University of Camerino

Scuola di Scienze e Tecnologie

Laurea Magistrale in
Physics (LM-17)

AI Rediscovery of Complex Quantum Networks

Supervisors

Prof. David Vitali

Prof. Mario Krenn

Candidate

Marcello Armezzani

Academic Year 2024-2025

Contents

List of Figures

Introduction

Chapter 1

Quantum Networks

1.1 A brief introduction to Quantum Networks

The world we live in is based on interconnected networks. Of all types of networks, the Internet is the one that most characterises our daily activities. Although its ultimate nature is digital, the Internet is based on a complex set of physical elements that enable it to function, and the various strategies that can be implemented with these elements (hereinafter referred to as *protocols*) allow us to perform different types of tasks. In line with the growing development of quantum technologies, it is natural to expect that the Internet and networks in general will also undergo a quantum evolution, especially since several tasks that cannot be implemented on classical networks are instead possible in a quantum context. These include key distribution [8, 17], clock synchronization [28], improving the baseline of telescopes [24], and several others [51]. Before getting into the actual discussion, let us take a few lines to introduce the basic concepts that we will be using throughout this work.

1.1.1 Photon encoding

From an experimental perspective, photons have established themselves as a leading quantum information carrier. This is due to the fact that they are easy to transmit over low-loss long distance channels, and moreover they can encode information in a relatively easy way, and actually we have several different degrees of freedom we can exploit in order to encode information into light. Two main categories of encoding exists: discrete variable (DV) and continuous variable (CV) encoding. The difference between the two lies in the cardinality of the corresponding Hilbert spaces: the state space of DV encodings is spanned by a finite number of orthogonal states, whilst the state space of CV encodings is spanned by an infinite number (possibly countable) of orthogonal states. In the present work we'll use just DV encodings, consistent with what often occurs in literature [4], essentially because CV encodings are particularly challenging when it comes to suppression of loss errors [38]. Another distinction we can make is the one between *single-rail* and *dual-rail* encoding. One can intend a rail as an optical mode used to characterize the information about a state. In particular, we will consider

- *Fock* encoding, as a single-rail encoding that indicates the occupation number

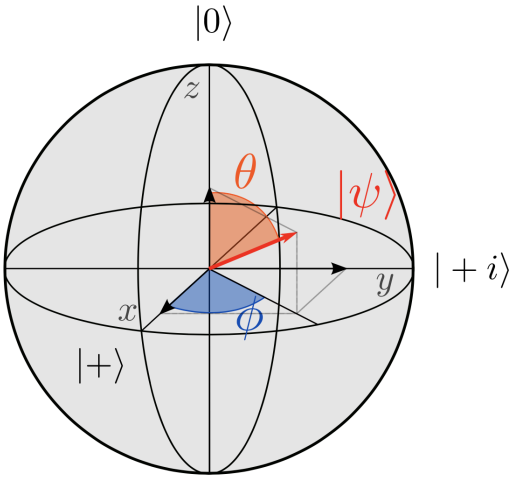


Figure 1.1: Bloch sphere representation of a qubit.

of photons of a given mode ($|0\rangle, |1\rangle, |2\rangle, ...$). In this representation, the qubit $|\psi\rangle$ is encoded into the Hilbert subspace of a single mode spanned by the vacuum state $|0\rangle$ and the single-photon state $|1\rangle$ ($|\psi\rangle = a|0\rangle + b|1\rangle$);

- *Polarization* encoding, as a dual-rail encoding that encodes a qubit $|\psi\rangle$ into the polarization state of a single photon, $|\psi\rangle = a|H\rangle + b|V\rangle$, where $|H\rangle = |10\rangle$ and $|V\rangle = |01\rangle$ (we use two optical modes to encode this information, therefore the name "dual").

For the single-rail encoding, with reference to Figure 1.1, we can always implement a rotation along the Z axis through a phase shifter, but the same cannot be done along the X axis, because state $|0\rangle$ and state $|1\rangle$ have different energies, and therefore non-linear optics operations are needed. On the other hand, it's always possible to perform a single qubit rotation through linear optics elements in dual-rail encoding [27]. Another notable difference is that for single-rail encoding it is possible to generate entanglement deterministically with linear optical resources, while the same process is necessarily probabilistic in dual-rail encoding. A good example of this difference is shown in Figure 1.2.

1.2 General scheme for a quantum network

A quantum network consists of essentially two components: processing nodes and quantum channels. A processing node is in general a node that can perform arbitrary local operations (LO) allowed by quantum mechanics [53, 37]. Concretely it could be a quantum computer or a repeater node, and it's the latter possibility the one we'll focus on in the following sections. The LO of a node X is mathematically described by a completely positive (CP) map, that takes in input a generic state ρ and returns a state σ_k with probability p_k in output, where $\sigma_k = \frac{M_k^X \rho (M_k^X)^\dagger}{p_k}$, where $\{M_k^X\}$ are Kraus operators [3]. On the other hand, a quantum channel between nodes X and Y can be implemented experimentally as an optical fiber, a superconducting microwave transmission line, or even optical free-space link; however, it

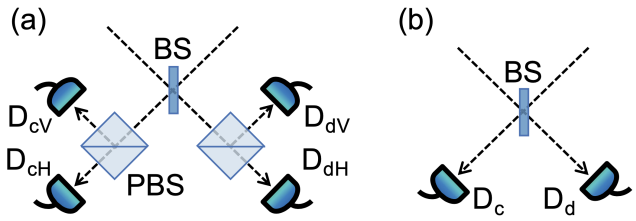


Figure 1.2: Examples of implementations of Bell state measurements (BSM). (a) Bell measurement for polarization-encoded qubits, spanned by horizontally and vertically polarized single-photon states $|H\rangle$ and $|V\rangle$. This is implemented using the application of a symmetric beam splitter (BS) on optical modes, followed by a polarization beam splitter (PBS) on each of the two output modes and then by photon detection at all the output modes. Clicks in detectors D_{cH} and D_{cV} or in D_{dH} and D_{dV} project the received pair of the qubits into the Bell state $|\Psi^+\rangle = \frac{1}{\sqrt{2}}\big(|HV\rangle + |VH\rangle\big)$, while clicks in detectors D_{cH} and D_{dV} or in D_{cV} and D_{dH} project the received pair of the qubits into the Bell state $|\Psi^-\rangle = \frac{1}{\sqrt{2}}\big(|HV\rangle - |VH\rangle\big)$. Notice that this Bell measurement can succeed only when the input two optical pulses have two or more photons in total. (b) Bell measurement for Fock-encoded qubits, spanned by the vacuum state $|0\rangle$ and the single-photon state $|1\rangle$. This is implemented by the application of a symmetric BS on optical modes, followed by photon counting at the output modes. Crucially, the Bell states in single-rail encoding live in different excitation subspaces. A click in the detector D_c (or D_d) at the constructive-interference (destructive-interference) mode projects the received pair of the qubits into the state $|\Psi^+\rangle = \frac{1}{\sqrt{2}}\big(|01\rangle + |10\rangle\big)$ $\left(|\Psi^-\rangle = \frac{1}{\sqrt{2}}\big(|01\rangle - |10\rangle\big)\right)$. Both implementations can distinguish $|\Psi^\pm\rangle$ from the other states only, and the success probabilities are thus $\frac{1}{2}$ even in the ideal cases.

