubits, initialized in known states or as problem inputs$|x_i\rangle$. Ancillary qubits, set to $|0\rangle$, provide workspace. A sequence of quantum gates is applied, leading to the desired operations. Finally, measurements are taken to retrieve the computation's outcome.

Boolean functions and the classical circuit Model

Before discussing the quantum model, it's helpful to first consider classical computation. Computation, whether classical or quantum, is fundamentally about evaluating functions. In the classical setting, a Boolean function maps an n-bit input to an m-bit output:

$$f : \{0,1\}^n \to \{0,1\}^m,$$

a function mapping a 2-bit input to 1-bit output is illustrated in Fig (1.2 a)). For simplicity, an m-bit function can be viewed as m independent single-bit Boolean functions. Thus, without loss of generality, we consider the Boolean function $f : \{0,1\}^n \to \{0,1\}$, where the output is a single bit. This separates n-bit strings into two sets [51]:

- Accepted inputs: where $f(x) = 1$
- Rejected inputs: where $f(x) = 0$

In the classical model of computation, circuits are used to compute these Boolean functions. A classical circuit is built from logic gates, and any Boolean function can be implemented using a universal gate set such as $\{\text{AND}, \text{OR}, \text{NOT}\}$ [52] or even the single gate NAND [53], which is functionally complete. One systematic way to represent any Boolean function is through Disjunctive Normal Form (DNF) [54]. The DNF is a normal form that expresses the function as a disjunction (OR gate $\vee$) of conjunctions (AND gates $\wedge$) of literals. Each literal is either a variable x_k or its negation $\neg x_k$ for some $k \in \{1,2,\ldots,n\}$. The DNF representation of a Boolean function $f(x_1, x_2, \ldots, x_n)$ is given by [54]:

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{i=1}^{k} \left(\bigwedge_{j=1}^{m_i} \ell_{ij} \right)$$

where ℓ_{ij} represents a literal (either x_k or $\neg x_k$ for some k), and the disjunction (OR) is over the k possible conjunctions (terms). This means the Boolean function is written as an OR of multiple AND clauses (terms). For example, consider the Boolean function $f(x_1, x_2, x_3)$ defined as:

$$f(x_1, x_2, x_3) = (x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_3)$$

This function is in DNF, where the disjunction (OR) is between two conjunctions (ANDs): $x_1 \wedge \neg x_2$ and $\neg x_1 \wedge x_3$. Any Boolean function can be converted into this form, though the resulting expression may grow exponentially in size for complex functions.

To further illustrate the universality of this representation, consider an interesting example: the majority function for three variables [55]. This function outputs 1 if at least two of the three inputs are 1, and 0 otherwise. In other words, $f(x_1, x_2, x_3) = 1$ when the inputs are

- $(1,1,0)$,
- $(1,0,1)$,
- $(0,1,1)$, or
- $(1,1,1)$.

For each of these cases, we can form an AND clause that is true only under that specific assignment:

Input Combination	**Conjunction Term**
$(1,1,0)$	$x_1 \wedge x_2 \wedge \neg x_3$
$(1,0,1)$	$x_1 \wedge \neg x_2 \wedge x_3$
$(0,1,1)$	$\neg x_1 \wedge x_2 \wedge x_3$
$(1,1,1)$	$x_1 \wedge x_2 \wedge x_3$

TABLE 1.1: Conjunction terms for the majority function in DNF.

Taking the OR of these terms, the DNF expression for the majority function is

$$f(x_1, x_2, x_3) = (x_1 \wedge x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge x_3) \vee (\neg x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_3).$$

Noting that the case where $x_1 = x_2 = x_3 = 1$ is already covered by the pairwise terms, this expression can be simplified to

$$f(x_1, x_2, x_3) = (x_1 \wedge x_2) \vee (x_1 \wedge x_3) \vee (x_2 \wedge x_3).$$

This example demonstrates that even a nontrivial Boolean function like majority voting can be expressed entirely using AND, OR, and NOT boolean algebra. In seminal work by Shannon these kind of operations were proved to be realisable via electrical swicthing circuits [56].

However, classical gates often discard information (e.g., AND gates lose input parity, two input bits are mapped to one), making them irreversible. To transition to quantum computation, reversibility becomes essential. Classical reversible gates, like the Toffoli gate ($\Lambda^2(X)$), preserve input information by ensuring bijectivity: every output uniquely determines its input [57]. For example, the Toffoli gate performs the transformation:

$$(x,y,z) \mapsto (x,y,z \oplus (x \wedge y)),$$

enabling universal reversible computation. Unlike an AND gate, which outputs a single bit and loses information about its inputs, the Toffoli gate retains all input bits while computing a controlled AND operation on the target bit. This ensures that the transformation is bijective, meaning each unique input maps to a unique output.

To illustrate this, consider the following mappings:

- $(0,0,0) \mapsto (0,0,0)$,
- $(0,0,1) \mapsto (0,0,1)$,
- $(0,1,0) \mapsto (0,1,0)$,
- $(1,1,0) \mapsto (1,1,1)$,
- $(1,1,1) \mapsto (1,1,0)$.

Since no two inputs produce the same output, the function is fully reversible. This reversibility mirrors the unitarity requirement in quantum mechanics, where operations must be invertible (i.e., $UU^\dagger = I$). Thus, reversible computation serves as the conceptual bridge between classical Boolean logic and quantum mechanics.

Quantum gates and universality

Building on classical reversible computation, quantum computation extends the concept of universality by leveraging unitary transformations—operations that preserve the norm of quantum states. Just as classical reversible gates like the Toffoli gate ($\Lambda^2(X)$) ensure bijectivity (every output maps uniquely to an input), quantum gates must satisfy unitarity ($U^\dagger U = I$), ensuring invertibility and compatibility with quantum mechanics: preserving the probabilistic interpretation of quantum states. Key gates include:

- **Pauli Gates (X, Y, Z)**: Fundamental single-qubit gates representing quantum analogs of classical bit-flip and phase-flip operations.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad X|0\rangle = |1\rangle, \quad X|1\rangle = |0\rangle.$$
$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Y|0\rangle = i|1\rangle, \quad Y|1\rangle = -i|0\rangle.$$
$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle.$$

- **Hadamard Gate (H)**: Creates superposition by mapping computational basis states to an equal-weighted superposition:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

- **Phase Gate (S)**: Introduces a relative phase to the $|1\rangle$ state:

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad S|0\rangle = |0\rangle, \quad S|1\rangle = i|1\rangle.$$

- **CNOT Gate**: A two-qubit entangling gate that flips the target qubit if the control qubit is $|1\rangle$:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

  Applied to basis states:

$$\text{CNOT}|00\rangle = |00\rangle, \quad \text{CNOT}|10\rangle = |11\rangle.$$

- **Toffoli Gate ($\Lambda^2(X)$)**: A three-qubit controlled gate that flips the third qubit if both the first and second qubits are $|1\rangle$. This generalizes the classical AND/-NAND operations and preserves reversibility:

$$\Lambda^2(X)|x,y,z\rangle = |x,y,z \oplus (x \wedge y)\rangle.$$

Its matrix representation is:

$$\text{Toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

The Toffoli gate is a universal classical reversible gate and can be decomposed using CNOT and T gates, making it crucial in both classical reversible circuits and quantum universality.

A quantum gate set is universal if it can approximate any unitary operation $U \in SU(2^n)$ (special unitary of degree(2^n)). This ensures quantum computers can execute any quantum algorithm, mirroring classical universality (e.g., NAND gates) from a finite set of local gates. Two forms of universality can be distuingished [48]:

Exact universality: ‘

A gate set is exactly universal if it constructs any unitary exactly. This requires:

1. Arbitrary single-qubit rotations (e.g., H, S, $R_z(\theta)$).
2. One entangling two-qubit gate (e.g., CNOT).

An example: The Toffoli gate ($\Lambda^2(X)$) decomposition into CNOT and T-gates. However, exact universality assumes perfect implementation of arbitrary continuous rotations (e.g., $R_z(\theta)$ for any real θ). In physical quantum hardware, operations are restricted to a finite gate set, and real-world constraints such as noise and control precision prevent exact implementations of arbitrary unitaries. Instead, an approximate universal set is used, where a finite number of gates can approximate any unitary to arbitrary precision.

Approximate universality: A finite gate set is approximately universal if it can approximate any unitary to arbitrary precision using a sequence of gates. One commonly used approximately universal set is the Clifford+T set:

$$\{H, S, \text{CNOT}, T\}.$$

The Clifford group consists of gates that preserve the Pauli operators under conjugation and includes the Hadamard (H), phase (S), and controlled-NOT (CNOT) gates. However, Clifford gates alone cannot generate arbitrary unitaries. The addition of the non-Clifford T-gate,

$$T = e^{-i\pi Z/8} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix},$$

breaks this limitation, allowing access to dense coverage of $SU(2^n)$ and making the set approximately universal [48].

To quantify how efficiently a gate set approximates an arbitrary unitary, we define circuit depth, the number of sequential gate layers required to implement a desired transformation. The Solovay-Kitaev theorem ensures that any unitary matrix can be approximated using Clifford+T gates with circuit depth scaling as: $O\left(\log^{3.97}(1/\epsilon)\right)$, where ϵ is the approximation error [58]. This logarithmic dependence guarantees that only a modest increase in gate count is required for higher precision.

While quantum computing traditionally relies on qubits implemented via two-level quantum systems, the ability to encode and process information in higher-dimensional Hilbert spaces ($d > 2$) offers several advantages. High-dimensional quantum systems, or qudits, generalize qubits by utilizing states labeled as $|0\rangle, |1\rangle, \ldots, |d-1\rangle$, where $d > 2$. These systems provide increased information density, as a single qudit can store more information than a qubit, enhancing computational encoding efficiency [35]. They also offer improved resilience to noise [59, 60], with certain high-dimensional encodings being more robust to errors than standard qubits, which benefits quantum communication and error correction. In classical and quantum optics, structured light fields provide a natural platform for high-dimensional encoding [61], leveraging degrees of freedom beyond polarization, such as spatial modes, where Laguerre-Gaussian (LG) and Hermite-Gaussian (HG) modes allow

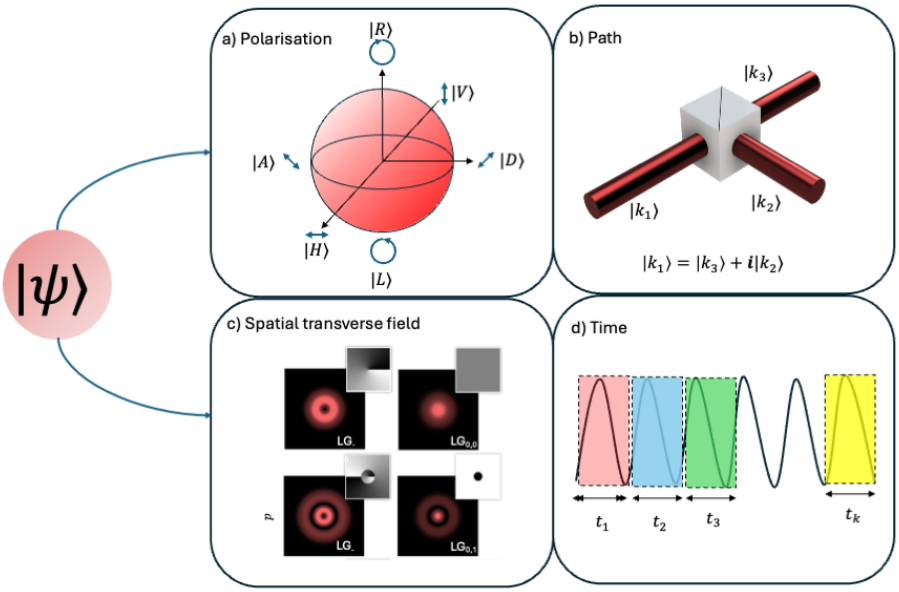


FIGURE 1.3: Light's different degrees of freedom. In a) we highlight the two-level polarisation on poincare sphere and how other orthogonal basis can be constructed from a superposition of fundamental eigenstates $|R\rangle$ and $|L\rangle$. b) Path DOF is illustrated by an optical mode entering a beam splitter and then split into two spatial paths. c) The LG mode basis which is orthonormal and complete is chosen to demonstrate how information can be encoded in the spatial DOF. d) We illustrate discretisation in the time-domain with information being encoded into time bins t_k

for information encoding in the spatial transverse field, and temporal modes, where time-bin encoding exploits time-separated photon pulses for qudit-based quantum communication. The following sections explore how information can be encoded and manipulated across these different degrees of freedom, forming the foundation for optical information processing and high-dimensional quantum computing.

1.2 Degrees of freedom of light for information encoding and manipulation

Light possesses multiple internal degrees of freedom that can be harnessed for encoding and manipulating information. A single photon propagating through free space can be described as an excitation of the electromagnetic field traveling at the speed of light. While photons exhibit particle-like behavior, they also display wave-like characteristics, with their probability amplitude—or wavefunction—distributed across space and time. This wavefunction incorporates several internal degrees of freedom, including polarization, path, spatial distribution and temporal structure (Fig 1.3).

A photon's quantum state can be described in terms of its internal degrees of freedom as follows [62]:

$$|\Psi\rangle = \int\int \Phi_{\text{spatial}}(r)\,\Omega_{\text{time}}(t)\,|\zeta_{\text{pol}}\rangle\,|r\rangle\,|t\rangle\,dt\,dr.$$

Here, $\Phi_{\text{spatial}}(r)$ represents the spatial probability amplitude of the photon, while $\Omega_{\text{time}}(t)$ denotes the temporal profile. The term $|\zeta_{\text{pol}}\rangle$ describes the photon's polarization state, and $|r\rangle$, $|t\rangle$ define the spatial and temporal basis, respectively. The temporal component is linked to the frequency domain through the Fourier transform:

$$\tilde{\Omega}(\omega) \propto \int \Omega_{\text{time}}(t) e^{-i\omega t}\,dt.$$

By engineering these internal degrees of freedom, it is possible to create structured light, which plays a central role in high-dimensional information encoding.

1.2.1 Polarization as a two-level encoding system

Polarization is a widely used degree of freedom for encoding quantum information because it is easy to generate, control, and manipulate with standard optical elements. In early quantum experiments—including Bell inequality tests [63], quantum key distribution [64], teleportation [65], and superdense coding [66]—polarization played a central role due to its experimental accessibility.

At the most basic level, polarization can be used to encode a qubit, a two-level quantum system. One natural choice is to use the two fundamental polarization eigenstates (Fig 1.3 a):

$$|0\rangle \equiv |R\rangle\,, \quad |1\rangle \equiv |L\rangle\,,$$

where $|R\rangle$ and $|L\rangle$ denote right- and left-circular polarization, respectively. Alternative basis can be constructed from superpositions of these states. For example, the horizontal and vertical polarization states are defined as:

$$|H\rangle = \frac{1}{\sqrt{2}}\left(|R\rangle + |L\rangle\right), \quad |V\rangle = \frac{1}{\sqrt{2}}\left(|R\rangle - |L\rangle\right).$$

Similarly, the diagonal and anti-diagonal states are given by:

$$|D\rangle = \frac{1}{\sqrt{2}}\left(|R\rangle + i\,|L\rangle\right), \quad |A\rangle = \frac{1}{\sqrt{2}}\left(|R\rangle - i\,|L\rangle\right).$$

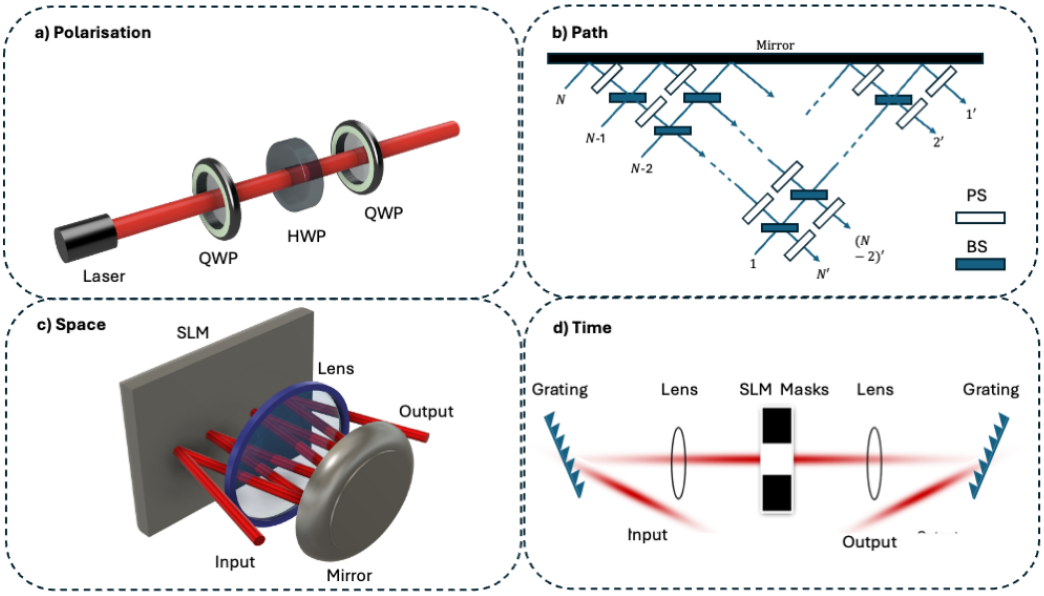


FIGURE 1.4: Manipulation of information in various DOFs. a) Polarisation: An experimental consisting of QWP and HWP capable of generating any $SU(2)$ transformation in polarisation encoding is illustrated. b) Path: an n-port interferometer is shown as a scheme for universal $SU(d)$ transformations in path domain, the setup is built from beams splitters (BS) and phase shiftters (PS). c) Spatial Tranverse Field: an MPLC consisting of multiple phase modulation encoded on the SLM and segments of phree space propagation between each bounce. d) Time domain: a setup for manipulating information time domain

Although polarization qubits are simple to manipulate, they are inherently limited to a two-dimensional Hilbert space. For encoding information in higher-dimensional systems ($d > 2$), alternative degrees of freedom must be explored.

Jones calculus and SU(2) transformations

In the framework of Jones calculus, the polarization state of light is represented by a two-component complex vector, and its evolution under optical transformations is described by SU(2) unitary matrices. A general polarization qubit state can be written as:

$$|\psi\rangle = \alpha\,|R\rangle + \beta\,|L\rangle\,, \tag{1.7}$$

here α, $\beta \in \mathbb{C}$ are the complex amplitudes. Optical elements, such as wave plates, introduce controlled phase delays between these components, enabling precise manipulation of the polarization state. Two common optical elements used for polarization control are the quarter-wave plate (QWP) and the half-wave plate (HWP) (Fig 1.4 A).

Quarter-Wave Plate (QWP)

A QWP introduces a $\pi/2$ phase difference between two orthogonal polarization components. Assuming its fast axis is aligned with the horizontal axis, its Jones matrix is given by:

$$\text{QWP} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}. \tag{1.8}$$

Half-Wave Plate (HWP)

A HWP introduces a π phase shift, effectively rotating the polarization axis. Its Jones matrix is:

$$\text{HWP} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{1.9}$$

For a wave plate oriented at an arbitrary angle θ, the transformation is expressed as:

$$M(\theta) = R(-\theta)\, M\, R(\theta), \tag{1.10}$$

where M is the Jones matrix of the wave plate in its principal axis, and $R(\theta)$ is the rotation matrix defined by:

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}. \tag{1.11}$$

Thus, the generalized forms for the half-wave plate and quarter-wave plate are:

$$\text{HWP}(\theta) = R(-\theta) \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} R(\theta) = \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix}, \tag{1.12}$$

$$\text{QWP}(\theta) = R(-\theta) \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} R(\theta) = \begin{bmatrix} \cos^2\theta + i\sin^2\theta & (1-i)\sin\theta\cos\theta \\ (1-i)\sin\theta\cos\theta & \sin^2\theta + i\cos^2\theta \end{bmatrix}. \tag{1.13}$$

These expressions are fully analogous to rotation gates in quantum computing. Just as single-qubit rotation gates act on the Bloch sphere to change the state's phase and orientation, the generalized QWP and HWP rotate the polarization state (or qubit) on the Poincaré sphere.

Minimal SU(2) setup for polarization transformations

Simon and Mukunda demonstrated that an optimized three-component setup, consisting of one half-wave plate and two quarter-wave plates, is sufficient to realize all possible SU(2) transformations on polarization states [67]. Their method requires only three wave plates arranged in the following sequence:

$$U = \mathrm{QWP}(\theta_1)\,\mathrm{HWP}(\theta_2)\,\mathrm{QWP}(\theta_3). \tag{1.14}$$

By adjusting the wave plate angles θ_1, θ_2, and θ_3, this compact setup can implement any desired polarization (universal single qubit) transformation. This highlights the efficiency and elegance of polarization control in optical systems.

1.2.2 Path encoding

Path encoding is based on the physical separation of optical modes into different spatial paths (Fig 1.3 b). This degree of freedom is particularly useful in quantum optics and interferometry, where different optical paths can be coherently manipulated to encode and process quantum information. Mathematically, path modes correspond to the Fourier transform of the spatial field distribution.

A light field propagating along the z-axis with transverse coordinates (x,y) can be represented by its field distribution $E(x,y,z)$. In the paraxial approximation, the Fourier transform of this field relates the spatial and angular components:

$$E(k_x,k_y) = \int\int E(x,y,0)e^{-i(k_x x+k_y y)}dxdy, \tag{1.15}$$

where (k_x,k_y) are the transverse wave vectors. This representation highlights how spatially distinct paths can be mapped onto different wave vector components, which are then manipulated using optical elements.

A significant realization of path information control is through n-port interferometers (Fig 1.4 B), which allow arbitrary unitary transformations of optical paths. As demonstrated by Reck et al. [68], any finite-dimensional unitary transformation can be implemented using a sequence of beam splitters and phase shifters arranged in a multiport interferometer. The fundamental building block is the two-mode beam

splitter, whose action is described by the unitary matrix:

$$B(\theta,\phi) = \begin{bmatrix} \cos\theta & e^{i\phi}\sin\theta \\ e^{-i\phi}\sin\theta & -\cos\theta \end{bmatrix}. \tag{1.16}$$

Using a combination of these beam splitters along with phase shifts, one can construct any desired unitary operation on a set of optical paths.

To decompose an arbitrary $N \times N$ unitary matrix U, we factorize it into a sequence of 2×2 transformations acting on pairs of modes. This is achieved by systematically applying beam splitters and phase shifters to eliminate off-diagonal elements, similar to Gaussian elimination:

$$U = T_{N,N-1} T_{N,N-2} \cdots T_{2,1} D, \tag{1.17}$$

where each $T_{i,j}$ is a two-mode transformation realized by a beam splitter, and D is a diagonal phase shift matrix. A recursive approach is used, where each step reduces the problem to a smaller unitary matrix until full decomposition is achieved. To illustrate this process explicitly, consider a 3×3 unitary matrix U [69]. In this case, the decomposition reads

$$U = T_{3,2}\, T_{3,1}\, T_{2,1}\, D,$$

with the individual matrices defined as follows:

$$T_{2,1} = \begin{pmatrix} \cos\theta_{21} & -e^{i\phi_{21}}\sin\theta_{21} & 0 \\ e^{-i\phi_{21}}\sin\theta_{21} & \cos\theta_{21} & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$T_{3,1} = \begin{pmatrix} \cos\theta_{31} & 0 & -e^{i\phi_{31}}\sin\theta_{31} \\ 0 & 1 & 0 \\ e^{-i\phi_{31}}\sin\theta_{31} & 0 & \cos\theta_{31} \end{pmatrix},$$

$$T_{3,2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{32} & -e^{i\phi_{32}}\sin\theta_{32} \\ 0 & e^{-i\phi_{32}}\sin\theta_{32} & \cos\theta_{32} \end{pmatrix},$$

and

$$D = \begin{pmatrix} e^{i\alpha_1} & 0 & 0 \\ 0 & e^{i\alpha_2} & 0 \\ 0 & 0 & e^{i\alpha_3} \end{pmatrix}.$$

In this scheme, $T_{2,1}$ first acts on modes 1 and 2 to eliminate one off-diagonal element; $T_{3,1}$ then mixes modes 1 and 3 to further zero an element in the first column; and finally, $T_{3,2}$ cleans up the remaining off-diagonal term by acting on modes 2 and 3. The diagonal matrix D adjusts the phases, ensuring that the final product is equivalent to the original unitary matrix U. This concrete 3×3 example mirrors the general approach for decomposing an arbitrary $N \times N$ unitary: by cascading simple two-mode operations and phase shifts, the full unitary transformation is built up recursively, analogous to the process of Gaussian elimination.

In the work of Cerf [36], which claimed the first practical simulation of quantum logic, this scheme was used to implement various gates and protocols. It now serves as a foundation for integrated photonics, where interferometers enable implementation of quantum circuits on photonic chips.

1.2.3 Continous temporal and spatial

Discrete basis for high-dimensional quantum encoding

Unlike polarization, which is inherently discrete, the temporal and spatial degrees of freedom of light exist in a continuous domain. To harness this continuous degree of freedom for high-dimensional quantum encoding, it is necessary to define a discrete basis within the continuous space. A valid discrete basis must satisfy two essential properties:

The basis states must be mutually orthogonal:

$$\langle i|j\rangle = \delta_{ij}, \quad \forall\, |i\rangle, |j\rangle \in B_d, \tag{1.18}$$

and must span the entire Hilbert space (i.e complete) so that any quantum state can be expressed as a superposition:

$$\sum_{j=0}^{d-1} |j\rangle\langle j| = I. \tag{1.19}$$

Given such a basis, any high-dimensional quantum state can be written as:

$$|\psi\rangle = \sum_{j=0}^{d-1} a_j |j\rangle , \tag{1.20}$$

with complex coefficients a_j satisfying the normalization condition:

$$\sum_{j=0}^{d-1} |a_j|^2 = 1. \tag{1.21}$$

Temporal degrees of freedom: encoding in time bins

Time-bin encoding exploits the temporal degree of freedom by discretizing the continuous temporal profile of a photon into distinct intervals or "time bins" (Fig 1.3 d). This encoding scheme is particularly useful in fiber-based quantum communication and high-dimensional quantum computing, where different time bins can be coherently manipulated to implement quantum gates and protocols. In the following sections, we describe both the basic formalism of time-bin encoding and several optical techniques for performing unitary transformations on time-bin states.

The temporal degree of freedom can be divided into distinct intervals. The photon's temporal wavefunction, $\Omega(t)$, can be partitioned into discrete modes corresponding to time bins, enabling a time-based encoding scheme. Mathematically, the continuous representation is given by:

$$|\Omega\rangle = \int_{-\infty}^{\infty} \Omega(t) |t\rangle \, dt,$$

which can be approximated by a discrete sum:

$$|\Omega\rangle \quad \longrightarrow \quad \sum_k b_k |t_k\rangle ,$$

where $|t_k\rangle$ represents the basis state corresponding to the k-th time bin, and b_k are the associated complex amplitudes. By partitioning the time domain into N intervals, one obtains a set of time-bin modes for encoding quantum information.

In its simplest form, time-bin encoding utilizes two distinct time bins to define a qubit. A general time-bin qubit state is given by (Fig 1.3d):

$$|\psi\rangle = \alpha |t_1\rangle + \beta |t_2\rangle , \tag{1.22}$$

where $|t_1\rangle$ and $|t_2\rangle$ represent two distinct time bins and $\alpha, \beta \in \mathbb{C}$ are the probability amplitudes. This idea naturally extends to higher dimensions:

$$|\psi\rangle = \sum_{j=1}^{d} \alpha_j |t_j\rangle, \tag{1.23}$$

where d is the number of time bins used for encoding, enabling high-dimensional quantum information processing.

Optical implementations for time-bin transformations

Several optical methods have been developed to perform unitary transformations on time-bin states. These include Fourier-domain pulse shaping. This form of pulse shaping allows for arbitrary transformations of time-bin states by modulating their spectral components. A typical setup involves a 4-f optical pulse shaper [70], which comprises a diffraction grating, a spatial light modulator (SLM), and a second grating to reassemble the modified spectrum (Fig 1.4d). The procedure is as follows:

1. The input temporal field $E_{\text{in}}(t)$ is Fourier-transformed to obtain the spectral field:
$$\tilde{E}(\omega) = \mathcal{F}\{E_{\text{in}}(t)\}. \tag{1.24}$$

2. A programmable transfer function $T(\omega)$ is applied in the frequency domain:
$$\tilde{E}'(\omega) = T(\omega)\tilde{E}(\omega). \tag{1.25}$$

3. The modified spectrum is then inverse Fourier-transformed back into the time domain:
$$E_{\text{out}}(t) = \mathcal{F}^{-1}\{\tilde{E}'(\omega)\}. \tag{1.26}$$

By carefully designing $T(\omega)$, one can implement controlled phase shifts, sort time bins, and perform arbitrary quantum gates on time-bin qubits and qudits.

Spatial basis

A straightforward approach to defining a discrete spatial basis is to divide the transverse plane into non-overlapping regions—essentially "pixelating" the photon's wavefunction (Fig 1.5a) [62]. In this representation, each discrete mode corresponds to a

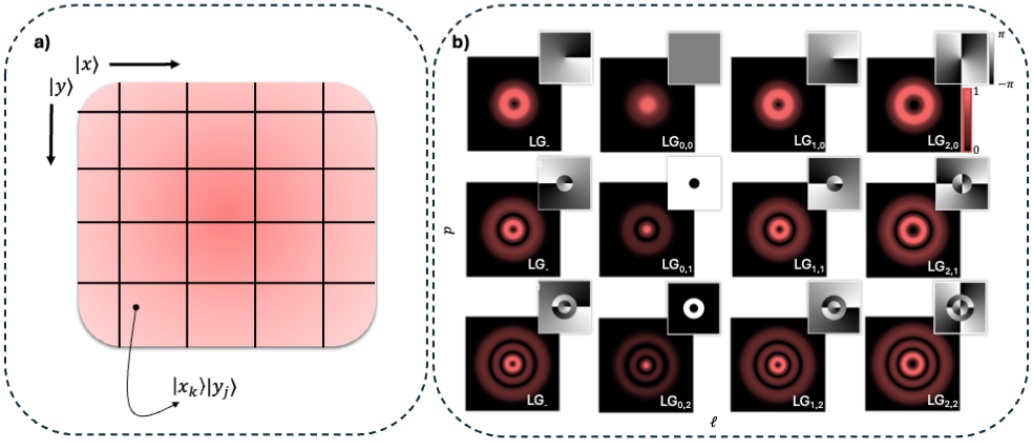


FIGURE 1.5: Discrete basis in the spatial transverse profile. a) Pixel basis: a straightforwad scheme for encoding information by discretising the tranverse field into pixels. Non overlapping pixels are orthogonal. b) illustrated is family of orthonal Laguerre-Gaussian (LG) modes which also form a basis for the transverse field

localized spatial region:

$$|\Phi\rangle = \int_{\mathbb{R}^2} \Phi(\mathbf{r})\,|\mathbf{r}\rangle\; d^2\mathbf{r} \quad \rightarrow \quad \sum_{k=0}^{N-1} a_k\,|\mathbf{r}_k\rangle\,. \tag{1.27}$$

However, this pixel-based approach is highly sensitive to the optical system's resolution and diffraction effects. A more robust alternative is to employ structured light modes that are propagation invariant. A widely useful spatial basis for high-dimensional encoding is the orbital angular momentum (OAM) basis (Fig 1.5b) , OAM eigenstates are characterized by an azimuthal phase dependence :

$$\Phi(r,\phi,z) = R(r)e^{il\phi},$$

where l is the topological charge and $R(r)$ describes the radial intensity distribution. Each OAM state carries a well-defined orbital angular momentum of $l\hbar$ per photon. The corresponding quantum state can be expressed as:

$$\left|LG_{l,p}\right\rangle = \int_{\mathbb{R}^2} LG_{l,p}(r)|r\rangle d^2r.$$

Here, p represents the radial mode number, and Laguerre-Gaussian (LG) modes form a natural orthonormal basis for spatial encoding. OAM modes provide several advantages: they are robust against diffraction, can be efficiently detected using phase-only techniques, and form a natural Schmidt basis for entangled states generated through SPDC (Spontaneous Parametric Down-Conversion).

Controlling the transverse field with Multi-Plane Light Conversion (MPLC)

Advances in spatial light modulator (SLM) technology have enabled precise control over the optical transverse field, paving the way for sophisticated high-dimensional quantum operations. Multi-plane light conversion (MPLC) is one such versatile approach that implements high-dimensional unitary transformations by using controlled phase manipulation and mode mixing (Fig1.4C).

MPLC systems consist of sequential phase modulation planes separated by segments of free-space propagation. In each stage, both phase modulation and diffraction contribute to mode mixing, gradually reshaping the transverse spatial profile of the light. In this way, MPLC acts similarly to multiport devices but operates on the spatial field rather than on discrete optical channels.

Each MPLC plane imparts a spatially varying phase to the incoming field. If we denote the phase modulation matrix for the m-th plane as P_m and the free-space propagation between planes as H, then for a system with M phase planes the overall unitary transformation U applied to an input field E_{in} can be written as [71]:

$$U = P_m H P_{m-1} \cdots H P_2 H P_1, \tag{1.28}$$

so that the output field is given by:

$$E_{\text{out}(x,y)} = U E_{\text{in}(x,y)}. \tag{1.29}$$

Achieving a target unitary transformation requires precise control over the phase at each plane. Typically, the phase profiles are optimized using methods such as wavefront matching (WFM) [72, 73]. In WFM, the input optical field $E_{\text{in}}(x,y)$ is propagated forward through the system to obtain the field distribution $E_m(x,y)$ at the m-th plane. Concurrently, the target output field $E_{\text{out}}(x,y)$ is back-propagated to yield the distribution $E'_m(x,y)$ immediately after the m-th plane. The optimal phase

profile for the m-th plane is then determined by [74]:

$$\Phi_m(x,y) = \arg\left[E'_m(x,y)E^*_m(x,y)\right], \tag{1.30}$$

where arg extracts the phase angle and $E^*_m(x,y)$ is the complex conjugate of $E_m(x,y)$. For multiple mode pairs, the phase for each plane is computed as a weighted sum over the corresponding phase differences:

$$\Phi_m(x,y) = \arg\left[\sum_j E'_{m,j}(x,y)E^*_{m,j}(x,y)\right], \tag{1.31}$$

with j indexing each set of input-output mode pairs. Through iterative forward and backward propagation, the algorithm converges to the phase profiles needed to realize the desired unitary transformation.

MPLC has been successfully employed in various high-dimensional quantum operations. For example, studies have demonstrated that using only a few phase modulation planes, MPLC can implement high-fidelity operations such as Pauli X-gates and Hadamard gates, achieving over 90% visibility for cyclic transformations and near-perfect process purities of around 99% [32]. This illustrates MPLC's effectiveness in controlling the spatial modes of light for quantum information processing.

Beyond quantum gate operations, MPLC is also capable of performing optical matrix multiplication—a subset of the linear transformations crucial for optical computing [72]. By carefully designing the sequence of phase modulation planes and propagation distances, MPLC can map an input vector of spatial modes to an output vector, thereby executing matrix multiplications in a fully optical manner.

The transition from continuous spatial degrees of freedom to a discrete basis is fundamental for high-dimensional quantum encoding. While a simple pixel-based discretization can be sensitive to system imperfections, employing propagation-invariant structured light modes offers a more robust approach. MPLC provides an effective method for realizing such structured modes, as it allows for precise control over the transverse field. By implementing tailored unitary transformations, MPLC can convert an arbitrary input field into a set of well-defined spatial modes that form

a discrete basis satisfying orthogonality and completeness. This integration of discrete spatial encoding with MPLC opens up new avenues for high-fidelity, high-dimensional quantum information processing and optical computing.

1.3 Dissertation outline

This work focuses on the development and characterization of optical circuits that use structured light to perform computationally relevant tasks—most notably, matrix-vector multiplication, which is central to implementing unitary operations in quantum circuits. By using spatial light modulators (SLMs) and lenses, I demonstrate that classical optical systems can simulate aspects of quantum computation. This not only bridges the gap between classical and quantum processing but also suggests new hybrid architectures that may ease the transition to scalable quantum technologies. The rest of the thesis is organised as follows:

- **Chapter 2: Experimental Techniques and Optical Toolbox**
  In Chapter 2 I detail the experimental techniques employed in this work. We described the design and operation of spatial light modulators (SLMs), the spatial Fourier-transforming property of lenses, and hologram construction methods to implement vector-matrix multiplication. This chapter concludes by introducing a practical alignment characterization technique based on modal decomposition thereby validating the experimental approach.

- **Chapter 3: Learning to solve a Useful Problem on a Quantum Computer: Reconstructing Quantum States**
  In chapter 3, we learn to apply the foundations of quantum computing introduced in this chapter 1 to solve a system of linear equations essential in quantum state tomography on a real quantum processor. In this chapter we aim to reconstruct quantum states produced from a source entangled photons, SPDC. The problem is formulated as a least sqaure optimisation problem, and variational quantum algorithms are utilized find the solution from experimental data.

- **Chapter 4: Emulating Quantum Computation with Optical Matrix Multiplication**

In chapter 4, we integrate the theoretical and experimental insights to demonstrate how optical matrix multiplication can be used to emulate quantum computing operations. We provide both a theoretical formulation and an experimental realization of quantum gate operations—specifically the Hadamard gate—and illustrate its application through the Deutsch-Jozsa algorithm. This chapter underscores the potential of OVMM (Optical Vector Matrix Multiplication) as a platform for simulating quantum processes using classical optical systems.

- **Chapter 5: Conclusion and future work**

  In chapter 5 we conclude by discussing future work and potential advancements of the work. We highlight three main areas that could be improved on universality, scalabilty and encoding efficiency.

Chapter 2

Experimental Techniques: Toolbox for optical matrix multiplication

Building on the quantum computing foundations introduced earlier, this chapter presents a versatile toolbox for optical matrix multiplication—our main tool for implementing optical gate operations. I begin by outlining the operational principles of spatial light modulators (SLMs) and discussing how various encoding schemes can be employed to manipulate light fields. Next, we examine the Fourier transforming properties of lenses, which play a crucial role in beam shaping and signal processing. By combining these optical tools, we demonstrate the assembly of an optical setup capable of computing with light patterns. Finally, we introduce an alignment verification procedure based on modal decomposition, ensuring the accurate generation and control of the desired light modes.

2.1 Spatial Light Modulators: Principles and Applications

2.1.1 Overview of SLM Operation

Spatial Light Modulators (SLMs) are widely used devices for manipulating light fields. Their primary function is to modulate the spatial phase, and in some cases, the amplitude of an incident electromagnetic field, enabling control over the light's spatial profile. In optical experiments, SLMs play a vital role in generating structured light fields, beam shaping and executing complex optical transformations [75]. SLMs typically consist of a pixelated array of liquid crystals (LCs), where each pixel can independently modulate the phase of the incident light (Fig 2.1). The phase shift

$\Delta\phi$ imparted by an SLM is determined by the relationship [76]:

$$\Delta\phi = \frac{2\pi}{\lambda}\Delta n \cdot d, \tag{2.1}$$

where:

- λ is the wavelength of the light,
- Δn is the change in the refractive index due to the reorientation of the liquid crystal molecules,
- d is the thickness of the liquid crystal layer.

Applying a voltage across the liquid crystal array changes the orientation of the molecules, altering the extraordinary refractive index. The modulation of the refractive index with respect to the reorientation angle θs given by:

$$n^2(\theta) = n_o^2 + (n_e^2 - n_o^2)\cos^2(\theta), \tag{2.2}$$

where n_o and n_e are the refractive indices along the ordinary and extraordinary axis of the liquid crystal. By controlling the voltage at each pixel, the phase of the light can be varied across the beam profile. SLMs typically modulate phase-only, which suits many optical applications, including beam shaping and wavefront manipulation in quantum optics and holography.

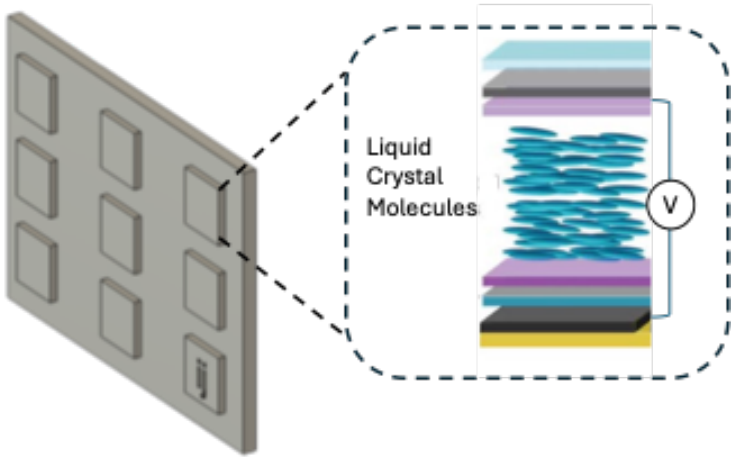


FIGURE 2.1: A simplified diagram of the SLM, showing a pixelated array of liquid crystals. A zoomed view of a pixel with the liquid crystal layer, demonstrating how the orientation changes based on applied voltage

2.1.2 Phase Modulation with SLMs

SLMs impose a spatially varying phase profile on an incident light field. The phase modulation is described by the transmission function $t(x,y)$, defined as:

$$t(x,y) = \exp(iH(x,y)), \tag{2.3}$$

where $H(x,y)$ is the phase hologram applied to the SLM at each pixel. Controlling the phase pixel-by-pixel allows the SLM to create complex wavefronts such as vortex beams, Bessel beams, and Laguerre-Gaussian beams.

2.1.3 Hologram Construction

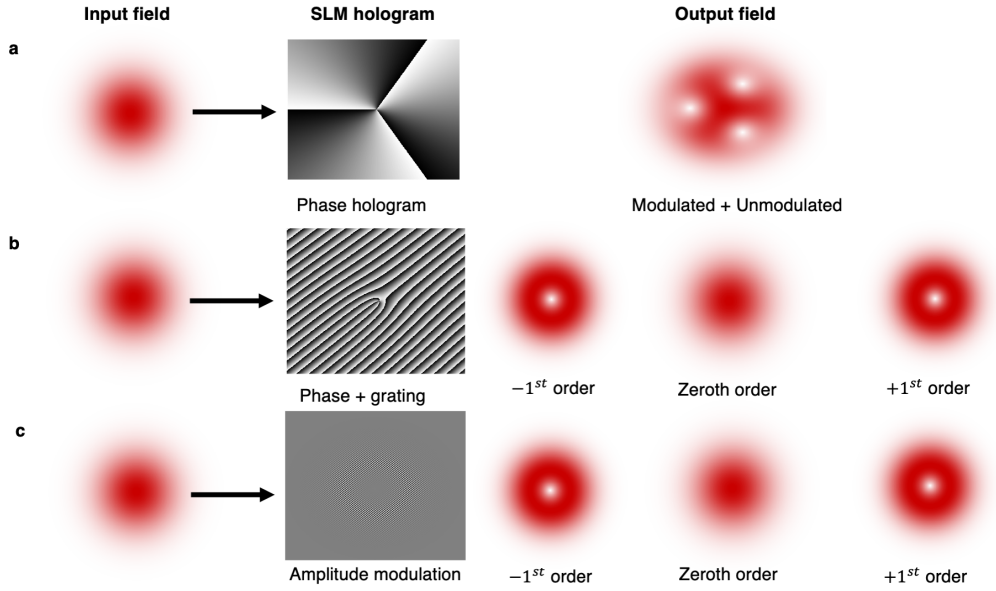


FIGURE 2.2: SLM hologram pattern construction for imparting orbital angular momentum (OAM) onto a Gaussian beam. (a) Encoding a spiral phase alone results in a superposition of multiple modes. (b) This is corrected by incorporating a linear grating, where the desired mode appears in the first-order diffraction. (c) Illustration of an amplitude modulation hologram used to control both phase and intensity.

Designing a hologram for an SLM involves tailoring the phase profile $H(x,y)$ to achieve a specific transformation of the incident light field. To demonstrate an SLM works, we aim to generate Laguerre-Gaussian (LG) beams, which are structured light fields carrying orbital angular momentum (OAM) from gaussian beams. A standard Gaussian beam has a symmetric intensity profile and a flat phase front. However, by imprinting a spiral phase structure onto the beam, we can transform it

into an LG mode. This is achieved by encoding a helical phase factor of the form:

$$R(r)f(z)e^{i\ell\phi} \tag{2.4}$$

where l is the topological charge, $R(r) = C\left(\frac{r\sqrt{2}}{w(z)}\right)^{|\ell|} L_p^{|\ell|}\left(\frac{2r^2}{w^2(z)}\right) e^{-\frac{r^2}{w^2(z)}}$ is the radial function, $f(z) = \frac{1}{w(z)} e^{-ikz - ik\frac{r^2}{2R(z)} + i\zeta(z)}$ is the axial function and $\phi(x,y)$ is the azimuthal angle. This modification results in a twisted wavefront, introducing a central phase singularity and a characteristic doughnut-shaped intensity profile.