is broadly speaking a completely positive trace preserving (CPTP) map that acts as $\mathcal{N}^{X\to Y}(\rho) = \mathrm{Tr}_{E'}[U^{XE\to YE'}(\rho \otimes |0\rangle_E\langle 0|)(U^{XE\to YE'})^\dagger]$, with $U^{XE\to YE'}$ as a unitary acting on Hilbert space $H_X \otimes H_E = H_Y \otimes H_{E'}$, where E, E' account for the environment. It is particularly useful to have such an abstract definition of a quantum channel, since this allows us to establish the limits of our quantum networks by studying the maximum capacity of the associated channels. Currently, it's still unknown how to compute the capacity of a general channel, and in many cases the best we can get are bounds [2, 40, 41]. However, for the lossy bosonic channel the capacity is known [42], and this is enough for our work, since the pure-loss bosonic channel is often used as a model for the optical fiber [4], and in our model we'll use precisely these as quantum channels. We can write it using a U_{XE} such that:

$$(U_{XE})a_X^\dagger(U_{XE}^\dagger) = \sqrt{\eta}a_X^\dagger + \sqrt{1-\eta}a_{E'}^\dagger, \tag{1.1}$$

where $a_X^\dagger$ is the creation operator in the bosonic system X and $0 \leq \eta \leq 1$ is the channel transmittance. Even if it has no practical purpose for the rest of the work, it is worth noticing that based on the structure we outlined for the quantum network, one can define a graph $G = (V, E)$ with a set V of vertices and a set E of directed edges, in such a way that vertices in the set V and directed edges in the set E correspond to quantum information processing nodes and quantum channels in the quantum network, respectively. A graphical representation of such a description is seen in Figure 1.3.

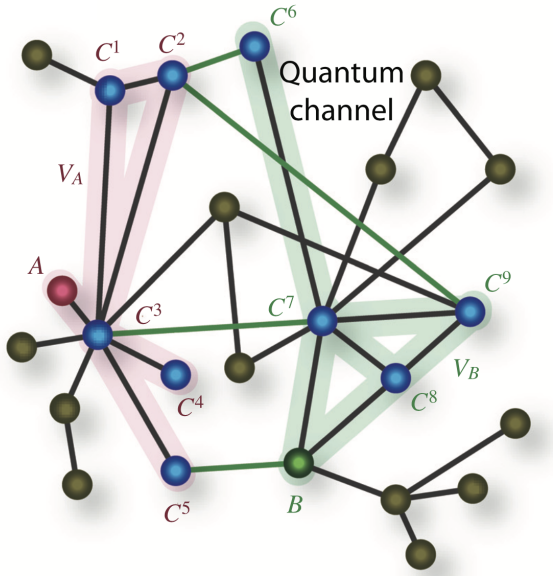


Figure 1.3: Depiction of a quantum network as a graph.

1.3 Communication over a quantum network

If we want to understand how to use the knowledge of the earlier sections, we should beforehand clarify what's the purpose of the quantum network we have in mind. What we want to achieve is fast and reliable communication between two

parties (A and B in Figure 1.3), meaning that we want to send information from a sender A to a receiver B with little (or in principle no) loss. One way to achieve this is through *quantum teleportation*, that we illustrate below.

1.3.1 Quantum teleportation

Quantum teleportation [7] consists in transferring quantum information between two distant parties without having to transfer the physical carrier of that information, and in order to achieve this the parties must share beforehand a classical communication channel and an entangled pair.

Suppose A owns a qubit in a generic state $|\psi\rangle_{A_1}$, and shares with B an entangled pair $|\Phi^+\rangle_{A_2B}$, that for the single-rail encoding is $|\Phi^+\rangle_{A_2B} = \frac{1}{\sqrt{2}}\big(|0\rangle_{A_2}|0\rangle_B + |1\rangle_{A_2}|1\rangle_B\big)$. If A performs a Bell-state measurement (remember Figure) on her two qubits A_1 and A_2, she will project B's qubit onto a state, that is exactly the state $|\psi\rangle$ up to a local rotation, established by the outcome of A's measurement according to:

$$|\Phi^+\rangle_{A_1A_2} \to |\psi\rangle_B, \tag{1.2}$$
$$|\Phi^-\rangle_{A_1A_2} \to Z_B|\psi\rangle_B, \tag{1.3}$$
$$|\Psi^+\rangle_{A_1A_2} \to X_B|\psi\rangle_B, \tag{1.4}$$
$$|\Psi^-\rangle_{A_1A_2} \to Z_BX_B|\psi\rangle_B. \tag{1.5}$$

Through the classical channel, A communicates to B the outcome of her measurement, so that B can apply the correct rotation and recover the state $|\psi\rangle$, ending the teleportation.
We therefore realize that the main goal to achieve reliable quantum communication is to share a Bell pair between two distant parties, and this is what we'll focus on in the next section.

1.4 Quantum repeaters

Now that we identify our goal with the sharing of a Bell pair between two parties A, B separated by a distance L, we ask what are the best strategies to achieve it. Before giving an answer, it's worth to show why a direct communication of a qubit over an optical fiber it's unpractical. With reference to Eq. 1.1, we know that the transmittance η of the optical fiber is related to its length L by

$$\eta = e^{-\frac{L}{L_{\text{att}}}}, \tag{1.6}$$

where L_{att} is the attenuation length, that in the case of typical optical fibers is 22km. With such a negative exponential scaling of the transmittance, for a distance $L = 500$km we would have $\eta \sim 10^{-10}$, that in the best hypothesis for the private capacity would guarantee a rate in the order of $1\frac{\text{bit}}{\text{s}}$. To give an idea of this value, a 24-hour communication would allow a 20-kilobyte email to be shared securely at best [4]. It is obvious that this makes it impossible to hope to base quantum networks on point-to-point communication.

Of course, simply generating a local Bell pair and send one qubit over the channel suffers from the same problem, therefore we are required to introduce some more elements in order to make quantum communication over long distances feasible. It seems that it's not possible to require for both photons of the entangled pair to be generated locally together and later distributed, because in the end the total amount of distance traveled by the Bell pair will be L, and we won't have any mitigation of the loss we just saw. We need therefore a procedure that allows us to generate entanglement between two photons, without the need of making them interact. *Entanglement swapping* serves the scope, as we see below.