To achieve this transformation experimentally, we use a spatial light modulator (SLM) to display a hologram that imparts the desired phase structure onto the incident beam. The phase hologram for generating an LG mode can be expressed as:

$$H(x,y) = l\phi(x,y) \mod 2\pi. \tag{2.5}$$

However, due to inefficiencies in phase-only modulation, the desired mode often appears superimposed with residual unmodulated light (Fig. 2.2a). To separate the modulated beam from unwanted diffraction orders, a linear phase grating is incorporated, resulting in the modified hologram:

$$H(x,y) = l\phi(x,y) + 2\pi(G_x x + G_y y) \mod 2\pi. \tag{2.6}$$

Some applications require simultaneous control over both amplitude and phase. As SLMs are phase-only devices, encoding techniques like Arrizón encoding are used to achieve both amplitude and phase modulation. Arrizón encoding is a technique that approximates the desired amplitude and phase by modulating the phase only, thus indirectly controlling the amplitude [77]. The desired light field is generally written as:

$$V(x,y) = A(x,y)\exp(i\Psi(x,y)), \tag{2.7}$$

where $A(x,y)$ is the amplitude distribution, and $\Psi(x,y)$ is the phase distribution. Arrizón encoding allows the SLM to modulate both by generating a hologram that approximates the complex field. The transmission function $t(A,\Psi)$, representing the modulation of the SLM, is written as:

$$t(A,\Psi) = \exp(iH(A,\Psi)), \tag{2.8}$$

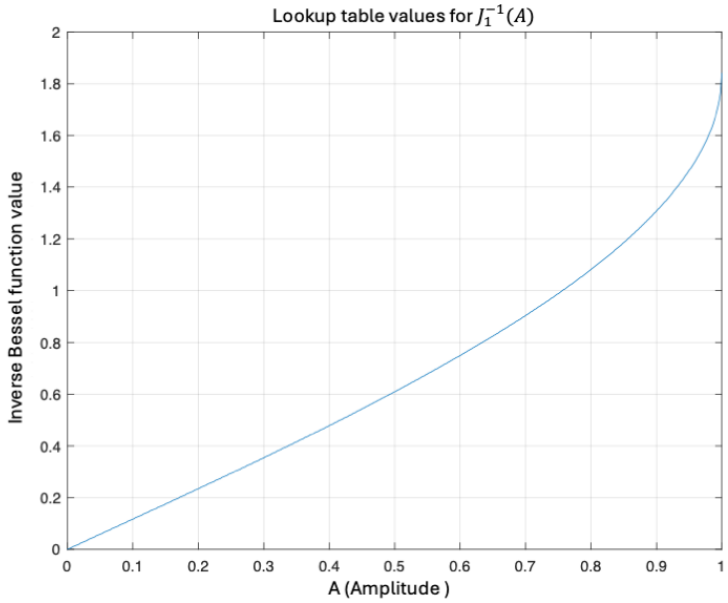


FIGURE 2.3: Plot of the inverse of the first-order Bessel function, $J_1^{-1}(A)$, over the amplitude range $0 \leq A \leq 1$. Since $J_1(A)$ is non-monotonic (i.e., it does not consistently increase or decrease over its full domain), only the monotonic region is used for inversion, and the inverse is implemented via a lookup table for Arrizón encoding.

where $H(A, \Psi)$ is the phase-only hologram applied to the SLM. To encode both amplitude and phase, Arrizón encoding uses a Fourier series expansion of the transmission function:

$$t(A, \Psi) = \sum_n t_n(A) \exp(in\Psi), \tag{2.9}$$

where $t_n(A)$ are the Fourier coefficients. The first-order term ($n = 1$) is sufficient to encode both the amplitude and phase. Arrizón encoding expresses the hologram $H(A, \Psi)$ as:

$$H(A, \Psi) = f(A) \sin(\Psi), \tag{2.10}$$

where $f(A)$ is a function modulating the amplitude via phase. This method leverages the Jacobi–Anger expansion, given by:

$$\exp(if(A)\sin(\Psi)) = \sum_m J_m(f(A)) \exp(im\Psi), \tag{2.11}$$

where $J_m()$ are Bessel functions of the first kind. The amplitude modulation is achieved by matching the first-order Bessel function with the desired amplitude, leading to the expression:

$$f(A) = J_1^{-1}(A), \tag{2.12}$$

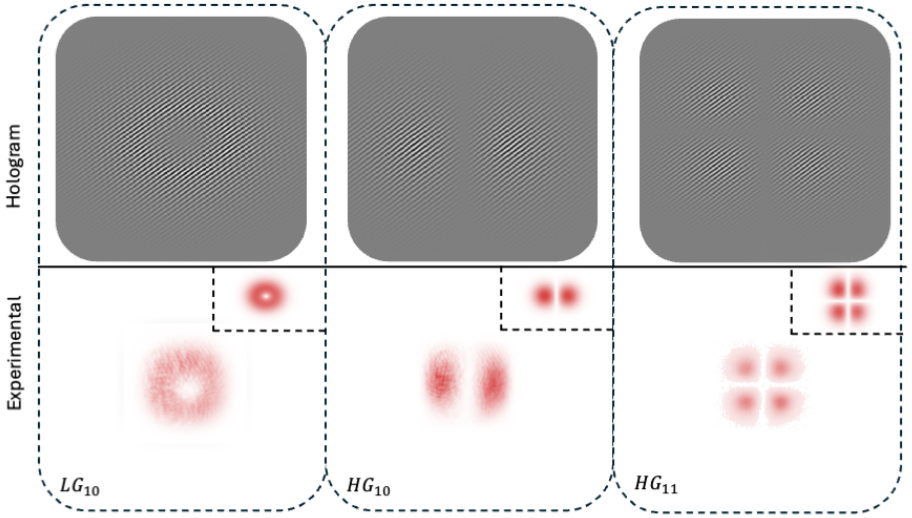


FIGURE 2.4: Experimentally generated modes LG_{10}, HG_{10}, and HG_{11}, along with their corresponding holograms using complex amplitude modulation. Theoretical modes are included as insets for visual comparison.

where $J_1^{-1}(A)$ is the inverse of the first-order Bessel function. Because $J_1(A)$ is non-monotonic (i.e., the function does not consistently increase or decrease over its entire domain) in general and does not have a simple closed-form inverse, we restrict A to its valid range $0 \leq A \leq 1$ and precompute $J_1^{-1}(A)$ using a numerical approach. In practice, these values are stored in a lookup table for rapid access. Figure 2.3 shows the resulting inverse function in this monotonic domain, illustrating how amplitude values are mapped to the phase offset needed for Arrizón encoding. This relationship ensures the desired amplitude and phase modulation is approximated by the phase-only hologram. The final hologram, Fig 2.2 c, used to modulate both the amplitude and phase is given by:

$$H(A, \Psi) = f(A) \sin\left(\Psi + 2\pi(G_x x + G_y y)\right) \mod 2\pi, \tag{2.13}$$

where G_x and G_y are the grating frequencies used to shift the modulated light away from the zeroth-order unmodulated component. In Fig 2.4 we illustrate experimentally generated modes and their holograms. Theoretical modes are included as insets for visual comparison.

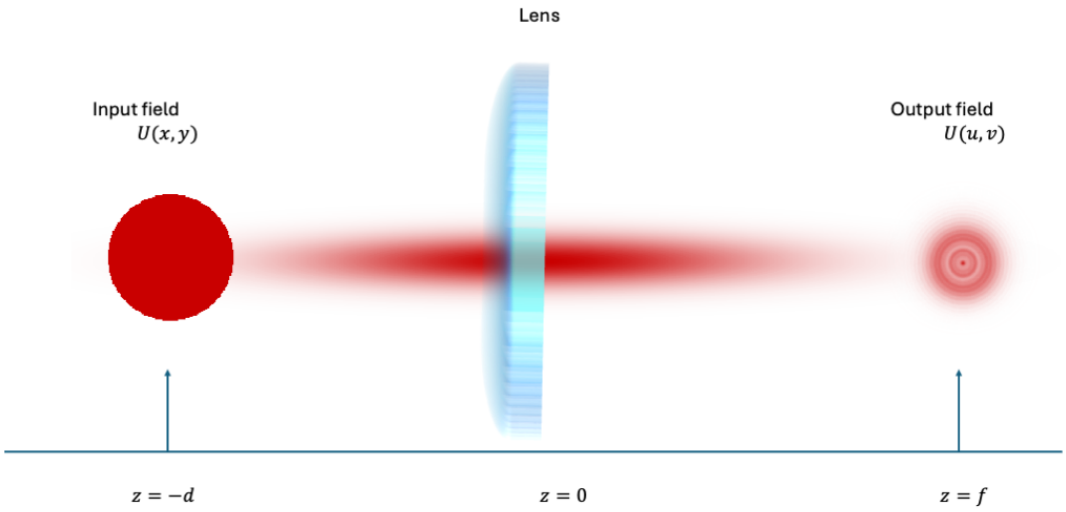


FIGURE 2.5: Simulation of a lens performing a spatial Fourier transform. The input field, a circular apperture undergoes free-space propagation and phase modulation, transforming into its Fourier representation at the focal plane. The aperture is placed at $z=-d$, and a lens at $z=0$ performs a spatial Fourier transform. The resulting diffraction pattern appears at the focal plane$z=f$, corresponding to the Fourier transform of the aperture function.

While SLMs provide a powerful tool for spatial phase modulation, optical transformations often require additional operations beyond direct modulation. In particular, Fourier optics enables complex spatial filtering, convolution, and mode decomposition—key steps in optical computation. In the next section, we explore how lenses perform spatial Fourier transforms, an essential concept for implementing optical matrix-vector multiplication

2.2 The Spatial Fourier Transforming Property of a Lens

Fourier transforms are ubiquitous in various information processing schemes. In an optical context, they can be realized with a lens-based setup that leverages free-space propagation and diffraction. This approach allows complex transformations to be achieved naturally within the optical system.

Consider a field $A(x,y)$ that interacts with an object placed at a distance d from a lens, imparting a spatially varying amplitude transmittance $I(x,y)$. The emerging field, $U(x,y)=A(x,y)I(x,y)$, thus carries encoded information about the object, as shown in Fig. 2.5. As this field propagates in free space from plane $z=-d$ to $z=0$ (where the lens is located), it can be described in terms of its angular spectrum components. The spatial frequency components f_x and f_y represent the Fourier domain

coordinates in the x- and y-directions, respectively. These correspond to the plane wave decomposition of the field in free space. The relationship between the field at these two planes is given by:

$$A(f_x, f_y, 0) = A(f_x, f_y, -d)\, e^{-i\pi\lambda d(f_x^2+f_y^2)}. \tag{2.14}$$

Each $A(f_x, f_y, z)$ represents the coefficients in the plane wave expansion of the field:

$$U(x,y,z) = \iint A(f_x, f_y, z)\, e^{i2\pi(f_x x + f_y y)}\, df_x\, df_y. \tag{2.15}$$

Upon reaching the lens, the field encounters a quadratic phase transmittance function:

$$t(x,y) = e^{-i\frac{k}{2f}(x^2+y^2)}, \tag{2.16}$$

where f is the focal length of the lens, with $k = \frac{2\pi}{\lambda}$. This results in a field distribution at $z = f$ (the focal plane), which can be expressed using the Fresnel diffraction integral:

$$U_f(u,v,f) = \frac{\exp\left[i\frac{k}{2f}(u^2+v^2)\right]}{i\lambda f} \iint_{-\infty}^{\infty} U_i'(x,y,0) \exp\left[-i\frac{2\pi}{\lambda f}(xu+yv)\right] dx\, dy, \tag{2.17}$$

where

$$U_i'(x,y,0) = U(x,y) e^{-i\frac{k}{2f}(x^2+y^2)}. \tag{2.18}$$

At the focal plane of the lens, the spatial coordinates u and v correspond to the transformed positions where the Fourier spectrum of the input field is observed. These are related to the spatial frequency coordinates f_x and f_y by:

$$f_x = \frac{u}{\lambda f} \quad \text{and} \quad f_y = \frac{v}{\lambda f}. \tag{2.19}$$

Rewriting the quadratic phase term in terms of f_x and f_y, we get:

$$\exp\left[i\frac{k}{2f}(u^2+v^2)\right] = \exp\left[i\pi\lambda f(f_x^2+f_y^2)\right]. \tag{2.20}$$

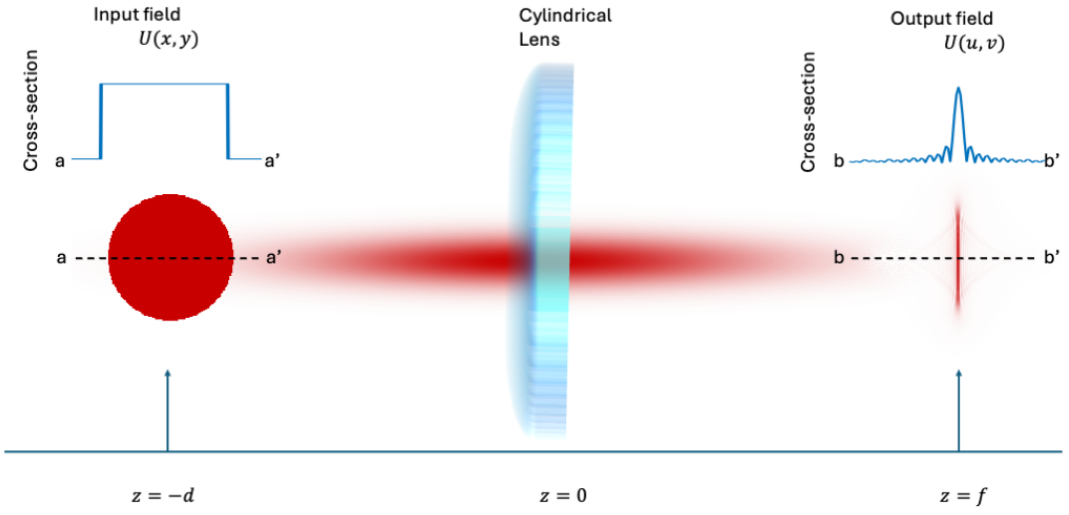


FIGURE 2.6: Schematic illustration of the 1D Fourier transform by a cylindrical lens. A circular aperture (left) is illuminated to form the input field $U(x,y)$. The cylindrical lens at $z=0$ imposes a 1D transform along the x axis, yielding the output field $U(u,v)$ at the focal plane $z=f$. Cross-sections (a–a and b–b) highlight how the transformation occurs solely along x, resulting in a sinc-like profile in that direction, while the y-axis remains unchanged.

Thus, the integral simplifies to:

$$U_f(u,v,f) = \frac{\exp\left[i\pi\lambda f(f_x^2+f_y^2)\right]}{i\lambda f}\iint_{-\infty}^{\infty} U(x,y,0)\,e^{-i2\pi(f_x x+f_y y)}\,dx\,dy$$
$$= \frac{\exp\left[i\pi\lambda f(f_x^2+f_y^2)\right]}{i\lambda f} A(f_x,f_y,0). \quad (2.21)$$

Utilizing the expression for $A(f_x,f_y,0)$ from Eq. (2.14), we obtain:

$$U_f(u,v,f) = \frac{\exp\left[i\pi\lambda f(f_x^2+f_y^2)\right]}{i\lambda f} A(f_x,f_y,-d)e^{-i\pi\lambda d(f_x^2+f_y^2)}. \quad (2.22)$$

When $d=f$, the phase factors cancel, yielding:

$$U_f(u,v,f) = \frac{1}{i\lambda f}A(f_x,f_y,-f) = \frac{1}{i\lambda f}\iint U(x,y,-f)\,e^{-i2\pi(f_x x+f_y y)}\,dx\,dy. \quad (2.23)$$

This confirms that the field at $z=f$ is the Fourier transform of the field at $z=-f$, with u and v serving as the transformed coordinates corresponding to the spatial frequencies f_x and f_y.

We now consider the action of a cylindrical lens with a focal length f along the x-axis as a special case. Such a lens applies a quadratic phase factor only in the

x-direction:

$$t(x,y) = e^{-i\frac{k}{2f}x^2}. \tag{2.24}$$

This phase factor results in a one-dimensional Fourier transform $\mathcal{F}_x$ along the x-axis, while leaving the y-axis unchanged. At the focal plane of the cylindrical lens, the transformation of the field $U(x,y)$ occurs such that only the x-axis spatial frequencies f_x are Fourier-transformed, while the y-axis spatial frequencies remain unchanged. For a field $U(x,y,0)$ incident on the cylindrical lens, the resulting field at the focal plane $z = f$ will be:

$$U_f(u,y,f) = \frac{\exp\left[i\frac{k}{2f}u^2\right]}{i\lambda f}\int_{-\infty}^{\infty} U(x,y,0)\exp\left[-i\frac{2\pi}{\lambda f}xu\right]dx = \frac{\exp\left[i\frac{k}{2f}u^2\right]}{i\lambda f}\mathcal{F}_x\{U(x,y,0)\}. \tag{2.25}$$

Here, u represents the x-coordinate in the focal plane, corresponding to the transformed spatial frequency $f_x = \frac{u}{\lambda f}$. The y-coordinate remains unchanged, effectively resulting in a one-dimensional Fourier transform along the x-axis alone. As illustrated in Fig. 2.6, a cylindrical lens at $z = 0$ performs a one-dimensional Fourier transform along the x-axis. The input field $U(x,y)$, formed by illuminating a circular aperture, undergoes transformation such that its output field $U(u,v)$ at the focal plane $z = f$ exhibits a sinc-like profile along x, while the y-axis remains unchanged. The cross-sections (a–a' and b–b') further demonstrate this selective transformation.

The spatial Fourier transform capability of a lens, combined with local phase modulation by Spatial Light Modulators (SLMs), enables the direct execution of fundamental linear algebra operations such as matrix-vector multiplication. By encoding computational inputs into an optical wavefront and utilizing Fourier optics for processing, we achieve highly parallelized mathematical transformations. In the following section, we present the mathematical framework for optical vector-matrix multiplication, integrating these two key components—SLMs and lenses—into the optical circuit design.

2.3 Combining all tools for vector matrix multplication

Building on the foundational tools (SLMs and lenses), we present an optical method for performing matrix–vector multiplication that leverages spatial light modulators (SLMs), an imaging system, and a cylindrical lens. The process is decomposed into

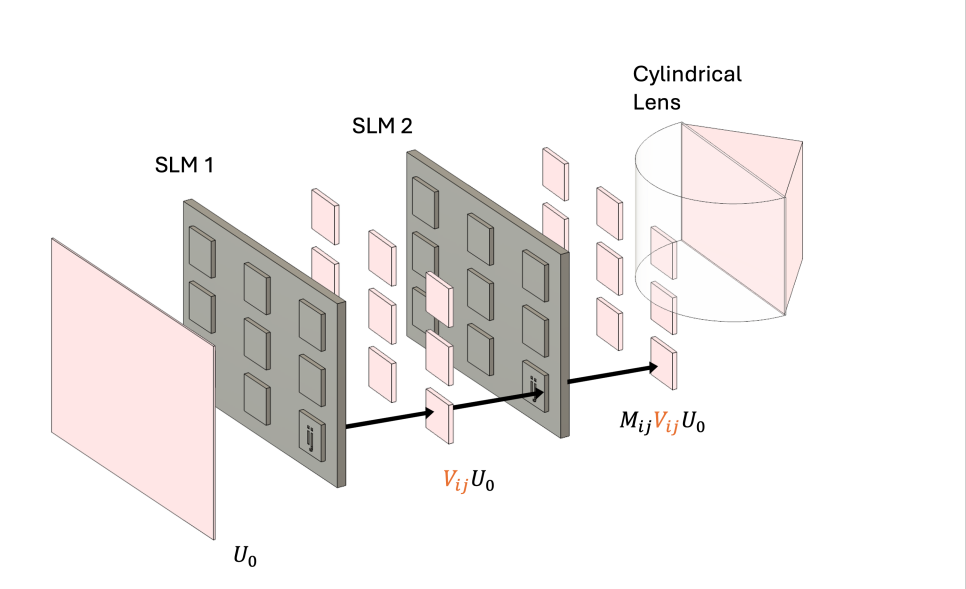


FIGURE 2.7: Schematic of the optical matrix-vector multiplication setup. A plane wave is first modulated by SLM1 to encode the input vector $\mathbf{v}$. The modulated beam then passes through SLM2, which encodes the matrix $\mathbf{M}$, enabling element-wise multiplication of $\mathbf{v}$ and $\mathbf{M}$. A cylindrical lens performs a one-dimensional Fourier transform along the x-axis, summing the products along each row. The output intensity distribution, captured by a CCD camera, corresponds to the squared result of the matrix-vector multiplication.

two main stages: (i) encoding and elementwise multiplication (Hadamard product), and (ii) Fourier-based summation.

The procedure begins by encoding both the matrix $\mathbf{M}$ and the vector $\mathbf{v}$ onto pixelated light modulators. To facilitate this encoding, we define an auxiliary matrix $\mathbf{V}$ where each element of $\mathbf{v}$ is repeated along every row:

$$V_{ij} = v_j. \tag{2.26}$$

This mapping is conveniently achieved using the outer product:

$$\mathbf{V} = |b\rangle\langle v|, \tag{2.27}$$

with $|b\rangle$ denoting an n-dimensional all-ones vector:

$$|b\rangle = (1,1,\ldots,1)^{\cdot} \tag{2.28}$$

Thus, $\mathbf{V}$ has identical rows, each containing the elements of $\mathbf{v}$:

$$\mathbf{V} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \\ v_1 & v_2 & \cdots & v_n \\ \vdots & \vdots & \ddots & \vdots \\ v_1 & v_2 & \cdots & v_n \end{bmatrix}. \tag{2.29}$$

Optically, a field is initialized as a uniform plane wave, denoted U_0 Fig 2.7. This field is first modulated by SLM1 to encode the vector $\mathbf{v}$ (or equivalently, the matrix $\mathbf{V}$) and is then relayed to SLM2, which encodes the matrix $\mathbf{M}$. An imaging system between the two SLMs ensures that the field modulated by SLM1 is directly multiplied—pointwise—by the modulation on SLM2. Mathematically, the resulting field is given by:

$$U_{ij} = M_{ij} V_{ij} U_0, \tag{2.30}$$

which represents the Hadamard (elementwise) product of $\mathbf{M}$ and $\mathbf{V}$. To complete the matrix–vector multiplication, the optical system must perform a summation along one dimension. This is achieved by passing the modulated field through a cylindrical lens. A lens with focal length f performs a Fourier transform of the incident field along its focusing axis (assumed here to be the x-axis). In the back focal plane of the lens, the field is expressed as the Fourier transform of the spatial profile of the input. Evaluating the field at the center of this focal plane (i.e., at $k_x = 0$) effectively computes the integral (or discrete sum) of the field along the x-direction:

$$\mathcal{F}\{U(x)\}\Big|_{k_x=0} = \int U(x)\,\mathrm{d}x. \tag{2.31}$$

By selecting the $k = 0$ component (the center of the Fourier plane), the cylindrical lens effectively sums the contributions from each row of the elementwise product $M_{ij}V_{ij}$. This yields the final optical output:

$$U_{\text{out},i} = \sum_j M_{ij} v_j U_0, \tag{2.32}$$

which is equivalent to the matrix–vector multiplication $(\mathbf{Mv})_i$ up to the constant factor U_0. While this technique has already been introduced in classical optics [27], in chapter 4 it will be reformulated for quantum computing, where the vector $\mathbf{v}$ now encodes a quantum state and $\mathbf{M}$ encodes a unitary gate operator.

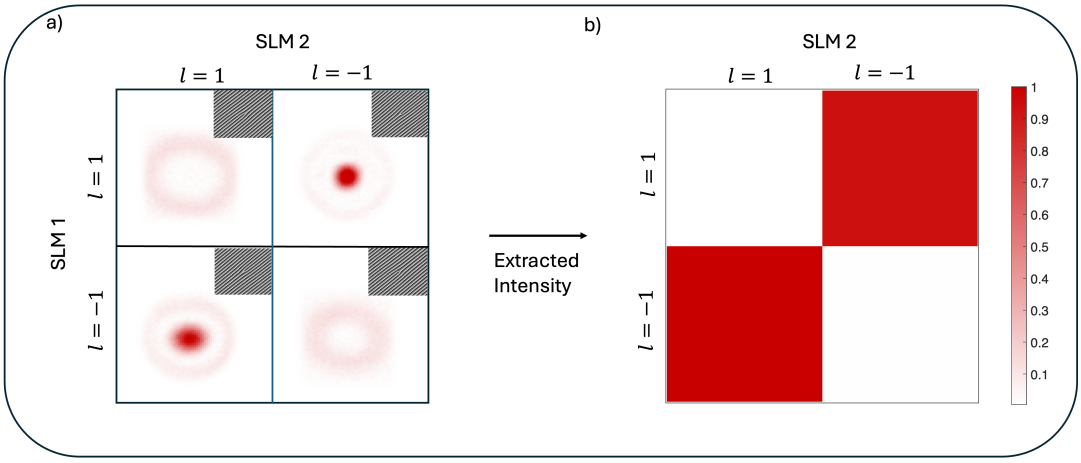


FIGURE 2.8: a) Experimental intensity distributions at the detection plane for different OAM modes, showing strong on-axis peaks when the phase on the SLM1 matches conjugation on SLM2. b) Extracted intensity matrix in which each row corresponds to the OAM mode encoded on the SLM1, and each column corresponds to the detected OAM mode. The red squares indicate high overlap (i.e., a strong on-axis signal), confirming accurate alignment when the encoded and detected modes match.

2.3.1 Alignment verification

To verify the alignment of the experimental setup, we employ a modal decomposition technique. The principle behind this method is that any arbitrary field in the transverse plane can be expressed as a linear superposition of a complete set of eigenfunctions. In our case, we consider orbital angular momentum (OAM) eigenfunctions, so that the transverse field $U(r,\theta)$ can be written as

$$U(r,\theta) = \sum_l c_l \, e^{il\theta}, \tag{2.33}$$

where $c_l$are the complex expansion coefficients corresponding to each OAM mode $e^{il\theta}$. Since these eigenfunctions are orthonormal, the coefficients c_l can be uniquely determined by the overlap integral

$$c_l = \frac{1}{2\pi} \int_0^{2\pi} U(r,\theta) \, e^{-il\theta} \, d\theta. \tag{2.34}$$

Experimentally, this overlap integral is implemented by propagating the light beam through a spatial light modulator (SLM) that encodes the conjugate phase $e^{-il\theta}$onto the incoming field. The modulated beam is then directed through a Fourier-transforming lens with focal length f, which performs a Fourier transform of the field. The field

at the back focal plane is given by

$$U_2 = C \int U_1 e^{-il\theta} e^{-i\frac{2\pi}{\lambda f}\mathbf{k}\cdot\mathbf{r}} d\theta, \tag{2.35}$$

where $C$is a constant, λ is the wavelength, and $\mathbf{k}$ represents the transverse spatial frequency. On the optical axis (i.e., when $\mathbf{k} = 0$), this expression simplifies to

$$U_2 = C \int U_1 e^{-il\theta} d\theta. \tag{2.36}$$

For the correctly detected mode, we expect a pronounced on-axis intensity peak. This occurs because the SLM applies the appropriate conjugate phase $e^{-il\theta}$, effectively flattening the helical phase of the incoming mode. Modes that do not match the applied phase yield significantly lower on-axis intensity, thus enabling modal discrimination. By measuring the on-axis intensity $|U_2|^2$with a CCD camera, we can directly infer the squared magnitude of the modal coefficient $|c_l|^2$ and thereby verify proper alignment of the optical system. Figure 2.8 shows a representative example of the measured beam profiles (a) and the corresponding cross-talk matrix (b). The matrix rows represent the OAM modes encoded on SLM1, while the columns represent the detected OAM modes on SLM2. The bright red squares along the diagonal highlight strong overlap, confirming proper alignment.

Having established a comprehensive toolbox for optical matrix multiplication—covering the operation of spatial light modulators, Fourier-transforming lenses, and phase modulation techniques—we now have all the essential components required to emulate quantum computing with structured light. These components will be put to use in Chapter 4 when we present our framework. In the next chapter, we elucidate foundational concepts of quantum computing by demonstrating a practical problem on current quantum hardware.

Chapter 3

Learning to solve a useful problem on a quantum computer: reconstructing quantum states

With a basic understanding of quantum computing principles and a toolbox for matrix multplication established in the chapters 1 and 2, this section serves as a guide to navigating practical quantum computing while exploring a valuable quantum optics characterization technique—Quantum State Tomography (QST). Using IBM's quantum processors, a variational quantum algorithm (VQA) [78] is applied to reconstruct quantum states. Analogous to constructing a 3D object from its 2D projections (Fig 3.1 a), QST (see Appendix A for an introductory discussion) aims to reconstruct the density matrix ρ of a quantum system from a set of complete measurements [79]. This approach not only demonstrates practical applications of quantum computing but also provides essential insights into the characterization and validation of quantum optical systems..

In this chapter, we present a method for reconstructing the density matrix of quantum states entangled in the orbital angular momentum (OAM) degree of freedom, denoted by $|l\rangle$, where l is the topological charge that defines the orbital angular momentum of a photon with $l\hbar$ [80]. These OAM-entangled states are generated through spontaneous parametric down-conversion (SPDC), a well-known source of entangled photons [81]. We approach the quantum state tomography (QST) problem by first formulating it as a least squares optimization problem. This method transforms the task of reconstructing the density matrix into a system of linear equations, where the projective measurements of the quantum state are related to the unknown density matrix components. We then solve this least squares problem using a hybrid

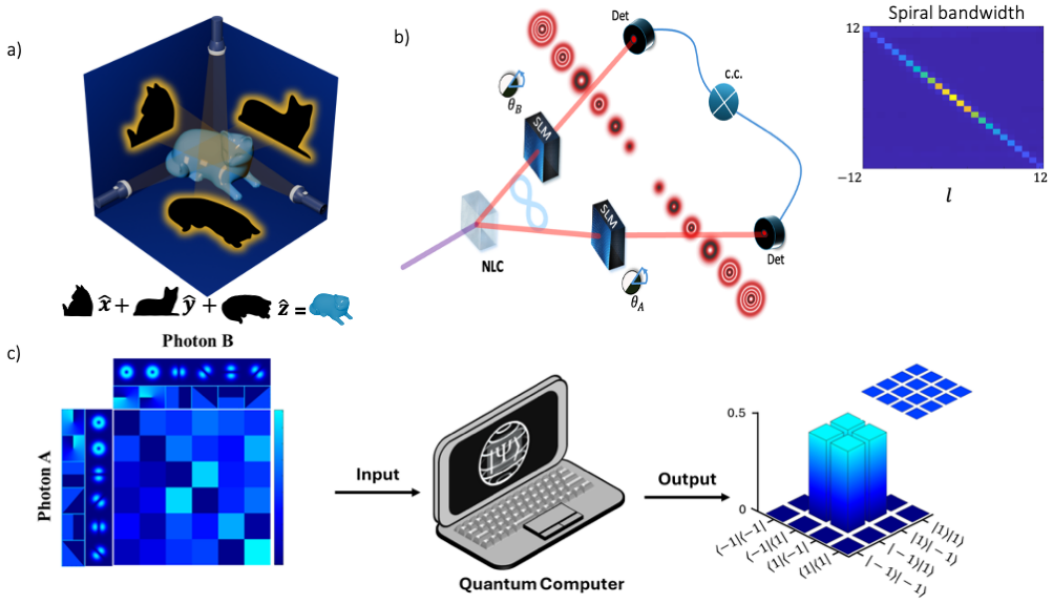


FIGURE 3.1: Concept figure on quantum state tomography. (a) We highlight concpetual layout of the tomography problem. Similar to reconstructing an image of a 3D object from its projections, in QST we aim to reconstruct the density matrix ρ (our version of the 3D object) from a set projection measurements. The quantum experiment is shown in b), entangled photons are generated from a non-linear crystal and two SLM's with single mode fibres perform projection measurements on them. In c) we illustrate typical measurements and our objective to utalise a quantum computer for obtaining the density matrix as an output.

classical-quantum approach, specifically by employing Variational Quantum Algorithms (VQAs). Our implementation encodes the least squares objective function into an Ising spin model [82], where different trial wavefunctions are tested using various parametrised ansatz circuits. The VQA iteratively adjusts the parameters of these circuits, minimizing the expectation value of the cost function until convergence is reached. To validate the performance of the proposed approach, we apply it to real experimental data from the SPDC source and test the reconstruction of the OAM-entangled quantum states. Through this work, we demonstrate the applicability of VQAs in reconstructing quantum states and provide insights into the effectiveness of different circuit configurations for this purpose.

3.1 Theory

3.1.1 State reconstruction as least square problem

Consider the problem of reconstructing the density matrix $\hat{\rho}$, which describes a quantum state $|\psi\rangle$, from a set of measurements m_k acquired through the application of a set of projection operators P_k. For a pure state, the density matrix is given by

$\hat{\rho} = |\psi\rangle\langle\psi|$. In this work we are interested in reconstructing the density matrix for entangled states produced by spontaneuos parameteric down conversion (SPDC). The produced two photon state is described by an entangled state ψ in the orbital angular momentum (OAM) basis $|l\rangle$:

$$\psi = \sum_l a_l |l\rangle_1 |-l\rangle_2,$$

here $|a_l|^2$ is the probability of finding photon 1 and 2 carrying OAM $l\hbar$ and $-l\hbar$, respectively. While in principle the OAM spectrum can extend to inifity in practice it's restctricted to a range of OAM values called the spiral bandwidth, we illlustrate this in Fig 3.1 b for an OAM range $l = \pm 12$. In this work we restrict ourselve to the biphoton states of dimension $d = 2$:

$$\psi = |-1\rangle_1 |1\rangle_2 + |1\rangle_1 |-1\rangle_2 \tag{3.1}$$

This restriction is solely based on the limited number of physical qubits on IBM's quantum platform.

The relationship between the obtained measurements and the projection operators acting on the physical state follows from Born's rule [83]:

$$m_k = \mathrm{tr}(P_k \hat{\rho}) = \bar{P}_k^\dagger \cdot \bar{\hat{\rho}}, \tag{3.2}$$

where the operation $\bar{\cdot}$ denotes the flattening of a matrix into a vector. Specifically, the flattening operator transforms a $d \times d$ matrix P_k into a $d^2 \times 1$ vector. For example, the flattened forms of the projection operator $P_1 = |0\rangle\langle 0|$ and the density matrix $\hat{\rho}$ are:

$$\bar{P}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}' \quad \text{and} \quad \bar{\hat{\rho}} = \begin{bmatrix} \rho_{11} \\ \rho_{12} \\ \rho_{21} \\ \rho_{22} \end{bmatrix}. \tag{3.3}$$

The goal is to reformulate the state reconstruction problem as a least squares optimization problem. By stacking the measurement outcomes m_k into a vector $\mathbf{m}$ and the flattened projection operators $\bar{P}_k$ into a matrix T, we arrive at the following

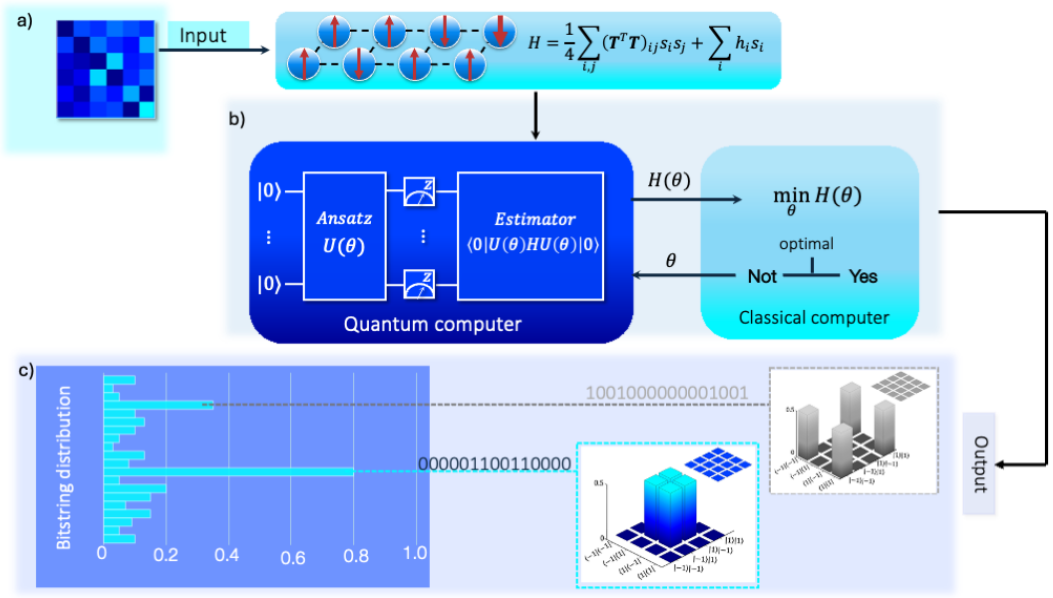


FIGURE 3.2: VQE Algorithm Workflow: a) The input variables are transformed into spin variables, converting the least squares problem into an Ising spin model. The cost function is expressed as a Hamiltonian in terms of spin observables. b) A quantum computer evaluates the Hamiltonian's expectation value based on ansatz circuit parameters, which are iteratively optimized by a classical computer. c) Once the expectation value is minimized, the optimized circuit parameters are used to sample the bitstring distribution through measurements in the computational basis. The density matrix is then reconstructed by reshaping the bitstrings with high counts into a density matrix representation.

system of linear equations:

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} = \begin{bmatrix} -\bar{P}_1^\dagger - \\ -\bar{P}_2^\dagger - \\ -\bar{P}_3^\dagger - \\ -\bar{P}_k^\dagger - \end{bmatrix} \begin{bmatrix} \rho_{11} \\ \rho_{12} \\ \rho_{21} \\ \rho_{22} \end{bmatrix} = \underbrace{\begin{bmatrix} P_{11}^* & P_{12}^* & P_{13}^* & P_{14}^* \\ P_{21}^* & P_{22}^* & P_{23}^* & P_{24}^* \\ P_{31}^* & P_{32}^* & P_{33}^* & P_{34}^* \\ P_{k1}^* & P_{k2}^* & P_{k3}^* & P_{k4}^* \end{bmatrix}}_{T} \begin{bmatrix} \rho_{11} \\ \rho_{12} \\ \rho_{21} \\ \rho_{22} \end{bmatrix}. \tag{3.4}$$

In this matrix form, P_{kj} represents the j-th element of the k-th projector. The Hermitian conjugate is denoted by $\dagger$, while $*$ represents complex conjugation. Thus, the measurement vector $\mathbf{m}$ can be expressed as:

$$\mathbf{m} = T\tilde{\rho}. \tag{3.5}$$

This form allows the tomography problem to be solved by minimizing the least squares cost function:

$$f(\bar{\hat{\rho}}) = |\mathbf{m} - T\bar{\hat{\rho}}|^2 = \bar{\hat{\rho}}^\dagger T^\dagger T\bar{\hat{\rho}} - 2\mathbf{m}^\dagger T\bar{\hat{\rho}} + \mathbf{m}^\dagger\mathbf{m}. \tag{3.6}$$

Thus, given a matrix T constructed from the action of the projection operators P_k on the physical state (3.1 b)to produce the measurements $\mathbf{m}$, minimizing equation 3.6 while iterating over guesses of $\hat{\rho}$ enables an estimation of the density matrix of the true state. The projections are performed using spatial light modulators (SLMs) and single-mode fibers. The projectors P_k, obtained from the action of these operators, contribute to the construction of the matrix T, which can be divided into two key sets: a) the computational basis $\mathcal{C} = \{|j\rangle, j = 1,\ldots,d\}$, representing the standard OAM states, and b) the superposition basis $\mathcal{S} = \left\{\frac{|j\rangle + e^{i\theta}|k\rangle}{\sqrt{2}}\right\}, k > j, k \leq d, \theta = 0, \pi, \frac{\pi}{2}, -\frac{\pi}{2}$, which represents superpositions of different OAM states with varying phases. The total number of projectors is given by $4\binom{d}{2} + d$, encompassing both the computational basis and superposition states. These projectors form a complete basis for quantum state reconstruction, ensuring that we capture all relevant information from the system. By stacking these projectors in their flattened form, we construct the matrix T used in the least squares optimization.

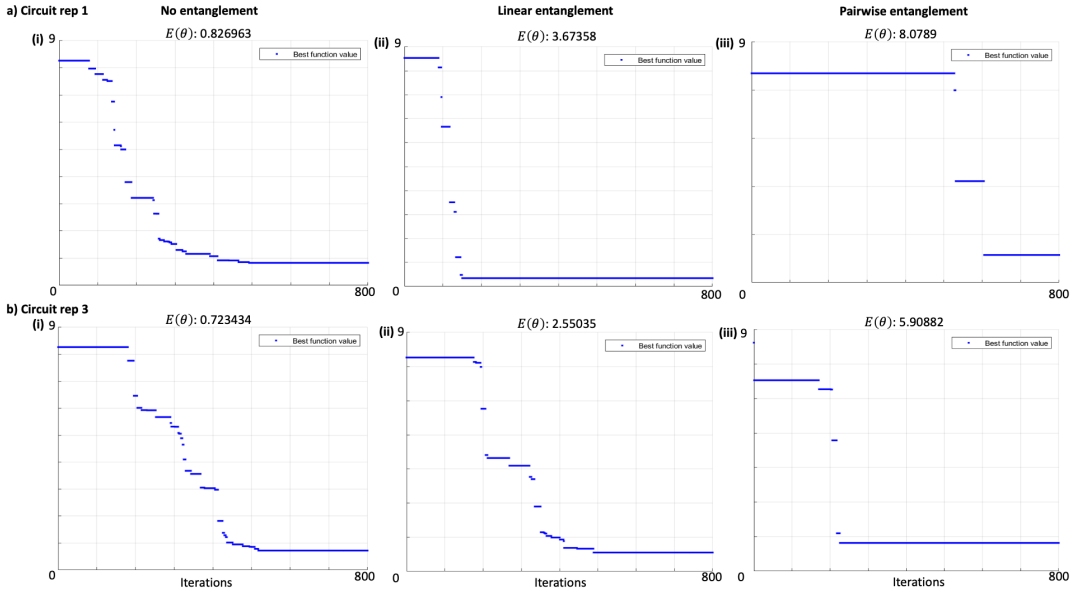


FIGURE 3.3: Training Data for Various Circuit Parameters: (a) Circuit depth 1 is investigated with different entanglement configurations: (i) no entanglement, (ii) linear entanglement, and (iii) pairwise entanglement. (b) The circuit depth is increased to 3. Results show that circuits without entanglement converge to the lowest cost function value.

3.1.2 VQE solution

The Variational Quantum Eigensolver (VQE) minimizes a cost function by encoding it into a quantum Hamiltonian, H, and finding its ground state. In this work, we map the least squares cost function onto an Ising model, where the ground state of the Hamiltonian corresponds to the optimal solution for the tomography problem. The algorithm workflow is illustrated in Fig 3.2.

To represent the cost function in terms of an Ising model, we express the components of the density matrix, $\boldsymbol{\rho}$, as spins s_i, where $s_i \in \{-1,1\}$ are the eigenvalues of the Pauli-Z operator. The relation $\rho_i = \frac{1-s_i}{2}$ maps the spin variables to the components of the density matrix. By substituting this representation into the least squares objective function, we can rewrite the cost function as follows:

$$\mathcal{L}(s) = \frac{1}{4}\sum_{i,j}(\mathbf{T}^T\mathbf{T})_{ij} - (\mathbf{T}^T\mathbf{T})_{ij}s_i - (\mathbf{T}^T\mathbf{T})_{ij}s_j + (\mathbf{T}^T\mathbf{T})_{ij}s_is_j - \sum_i(\mathbf{T}^T\mathbf{M})_i + \sum_i(\mathbf{T}^T\mathbf{M})_is_i + \mathbf{M}^T\mathbf{M}. \quad (3.7)$$

In this formulation, the quadratic term $\frac{1}{4}\sum_{i,j}(\mathbf{T}^T\mathbf{T})_{ij}s_is_j$ represents the interaction between spins, analogous to the coupling constants J_{ij} in the Ising model. The linear terms, $-\frac{1}{4}\sum_{i,j}(\mathbf{T}^T\mathbf{T})_{ij}s_i - \frac{1}{4}\sum_{i,j}(\mathbf{T}^T\mathbf{T})_{ij}s_j + \sum_i(\mathbf{T}^T\mathbf{M})_is_i$, correspond to the effective magnetic field acting on each spin s_i, which can be expressed as:

$$h_i = -\frac{1}{2}\sum_j(\mathbf{T}^T\mathbf{T})_{ij} + (\mathbf{T}^T\mathbf{M})_i.$$

Finally, the offset or constant terms are given by $\frac{1}{4}\sum_{i,j}(\mathbf{T}^T\mathbf{T})_{ij} + \mathbf{M}^T\mathbf{M} - \sum_i(\mathbf{T}^T\mathbf{M})_i$. This leads to the Ising Hamiltonian representation:

$$\mathcal{H}(s) = \frac{1}{4}\sum_{i,j}(\mathbf{T}^T\mathbf{T})_{ij}s_is_j + \sum_i h_is_i + \text{constant terms}.$$

The goal of the VQE algorithm is to find the spin configuration s that minimizes this Hamiltonian. By using a quantum-classical hybrid approach, we prepare trial wavefunctions with a parameterized quantum circuit, and iteratively minimize the expectation value of the Hamiltonian, $\langle H\rangle = \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$, using a classical optimizer. The variational parameters $\boldsymbol{\theta}$ are adjusted to find the lowest energy state,

which corresponds to the optimal solution to the tomography problem. The solution is extracted as a bitstring, corresponding to the optimal configuration of spins. Given that we employ basis encoding, the optimal solution is inherently restricted to binary bitstrings, reflecting the eigenstates of the qubits, $|0\rangle$ or $|1\rangle$. This encoding directly correlates with the components of the density matrix, allowing us to ascertain whether a specific component is present or absent.

To show the steps involved in the Variational Quantum Eigensolver approach, we present the following pseudocode, which outlines the algorithmic framework used to find the optimal parameters that minimize the expectation value of the Hamiltonian. This pseudocode details the iterative process of preparing the quantum circuit with the current set of parameters, measuring the expectation value, extracting the resulting bitstrings, and updating the parameters accordingly until convergence is achieved. The full matlab script is attached Apendix C. By following this structured approach, we can efficiently approximate the ground state of the quantum system while obtaining the necessary binary solutions indicative of the presence or absence of components in the density matrix.

Algorithm 1 Variational Quantum Eigensolver (VQE) Algorithm

1: **Input:** Initial parameters θ
2: **while** not converged **do**
3: Prepare quantum circuit with parameters θ
4: Measure the expectation value $\langle H \rangle$
5: Update parameters θ to minimize $\langle H \rangle$
6: **Output:** Optimal parameters θ^* corresponding to the minimal expectation value and extracted bitstrings

3.1.3 Results and discussion

We began by investigating an efficient ansatz that optimally represents our quantum states for the variational approach. To achieve this, we tested three different entanglement strategies, illustrated in Fig. (3.4): no entanglement, linear entanglement, and pairwise entanglement. In the no-entanglement case, each qubit undergoes independent local rotations without any entangling operations, serving as a baseline where correlations arise purely from measurement statistics rather than circuit connectivity. The linear entanglement strategy connects qubits sequentially, where each

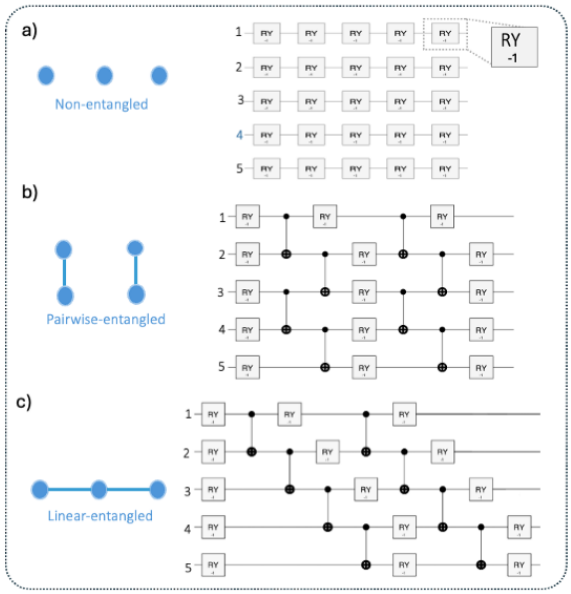


FIGURE 3.4: Variational circuit ansätze with different entanglement structures. (a) No entanglement: each qubit undergoes independent local rotations without entangling operations. (b) Pairwise entanglement: qubits are entangled in disjoint pairs, ensuring controlled entanglement without introducing long-range correlations. (c) Linear entanglement: qubits are entangled sequentially, interacting only with their nearest neighbor. These ansätze were tested for their effectiveness in representing the target quantum state in the variational approach.

interacts only with its nearest neighbor, efficiently distributing entanglement while keeping circuit depth relatively low. In contrast, the pairwise entanglement approach entangles qubits in disjoint pairs, ensuring controlled entanglement without introducing long-range correlations found in more complex entanglement schemes.

To assess the performance of each ansatz, we simulated the measurement matrix M for the state

$$\psi = |-1\rangle_1 |1\rangle_2 + |1\rangle_1 |-1\rangle_2 .$$

The variational circuits, illustrated in Figure 3.4, depict the different entanglement structures tested. These circuits were then deployed on the IBM Hanoi quantum processor, utilizing MATLAB's built-in surrogate model as the classical optimizer to refine the variational parameters.