1.4.1 Entanglement swapping

Entanglement swapping [39] consists in entangling two qubits that never interact. To realize it, we need two Bell pairs, one shared between A and C and the other between C and B. The initial state $|\Phi^+\rangle_{AC_1}|\Phi^+\rangle_{C_2B}$ can be rewritten as

$$|\Phi^+\rangle_{AC_1}|\Phi^+\rangle_{C_2B} = \frac{1}{2}\big(|\Phi^+\rangle_{AB}|\Phi^+\rangle_{C_1C_2}+|\Phi^-\rangle_{AB}|\Phi^-\rangle_{C_1C_2}+|\Psi^+\rangle_{AB}|\Psi^+\rangle_{C_1C_2}+|\Psi^-\rangle_{AB}|\Psi^-\rangle_{C_1C_2}\big), \tag{1.7}$$

and by applying a Bell-state measurement, C will individuate one of the four Bell pairs, projecting also the qubits shared between A and B in the same Bell pair. This procedure is remarkable for two reasons: first, it generates a Bell pair between A and B without making the qubits of the pair ever interact, and second, the generated Bell pair is ready to be used for further processing, and not immediately consumed. This latter property is achieved because the swapping procedure we just described is an heralding generation of states [19], where we detect just a portion of the the total photons, leaving the undetected ones ready to be further used, opposed to post-selected states, that are instead generated by measuring all the photons in the system, validating the generation of the state but consuming it at the same time.

Now, how can we put entanglement swapping in the picture? A possibility is shown in Figure 1.4.

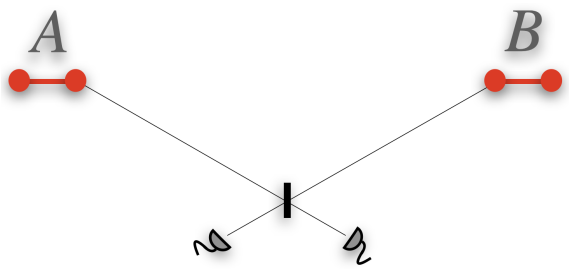


Figure 1.4: Two parties A and$B$ separated by a distance L each own an entangled pair. They perform entanglement swapping by performing a BSM at the middle station. In this case, contrary to what we showed in Figure 1.2, the success probability is 25%, because with a coincidence detection we are able to distinguish just the state$|\Psi^-\rangle$.

Doing a Bell-state measurement in the middle allows us to make the photons travel

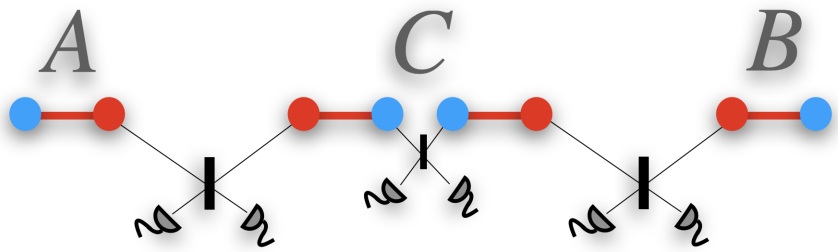


Figure 1.5: We add a repeater in the middle. In light blue we represent the memories, that don't immediately release information, but can store it for some time.

just half the length, $\frac{L}{2}$, but this is not enough to improve with respect to the direct transmission. In fact, requiring that both photons successfully reach the middle measurement station means that the total probability of generating an entangled pair between A and B is

$$p_{AB} = (e^{-\frac{L}{2L_{\text{att}}}})(e^{-\frac{L}{2L_{\text{att}}}})p_s = e^{-\frac{L}{L_{\text{att}}}}p_s = \eta p_s, \tag{1.8}$$

where p_s is the swapping probability, i.e. the probability that C can successfully detect its Bell pair. Even when $p_s = 1$ (and we saw this is not typically true), still our probability p registers no gain with respect to the point-to-point communication. If we express this result in terms of the *expected number of trials* E, that is the number of times we expect to repeat our protocol, we would say that E is still

$$E = \frac{1}{p_{AB}} = \frac{1}{\eta} \sim e^L. \tag{1.9}$$

Entanglement swapping goes in the right direction, but we still miss an ingredient to increase the efficiency of the entanglement sharing, and this is *quantum memories*.

1.4.2 Quantum memories

Quantum memories [36] are devices that, similarly to what happens in the classical context, are able to store information (encoded in our case by qubits) for a certain amount of time, and release it on demand. As of today, they are mainly realized either through delay lines, optical cavities or most notably atomic ensembles [15]. However, here we are not interested in the different possible experimental realizations, but just on their physical ability to store states. With quantum memories, we can upgrade the scheme in Figure 1.4 to the scheme in Figure 1.5. Now, at the cost of using more resources (photons, memories, detectors), we build a protocol that consists of the parallel generation of two Bell pairs through entanglement swapping between A, C and C, B, where C is called *quantum repeater*. When both pairs are successfully generated, we can perform the last swapping at the repeater, and if it is successful, we have a Bell pair shared between A and B. From the computational point of view, nothing new is happening, and what we observed in Eq. 1.7 is enough knowledge to recover the result with more resources; the real novelty is given by the use of memories, that allow for a good Bell pair generated either between A, C or

C, B to be stored. Without memories, this protocol could be successful only when both pairs are correctly generated in the same trial, and we would find again the result in 1.9. Now instead, we allow for the pairs between A, C and C, B to be generated in the same trial, but also over different trials, so that if the generation of a Bell pair on one side is successful and the other fails, we don't have to restart from scratch, but we keep the partial success stored and we just repeat the generation of the other pair until that is also successful.

The probability of generating an entangled pair through swapping between one of the parties and the repeater is $p = (e^{-\frac{L}{4L_{\text{att}}}})(e^{-\frac{L}{4L_{\text{att}}}})p_s = (e^{-\frac{L}{2L_{\text{att}}}})p_s$. Now, since it's not required to get both pairs in the same trial, we don't simply write $E = \frac{1}{p^2 p_s}$ (this would correspond to both pairs generated and successful final swapping in the same trial), but instead [46]:

$$E = \frac{3 - 2p}{p(2-p)p_s} \sim e^{\frac{L}{2}}. \tag{1.10}$$

Comparing the results in 1.9 and 1.10 we see how the use of quantum memories wins us a square root improvement in the generation of a Bell state between A and B. Different methods can be used to compute the expression in 1.10; at the moment we just send back to the reference [4], but in the following chapters we'll show our personal derivation of this result.

1.4.2.1 Nested quantum repeaters

What we presented in Figure 1.5 is typically called *quantum repeater protocol*, and in particular one that makes use of a single repeater. However, the structure of the setup associated with the functioning logic are so general that we can immediately think of putting more repeaters in between A and B, concatenating them, reaching structures as those in Figure.

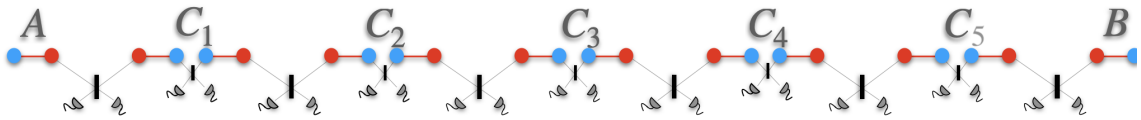


Figure 1.6: Nested repeater with 5 intermediate repeater nodes. Whenever two adjacent segments are ready, a node will attempt a swapping to share entanglement over a longer distance.