Figure 3.3 shows the training data, tracking the progress of the energy expectation value $\langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$, our cost function, as it minimizes toward the approximate solution. The cost function values for circuit depths 1 and 3 are summarized in Table 3.1. Notably, the ansatz without any entanglement converges to the solution more effectively than the other configurations, providing an accurate approximation of the target state. This result highlights that, for our specific problem, introducing

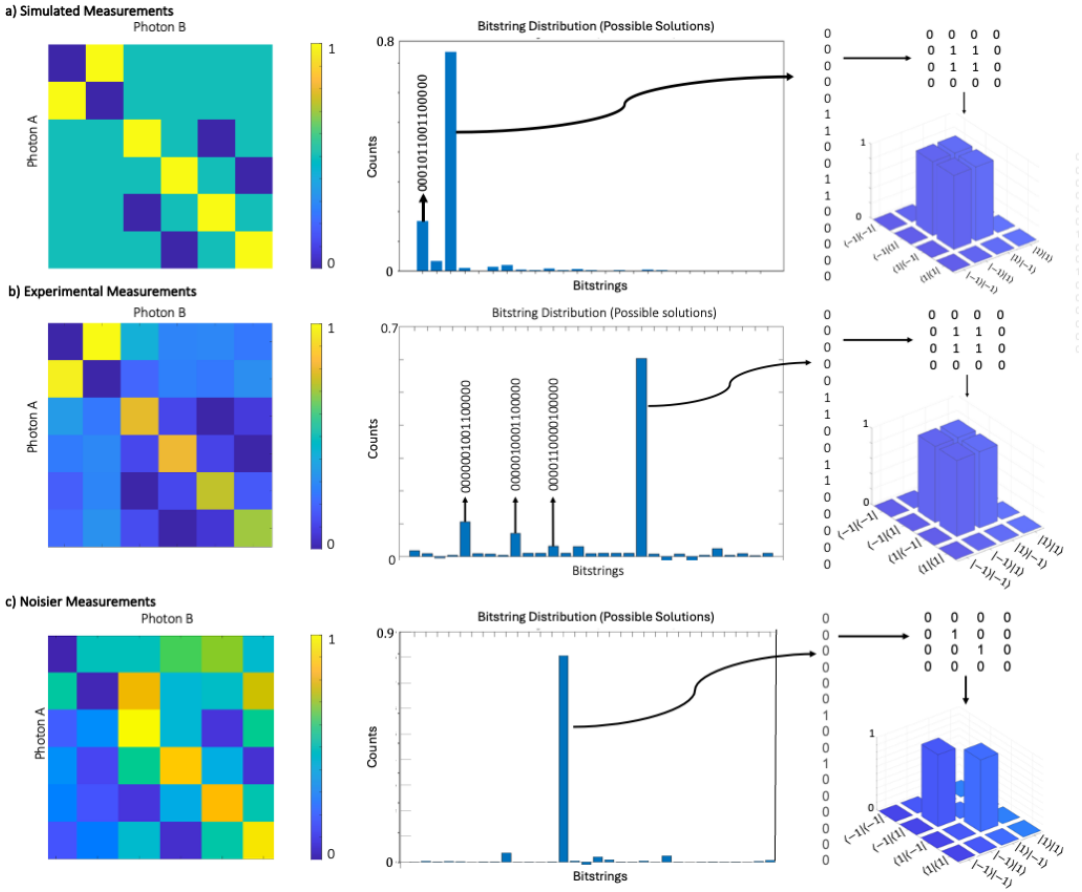


FIGURE 3.5: Bitstring Quasi-Probability Distribution and Experimental Density Matrix Reconstruction: (a) Quasi-probability distribution obtained by running the circuit with optimal parameters for 100 shots. The solution is extracted from the bitstring with the highest counts and reshaped into a 4×4 matrix, demonstrating the algorithm's ability to identify nonzero density matrix elements reliably. (b) Extension to experimental measurements from an SPDC source, reconstructing the anti-symmetric density matrix state $\psi = |1\rangle_a |-1\rangle_b + |-1\rangle_a |1\rangle_b$. The results show that even with real data, the scheme can reliably identify the presence of key density matrix elements. c) Test with noisier data, our vqe scheme performs reconstruction with fidelity 0.88.

entanglement (either linear or pairwise) does not improve convergence. Instead, the simpler non-entangled ansatz achieves the lowest cost function value. We also note that while the linear entanglement ansatz converges faster than the other entangled configurations, its cost function fails to approach zero. This could be due to noisy cost function evaluations, as entangling operations are difficult to implement reliably, or because the trial states lead to regions of the state space that do not contain our solution.

In Figure 3.5a, we illustrate a bitstring quasi-probability distribution from simulated data. This is obtained by running the circuits with the optimal parameters for 100 shots to build statistics, and we derive the solution from the bitstring with the

Ansatz Type	Circuit Depth	Final Cost Function Value
No Entanglement	1	0.83
No Entanglement	3	0.72
Linear Entanglement	1	3.67
Linear Entanglement	3	2.55
Pairwise Entanglement	1	8.08
Pairwise Entanglement	3	5.91

TABLE 3.1: Summary of cost function convergence for different ansatz strategies at circuit depths 1 and 3. The no-entanglement ansatz achieves the lowest cost function, while entangled ansätze fail to reach zero.

highest counts. We compute the fidelity of our approach using

$$F = 1 - \frac{1}{N}\sum_{i=1}^{N} d_i = 1 - \frac{1}{N}\sum_{i=1}^{N} |y_i - x_i| \tag{3.8}$$

where $\frac{1}{N}\sum_{i=1}^{N} d_i$ represents the normalized distance between the ideal bitstring and the computed result. A fidelity value close to zero indicates large errors in the computation, while a value of 1 suggests that the VQE scheme reliably reconstructs the density matrix bitstring.

Our VQE solution performed reliably on simulated measurements, achieving a fidelity value of 1 with probabilty 0.75. The extracted bitstring is then reshaped into a 4×4 matrix, demonstrating that the algorithm can correctly identify which elements of the density matrix are present. The scheme was then extended to noisy experimental measurements obtained from the lab, as shown in Figure 3.5b. Even with real data, the method successfully identified the nonzero elements of the density matrix, maintaining a fidelity value of 1 with probabilty equal to 0.63. Additionally, we tested our approach on noisier experimental measurements (Fig. 3.5c), where the computation incorrectly predicted two bitstrings, leading to an overall fidelity of 0.88 with probabilty of 0.8. This suggests that while our method is robust under moderate noise, extreme noise levels can introduce inaccuracies in the reconstructed density matrix. In Fig 3.5, we have only labelled bitststrings with major counts contributions. The plots with fully labelled bitsrings can be found in Appendix D.

Potential errors in our approach arise from multiple factors, including ansatz limitations, hardware noise, and optimization challenges. The choice of ansatz may constrain the optimization process to a subspace that does not contain the true ground state, limiting the expressibility of the variational circuit and preventing convergence. Additionally, entangling operations introduce a more complex optimization

landscape with potential local minima, making it harder for classical optimizers to find the global minimum. The presence of noise in real quantum hardware further affects performance, as two-qubit gates are more error-prone than single-qubit operations, potentially leading to inaccurate cost function evaluations. Furthermore, shot noise from finite sampling and inaccuracies in the surrogate model could contribute to deviations from the expected results. To mitigate these issues, future work could explore alternative ansätze, improve noise mitigation techniques, and refine classical optimization strategies.

Having illustrated how quantum computing can be applied to a practical problem, using a variational algorithm for quantum state reconstruction. By formulating the tomography problem as a least squares optimization and solving it with a hybrid classical-quantum approach, we demonstrated the potential of VQAs in reconstructing entangled states. In the following chapter, we transition from quantum computing to optical computation, introducing the experimental toolbox necessary for optical matrix multiplication.

Chapter 4

Emulating quantum computing with Matrix Multiplication

This chapter is based on the following published work:

Koni, Mwezi, Hadrian Bezuidenhout, and Isaac Nape. "Emulating quantum computing with optical matrix multiplication." APL Photonics 9.10 (2024).

4.1 Introduction

In this chapter, we use optical matrix multiplication as a foundation for emulating quantum algorithms, therefore interacting quantum computing with structured light. We unveil the inherent tensor product structure of transverse spatial modes that is exploited in coherent optical matrix-vector multiplication and show that it can be harnessed as a tool to emulate quantum computing algorithms. Our logical basis is constructed from lattices of displaced Gaussian modes, eigenmodes of free space, that are locally modulated with digital holograms encoded on spatial light modulators. We treat the Cartesian location coordinates x and y of each mode as our degrees of freedom, forming the equivalent of two qudit registers. The x component is used to embed our state vectors, while the y-components acts as an ancilla that facilitates the unitary gate operator implementation. After a cylindrical lens integrates over the x-component, the final resulting state is measured in the y-coordinate. We test this using our basis and the Hadamard transform as example and demonstrate the Deutsch-Jozsa algorithm with our encoding scheme, showing it can be used to query balanced and constant function, with average fidelity above 90%. In our scheme, we

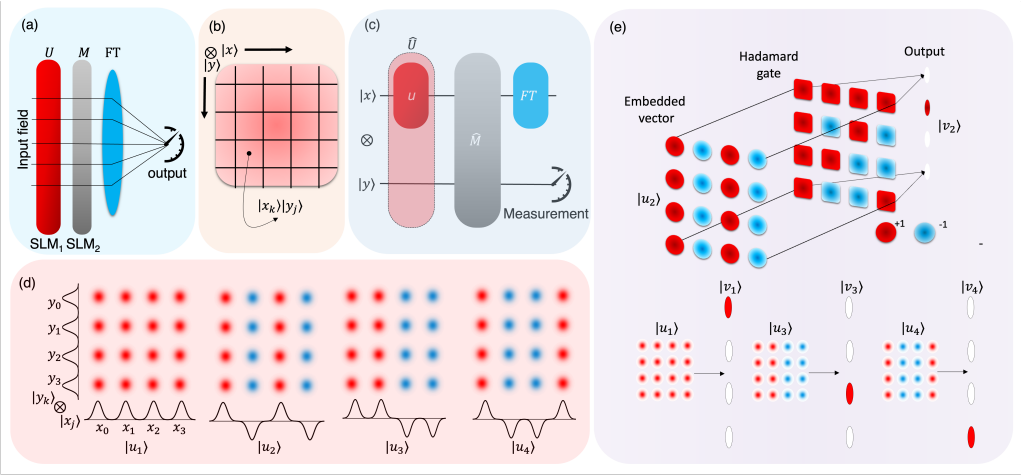


FIGURE 4.1: (a) Analogue of quantum computing using optical vector-matrix multiplication. SLM_1 encodes the lattice state via an operation, U, by copying the vector components encoded in the x-direction into the y-direction following Ref. [27]. The second SLM encodes the matrix operation M. After this a cylindrical lens focuses the field in x-direction to produce the final matrix-vector product at the output farfield plane. (b) The encodings performed by the spatial light modulators (SLMs) operate on the tensor product states $|x_k\rangle \otimes |y_j\rangle$ corresponding to the discrete position states that can be obtained by partitioning the Cartesian plane ($\mathbb{R}_2$) into non-overlapping segments. (c) A quantum circuit for performing matrix multiplication using our basis states $|x_k\rangle \otimes |y_j\rangle$, corresponding to the coordinates of the partitions. The operations are represented as operators that act on the x-coordinate and y-coordinate degrees of freedom. (d) An example of the Hadamard basis states, $|u_j\rangle$, encoded onto the lattice states for N=4. Here, the lattice is formed from propagation invariant Gaussian modes, each encoded into the independent partitions. To demonstrate that they form a tensor product space, $\mathcal{H}_N \otimes \mathcal{H}_N$, we show the individual states for the x and y coordinates, where the final lattice is obtained from their tensor products. These states are orthorgonal. (e) Example of one of the states being acted on by the Hadamard gate and then focused to perform the column wise summation. The embedded state vector is modulated, pointwise, with an SLM encoding the Hadamard matrix/gate. This is then focused in the farfield with a cylindrical lens where the zero order component of the pattern contains the result in the vertical direction. The outcomes are shown for other states in the Hadamard basis mapping from $|u_j\rangle$ to the $|v_j\rangle$.

exploit the inherent parallelism offered by optical matrix-vector multiplication, allowing to prepare multiple identical spatial modes as a superposition, and encode the oracle matrix as local phase transformations on each state.

4.2 Theory

4.2.1 Formulation using quantum states

In our approach, we emulate quantum computing by formulating optical vector-matrix multiplication in the language of quantum mechanics by representing the

optical fields that encode the vectors as states on Hilbert space and operations acting on the fields to represent the matrices that enact the equivalent of gate operations as shown in Fig. 4.1 (b) and (c). Firstly, the x-y coordinate system can be decomposed into a tensor-product space using the continuous degrees of freedom mapping the states $|x\rangle \in \mathcal{H}_\infty$ and $|y\rangle \in \mathcal{H}_\infty$ so that any field $\psi(x,y)$ is described by the state $|\psi\rangle = \int \psi(x,y)\,|x\rangle\,|y\rangle\,dxdy$ where the basis states satisfy $\langle x|x'\rangle = \delta(x-x')$ and $\langle y|y'\rangle = \delta(y-y')$. This means that each photon in a laser field is defined on a Hilbert space $\mathcal{H}_\infty \otimes \mathcal{H}_\infty$ spanned by the Cartesian coordinate internal degrees of freedom (i.e. the coordinate states). By partitioning the transverse plane into N bounded and non-overlap intervals , $|x\rangle \to |x_j\rangle$ and $|y\rangle \to |y_k\rangle$, shown in Fig. 4.1 (b), we can now define qudit states that form a high dimensional qudit space on the combined Hilbert space, i.e. $\mathcal{H}_N \otimes \mathcal{H}_N$ that has dimensions N^2. This has been discussed in recent work [84, 85], showing that non-overlapping regions of transverse optical fields can be expressed using a basis formed from tensor product states, much like distinct particles that occupy their own Hilbert spaces. This ensures that the spatial modes corresponding to the x and y coordinates function as independent degrees of freedom. From these states, a uniform superposition can be prepared as

$$|\Psi\rangle = \frac{1}{N}\sum_{jk} |x_k\rangle \otimes |y_j\rangle\,, \tag{4.1}$$

where each state in the partition has a non-zero coefficient. From this description, we see that the operations that can be applied on them can be enacted on each individual degree of freedom or on both simultaneously. Using our matrix vector product, we can show that using such states and treating the x and y degrees of freedom as analogues of qudit registers, that we can execute matrix vector (equivalently, operator and state) products. For instance, we can map the matrix U that was defined earlier onto an operator that encodes information into the lattice as,

$$\hat{U} = \sum_m u_m\,|x_m\rangle\,\langle x_m| \otimes \mathbb{I}, \tag{4.2}$$

this is the analogue for the matrix $\hat{V}$ in Eq. (2.29) - assuming that the initial state is the superposition $|\Psi\rangle$. This approach leverages multiple channels (Gaussian arrays) to carry independent streams of information simultaneously, reminescent of multiplexing approaches [61]. Here, the coefficients mark the x-components but copy

all the elements, u_m, across the y-components - this still leaves the y-components independent of the x-components. Similarly, for matrix M we have the operator

$$\hat{M} = \sum_{mn} M_{nm} |x_m\rangle |y_n\rangle \langle y_n| \langle x_m|, \tag{4.3}$$

which instead interacts with both components (registers), therefore encoding the matrix component's M_{nm} into each state $|x_m\rangle |y_n\rangle$ of the lattice. The combined operation $\hat{M}\hat{U}$ produces the state,

$$|\Psi\rangle = \frac{1}{N} \sum_{jk} M_{jk} u_k |x_k\rangle \otimes |y_j\rangle. \tag{4.4}$$

While the result of the multiplication is encoded into the lattice, which is $N \times N$ in dimensions, the result can be read out as a column in the y-direction once the summation k (in the x-component) is contracted. This can be done using the Fourier transforming (FT) lens in the x-direction - the analogue of a quantum Fourier transform, therefore completing the circuit in Fig. 4.1 (c). The result of this is an interference pattern, having a zero order component that corresponds to the amplitude, $\sum_{jk} e^{if_{x_l} x_k} M_{jk} v_k$ for the frequency component $f_{x_l} = 0$. Measuring the zeroth order interference pattern in x-direction integrates the x-component, therefore leaving the y-components in the state

$$|v\rangle = \frac{1}{N} \sum_{jk} M_{jk} u_k |y_j\rangle. \tag{4.5}$$

as expected. Next, measuring each y-components produces the probability amplitudes $v_j = \sum_k M_{jk} u_k$. Because this scheme allows us to express operations of the x-y coordinates analogous to gate based computing platforms, we will exploit it for quantum computing, where the matrix mechanics that describes quantum computing can be formulated using the transverse spatial modes in Cartesian coordinates. Next we introduce our basis of choice for doing this and illustrate the Hadamard gate.

4.3 Hadamard Gate

Before we introduce the Hadamard gate we first present our encoding basis. Instead of using arrays of square pixel like states for each bounded interval on the x-y plane,

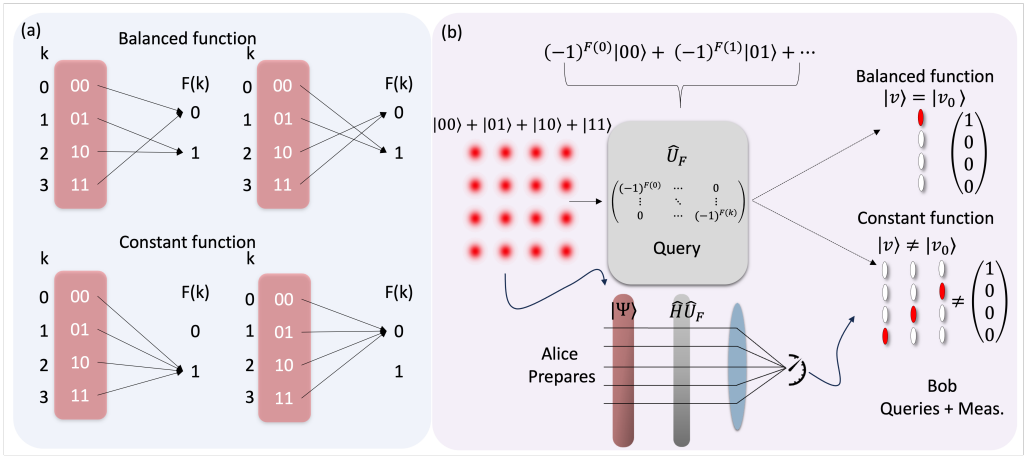


FIGURE 4.2: (a) Conceptual illustration of the balanced and constant functions involved in the Deutsch-Jozsa Algorithm for binary strings of length two, i.e. 00, 01, 10, 11. The two top figures showcase a balanced function which maps half of its inputs to 0 and the other half to 1 and the two bottom figure, a constant function that maps all its inputs to either 0 or to 1. (b) Illustration of our optical implementation of the Deutsch-Jozsa Algorithm. A lattice of Gaussian beams, representing the superposition of states, is simultaneously prepared by Alice. The state is then queried by Bob in parallel by the action of the compound oracle and hadarmard operations, which impart function evaluations spatially on the state through phase transformations and map us back into the computational basis. A constant function returns the state mapping onto the binary string 00, indicated by the vector lighting up on the first elements. A balanced function returns a state orthogonal to 00, shown by light lobes at different positions.

we encode arrays of Gaussian modes, as shown in Fig. 4.1(c). The illustrations show that our Gaussian lattices/arrays are encoded as tensor products of arrays of one dimensional Gaussian modes, where $(x_j, y_k) \rightarrow |x_j\rangle |y_k\rangle$ denote the centres of each Gaussian mode given by the states,

$$\begin{aligned} |x_j\rangle |y_k\rangle &\propto \int_{A_j} e^{-\frac{(x-x_j)^2}{w^2}} |x\rangle \, dx \otimes \int_{A_k} e^{-\frac{(y-y_k)^2}{w^2}} |y\rangle \, dy \\ &= \int_{A_j \times A_k} e^{-\frac{(x-x_j)^2+(y-y_k)^2}{w^2}} |x\rangle |y\rangle \, dxdy, \end{aligned} \tag{4.6}$$

where each mode is centred at coordinates (x_j, y_k) within a closed boundaries $A_j \times A_k \subset \mathbb{R}_2$ that are non overlapping. In this way, uniform superpositions such as in Eq. (4.1) can also be prepared, i.e. resulting in the field in the first panel of Fig. 4.1 (d). The rest of the fields in the figure represent instance of the Hadamard basis states, which include the uniform superposition. Next, we show how the Hadamard gate can be enacted on these fields.

The N-dimensional Hadamard gate, $\hat{H}_N$, is commonly used for quantum computing to map between the logical and the superposition basis (or vice-versa). For example, for two-dimensional states ($N = 2$), we can describe the logical basis as

$|v_0\rangle \equiv |0\rangle = (1,0)$ and $|v_1\rangle \equiv |1\rangle = (0,1)$,. In this basis the Hadamard gate is given by, $\hat{H}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. Applying the Hadamard gate maps the logical basis states onto the Hadamard basis states $|u_{1,2}\rangle = 1/\sqrt{2}(|0\rangle \pm |1\rangle)$, respectively. The states $|u_{1,2}\rangle$ are orthogonal and form a basis. The Hadamard gate performs a change of basis, however when applied twice it leaves the state unchanged, i.e $\hat{H}\hat{H} = \mathbb{I}$, true for all dimensions N. For qudits, $N > 2$, the Hadamard matrix can be computed for dimensions $N = 2^n$ (n power of 2) as $\hat{H}_N = \hat{H}^{\otimes n}$, using tensor products of n Hadamard gates. Thus, the k^{th} Hadamard basis state can be obtained by applying the Hadamard gate to the k^{th} state in the logical basis following,

$$\begin{aligned} \hat{H}_N |v_k\rangle &= |u_k\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} (-1)^{m \cdot k} |m\rangle , \end{aligned} \tag{4.7}$$

where the states $|m\rangle$ are the standard basis states with coefficients $(-1)^{m \cdot k}$ with $m \cdot k$ corresponding to the dot product of the binary representation of the integers m and k. Accordingly, the Hadamard basis states can be mapped onto our lattice states as

$$|u_k\rangle = \frac{1}{N} \sum_{mn=0}^{N-1} (-1)^{m \cdot k} |x_m\rangle |y_n\rangle . \tag{4.8}$$

Figure 4.1 (d) shows examples of the lattice states that have been encoded with the Hadamard basis states in $N = 4$ dimensions. To encode these coefficients into the lattice, we use the operator $\hat{U}$ in Eq. 4.2, but with the coefficients corresponding to our Hadamard basis states. Applying the Hadamard gate to each of the states in the lattice should map them back to the logical basis states, following $\hat{H}_N |u_k\rangle = |v_k\rangle$. Each basis state $|u_k\rangle$ that is embedded in the lattice (as in Fig. 4.1 (d)) can be mapped onto a unique element vector with one non-zero entry, as shown in Fig. 4.1 (e). To encode the Hadamard gate, we can map it's matrix components onto the lattice, using Eq. (4.3) where M_{jk} will represent the matrix elements of the Hadamard gate. Thereafter, we apply the Fourier transform and the zero order component filtering to complete the operator-state (matrix-vector) multiplication.

Next we show how this scheme can be used to perform a well known quantum algorithm.

4.3.1 Application: Deutsch-Jozsa algorithm

The Deutsch-Jozsa algorithm [86] provides an efficient quantum solution for determining whether a boolean function, $f(\cdot)$, that takes binary values as inputs, is balanced (half of the inputs map to 1 and the other half to 0) or constant (all inputs either map to 0 or 1). To illustrate this, consider a collection of 2-bit input strings chosen from the set $\mathcal{B} = \{00,01,10,11\}$; each element $k \in \mathcal{B}$ can be used an input to the function, i.e. $F(k)$. A balanced function would return 0 for half the entries, e.g., $k \in \{00,11\}$, and 1 for the other half, $k \in \{01,10\}$, as shown in the first left panel of Fig. 4.2 (a). The second example (top right panel of Fig. 4.2 (a)) shows an alternative combination that can also represent a balanced function. For the constant case, all binary values map to a single output, i.e., $\{00,01,10,11\} \rightarrow 0$ or $\{00,01,10,11\} \rightarrow 1$ as shown in the bottom panel of Fig. 4.2 (b) for two instances of a such a function.

To query the nature of such a function, the quantum computer prepares a superposition of all possible binary values in $\mathcal{B}$, queries the function values simultaneously, and through interference it returns a result indicating whether the function is balanced or not. We adapt the algorithm to our encoding scheme using the lattice basis and our operator-state (matrix-vector) multiplication scheme.

We illustrate the concept using four-dimensional (N=4) states as an example. We can map the binary values of the inputs to the function $F(\cdot)$ in the set $\mathcal{B}$ to the logical basis states

$$
\begin{aligned}
|0\rangle \rightarrow |00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |1\rangle \rightarrow |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \\
|2\rangle \rightarrow |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, |3\rangle \rightarrow |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.
\end{aligned} \tag{4.9}
$$

Using this basis, we can prepare all the possible inputs as the un-normalised superposition state $|0\rangle + |1\rangle + |2\rangle + |3\rangle$ where the binary values are mapped onto their integer representations. These correspond to the logical basis states $|v_k\rangle$ from

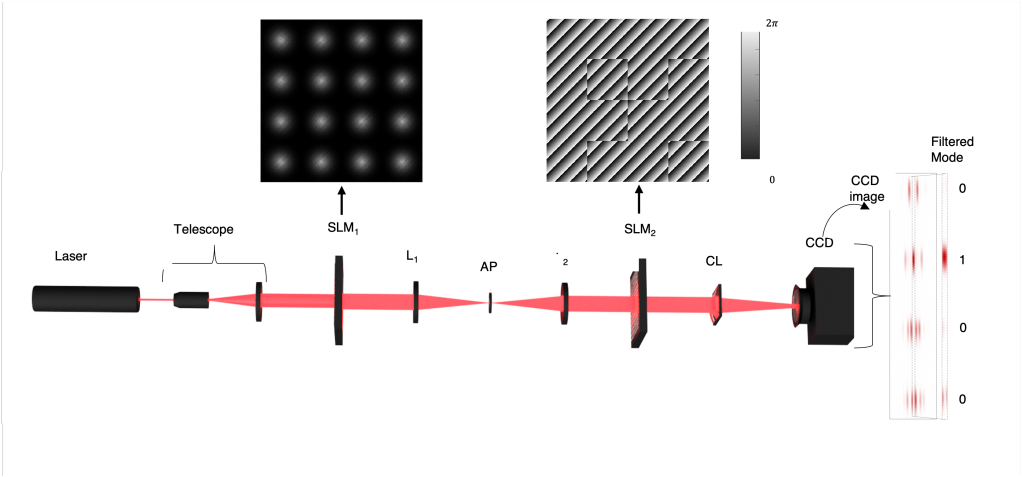


FIGURE 4.3: A laser beam is expanded using a telescope to overfill the active area of the first Spatial Light Modulator (SLM 1). At this point, the beam is transformed to encode the input vector as repeated rows in a matrix of Gaussian beams. This output is then relayed to SLM 2 using the 4f system of lenses (L_2 and L_1), where the lattice undergoes an additional phase encoding. An aperture (AP) at the focal plane of lens L_1 filters out the first-order diffraction pattern. Subsequently, the beam is propagated through a cylindrical lens (CL) and undergoes a one-dimensional Fourier transform in the x-coordinate. The final outcome is captured by a CCD camera, with the result of the multiplication found on the central fringe, which is filtered out (as shown in the inset). The filtered mode image on the CCD camera displays the result of the matrix-vector multiplication, where regions of no intensity correspond to zero elements, and bright regions correspond to non-zero elements. The holograms encoded on each SLM are shown as grayscale images.

the previous section. In the algorithm, we will see that the mappings of the gates convert between the Hadamard basis states ($|u_k\rangle$) and the logical basis states ($|v_k\rangle$). To prepare the uniform superposition ($|0\rangle + |1\rangle + |2\rangle + |3\rangle$) using our lattice, we can use the first element of the Hadamard basis following Eq. (4.8)

$$|u_0\rangle = \frac{1}{N}\sum_{jk=0}^{N-1} |x_k\rangle \otimes |y_j\rangle . \tag{4.10}$$

This indeed encodes the uniform superposition into our lattice as shown in Fig. 4.2 (b). Next, the features of the function $F(k)$, are encoded onto the unitary operator as U_F so that the states in the superposition become $\sum_k (-1)^{F(k)} |k\rangle$, showing that U_F is diagonal, and has the form

$$U_F = \begin{pmatrix} (-1)^{F(0)} & 0 & \dots & 0 \\ 0 & (-1)^{F(1)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (-1)^{F(N-1)} \end{pmatrix}, \tag{4.11}$$

where each state in the superposition is marked with a coefficient $(-1)^{F(k)}$. As such, we see that U_F encodes a π phases depending on whether $F(k)$ is 0 or 1. Motivated by this, we encode the unitary as $\hat{U}_F = \sum_k (U_F)_{kk} |x_k\rangle \langle x_k| \otimes \mathbb{I}$ onto our lattice, where $(U_F)_{kk}$ are the diagonal elements of U_F , marking the x-components of the lattice with the desired phases. In this way we do not need ancillary particles thanks to the higher dimensional nature of our degrees of freedom.

Accordingly, given the uniform superposition of the lattice, we obtain

$$\hat{U}_F |u_0\rangle = \frac{1}{N}((-1)^{F(0)} |x_0\rangle |y_0\rangle + (-1)^{F(1)} |x_1\rangle |y_0\rangle + ..) \tag{4.12}$$

$$= \frac{1}{N} \sum_{jk=0}^{N-1} (-1)^{F(k)} |x_k\rangle |y_j\rangle . \tag{4.13}$$

As a result, we can express the state as a piece wise function

$$\hat{U}_F |u_0\rangle = \begin{cases} \pm |u_0\rangle , & \text{Constant} \\ \frac{1}{N} \sum_{jk} (-1)^{F(k)} |x_k\rangle |y_j\rangle & \text{Balanced,} \end{cases} \tag{4.14}$$

showing that the state remains unchanged if the function is constant, while it can be in an arbitrary superposition if it is balanced. Finally, encoding the Hadamard matrix and applying the cylindrical lens and performing the filtering leaves the y-coordinate in the state

$$|v\rangle = \begin{cases} \pm |y_0\rangle , & \text{Constant} \\ \frac{1}{\sqrt{N}} \sum_{j \neq 0} u_j |y_j\rangle & \text{Balanced,} \end{cases} \tag{4.15}$$

illustrated graphically in Fig. 4.2 (b), showing the expected output intensities mapped onto the vertical rows of the output field. For the constant case we have that the algorithm returns the first element of the basis , $|v_0\rangle \equiv |y_0\rangle$. In the N=4 binary example, this is similar to obtaining the $|v_0\rangle \equiv |00\rangle$ state. The balanced case returns an arbitrary superposition of the logical basis states with coefficients given by $u_j = \frac{1}{\sqrt{N}} \sum_m (-1)^{m \cdot j + F(j)}$ where $m \cdot j$ is the binary inner product using the binary representation of m and j. For example, for a balanced function that maps

as $\{00,11\} \rightarrow 1$ and $\{01,10\} \rightarrow 0$, corresponds to a unitary

$$U_F = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \tag{4.16}$$

represented in the logical basis. The output for this example is $-|v_3\rangle \equiv -|11\rangle$. In general, the output state is one of the logical basis states that exclude, $|v_0\rangle \equiv |00\rangle$ as shown in Fig. 4.2 (b). In our setup, we execute the protocol by encoding $|u_0\rangle$ on the first SLM and then subsequently encode $\hat{M} = \hat{H}_N\hat{U}$ onto the second SLM.

4.4 Experiment

In Fig. 4.3 we illustrate the setup used for our demonstration. To ensure a uniform field intensity across the transverse plane, we overfilled the first spatial light modulator (SLM_1) by magnifying a laser beam from a Helium Neon (HeNe) source with a wavelength of 633 nm, using the telescope. On SLM_1, the input vector for the lattice of Gaussian modes was encoded as an array of displaced Gaussian beams, producing the superposition state, $|\Psi\rangle \equiv |u_0\rangle$ from Eq. (4.1). To alter the coefficients of the lattice, each row of the matrix (U) mapping the vector was encoded using a digital hologram that encodes amplitudes and phases [77] onto each of the Gaussian modes. Because the SLM encodes the vector state as a matrix, U, given any state $|u\rangle$, the elements were copied into its rows following the procedure outlined in the theory.

The matrix generated on SLM_1 was then transferred to SLM_2 using an imaging system consisting of lenses L_1 and L_2. An aperture (AP) was placed between the two lenses to filter the modulated first order diffraction pattern from SLM_1. On SLM_2, we applied an additional phase encoding to the lattice/matrix elements, thereby completing the optical representation of the matrix-vector multiplication process. Following the encoding on SLM_2, the beam was passed through a cylindrical lens (CL) to undergo a one-dimensional Fourier transform. This operation focused the outcome of the matrix-vector multiplication into the central fringe (shown as an inset in Fig. 4.3), specifically at the zeroth order Fourier component in coordinate space.

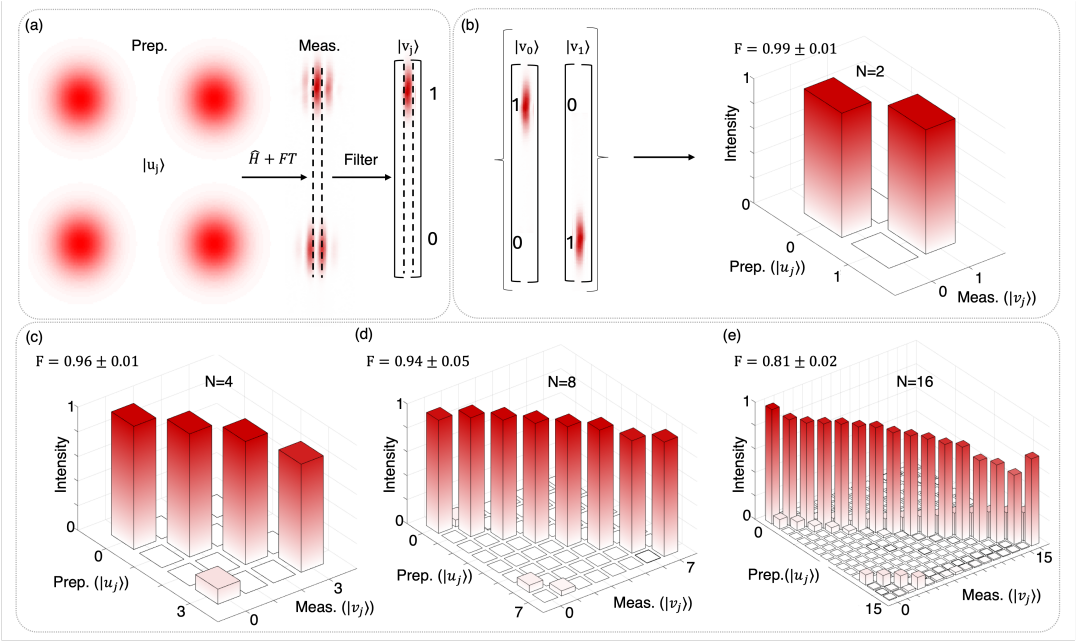


FIGURE 4.4: Setup characterization: Intensity images collected by the CCD camera when encoding a Hadamard matrix on the second SLM and a vector built from Hadamard columns on the first SLM. (a) Extraction of the central lobe containing our solution vector from the multiplication of a 2×2 Hadamard matrix with its first column. (b) Stacking these to form a diagonal matrix consisting of the light fringes as our elements. This diagonal matrix is then used to form a crosstalk matrix, from which the fidelity of the computation is computed. (c-e) Extension of this process to higher-dimensional matrices, up to 16D.

To extract this outcome accurately, we employed digital filtering, enabling precise capture of the resultant vector as shown in the inset of Fig. 4.3.

4.5 Results

4.5.1 Matrix-Multiplication

To characterise our system, we exploit the orthogonality of the Hadamard basis (this is presented in detail in Appendix B along with fidelity calculation procedure). In order to do this, we prepared the Hadamard basis states $|u_j\rangle$ on SLM_1 and encoded the Hadamard transform on SLM_2. Thereafter, we propagated the field through the cylindrical lens and collected the resulting interference pattern on a CCD camera. The multiplication between the Hadamard basis states and the Hadamard gate enable us to clearly distinguish between different columns of the output vector state. Accordingly, we obtain orthogonal states ($|v_j\rangle$) with distinct positions marked by bright lobes. We show a measured example of this for $N = 2$ dimensions in Fig. 4.4 (a) where the first basis state, $|u_0\rangle$, is encoded on the lattice and measured using

the Hadamard gate. The outcome is a measured interference pattern, where the resulting filtered zero order produces a bright lobe in the first entry - for this specific example, the result corresponds to the state $|v_0\rangle$. Summing and normalising the intensity for each region of the output vector, corresponded to extracting the output vector elements.

We repeated the process for all the basis states for a given dimension. In Fig. 4.4 (b) we show the measured lobes for all the states in $N = 2$, confirming that the Hadamard basis states ($|u_0\rangle$ and $|u_1\rangle$) are mapped on to the basis states $|v_0\rangle$ and $|v_1\rangle$. From the measured lobes, we extracted the intensity from each component and produce a crosstalk matrix (C.M). For $N = 2$. The cross talk matrix resembles an identity matrix showing that the measure basis modes are distinguishable and therefore confirms orthogonality between the measured vectors. To quantify this, we measured a fidelity of F = 0.99 ± 0.01, showing that the Hadamard matrix can map the basis elements $|u_j\rangle$ onto the basis states $|v_j\rangle$. This was calculated by normalising each row, and computing the average over the diagonal components.

Figures. 4.4 (c)-(e) show the extension of this process to higher-dimensional matrices. For dimensions $N = 4$, $N = 8$ and $N = 16$ we obtain average fidelities of 0.96 ± 0.01, 0.94 ± 0.05 and 0.81 ± 0.02 multiplication, respectively. However, as the dimensions increase, we note that the fidelity decreases. We attribute this to the diffraction of Gaussian beams. Increasing the dimension or number of Gaussian modes for the same physical size of the matrix means decreasing the beam waist w_0 of each mode. As the beam waists get smaller, they expand at a faster rate during propagation, according to $w(z) = w_0\sqrt{1 + \left(\frac{\lambda z}{\pi w_0^2}\right)^2}$. Given that the cylindrical lens focuses light in one direction while neglecting the other, we expect the unfocused direction to continue expanding, more rapidly as w_0 gets smaller, resulting in overlapping vector elements which are hard to distinguish. This diffraction effect impacts the fidelity of higher-dimensional computations.

4.5.2 Deutsch-Jozsa Algorithm demonstration

Using our scheme, we demonstrate the Deutsch-Jozsa algorithm for a function that encodes information using $N = 2$, $N = 4$ and $N = 8$ basis elements, shown in Fig. 4.5 (a)-(c), respectively. We initialized our system in the superposition state $|u_0\rangle$ by digitally displaying a hologram encoding the uniform lattice of Gaussian modes on SLM_1. The action of the oracle phase transformation, and the subsequent Hadamard

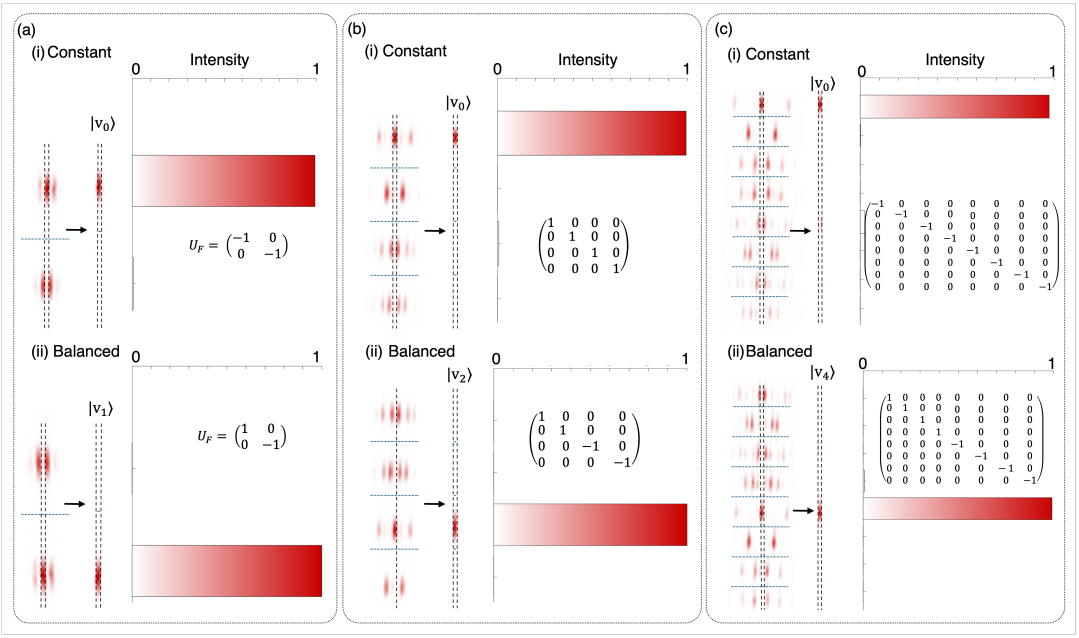


FIGURE 4.5: Deutsch-Jozsa algorithm results: Algorithm results for the Deutsch-Jozsa algorithm implementation. Panels (a), (b), and (c) show the intensity distributions for single-qubit, two-qubit, and three-qubit systems, respectively, with the form of the oracle diagonal matrix U_F. In each panel: (1) Unfiltered images representing the initial light distribution, and (2) Filtered intensity images corresponding to the central fringe at $k = 0$. Subpanels (i) correspond to the constant function, and subpanels (ii) correspond to the balanced function. The bar graphs depict the normalized intensity against the row position, allowing the determination of the function type. The presence or absence of light in the first vector element's spatial position indicates whether the function is balanced or constant.

gate are encoded as a compound unitary transformation $H\hat{U}_F$ on the second (SLM_2), the light beam is then sent through a cylindrical lens which acts as a summing operator. The resulting interference pattern that was measured with the CCD camera and filtered zero order vector components are shown as insets in each figure.

From these images, we then plot the cross-section of the normalized intensity against the vector element number. In this case, we can distinguish whether the function is balanced or constant by interrogating the presence or absence of light in the spatial position of the vector element. Using our basis vectors from the characterisation, this means a constant function produces the vector $|v_0\rangle$ and if the function is balanced $|v_{j\neq 0}\rangle$. The instance of the oracle operator ($\hat{U}_F$) used for each type of function is shown in each figure; this is shown as an inset in the intensity plots representing the measured components.

For each case, we accurately determined whether the function encoded in the unitary operation is constant or balanced. The average fidelity of these measurements exceeded 90% across all cases. Specifically, for N=2, 4, and 8, we achieved fidelities of 0.99 ± 0.01, 0.97 ± 0.01, and 0.93 ± 0.05, respectively.

4.6 Discussion and Conclusion

In this study, we utilized optical matrix-vector multiplication to emulate quantum computing by treating the x-y components of a coherent field as a tensor product space, $\mathcal{H}_N \otimes \mathcal{H}_N$, derived from individual coordinates. The encoding basis was defined within this Hilbert space using a lattice of Gaussian modes, where the positions of the centers in the x-y coordinates established localized states. Vector states were encoded along the x-coordinate, while both coordinates were employed to encode the matrix operator. The output vector was measured along the y-coordinate after filtering out the zero-order component of the output interference pattern. We demonstrated the use of Hadamard basis elements and the Hadamard gate, achieving encoding and decoding fidelities of high as 95% for $N = 8$ dimensions and above 80% for $N = 16$.

Further, our implementation of the Deutsch-Jozsa algorithm using optical vector matrix multipliers leverages the representation of quantum states as vectors and quantum operators as matrices that can be imprinted on our lattice basis. Optically, we prepare multiple identical Gaussian modes in parallel, with each mode representing an element of the states that are inputs to the function $F(k)$. The high-dimensional nature of our encoding scheme enables the oracle function in our demonstration to be encoded as a diagonal operator.

However, we note spatial resources required to perform computations using structured light in higher dimensions are fundamentally limited by the diffraction limit. The minimum separation between distinguishable spatial modes is constrained by the wavelength of light and the numerical aperture of the optical system, leading to crosstalk between modes when attempting to scale up to higher dimensions. In our current setup, we have explored up to 16 distinguishable spatial modes, equivalent to a 4-qubit system. J. Spall's vector-matrix demonstration reached up to ≈ 50 dimensions of encoding - in principle, every pixel of the SLM can be utalised as an encoding channel but would require much effort. Moreover, as the number of spatial modes increases, the complexity of addressing and manipulating these modes grows significantly. In quantum systems, the number of operations required for a computation scales polynomially with the number of qubits, whereas in classical systems, the need to individually resolve and address each spatial mode results in a linear scaling with the number of available modes. This ultimately limits the scalability of

classical systems in performing quantum-like operations .

Additionally the preparation of a Gaussian lattices and their unitary transformation matrices on an SLM typically involves NxN operations to generate the necessary hologram for the spot array, which can be computationally expensive and inefficient when scaling to higher dimensions. An alternative approach is the use of fan-out operations to generate the lattice of Gaussian modes [87, 88]. Techniques such as cylindrical lenses [89] or diffractive optical elements [90] can directly produce the desired lattice by splitting or reshaping the input beam, reducing the need for pixel-by-pixel modulation on the SLM and significantly improving efficiency. Recent advances in trained diffractive optical elements provide another potential solution by allowing computationally designed optical masks to perform the necessary optical transformations [34]. This approach leverages inverse design techniques to optimize optical circuits, embedding high-dimensional transformations in complex media without requiring precise control over individual elements. Such masks streamline the process by reducing the need for real-time computation and re-calibration, ensuring scalability and maintaining the accuracy of the transformations.

We acknowledge the proposal by Perez-Garcia et al. [40], which employs classical light following Dragoman's method. However, this approach necessitates multiple optical elements to prepare the superposition state. In contrast, our method efficiently prepares the state in a single step by projecting light onto a spatial light modulator (SLM) in a reprogrammable manner - the digital encoding allows for dynamic control and can therefore be adapted for any operation. Therefore, we anticipate that any transformation (operator/gate) can be encoded, including X-gate, Z-gate, etc., because the method allows any matrix operation to be encoded into the lattice.

Chapter 5

Conclusion and Future Work

This thesis has demonstrated that classical optical systems can be harnessed to emulate quantum computing operations through the novel application of optical vector-matrix multiplication (OVMM). By integrating both theoretical formulations and experimental methodologies, we have bridged the gap between traditional quantum computation concepts and the rich potential offered by structured light.

Summary of Contributions

- **Chapter 1: Foundations and Optical Degrees of Freedom**
  We began by reviewing a robust framework for quantum computing, covering essential concepts such as quantum bits, multi-qubit systems, and quantum gate operations within the circuit model. This chapter also reviewed the various degrees of freedom available in light—polarization, path, transverse spatial modes, and temporal bins—which can be effectively utilized for information encoding and manipulation in optical systems.

- **Chapter 2: Experimental Techniques and Optical Toolbox**
  Building on the theoretical groundwork, Chapter 3 detailed the experimental techniques employed in this work. We described the design and operation of spatial light modulators (SLMs), the spatial Fourier-transforming properties of lenses, and hologram construction methods to implement vector-matrix multiplication. This chapter concluded by introducing a practical characterization technique through the implementation of the Hadamard gate, thereby validating the experimental approach.

- **Chapter 3: Learning to Solve a Useful Problem on a Quantum Computer: Reconstructing Quantum States**

In this chapter, we demonstrated how current quantum hardware can be leveraged to solve systems of linear equations that are central to Quantum State Tomography. By employing variational quantum algorithms, we successfully reconstructed density matrices for biphoton entangled states produced from SPDC.

- **Chapter 4: Emulating Quantum Computation via Matrix Multiplication**
  In the final chapter, we integrated the theoretical and experimental insights to demonstrate how optical matrix multiplication can be used to emulate quantum computing operations. We provided both a theoretical formulation and an experimental realization of quantum gate operations—specifically the Hadamard gate—and illustrated its application through the Deutsch-Jozsa algorithm. This chapter underscores the potential of OVMM as a platform for simulating quantum processes using classical optical systems.

Implications and Future Directions

We have demonstrated that optical vector-matrix multiplication (OVMM) can emulate aspects of quantum computing by utilizing the x- and y-components of coherent fields as a tensor product space, $\mathcal{H}_N \otimes \mathcal{H}_N$. Our experiments achieved encoding/decoding fidelities of 95% for $N = 8$ and over 80% for $N = 16$, showcasing the potential of OVMM. However, challenges related to scalability, universality, and encoding efficiency remain. The following refined questions and discussion points provide a roadmap for formalizing and advancing OVMM as a computational tool.

5.1 Universality

A key question is whether OVMM can approximate arbitrary unitary transformations. Current linear optical methods inherently lack the nonlinearity required for universal quantum computation, limiting their ability to directly realize a universal set of operations. However, by integrating measurement feedback—as employed in Linear Optical Quantum Computing (LOQC)—or by utilizing nonlinear media (e.g., materials with Kerr effects), [91] it may be possible to overcome these limitations and bridge the gap towards universality. Recently, there has been growing interest in ferroelectric nematic nonlinear crystals, which exhibit strong nonlinear

interactions [92, 93]. Spatial Light Modulators (SLMs) made from these materials could potentially revolutionize optical systems, offering new avenues for achieving the necessary nonlinearity and significantly changing the landscape of OVMM and optical quantum computing.