These are referred to as *nesting levels* of the repeater chain, and even if an exact expression for the expected number of trials is still not known in this general case (strategies exist to numerically compute the expectation values at higher levels [41]), a good approximation is easy to extrapolate [4]:

$$E_n \sim \frac{3^n}{2^n p_s^n p(-\frac{L}{2^n L_{\text{att}}})}, \tag{1.11}$$

where $p(-\frac{L}{2^n L_{\text{att}}}) = (e^{-\frac{L}{2^n L_{\text{att}}}})p_s$ and n is the nesting level. For completeness, we also indicate that in the case of a deterministic swapping, i.e. $p_s = 1$ (we already

argued several times about never reaching this value in practice), a general solution is known:

$$E_n = \sum_{k=1}^{n} \binom{n}{k} \frac{(-1)^{k+1}}{1 - (1 - p(-\frac{L}{2^n L_{\text{att}}}))^k}. \tag{1.12}$$

Chapter 2

The AI approach

2.1 Introducing AI to tackle Quantum Networks

The study of quantum systems, or physics in general, through the use of AI is an increasingly common activity in the scientific community. Great variability in the analysis of these systems is guaranteed by an ever-increasing number of machine learning (or deep learning) techniques, but we believe that great importance should be assigned to the underlying philosophy that guides the process of discovery and understanding of the results [33]. Remaining faithful to our subject of study, we will not delve into a general analysis of the various AI techniques applied to quantum technologies, for which we refer to [34], but we will limit ourselves to retracing some important steps in the design of quantum optical circuits through AI until we arrive at our current software.

2.1.1 Evolution of AI in quantum optics

There is some unbreakable discrete nature when it comes to generating blueprint for experiments. In fact, experimental setups are composed by a finite number of elements, that in the context of quantum optics are lasers, nonlinear crystals, single photon sources, beam splitters, phase shifters, photon detectors, and so on. A given combination of such elements could lead to a very useful setup, for example to generate some complex high-dimensional state, but exchange one single element and you will find yourself with completely different, and most probably useless, setup. In the past, it was the task of expert quantum physicists to combine these elements in such a way as to provide the desired result, but in recent years machines have emerged that are capable of performing automated discovery from scratch. The first example of this kind is presented in [31]; the AI named *Melvin*, like any other scientist, has its own toolbox full of the experimental components we listed above. He puts optical elements from the toolbox onto a virtual optical table. The algorithm initially starts to put elements from the toolbox in random order, and if the candidate setup satisfies all sanity checks, a simulator computes the full experimental output. If the setup produces the desired quantum state, it is automatically simplified and reported to the user. In addition, the setup is then stored as a new part of the toolbox. In that way, over time, the algorithm learns useful macro components

which it can use in subsequent iterations. Thereby, it can already access useful operations that significantly speed up the discovery process. For details see Figure 2.1.

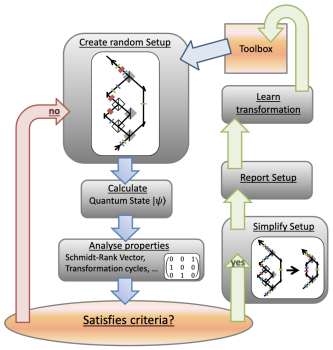


Figure 2.1: Melvin workflow.

Melvin succeeded in individuating several new blueprints that where also tested in the laboratory [30]. Not only these discoveries surpass sometimes the best human implementations, they are also often quite unintuitive for the scientists that find them. Being such a young field, there's still a quite strong debate over the reasons that lead sometimes to such a mismatch between humans and machines when it comes to finding new experiments. Our idea is that we can imagine the space of all the possible realizable experiments under given physical rules as the one in Figure 2.2.

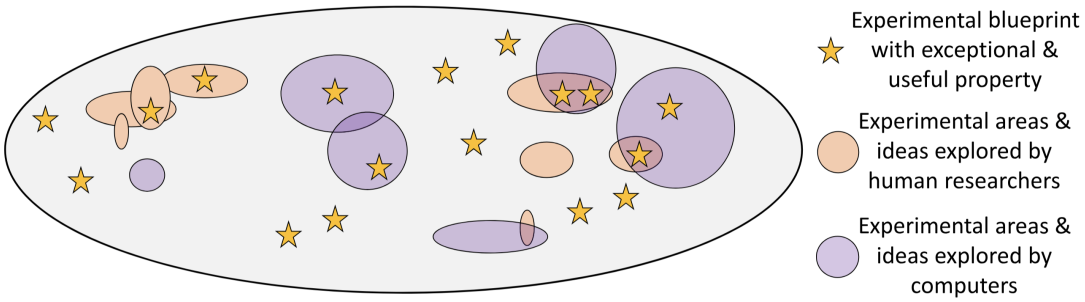


Figure 2.2: Abstract space of all experimental configurations. This space contains all possible experimental setups, including all configurations with the exceptional properties, such as quantum networks that can distribute entanglement (indicated by stars). In this space, creative and experienced human researchers have found many setups (orange). However, other useful but unorthodox designs might never be detected in this way. Here, AI-based design might help (violet).

Our human intuition, theoretical background and habit to a certain specific approach to analysis naturally restrict ourselves to exploring just portions of the space. On the other hand, machines will suffer in part from different biases, and this can potentially lead to the exploration of different regions of the space.

What's clear is that we still need to understand what the machine is doing, and the great interpretability of *Melvin* allowed human scientists to generalize some of

the discoveries, unravelling previously unknown concepts, one of which was particularly useful to create the second algorithm we mention: *PyTheus* [45]. It was discovered in fact that many quantum optical experiments can be translated into a graph-based representation [32], as shown in Figure 2.3.

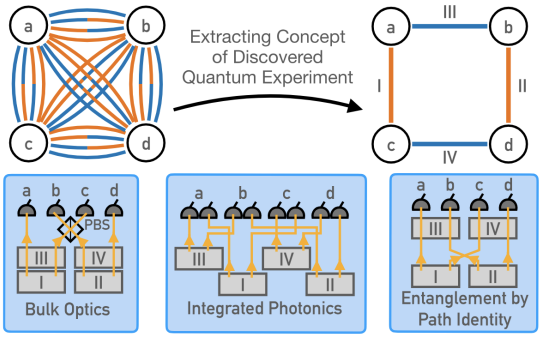


Figure 2.3: Discovery of quantum experiments. Quantum optics experiments can be represented by colored graphs. Using the most general, complete graph as a starting representation, the AI's goal is to extract the conceptual core of the solution, which can then be understood by human scientists. The solution can then be translated to numerous different experimental configurations.

Now, the problem of finding a certain quantum setup can be directly translated into discovering a graph with certain properties. Being now the parameters continuous, one can use gradient-based optimization algorithms to find the solution, making the model orders of magnitudes faster than the previous technique. This marks an important shift in the approach: we try to break the discrete nature of the problem by embedding it in a more general continuous space. To conclude, the graph's topology is eventually simplified, such that the human researcher is not only presented with a solution, but can immediately understand why and how the solution works. The algorithm has been used to answer numerous open questions and has led to new concepts, some of which have already been experimentally realized [50].