5.2 Encoding Efficiency

While OVMM offers a powerful approach for emulating quantum computation, the efficiency of state encoding on Spatial Light Modulators (SLMs) remains a critical concern. Encoding quantum states on an SLM requires $N \times N$ operations, which introduces significant computational overhead, particularly for high-dimensional states. To address this challenge, future research should focus on improving encoding efficiency through several strategies. One approach involves using fan-out beam splitters, such as cylindrical lenses, to generate Gaussian lattices without the need for pixel-by-pixel encoding, thereby reducing computational cost. Another promising direction is the use of inverse-designed diffractive optical elements (DOEs), which can implement fixed unitary transformations with minimal computational overhead, further enhancing the scalability and efficiency of OVMM.

5.3 Scalability

Our experiments demonstrated OVMM with $N = 16$ spatial modes. However, scaling to a larger number of modes presents significant challenges, primarily due to diffraction and mode crosstalk. The minimum resolvable spatial mode size is inherently limited by both the wavelength of the light and the numerical aperture of the optical system. To address these limitations, future work should explore strategies to mitigate diffraction and improve mode isolation. While classical OVMM provides a powerful platform for high-dimensional state manipulation, it lacks the quantum entanglement that is essential for certain quantum computational advantages. To harness the exponential scaling potential of quantum systems, future research should investigate single-photon OVMM architectures, which could pave the way for achieving true quantum-enhanced capabilities in optical computation.

5.4 Query Complexity and Parallelization

5.4.1 Standard Query Model

In the standard query model, a function $f : \{0,1\}^n \to \{0,1\}$ is treated as an opaque black box. The objective is to determine a property of f with the minimal number of queries. For classical queries, the worst-case complexity scales exponentially with n, i.e.,

$$Q_C(n) = \Omega(2^n).$$

Quantum algorithms, by contrast, leverage superposition and interference to achieve dramatic reductions in query complexity. For example, the Deutsch-Jozsa problem—determining whether a function is constant or balanced—requires $2^{n-1} + 1$ queries classically, yet a quantum algorithm accomplishes this with a single query:

$$Q_Q(n) = O(1).$$

Here, the quantum oracle is implemented by a unitary U_f acting as

$$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle ,$$

and, through phase kickback, it can be reformulated as a phase oracle:

$$U_f |x\rangle = (-1)^{f(x)} |x\rangle .$$

5.4.2 OVMM as a Parallel Query Model

Our OVMM framework similarly encodes the evaluation of $f(x)$ into structured light. By mapping all inputs x into a single optical state $|\psi\rangle$, OVMM enables parallel evaluation:

$$|\psi'\rangle = M_f|\psi\rangle,$$

where M_f is an optical matrix that implements the function f. This operation occurs simultaneously over the entire input space, suggesting that for certain tasks the effective query complexity could be significantly reduced. In analogy to quantum computing, we hypothesize that:

$$Q_{\text{OVMM}}(n) \approx O(1),$$

for functions that are particularly amenable to optical parallelism. Formalizing this hypothesis, deriving analytical bounds, and validating them experimentally represents an exciting direction for future work.

This thesis has laid the groundwork for using OVMM as a viable computational tool. While challenges remain in scalability, universality, and encoding efficiency, the demonstrated ability to emulate quantum operations using classical optics is a promising step towards future high-dimensional and potentially quantum-enhanced optical computing.

Appendix A

Introduction to Quantum State Tomography

In this section an introductory discussion of quantum state tomography is provided. The tomography problem is first formulated and then a list of well known techniques for solving the problem are listed. This section is relvant for understanding the technique we introduce in Chapter 2. The objective of QST is to reconstruct a quantum state from a set of tomographically complete measurements—those that provide sufficient data to fully infer the state of a quantum system. In this context, we consider a two-photon qubit system represented by the density matrix ρ. The measurement observables chosen are the Pauli operators, which form a complete basis for single-qubit states. By combining these operators, we can obtain the projections necessary for tomographic reconstruction.

A.1 Formulating the Quantum State Estimation Problem

To reconstruct the density matrix ρ for a biphoton state in a Hilbert space of dimension d, we express it as a linear combination of an orthonormal basis of traceless Hermitian operators $\{B_i\}_{i=1}^{d^2-1}$:

$$\rho = \frac{1}{d} I + \sum_{i=1}^{d^2-1} q_i B_i, \tag{A.1}$$

where I is the identity operator, and q_i are real coefficients that correspond to the expectation values of the observables B_i. This representation allows us to fully describe the quantum state using basis operators, such as the Pauli matrices for qubits

or their generalizations for higher dimensions. It is important to note that the coefficients q_i must be constrained to ensure that ρ remains a valid density matrix (i.e., it is Hermitian and positive semi-definite).

A.1.1 Measurement and Basis Selection Using Pauli Operators

For a qubit-based system, measurements are typically implemented using the Pauli operators σ_x, σ_y, and σ_z. These operators serve distinct roles in QST:

- **Computational Basis Measurement**: The operator σ_z corresponds to measurements in the computational basis $\{|0\rangle, |1\rangle\}$.
- **Superposition Basis Measurement**: The operators σ_x and σ_y project onto superposition states, such as $|+\rangle$ and $|-\rangle$, as well as complex phase states like $\frac{|0\rangle+i|1\rangle}{\sqrt{2}}$. These superpositions are crucial for capturing quantum coherence and entanglement.

The probability of obtaining a measurement outcome corresponding to a particular projector P_j (constructed from Pauli operators) is governed by the Born rule:

$$p_j = \mathrm{Tr}(\rho P_j). \tag{A.2}$$

This equation shows how the density matrix ρ predicts the probability of observing a specific measurement outcome associated with the projector P_j. Each projector P_j can be expressed as:

$$P_j = \frac{1}{d} I + \sum_{i=1}^{d^2-1} p_{j,i} B_i, \tag{A.3}$$

which implies that the probability vector $\vec{p}$ associated with the outcomes of an informationally complete set of measurements relates to the coefficient vector $\vec{q}$ via a matrix X that encodes these projections:

$$\vec{p} = \frac{1}{d}\vec{1} + X\vec{q}, \tag{A.4}$$

where $X_{ji} = \mathrm{Tr}(B_i P_j)$.

Upon performing measurements yielding empirical frequencies $\hat{p}$, our objective is to solve for the vector $\vec{q}$ such that ρ best fits the observed data:

$$\hat{\vec{p}} \approx \frac{1}{d}\vec{1} + X\vec{q}. \tag{A.5}$$

Approaches to State Reconstruction

1. **Linear Inversion**: This direct approach involves solving the linear system for q using least squares fitting, while ensuring that the resulting ρ remains physical (i.e., positive semi-definite).

2. **Maximum Likelihood Estimation (MLE)**: This method maximizes the likelihood that ρ would produce the observed data, resulting in a physical ρ that is often more robust to noise.

3. **Bayesian Estimation**: A Bayesian approach updates a prior distribution on ρ with new data to produce a posterior distribution, effectively handling uncertainty in state estimation.

Appendix B

Characterisation Based on the Hadamard Gate/Matrix

To validate the performance of our optical matrix multiplication scheme in Chapter 4, we outline a characterization method based on the Hadamard matrix. This approach quantitatively assesses the orthogonality and fidelity of the implemented operation. Consider an $n \times n$ Hadamard matrix H with elements H_{ij} (with $i, j = 1, \ldots, n$):

$$H = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{1n} \\ H_{21} & H_{22} & \cdots & H_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ H_{n1} & H_{n2} & \cdots & H_{nn} \end{pmatrix}. \tag{B.1}$$

We extract the j-th column vector v from H as:

$$v = H_{\cdot j} = \begin{pmatrix} H_{1j} \\ H_{2j} \\ \vdots \\ H_{nj} \end{pmatrix}. \tag{B.2}$$

When performing the optical matrix multiplication, the i-th element of the output vector o is computed as:

$$o_i = \sum_{k=1}^{n} H_{ik} H_{kj}. \tag{B.3}$$

Because H is an orthogonal matrix (up to the normalization factor), it satisfies

$$HH^{\top} = nI, \tag{B.4}$$

which means that

$$\sum_{k=1}^{n} H_{ik} H_{jk} = n \delta_{ij}. \tag{B.5}$$

In our multiplication, we have

$$o_i = \sum_{k=1}^{n} H_{ik} H_{kj} = n \delta_{ij}, \tag{B.6}$$

so that the expected output vector is all zeros except at the $i = j$ component, where the value is n. We can write the multiplication explicitly as:

$$H \times v = \begin{pmatrix} \sum_{k=1}^{n} H_{1k} H_{kj} \\ \sum_{k=1}^{n} H_{2k} H_{kj} \\ \vdots \\ \sum_{k=1}^{n} H_{nk} H_{kj} \end{pmatrix} = \begin{pmatrix} n \delta_{1j} \\ n \delta_{2j} \\ \vdots \\ n \delta_{nj} \end{pmatrix}. \tag{B.7}$$

To quantify performance, we use a fidelity measure based on the overlap of output vectors. We sequentially send the column vectors v (each extracted from the columns of an n-dimensional Hadamard matrix) through the optical system. Let the resulting normalized output vectors be $\{o^{(j)}\}_{j=1}^{n}$. Ideally, the orthogonality of the Hadamard matrix ensures that

$$o^{(i)} \cdot o^{(j)} = \delta_{ij}. \tag{B.8}$$

We then define the generalized fidelity F as the complement of the average squared overlap between distinct output vectors:

$$F = 1 - \frac{1}{n(n-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \left| o^{(i)} \cdot o^{(j)} \right|^2. \tag{B.9}$$

In the ideal case, where all pairwise overlaps vanish for $i \neq j$, $F = 1$. This fidelity measure quantitatively assesses how well the optical system preserves the orthogonality of the encoded information.

Appendix C

Variational Quantum Algorithm code

A matlab script for executing the variational quantum eigensolver for quantum state reconstruction is attached below. This is the main implementation of the techique we introduce to solve a useful problem on a quantum computer in Chapter 2. It is based on the following psuedocode:

Algorithm 2 Variational Quantum Eigensolver (VQE) Algorithm

1: **Input:** Initial parameters θ
2: **while** not converged **do**
3: Prepare quantum circuit with parameters θ
4: Measure the expectation value $\langle H \rangle$
5: Update parameters θ to minimize $\langle H \rangle$
6: **Output:** Optimal parameters θ^* corresponding to the minimal expectation value and extracted bitstrings

```
% A part of this  code is part of MATLAB-Qiskit Runtime
   Primitives.
% (C) Copyright IBM 2023.
```

```
clc;
clear all;
```

```
close all;

%% Setup IBM Quantum cloud credentials

apiToken=""




%% Setup IBM Quantum Platform credentials
channel="ibm_quantum"
service=QiskitRuntimeService(channel,apiToken,[]);

%%Define backend and access

service.Start_session=true;
backend="ibmq_qasm_simulator"
service.hub="ibm-q-wits"
service.group="internal"
service.project="physics"
%%
service.program_id = "estimator";
service.Start_session = true;


%% Enable the session and Estimator
session = Session(service, backend);
estimator = Estimator(session=session);

%%
% Setup for Two-Qubit Quantum State Tomography Using VQE

%%%% this function generates the complete set of
   projections that are
```

```
%%%% needed for state tomography where the projectors are
   flattened
%%%% we assume this is done for two photons

close all
twophoton = "T";
dim=2;
numParticles = 2;

numMeasurements = 4*nchoosek(dim, 2)+dim; %%% number of
   measurements needed

T = ProduceProjectionsmat(dim, twophoton);
T=double(T);
TestState = reshape(fliplr(eye(dim)), [dim^numParticles,
   1]);

Testrho = kron(TestState, TestState');

rhoflat = reshape(Testrho, [(dim^numParticles)^2, 1]); %
   flattened density matrix
%% compute

Measurements = (T*rhoflat);
% M=Measurements;
% Input_exp=readmatrix('QST 2D.xlsx');
Measurement_exp=inter(Measurements);
M=Measurements(:);
[MeasurementsFor_Disp, DensityMat_For_Disp] =
   MatricesForplots(M, rhoflat, dim,numMeasurements);


%visualises the measurements
```

```
MeasurementsFor_Disp = reshape(M, [numMeasurements,
    numMeasurements]);
imagesc(abs(MeasurementsFor_Disp));colorbar

% densitymatrix_display
%DensityMat_For_Disp = reshape(rhoflat, [dim^2, dim^2])
% imagesc(DensityMat_For_Disp)

%%


%% 1. Mapping the problem to qubits/Quantum Hamiltonian

%%%% Here we use the QuadraticProgram class to convert into
     the Pauli terms
Linear    = -real(T'* M + (M' * T)');
Quadratic = T'* T;
constant  = M'* M;


Qubo = QuadraticProgram(Linear,Quadratic,constant);
[Pauli_terms,Pauli_Coeff, Offset_value] = Qubo.To_ising(
    Linear,Quadratic,constant);
hamiltonian.Pauli_Term = Pauli_terms;
hamiltonian.Coeffs = Pauli_Coeff;

%% 2. Choosing the ansatz circuit/s
circuit.reps=3;
circuit.entanglement = "pairwise";
circuit.number_qubits = strlength(Pauli_terms(1));
circuit.num_parameters = (circuit.reps+1)*circuit.
    number_qubits;

%% Arguments for the optimizer
```

```
arg.hamiltonian = hamiltonian;
arg.circuit     = circuit;
arg.estimator   = estimator;

%% Define the cost function
cost_func = @(theta) cost_function(theta,arg);

%% Set the parameters for MATLAB surrogateopt global
   optimizer
x0 = -5*ones(circuit.num_parameters,1);
max_iter =1500;

lower_bound = repmat(-2*pi,circuit.num_parameters,1);
upper_bound = repmat( 2*pi,circuit.num_parameters,1);

options = optimoptions("surrogateopt",...
"MaxFunctionEvaluations",max_iter, ...
"PlotFcn","optimplotfval",...
"InitialPoints",x0);


%% 3. Executing the quantum VQE algorithm using the qiskit
   Estimator primititve and MATLAB
% optimizer to find the ground state energy of H2 molecule

rng default %% For reproducibility
%
[angles,minEnergy] = surrogateopt(cost_func,lower_bound,
   upper_bound,options);


fprintf('Ground state energy: [ %s ]\n', minEnergy);
%%
```

```
ansatz = Twolocal(circuit, angles);

plot(ansatz)
sampler = Sampler(session=session);



job = sampler.run(ansatz,sampler.options.service);
results = sampler.Results(job.id);
%%% extract the bitstring
string_data = string(fieldnames(results.quasi_dists));
bitstring_data = replace(string_data,"x","");
%%% Extract the probabilitites
probabilities = cell2mat(struct2cell(results.quasi_dists));

%%% Find the bitstring with the highest probability
bitstring_maxprobability = string_data(find(probabilities==
   max(probabilities)));
fprintf('The quantum solution for vqe is: [ %s ]\n',
   bitstring_maxprobability);

%%%% plot the results and color the graph using the
   received bit-string
%%%% (solution)

%%
bitstrings = fieldnames(results.quasi_dists);
counts = struct2array(results.quasi_dists);


% Create the bar plot
bar(counts)
title('Bitstring Distribution')
```

```
xlabel('Bitstrings')
ylabel('Counts')

% Label the x-axis with bit strings
set(gca, 'XTick', 1:length(bitstrings), 'XTickLabel',
   bitstrings)
xtickangle(45)  % Rotate labels for clarity
%%
[Ary]=strconvert(bitstring_maxprobability)
%% Define the cost function to calculate the expectation
   value of the derived Hamiltonian
function [energy] = cost_function(parameters,arg)

% Construct the variational circuit
ansatz = Twolocal(arg.circuit, parameters);

estimator = arg.estimator;
job       = estimator.run(ansatz,arg.hamiltonian);

%%%% Retrieve the results back
results   = estimator.Results(job.id);
energy    = results.values;
end

%%
function [pauli_terms, coefficients] = compute_pauli_terms(
   T, M)
% Compute T^T T for convenience
TTT = T'*T;
TM=T'*M;
% Dimensions
N = size(T, 2);

% Coefficients and terms
```

```
coefficients = [];
pauli_terms = {};

% Calculate the J_{ij} coefficients and generate the
   associated Pauli terms
for i = 1:N
for j = i+1:N
J_ij = 0.25 * TTT(i,j);

if J_ij ~= 0
coefficients = [coefficients; J_ij];
term = generate_pauli_string(i, j, N);
pauli_terms{end+1} = term;
end
end
end

% Calculate the h_i coefficients and generate the
   associated Pauli terms
for i = 1:N
h_i = -0.5 * sum(TTT(i,:)) +TM(i);

if h_i ~= 0
coefficients = [coefficients; h_i];
term = generate_pauli_string(i, 0, N);  % 0 indicates a
   linear term
pauli_terms{end+1} = term;
end
end

% Add the offset term
offset = (0.25 * sum(sum(TTT)))+M'*M-sum(T'*M);
coefficients = [coefficients; offset];
```

```
pauli_terms{end+1} = generate_pauli_string(0, 0, N);  %
   Using a special case in generate_pauli_string to get '
   IIII...'
end

function term = generate_pauli_string(i, j, N)
term = repmat('I', 1, N);

if i == 0 && j == 0  % Offset term
return;
elseif j == 0  % Linear term
term(i) = 'Z';
else  % Quadratic term
term(i) = 'Z';
term(j) = 'Z';
end
end


%Tomography function
%%

function [Ary]=strconvert(bitstring_data)
sz=size(bitstring_data)
if sz(1)==1
sol=convertStringsToChars(bitstring_data)
Ary=zeros(length(sol)-1,1);
for k=1:length(Ary)
Ary(k)=str2num(sol(k+1));
end
l=sqrt(length(sol)-1);

imagesc(Ary)
else
```

```
for r=1:sz(1)
Res=bitstring_data(r,:);
sol=convertStringsToChars(Res)
Ary=zeros(length(sol)-1,1);
for k=1:length(Ary)
Ary(k)=str2num(sol(k+1));
end
l=sqrt(length(sol)-1);


subplot(sz(1),1,r)
imagesc(Ary)
end
end
end

function T = ProduceProjectionsmat(dim, twophoton)
Basis = eye( dim );

c=1;
for k = 1:dim
State = Basis(:, k);
Projjsinglephoton{c} = vpa(kron(State, conj(transpose(State
   ))));
c=c+1;
end

for k = 1:dim
j = k+1;
while j<=dim
for al =  [1, 1i, -1, -1i]
State =vpa((Basis(:, k) + al*Basis(:, j)) ./ (sqrt(2)));
Projjsinglephoton{c} =vpa(kron(State, State'));
c = c+1;
```

```
end
j = j+1;
end
end

numMeasureSinglePhoton = 4*nchoosek(dim, 2) + dim;

T = [];

if twophoton == "T"

for j = 1 : numMeasureSinglePhoton

ProjPhoton1 = Projjsinglephoton{j};

for k = 1 : numMeasureSinglePhoton
ProjPhoton2 = Projjsinglephoton{k};
Proj12 = vpa(kron(ProjPhoton1, ProjPhoton2));

T = [T, reshape(Proj12, [dim^4, 1])];
end
end
else
for j = 1 : numMeasureSinglePhoton
T= [T,reshape(Projjsinglephoton{j}, [dim^2, 1])];
end
end
T = T';
end

function [MeasurementsFor_Disp, DensityMat_For_Disp] =
   MatricesForplots(Measurements, rhoflat, dim,
   numMeasurements)
```

```
%visualises the measurements
MeasurementsFor_Disp = reshape(Measurements, [
   numMeasurements, \\ numMeasurements]);
%imagesc(MeasurementsFor_Disp)

% densitymatrix_display
DensityMat_For_Disp = reshape(rhoflat, [dim^2, dim^2]);
imagesc(DensityMat_For_Disp)
end
function C=inter(A)


max_val = max(A(:));
B=A-0.15*(max_val);
B(B<0)=0;
max_val = max(B(:));
min_val = min(B(:));
% Normalize the matrix to [0, 1]
C = (B - min_val) / (max_val - min_val);

end
```

Appendix D

Exprimental results with fully labelled bitstring distribution

In this section we provide fully labelled bitstring distributions one would expect preprocessing counts from the quantum computer. In chapter 2, we only label distributions with siginificant counts.

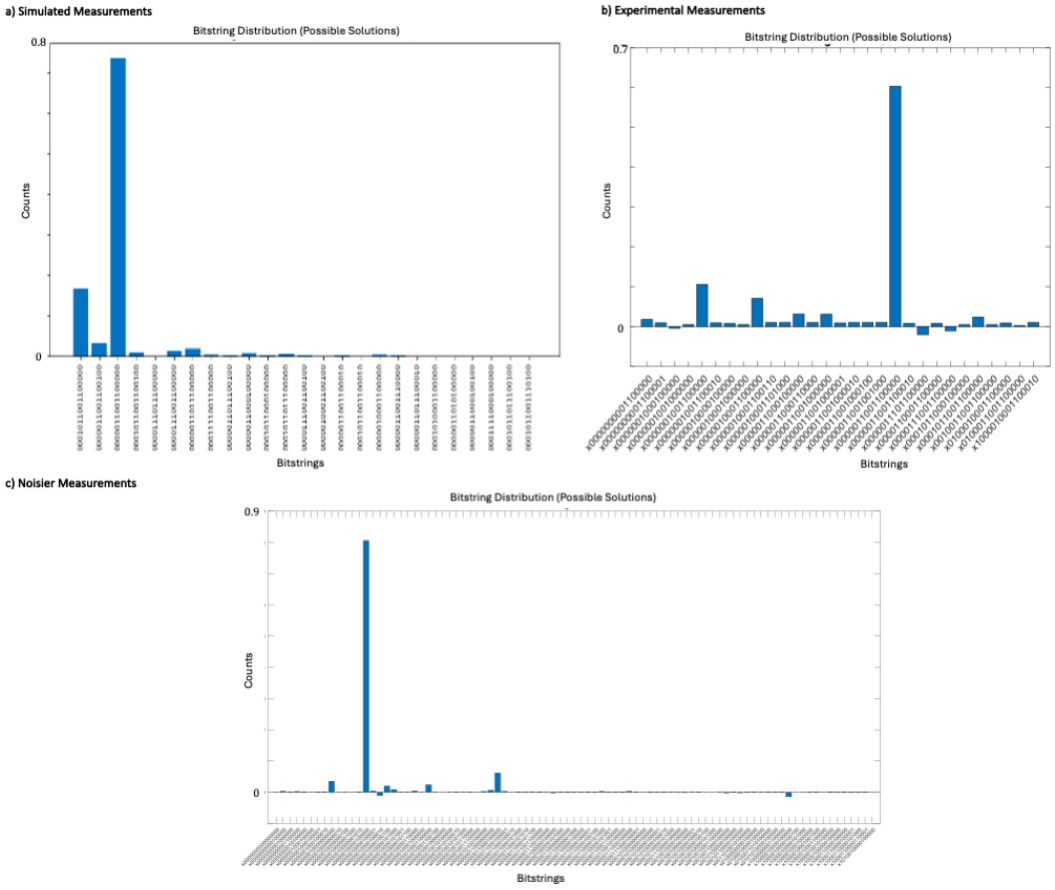


FIGURE D.1: Predicted bitstring distribution with all labelled measured bitstrings

Bibliography

[1] P. Cameron, B. Courme, C. Vernière, R. Pandya, D. Faccio, and H. Defienne, "Adaptive optical imaging with entangled photons," *Science*, vol. 383, no. 6687, pp. 1142–1148, 2024.

[2] C. Moodley and A. Forbes, "Advances in quantum imaging with machine intelligence," *Laser & Photonics Reviews*, p. 2300939, 2024.

[3] J. Geng, "Structured-light 3d surface imaging: a tutorial," *Advances in optics and photonics*, vol. 3, no. 2, pp. 128–160, 2011.

[4] B. I. Erkmen and J. H. Shapiro, "Ghost imaging: from quantum to classical to computational," *Advances in Optics and Photonics*, vol. 2, no. 4, pp. 405–450, 2010.

[5] F. Ferri, D. Magatti, L. Lugiato, and A. Gatti, "Differential ghost imaging," *Physical review letters*, vol. 104, no. 25, p. 253603, 2010.

[6] D. Zia, N. Dehghan, A. D'Errico, F. Sciarrino, and E. Karimi, "Interferometric imaging of amplitude and phase of spatial biphoton states," *Nature Photonics*, vol. 17, no. 11, pp. 1009–1016, 2023.

[7] G. Kim, Y. Kim, J. Yun, S.-W. Moon, S. Kim, J. Kim, J. Park, T. Badloe, I. Kim, and J. Rho, "Metasurface-driven full-space structured light for three-dimensional imaging," *Nature Communications*, vol. 13, no. 1, p. 5920, 2022.

[8] M. Mirhosseini, O. S. Magaña-Loaiza, M. N. O'Sullivan, B. Rodenburg, M. Malik, M. P. Lavery, M. J. Padgett, D. J. Gauthier, and R. W. Boyd, "High-dimensional quantum cryptography with twisted light," *New Journal of Physics*, vol. 17, no. 3, p. 033033, 2015.

[9] J. Wang, J.-Y. Yang, I. M. Fazal, N. Ahmed, Y. Yan, H. Huang, Y. Ren, Y. Yue, S. Dolinar, M. Tur, *et al.*, "Terabit free-space data transmission employing orbital angular momentum multiplexing," *Nature photonics*, vol. 6, no. 7, pp. 488–496, 2012.

[10] M. Krenn, R. Fickler, M. Fink, J. Handsteiner, M. Malik, T. Scheidl, R. Ursin, and A. Zeilinger, "Communication with spatially modulated light through turbulent air across vienna," *New Journal of Physics*, vol. 16, no. 11, p. 113028, 2014.

[11] J. Wang, J. Liu, S. Li, Y. Zhao, J. Du, and L. Zhu, "Orbital angular momentum and beyond in free-space optical communications," *Nanophotonics*, vol. 11, no. 4, pp. 645–680, 2022.

[12] A. Forbes, M. Youssef, S. Singh, I. Nape, and B. Ung, "Quantum cryptography with structured photons," *Applied Physics Letters*, vol. 124, no. 11, 2024.

[13] M. P. Lavery, C. Peuntinger, K. Günthner, P. Banzer, D. Elser, R. W. Boyd, M. J. Padgett, C. Marquardt, and G. Leuchs, "Free-space propagation of high-dimensional structured optical fields in an urban environment," *Science Advances*, vol. 3, no. 10, p. e1700552, 2017.

[14] B. Perez-Garcia, R. I. Hernandez-Aranda, A. Forbes, and T. Konrad, "The first iteration of Grover's algorithm using classical light with orbital angular momentum," *Journal of Modern Optics*, vol. 65, no. 16, pp. 1942–1948, 2018.

[15] 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.

[16] S. Konno, W. Asavanant, F. Hanamura, H. Nagayoshi, K. Fukui, A. Sakaguchi, R. Ide, F. China, M. Yabuno, S. Miki, *et al.*, "Logical states for fault-tolerant quantum computation with propagating light," *Science*, vol. 383, no. 6680, pp. 289–293, 2024.

[17] P. L. McMahon, "The physics of optical computing," *Nature Reviews Physics*, vol. 5, no. 12, pp. 717–734, 2023.

[18] I. Nape, B. Sephton, P. Ornelas, C. Moodley, and A. Forbes, "Quantum structured light in high dimensions," *APL Photonics*, vol. 8, no. 5, 2023.

[19] A. Forbes, M. de Oliveira, and M. R. Dennis, "Structured light," *Nature Photonics*, vol. 15, no. 4, pp. 253–262, 2021.

[20] H. Rubinsztein-Dunlop, A. Forbes, M. V. Berry, M. R. Dennis, D. L. Andrews, M. Mansuripur, C. Denz, C. Alpmann, P. Banzer, T. Bauer, *et al.*, "Roadmap on structured light," *Journal of Optics*, vol. 19, no. 1, p. 013001, 2016.

[21] M. Piccardo, V. Ginis, A. Forbes, S. Mahler, A. A. Friesem, N. Davidson, H. Ren, A. H. Dorrah, F. Capasso, F. T. Dullo, *et al.*, "Roadmap on multimode light shaping," *Journal of Optics*, vol. 24, no. 1, p. 013001, 2021.

[22] K. Y. Bliokh, E. Karimi, M. J. Padgett, M. A. Alonso, M. R. Dennis, A. Dudley, A. Forbes, S. Zahedpour, S. W. Hancock, H. M. Milchberg, *et al.*, "Roadmap on structured waves," *Journal of Optics*, vol. 25, no. 10, p. 103001, 2023.

[23] P. O. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan, "A new universal and fault-tolerant quantum basis," *Information Processing Letters*, vol. 75, no. 3, pp. 101–107, 2000.

[24] A. Y. Kitaev, A. Shen, and M. N. Vyalyi, *Classical and quantum computation*. No. 47, American Mathematical Soc., 2002.

[25] Y. Wang, Z. Hu, B. C. Sanders, and S. Kais, "Qudits and high-dimensional quantum computing," *Frontiers in Physics*, vol. 8, p. 589504, 2020.

[26] S. Ecker, F. Bouchard, L. Bulla, F. Brandt, O. Kohout, F. Steinlechner, R. Fickler, M. Malik, Y. Guryanova, R. Ursin, *et al.*, "Overcoming noise in entanglement distribution," *Physical Review X*, vol. 9, no. 4, p. 041042, 2019.

[27] J. Spall, X. Guo, T. D. Barrett, and A. Lvovsky, "Fully reconfigurable coherent optical vector–matrix multiplication," *Optics Letters*, vol. 45, no. 20, pp. 5752–5755, 2020.

[28] J. C. Garcia-Escartin and P. Chamorro-Posada, "Quantum computer networks with the orbital angular momentum of light," *Physical Review A*, vol. 86, no. 3, p. 032334, 2012.

[29] J. L. O'brien, "Optical quantum computing," *Science*, vol. 318, no. 5856, pp. 1567–1570, 2007.

[30] X. Gao, M. Krenn, J. Kysela, and A. Zeilinger, "Arbitrary d-dimensional pauli x gates of a flying qudit," *Physical review A*, vol. 99, no. 2, p. 023825, 2019.

[31] N. K. Fontaine, J. Carpenter, S. Gross, S. Leon-Saval, Y. Jung, D. J. Richardson, and R. Amezcua-Correa, "Photonic lanterns, 3-d waveguides, multiplane light conversion, and other components that enable space-division multiplexing," *Proceedings of the IEEE*, vol. 110, no. 11, pp. 1821–1834, 2022.

[32] F. Brandt, M. Hiekkamäki, F. Bouchard, M. Huber, and R. Fickler, "High-dimensional quantum gates using full-field spatial modes of photons," *Optica*, vol. 7, no. 2, pp. 98–107, 2020.

[33] Q. Wang, J. Liu, D. Lyu, and J. Wang, "Ultrahigh-fidelity spatial mode quantum gates in high-dimensional space by diffractive deep neural networks," *Light: Science & Applications*, vol. 13, no. 1, p. 10, 2024.

[34] S. Goel, S. Leedumrongwatthanakun, N. H. Valencia, W. McCutcheon, A. Tavakoli, C. Conti, P. W. Pinkse, and M. Malik, "Inverse design of high-dimensional quantum optical circuits in a complex medium," *Nature Physics*, pp. 1–8, 2024.

[35] O. Lib and Y. Bromberg, "Quantum light in complex media and its applications," *Nature Physics*, vol. 18, no. 9, pp. 986–993, 2022.

[36] N. J. Cerf, C. Adami, and P. G. Kwiat, "Optical simulation of quantum logic," *Physical Review A*, vol. 57, no. 3, p. R1477, 1998.

[37] R. J. Spreeuw, "Classical wave-optics analogy of quantum-information processing," *Physical Review A*, vol. 63, no. 6, p. 062302, 2001.

[38] P. Londero, C. Dorrer, M. Anderson, S. Wallentowitz, K. Banaszek, and I. Walmsley, "Efficient optical implementation of the bernstein-vazirani,"

[39] G. Kaur, G. Narang, *et al.*, "Optical implementations, oracle equivalence, and the Bernstein-Vazirani algorithm," *JOSA B*, vol. 24, no. 2, pp. 221–225, 2007.

[40] B. Perez-Garcia, M. McLaren, S. K. Goyal, R. I. Hernandez-Aranda, A. Forbes, and T. Konrad, "Quantum computation with classical light: implementation of the Deutsch–Jozsa algorithm," *Physics Letters A*, vol. 380, no. 22-23, pp. 1925–1931, 2016.

[41] R. Jozsa, "Entanglement and quantum computation," *arXiv preprint quant-ph/9707034*, 1997.

[42] R. Jozsa and N. Linden, "On the role of entanglement in quantum-computational speed-up," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 459, no. 2036, pp. 2011–2032, 2003.

[43] Y. Wang, V. Potoček, S. M. Barnett, and X. Feng, "Programmable holographic technique for implementing unitary and nonunitary transformations," *Physical Review A*, vol. 95, no. 3, p. 033827, 2017.

[44] K. Nitta, O. Matoba, and T. Yoshimura, "Parallel processing for multiplication modulo by means of phase modulation," *Applied optics*, vol. 47, no. 4, pp. 611–616, 2008.

[45] P. N. Tamura and J. C. Wyant, "Two-dimensional matrix multiplication using coherent optical techniques," *Optical Engineering*, vol. 18, no. 2, pp. 198–204, 1979.

[46] 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.

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

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

[49] E. Bernstein and U. Vazirani, "Quantum complexity theory," in *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pp. 11–20, 1993.

[50] A. C.-C. Yao, "Quantum circuit complexity," in *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pp. 352–361, IEEE, 1993.

[51] J. Preskill, "Lecture notes for physics 229: Quantum information and computation," *California institute of technology*, vol. 16, no. 1, pp. 1–8, 1998.

[52] F. J. Pelletier and N. M. Martin, "Post's functional completeness theorem," *Notre Dame Journal of Formal Logic*, vol. 31, no. 3, pp. 462–475, 1990.

[53] H. M. Sheffer, "A set of five independent postulates for boolean algebras, with application to logical constants," *Transactions of the American mathematical society*, vol. 14, no. 4, pp. 481–488, 1913.

[54] J. E. Savage, *Models of computation*, vol. 136. Addison-Wesley Reading, MA, 1998.

[55] L. Amarù, E. Testa, M. Couceiro, O. Zografos, G. De Micheli, and M. Soeken, "Majority logic synthesis," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–6, IEEE, 2018.

[56] C. E. Shannon, "A symbolic analysis of relay and switching circuits," *Electrical Engineering*, vol. 57, no. 12, pp. 713–723, 1938.

[57] T. Toffoli, "Reversible computing," in *International colloquium on automata, languages, and programming*, pp. 632–644, Springer, 1980.

[58] C. M. Dawson and M. A. Nielsen, "The Solovay-Kitaev algorithm," *arXiv preprint quant-ph/0505030*, 2005.

[59] M. Mafu, A. Dudley, S. Goyal, D. Giovannini, M. McLaren, M. J. Padgett, T. Konrad, F. Petruccione, N. Lütkenhaus, and A. Forbes, "Higher-dimensional orbital-angular-momentum-based quantum key distribution with mutually unbiased bases," *Physical Review A—Atomic, Molecular, and Optical Physics*, vol. 88, no. 3, p. 032305, 2013.

[60] N. J. Cerf, M. Bourennane, A. Karlsson, and N. Gisin, "Security of quantum key distribution using d-level systems," *Physical review letters*, vol. 88, no. 12, p. 127902, 2002.

[61] C. He, Y. Shen, and A. Forbes, "Towards higher-dimensional structured light," *Light: Science & Applications*, vol. 11, no. 1, p. 205, 2022.

[62] I. Nape, B. Sephton, P. Ornelas, C. Moodley, and A. Forbes, "Quantum structured light in high dimensions," *APL Photonics*, vol. 8, no. 5, 2023.

[63] A. Aspect, P. Grangier, and G. Roger, "Experimental realization of Einstein-Podolsky-Rosen-Bohm gedankenexperiment: a new violation of bell's inequalities," *Physical review letters*, vol. 49, no. 2, p. 91, 1982.

[64] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theoretical computer science*, vol. 560, pp. 7–11, 2014.

[65] 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, p. 1895, 1993.

[66] C. H. Bennett and S. J. Wiesner, "Communication via one-and two-particle operators on Einstein-Podolsky-Rosen states," *Physical review letters*, vol. 69, no. 20, p. 2881, 1992.

[67] R. Simon and N. Mukunda, "Minimal three-component su (2) gadget for polarization optics," *Physics Letters A*, vol. 143, no. 4-5, pp. 165–169, 1990.

[68] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental realization of any discrete unitary operator," *Physical review letters*, vol. 73, no. 1, p. 58, 1994.

[69] C. Myers and R. Laflamme, "Linear optics quantum computation: an overview," *Quantum Computers, Algorithms and Chaos*, pp. 45–93, 2006.

[70] A. M. Weiner, "Femtosecond pulse shaping using spatial light modulators," *Review of scientific instruments*, vol. 71, no. 5, pp. 1929–1960, 2000.

[71] H. Kupianskyi, S. A. Horsley, and D. B. Phillips, "High-dimensional spatial mode sorting and optical circuit design using multi-plane light conversion," *APL Photonics*, vol. 8, no. 2, 2023.

[72] H. Zhou, J. Dong, J. Cheng, W. Dong, C. Huang, Y. Shen, Q. Zhang, M. Gu, C. Qian, H. Chen, *et al.*, "Photonic matrix multiplication lights up photonic accelerator and beyond," *Light: Science & Applications*, vol. 11, no. 1, p. 30, 2022.

[73] N. K. Fontaine, R. Ryf, H. Chen, D. T. Neilson, K. Kim, and J. Carpenter, "Laguerre-Gaussian mode sorter," *Nature communications*, vol. 10, no. 1, p. 1865, 2019.

[74] J. Cheng, H. Zhou, and J. Dong, "Photonic matrix computing: from fundamentals to applications," *Nanomaterials*, vol. 11, no. 7, p. 1683, 2021.

[75] A. Forbes and I. Nape, "Quantum mechanics with patterns of light: progress in high dimensional and multidimensional entanglement with structured light," *AVS Quantum Science*, vol. 1, no. 1, 2019.

[76] C. Rosales-Guzmán and A. Forbes, "How to shape light with spatial light modulators," Society of Photo-Optical Instrumentation Engineers (SPIE), 2017.

[77] V. Arrizón, U. Ruiz, R. Carrada, and L. A. González, "Pixelated phase computer holograms for the accurate encoding of scalar complex fields," *JOSA A*, vol. 24, no. 11, pp. 3500–3507, 2007.

[78] 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.

[79] M. Agnew, J. Leach, M. McLaren, F. S. Roux, and R. W. Boyd, "Tomography of the quantum state of photons entangled in high dimensions," *Physical Review A—Atomic, Molecular, and Optical Physics*, vol. 84, no. 6, p. 062101, 2011.

[80] L. Allen, M. W. Beijersbergen, R. Spreeuw, and J. Woerdman, "Orbital angular momentum of light and the transformation of Laguerre-Gaussian laser modes," *Physical review A*, vol. 45, no. 11, p. 8185, 1992.

[81] A. Mair, A. Vaziri, G. Weihs, and A. Zeilinger, "Entanglement of the orbital angular momentum states of photons," *Nature*, vol. 412, no. 6844, pp. 313–316, 2001.

[82] A. Lucas, "Ising formulations of many np problems," *Frontiers in physics*, vol. 2, p. 5, 2014.

[83] C. Granade, C. Ferrie, and S. T. Flammia, "Practical adaptive quantum tomography," *New Journal of Physics*, vol. 19, no. 11, p. 113017, 2017.

[84] Y. Shen and C. Rosales-Guzmán, "Nonseparable states of light: from quantum to classical," *Laser & Photonics Reviews*, vol. 16, no. 7, p. 2100533, 2022.

[85] A. Aiello, F. Töppel, C. Marquardt, E. Giacobino, and G. Leuchs, "Quantum-like nonseparable structures in optical beams," *New Journal of Physics*, vol. 17, no. 4, p. 043024, 2015.

[86] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 439, no. 1907, pp. 553–558, 1992.

[87] A. G. Kirk, H. T. Imam, K. Bird, and T. J. Hall, "Design and fabrication of computer-generated holographic fan-out elements for a matrix/matrix interconnection scheme," in *Intl Colloquium on Diffractive Optical Elements*, vol. 1574, pp. 121–132, SPIE, 1991.

[88] S. Zhou, S. Campbell, P. Yeh, and H.-k. Liu, "Modified-signed-digit optical computing by using fan-out elements," *Optics letters*, vol. 17, no. 23, pp. 1697–1699, 1992.

[89] D. E. Tamir, N. T. Shaked, P. J. Wilson, and S. Dolev, "High-speed and low-power electro-optical dsp coprocessor," *JOSA A*, vol. 26, no. 8, pp. A11–A20, 2009.

[90] H. Dammann and K. Görtler, "High-efficiency in-line multiple imaging by means of multiple phase holograms," *Optics communications*, vol. 3, no. 5, pp. 312–315, 1971.

[91] P. Kok, W. J. Munro, K. Nemoto, T. C. Ralph, J. P. Dowling, and G. J. Milburn, "Linear optical quantum computing with photonic qubits," *Reviews of modern physics*, vol. 79, no. 1, pp. 135–174, 2007.

[92] V. Sultanov, A. Kavčič, E. Kokkinakis, N. Sebastián, M. V. Chekhova, and M. Humar, "Tunable entangled photon-pair generation in a liquid crystal," *Nature*, vol. 631, no. 8020, pp. 294–299, 2024.

[93] Y. Liu, L. Zhou, M. Guo, Z. Xu, J. Ma, Y. Wen, N. M. Litchinitser, Y. Shen, J. Sun, and J. Zhou, "Broadband spin and orbital momentum modulator using self-assembled nanostructures," *Advanced Materials*, vol. 36, no. 45, p. 2412007, 2024.

[94] R. A. Horn, "The hadamard product," in *Proc. Symp. Appl. Math*, vol. 40, pp. 87–169, 1990.

[95] D. F. James, P. G. Kwiat, W. J. Munro, and A. G. White, "Measurement of qubits," *Physical Review A*, vol. 64, no. 5, p. 052312, 2001.

[96] M. Schlosshauer, "Decoherence, the measurement problem, and interpretations of quantum mechanics," *Reviews of Modern physics*, vol. 76, no. 4, p. 1267, 2005.

[97] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, pp. 802–803, 1982.

[98] M. Hendrych, R. Gallego, M. Mičuda, N. Brunner, A. Acín, and J. P. Torres, "Experimental estimation of the dimension of classical and quantum systems," *Nature Physics*, vol. 8, no. 8, pp. 588–591, 2012.

[99] N. Brunner, S. Pironio, A. Acin, N. Gisin, A. A. Méthot, and V. Scarani, "Testing the dimension of hilbert spaces," *Physical review letters*, vol. 100, no. 21, p. 210503, 2008.

[100] G. M. D'Ariano, M. G. Paris, and M. F. Sacchi, "Quantum tomography," *Advances in imaging and electron physics*, vol. 128, no. 10.1016, pp. S1076–5670, 2003.

[101] M. Paris and J. Rehacek, *Quantum state estimation*, vol. 649. Springer Science & Business Media, 2004.

[102] J. B. Altepeter, E. R. Jeffrey, and P. G. Kwiat, "Photonic state tomography," *Advances in atomic, molecular, and optical physics*, vol. 52, pp. 105–159, 2005.

[103] A. I. Lvovsky and M. G. Raymer, "Continuous-variable optical quantum-state tomography," *Reviews of modern physics*, vol. 81, no. 1, p. 299, 2009.

[104] E. Toninelli, B. Ndagano, A. Vallés, B. Sephton, I. Nape, A. Ambrosio, F. Capasso, M. J. Padgett, and A. Forbes, "Concepts in quantum state tomography and classical implementation with intense light: a tutorial," *Advances in Optics and Photonics*, vol. 11, no. 1, pp. 67–134, 2019.

[105] R. T. Thew, K. Nemoto, A. G. White, and W. J. Munro, "Qudit quantum-state tomography," *Physical Review A*, vol. 66, no. 1, p. 012303, 2002.

[106] R. De Wolf, "Quantum computing: Lecture notes," *arXiv preprint arXiv:1907.09415*, 2019.

[107] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[108] 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.

[109] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

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

[111] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.

[112] F. Glover, G. Kochenberger, and Y. Du, "Quantum bridge analytics i: a tutorial on formulating and using qubo models," *4or*, vol. 17, no. 4, pp. 335–371, 2019.

[113] J. T. Barreiro, N. K. Langford, N. A. Peters, and P. G. Kwiat, "Generation of hyperentangled photon pairs," *Physical review letters*, vol. 95, no. 26, p. 260501, 2005.

[114] L. Allen, M. W. Beijersbergen, R. Spreeuw, and J. Woerdman, "Orbital angular momentum of light and the transformation of Laguerre-Gaussian laser modes," *Physical review A*, vol. 45, no. 11, p. 8185, 1992.

[115] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, *et al.*, "Advances in quantum cryptography," *Advances in optics and photonics*, vol. 12, no. 4, pp. 1012–1236, 2020.

[116] J. L. O'brien, A. Furusawa, and J. Vučković, "Photonic quantum technologies," *Nature Photonics*, vol. 3, no. 12, pp. 687–695, 2009.

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

[118] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien, "Quantum computers," *nature*, vol. 464, no. 7285, pp. 45–53, 2010.

[119] R. Renner, "Security of quantum key distribution," *International Journal of Quantum Information*, vol. 6, no. 01, pp. 1–127, 2008.

[120] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, "The security of practical quantum key distribution," *Reviews of modern physics*, vol. 81, no. 3, pp. 1301–1350, 2009.

[121] 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.

[122] E. Toninelli, B. Ndagano, A. Vallés, B. Sephton, I. Nape, A. Ambrosio, F. Capasso, M. J. Padgett, and A. Forbes, "Concepts in quantum state tomography and classical implementation with intense light: a tutorial," *Advances in Optics and Photonics*, vol. 11, no. 1, pp. 67–134, 2019.

[123] T. Opatrnỳ, D.-G. Welsch, and W. Vogel, "Least-squares inversion for density-matrix reconstruction," *Physical Review A*, vol. 56, no. 3, p. 1788, 1997.

[124] Z. Hradil, "Quantum-state estimation," *Physical Review A*, vol. 55, no. 3, p. R1561, 1997.

[125] V. Bužek, R. Derka, G. Adam, and P. L. Knight, "Reconstruction of quantum states of spin systems: From quantum bayesian inference to quantum tomography," *Annals of Physics*, vol. 266, no. 2, pp. 454–496, 1998.

[126] K. Jones, "Principles of quantum inference," *Annals of Physics*, vol. 207, no. 1, pp. 140–170, 1991.

[127] S. Lohani, B. T. Kirby, M. Brodsky, O. Danaci, and R. T. Glasser, "Machine learning assisted quantum state estimation," *Machine Learning: Science and Technology*, vol. 1, no. 3, p. 035007, 2020.

[128] M. Neugebauer, L. Fischer, A. Jäger, S. Czischek, S. Jochim, M. Weidemüller, and M. Gärttner, "Neural-network quantum state tomography in a two-qubit experiment," *Physical Review A*, vol. 102, no. 4, p. 042604, 2020.

[129] Y. Zheng, C. Song, M.-C. Chen, B. Xia, W. Liu, Q. Guo, L. Zhang, D. Xu, H. Deng, K. Huang, *et al.*, "Solving systems of linear equations with a superconducting quantum processor," *Physical review letters*, vol. 118, no. 21, p. 210504, 2017.

[130] J. Pan, Y. Cao, X. Yao, Z. Li, C. Ju, H. Chen, X. Peng, S. Kais, and J. Du, "Experimental realization of quantum algorithm for solving linear systems of equations," *Physical Review A*, vol. 89, no. 2, p. 022313, 2014.

[131] A. Mair, A. Vaziri, G. Weihs, and A. Zeilinger, "Entanglement of the orbital angular momentum states of photons," *Nature*, vol. 412, no. 6844, pp. 313–316, 2001.

[132] J. Du and J. Wang, "High-dimensional structured light coding/decoding for free-space optical communications free of obstructions," *Optics Letters*, vol. 40, no. 21, pp. 4827–4830, 2015.

[133] A. Forbes, A. Dudley, and M. McLaren, "Creation and detection of optical modes with spatial light modulators," *Advances in optics and photonics*, vol. 8, no. 2, pp. 200–227, 2016.