2.1.2 Improving on the existing models

Our brief historical review of AI models used for automated discoveries of experimental setups in quantum optics, left us with two main observations: firstly, we have now a clear image of the space we are operating into (see Figure 2.2), and secondly, we know we want to make this space continuous, since this will win us orders of magnitudes in execution time. What's left to understand is which schematization are we using to represent our physical system, if oriented to the real experimental tools or instead more abstract.

As some works already show [29, 44], whenever we are in possess of a very fast and reliable simulator, a good representation of the general solution space is all we need to perform an optimization. We call this good initial representation an *ansatz*. However, the time required by the simulator and the optimizer to compute states is strictly related to the parametrization we make of that space. Although *PyTheus* has improved *Melvin*'s execution times by introducing continuous parameterisation, the

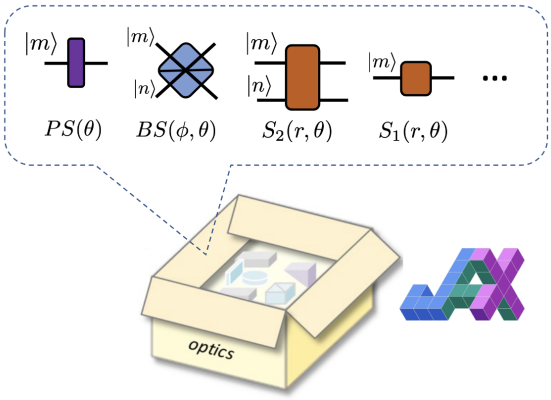


Figure 2.4: *EsQueranto* toolbox with elements built in the number basis. Credits: [25]

.

abstraction of graph formulation limits us in the type of operations we can perform and the type of tools we can use (for example, it is unclear how to introduce quantum memories). It therefore seems convenient to come back to the real experimental tools, keeping the continuous description. This is precisely what *EsQueranto* does.

2.2 *EsQueranto*

EsQueranto is a JAX-based[10] software originally proposed and designed in [25] for the simulation of quantum optics setups. Despite it still being under development, we will show in the remaining part of this chapter some examples of its possible applications, to be intended as a playground to get familiar with its syntax and structures (see Figure 2.4).
Later, in the last two chapters, we'll apply it explicitly to quantum networks and we'll show how to perform optimizations with it. Before showing the examples, let's spend some time on introducing JAX, in order to understand why it is a particularly good choice for automated discovery.

2.2.1 JAX: automatic differentiation for machine learning

JAX is a modern Python library for high-performance numerical computing that has rapidly gained popularity in the fields of machine learning, scientific computing, and quantum simulation. At its core, JAX provides a `NumPy`-like API, but it is designed from the ground up to leverage accelerators such as GPUs and TPUs, while also exposing powerful tools for automatic differentiation, compilation, and parallelization. In practice, JAX allows researchers to write code that is both mathematically expressive and computationally efficient, making it particularly suited to the kind of large-scale optimization tasks that arise in machine learning and physics, precisely as the one we are tackling in this work.

One of the most distinctive features of JAX is its support for **automatic differentiation (autodiff)**, implemented through the `grad` and `value_and_grad` primi-

tives. Unlike symbolic differentiation or numerical finite differences, JAX's autodiff is based on *reverse-mode differentiation*, the same technique used by deep learning frameworks such as TensorFlow or PyTorch, but implemented in a functional and composable way. Given a function

$$f(x) = x^2 + 3x + \sin(x), \tag{2.1}$$

JAX can automatically provide the derivative

$$\nabla f(x) = 2x + 3 + \cos(x), \tag{2.2}$$

without requiring symbolic manipulation or explicit finite differences. More generally, if $f : \mathbb{R}^n \to \mathbb{R}$, JAX's reverse-mode differentiation computes $\nabla f(x)$ in time $O(1)$ (i.e. constant) relative to evaluating $f(x)$, which is ideal for optimization. In contrast, forward-mode differentiation, accessible through `jax.jvp`, is more efficient for $f : \mathbb{R} \to \mathbb{R}^m$, since it propagates derivatives along input directions. JAX supports both modes and even higher-order derivatives. For instance, the Hessian of a function can be computed as

$$H_f(x) = \nabla^2 f(x) = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \tag{2.3}$$

by simply nesting transformations like `jax.grad(jax.grad(f))`.

Another central component is the **just-in-time (JIT) compilation**, implemented via the `@jit` decorator. When applied, JAX traces the Python function, builds a computation graph, and hands it off to *XLA* (Accelerated Linear Algebra) [23]. XLA generates optimized code for the target backend (CPU, GPU, or TPU), fusing operations into efficient kernels and reducing memory overhead. This allows researchers to write clean Python functions like

```
@jit
def step(params, x):
    return jnp.dot(params, x)
```

and obtain performance comparable to low-level CUDA or C++ implementations.

Beyond differentiation and compilation, JAX provides primitives for **vectorization and parallelization**. The `vmap` transformation automatically lifts a function $f : \mathbb{R}^n \to \mathbb{R}^m$ to operate on batches:

$$\texttt{vmap}(f)(x_1, x_2, \ldots, x_B) \;\equiv\; \big(f(x_1), f(x_2), \ldots, f(x_B)\big), \tag{2.4}$$

removing Python loops and generating efficient batched GPU kernels. Similarly, `pmap` distributes computations across multiple devices, e.g. many GPUs, enabling near-linear speedups for large-scale training or simulation.

An additional strength of JAX lies in its **hardware compatibility**. The same Python code runs seamlessly on CPU, GPU, or TPU without modification, provided the appropriate backend is installed. This portability makes it appealing in quantum information, where computations may scale from local prototyping on a laptop to distributed clusters. Tight integration with CUDA and ROCm ensures efficient GPU memory usage, while XLA provides low-level optimizations such as kernel fusion.

Finally, JAX follows a design philosophy based on **composable function transformations**. Transformations like `grad`, `jit`, `vmap`, and `pmap` can be arbitrarily combined, for instance:

$$\texttt{pmap}\big[\texttt{jit}\big[\texttt{vmap}\big[\texttt{grad}(f)\big]\big]\big], \tag{2.5}$$

allowing one to compute gradients of a batched function across multiple devices with minimal changes to the original code. This functional approach has been highlighted as one of the core innovations of JAX [10].

In summary, JAX offers a unique combination of features—automatic differentiation in both forward and reverse modes, JIT compilation to XLA backends, efficient vectorization through `vmap`, scalable parallelism via `pmap`, and hardware portability.

2.2.2 First example: generation of a N00N state

A paradigmatic resource in quantum optics is the so-called *N00N state*, which for N photons has the form

$$|\psi_{\mathrm{N00N}}\rangle \;=\; \frac{1}{\sqrt{2}}\left(|N,0\rangle + e^{iN\alpha}|0,N\rangle\right), \tag{2.6}$$

where $|N,0\rangle$ denotes all N photons in mode a and none in mode b, $|0,N\rangle$ the reverse situation, and α is a phase. Such states are central in quantum metrology for phase super-resolution and approaching the Heisenberg limit [14, 20, 21].

Beam splitter convention. We adopt the standard lossless 2×2 unitary for a beam splitter acting on the *annihilation* operators,

$$\begin{pmatrix}\hat{a}_{\mathrm{in}}^\dagger\\ \hat{b}_{\mathrm{in}}^\dagger\end{pmatrix} = \underbrace{\begin{pmatrix}e^{i\phi}\sin\theta & \cos\theta\\ \cos\theta & -e^{-i\phi}\sin\theta\end{pmatrix}}_{U_{\mathrm{BS}}(\theta,\phi)} \begin{pmatrix}\hat{a}_{\mathrm{out}}^\dagger\\ \hat{b}_{\mathrm{out}}^\dagger\end{pmatrix}, \qquad \theta\in[0,\tfrac{\pi}{2}],\ \phi\in[0,2\pi), \tag{2.7}$$

which satisfies $U_{\mathrm{BS}}^\dagger U_{\mathrm{BS}} = \mathbb{I}$ and has intensity transmittance $T = \cos^2\theta$ and reflectance $R = \sin^2\theta$ [11, 35]. The corresponding action on *creation* operators reads:

$$\hat{a}_{\mathrm{in}}^\dagger \;\longrightarrow\; e^{i\phi}\sin\theta\,\hat{a}_{\mathrm{out}}^\dagger + \cos\theta\,\hat{b}_{\mathrm{out}}^\dagger, \tag{2.8}$$

$$\hat{b}_{\mathrm{in}}^\dagger \;\longrightarrow\; \cos\theta\,\hat{a}_{\mathrm{out}}^\dagger - e^{-i\phi}\sin\theta\,\hat{b}_{\mathrm{out}}^\dagger. \tag{2.9}$$

For a balanced beam splitter we set $\theta = \frac{\pi}{4}$; an overall device phase can be absorbed and is physically irrelevant.

Derivation for $N = 2$. To illustrate how a N00N state can be generated, consider the simplest non-trivial case with $N = 2$. We begin with two single photons entering the two input ports of a balanced 50:50 beam splitter. The input state is

$$|1\rangle_a \otimes |1\rangle_b \;\equiv\; |1,1\rangle, \tag{2.10}$$

meaning that there is one photon in mode a and one in mode b.

Using the balanced version of (2.7) with $\phi = 0$ (a common, convenient choice), the creation operators transform as

$$\hat{a}^\dagger \rightarrow \tfrac{1}{\sqrt{2}}\left(\hat{a}^\dagger + \hat{b}^\dagger\right), \tag{2.11}$$

$$\hat{b}^\dagger \rightarrow \tfrac{1}{\sqrt{2}}\left(\hat{a}^\dagger - \hat{b}^\dagger\right). \tag{2.12}$$

Applying this to $\hat{a}^\dagger\hat{b}^\dagger|0,0\rangle$ gives

$$\hat{a}^\dagger\hat{b}^\dagger|0,0\rangle \rightarrow \tfrac{1}{2}\Big(\hat{a}^\dagger + \hat{b}^\dagger\Big)\Big(\hat{a}^\dagger - \hat{b}^\dagger\Big)|0,0\rangle \tag{2.13}$$

$$= \tfrac{1}{2}\Big(\hat{a}^{\dagger 2} - \hat{b}^{\dagger 2}\Big)|0,0\rangle. \tag{2.14}$$

Thus, the output state after the beam splitter is

$$|\psi_{\text{out}}\rangle = \frac{1}{\sqrt{2}}\Big(|2,0\rangle - |0,2\rangle\Big), \tag{2.15}$$

which is the two-photon N00N state in 2.6 with $\alpha = \frac{\pi}{2}$. Physically, the two photons always exit together in the same output mode but never one per mode: the destructive interference of the $|1,1\rangle$ output term is the hallmark of the Hong–Ou–Mandel effect [26].

In the following, we will use our software *EsQueranto* to simulate this situation and verify numerically the generation of the NOON state.

Listing 2.1: Simulation of a 2-photon N00N state with EsQueranto.

```
from optical_elements import *

cutoff = 3
num_modes = 2

# Define the input state |1,1> as a sparse BCOO tensor
input_idx  = jnp.array([[1,1]], dtype=jnp.int32)
input_data = jnp.array([1.0+0j], dtype=DTYPE_COMPLEX)
input = jsparse.BCOO((input_data, input_idx), shape=(cutoff,)*
   num_modes)

# Define a balanced beam splitter (theta = pi/4, phi = 0)
bs_params = jnp.array([jnp.pi/4, 0.0])
bs_modes  = (0,1)

# Apply the beam splitter
output = apply_bs(input, bs_params, bs_modes)

print(output)
```

The printed output corresponds to the two-photon NOON state:

```
[[0 2]
 [2 0]]
[-7.0710635e-01+0.j   7.0710635e-01+0.j]
```

This means that the state is a superposition of

$$-\frac{1}{\sqrt{2}}|0,2\rangle \ + \ \frac{1}{\sqrt{2}}|2,0\rangle, \tag{2.16}$$

which matches the theoretical result obtained from the beam splitter transformation.

There are some observations about the software worth to make. As one can see, first thing we do is to introduce a `cutoff`, which indicates the number of allowed states of a single mode. Being in the Fock representation, the state of a mode is represented by the number of photons in that mode, and if we want to allow for a maximum number n of photons in a single mode, then `cutoff` $= n + 1$ (we need one state for $|0\rangle$). That's why in the example above we use `cutoff` $= 3$, having to describe modes with $|0\rangle, |1\rangle, |2\rangle$. Also we make use of the `sparse` module built in JAX [48], that we introduce basically to keep memory usage under control. In fact, given a `cutoff` C and a `num_modes` M, the whole Hilbert space has a dimension d of

$$d = C^M, \tag{2.17}$$

which for the case above corresponds to the case $d = 9$, where we have $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, $|02\rangle, |20\rangle, |12\rangle$, $|21\rangle, |22\rangle$. If we represented the whole space through a dense array, we would have to carry around all these kets, even if they have coefficient zero. With the `sparse` representation instead, we just carry the non-zero elements, which in the case of the N00N state are two ($|02\rangle, |20\rangle$).

2.2.3 Second example: quantum teleportation

As a second example we show the realization of a quantum teleportation, that we already theoretically discussed in[], but this time we work in the polarization encoding. In particular we'll replicate the original experiment presented for the first time in [9]. A sketch of the experimental setup can be seen in Figure 2.5. To prepare the initial photon and the initial entangled pair one does simply:

Listing 2.2: Preparation of initial resources.

```
from optical_elements import *

cutoff = 3
num_modes_photon_1 = 2 #(1 physical mode in the circuit, 2
    modes in the code)

# Define the input state 1/sqrt(2)*(|H>+|V>)
photon_1_idx  = jnp.array([[1,0], [0,1]], dtype=jnp.int32)
photon_1_data = jnp.array([1/jnp.sqrt(2)+0j, 1/jnp.sqrt(2)+0j
    ], dtype=DTYPE_COMPLEX)
photon_1 = jsparse.BCOO((photon_1_data, photon_1_idx), shape=(
    cutoff,)*num_modes_photon_1)


num_modes_entangled_pair = 4  #(2 physical modes in the
    circuit, 4 modes in the code)
```

```
# Define the entangled pair 1/sqrt(2)*(|HV>-|VH>)
entangled_pair_idx  = jnp.array([[1,0,0,1], [0,1,1,0]], dtype=
    jnp.int32)
entangled_pair_data = jnp.array([1/jnp.sqrt(2)+0j, -1/jnp.sqrt
    (2)+0j], dtype=DTYPE_COMPLEX)
entangled_pair = jsparse.BCOO((entangled_pair_data,
    entangled_pair_idx), shape=(cutoff,)*
    num_modes_entangled_pair)

state = outer_product(photon_1, entangled_pair) #Complete
    system of 6 modes
```

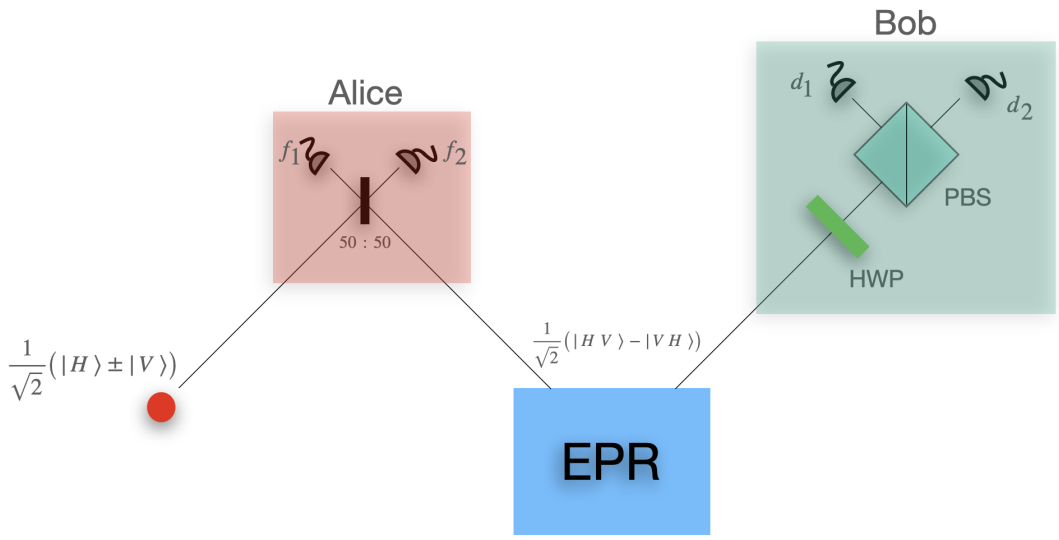


Figure 2.5: Experimental setup to perform a proof-of-principle about quantum teleportation. The single photon can be in both $+45°$ or $-45°$ polarizations.

We immediately notice that due to the choice of working with polarization encoding, we have to duplicate the rails, in order to represent that particular information. No new implementation is needed at software level: the representation of states in the number basis is so general that we can go from single-rail to dual-rail smoothly. Now, at Alice's measurement station (red in Figure 2.5) we perform a BSM as already presented in Figure. Since this is the simple implementation of the BSM, we expect a 25% success probability, because we are able to distinguish just the Bell pair $|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|HV\rangle - |VH\rangle)$ (a more detailed analysis of the BSM through the software is presented in Appendix). In the code this reads:

Listing 2.3: BSM at Alice station.

```
# Define a balanced beam splitter (theta = pi/4, phi = 0)
bs_params = jnp.array([jnp.pi/4, 0.0])
bs_modes_H  = (0,2)
bs_modes_V  = (1,3)
```

```
# Apply the beam splitter on both horizontal and vertical
   modes
state = apply_bs(state, bs_params, bs_modes_H)
state = apply_bs(state, bs_params, bs_modes_V)
```

We should pay attention to the code above, because now that we are in polarization encoding, to apply a BS means to apply it between horizontally polarized and vertically polarized modes, as shown in Figure 2.6.

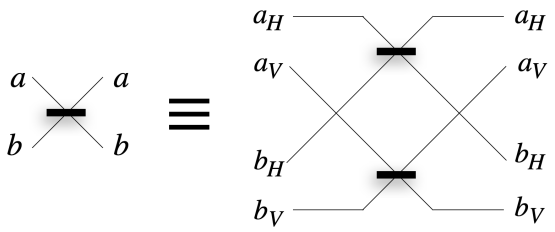


Figure 2.6: Action of a BS in polarization encoding.

At Bob's measurement station we have to analyze photon 3 from the entangled pair, and since we want the teleportation of a $+45°$ oriented photon, we should measure through a PBS oriented at $45°$. To do so, we place an HWP oriented at $\omega = 22.5°$ before a PBS. The HWP can be implemented in the code through a BS (Figure 2.7), due to the fact that it's represented by

$$\begin{pmatrix} \hat{a}^\dagger_{\mathrm{H}} \\ \hat{a}^\dagger_{\mathrm{V}} \end{pmatrix} = \underbrace{\begin{pmatrix} \cos(2\omega) & \sin(2\omega) \\ \sin(2\omega) & -\cos(2\omega) \end{pmatrix}}_{U_{\mathrm{BS}}(\theta,\phi)} \begin{pmatrix} \hat{a}^\dagger_{\mathrm{H}} \\ \hat{a}^\dagger_{\mathrm{V}} \end{pmatrix}, \qquad \omega \in [0, \tfrac{\pi}{2}], \tag{2.18}$$

and comparing 2.18 with 2.7 we notice that they are equal for $\theta = \frac{\pi}{2} - 2\omega,\ \phi = 0$. Also the PBS can be implemented through a BS in the code (Figure 2.8), being simply a swap over the horizontal modes. We write therefore:

Listing 2.4: Measurement at Bob station.

```
# Define a HWP oriented at 22.5^\circ --> theta = pi/4, phi=0
hwp_params = jnp.array([jnp.pi/4, 0.0])
hwp_modes = (4,5)

state = apply_bs(state, hwp_params, hwp_modes)

# The PBS has a vacuum entry, so we need to create a two modes
    vacuum first
vac = create_vac(2)  #function that generates a n-modes vacuum
    simply calling create_vac(n)

state = outer_product(state, vac) #Now 8 modes in total

# Apply the PBS
pbs_params = jnp.array([0.0, 0.0])
```

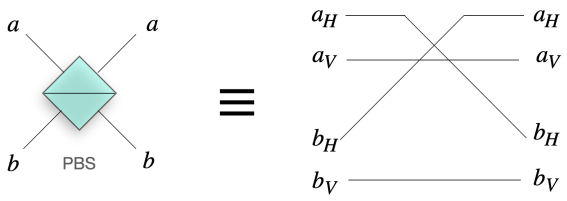


Figure 2.8: Action of a PBS in polarization encoding.

```
pbs_modes = (4,6)
state = apply_bs(state, pbs_params, pbs_modes) #no action on
    the vertical modes
```

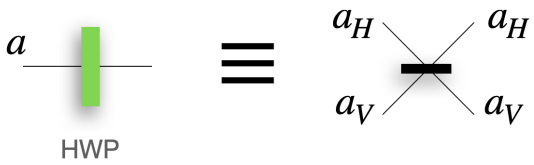


Figure 2.7: Action of a HWP in polarization encoding.

Now it's left to measure the three-fold coincidences at the detectors. Starting with a $+45°$ polarized photon, we expect to measure the teleported photon in detector d_2, while measuring dark in detector d_1. This of course given that we are measuring a coincidence detection in f_1 and f_2. We simply write

Listing 2.5: Detection to verify teleportation.

```
# Define the detectors modes
det_modes = jnp.array([[0,1], [2,3], [4,5], [6,7]]) #We
    indicate the mode pairs over which each single detector
    measures: f_1, f_2, d_1, d_2

#We measure with threshold detectors
probs = det_threshold(state, det_modes)

print(probs)
```

The printed output corresponds to our expectations:

```
1101: 0.250
1110: 0.000
```

where 1 indicates a measurement in the corresponding detector, and 0 means dark. Of course, if we restarted the procedure by choosing as initial state $\frac{1}{\sqrt{2}}(|H\rangle - |V\rangle)$, so the photon polarized at $-45°$, we would get by the same procedure swapped final detections:

```
1101: 0.000
1110: 0.250
```

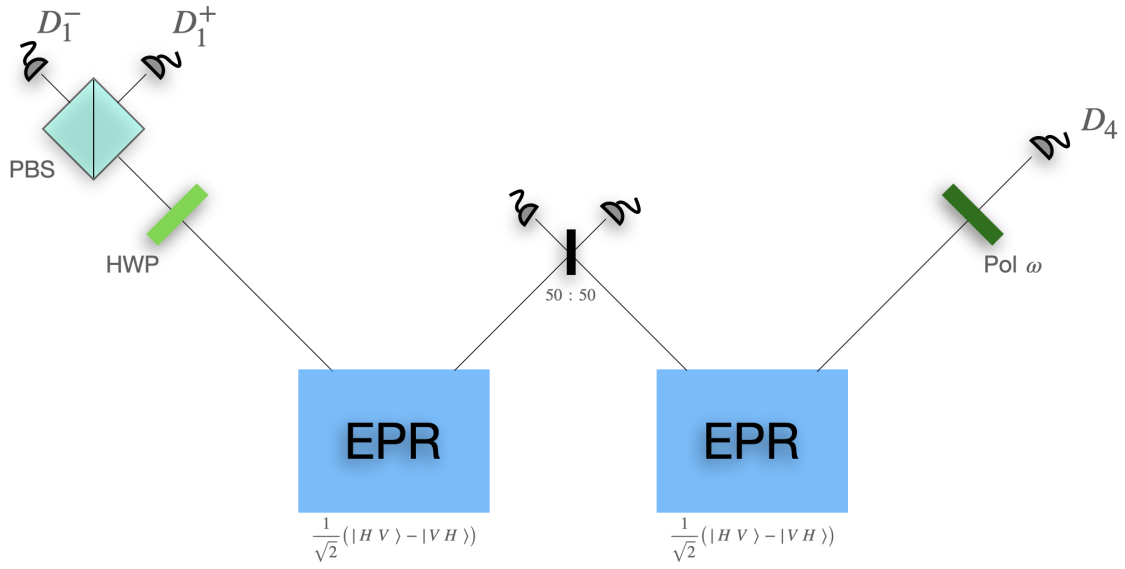


Figure 2.9: Experimental setup for the proof of principle of entanglement swapping.

2.2.4 Third example: entanglement swapping

As a third and last example we show how to realize another major experiment: the first realization of entanglement swapping, presented in [39]. The sketch for the experiment is shown in Figure 2.9. We already know how to implement everything beside the polarizing filter, but we'll just use an HWP considering the horizontal output as the surviving part. Since the initial state is $|\Psi^-\rangle_{12}|\Psi^-\rangle_{34}$, analogously to what we saw in [] we can rewrite it as

$$|\Psi^-\rangle_{12}|\Psi^-\rangle_{34} = \frac{1}{2}\big(|\Phi^+\rangle_{14}|\Phi^+\rangle_{23} + |\Phi^-\rangle_{14}|\Phi^-\rangle_{23} + |\Psi^+\rangle_{14}|\Psi^+\rangle_{23} + |\Psi^-\rangle_{14}|\Psi^-\rangle_{23}\big), \tag{2.19}$$

and again with a 25% BSM we can individuate the state $|\Psi^-\rangle_{23}$, which implies that photons 1 and 4 should be in state $|\Psi^-\rangle_{14}$. If this is true, then their polarizations should be orthogonal in any polarization basis, and in particular we decide to measure photon 1 along the $+45°$ (D_1^+), $-45°$ (D_1^-) axes. At a varying value of ω one should observe for the two four-fold coincidence detections of $D_1^+D_4$ and $D_1^-D_4$ two sine curves out of phase of $\frac{\pi}{2}$. We write

Listing 2.6: Entanglement swapping in EsQueranto.

```
# Define the initial state
cutoff=3
num_modes_entangled_pair = 4  #(2 physical modes in the
   circuit, 4 modes in the code)
entangled_pair_idx  = jnp.array([[1,0,0,1], [0,1,1,0]], dtype=
   jnp.int32)
entangled_pair_data = jnp.array([1/jnp.sqrt(2)+0j, -1/jnp.sqrt
   (2)+0j], dtype=DTYPE_COMPLEX)
entangled_pair = jsparse.BCOO((entangled_pair_data,
   entangled_pair_idx), shape=(cutoff,)*
```

```
    num_modes_entangled_pair)

state = outer_product(entangled_pair, entangled_pair) #
    Complete system of 8 modes

#BSM on photons 2&3
bs_params = jnp.array([jnp.pi/4, 0.0])
bs_modes_H  = (2,4)
bs_modes_V  = (3,5)
state = apply_bs(state, bs_params, bs_modes_H)
state = apply_bs(state, bs_params, bs_modes_V)

#Measurement on photon 4
hwp_params = jnp.array([3*jnp.pi/4, 0.0])
hwp_modes = (0,1)
pbs_params = jnp.array([0.0, 0.0])
pbs_modes = (8,9)

state = apply_bs(state, hwp_params, hwp_modes)
vac = create_vac(2)
state = outer_product(state, vac)
state = apply_bs(state, pbs_params, pbs_modes)


#Measurement on photon 1 and four-fold coincidence
det_modes = jnp.array([[0,1], [2,3], [4,5], [6], [8,9]])

def run_for_omega(state, omega):
    # BS params depending on omega
    pol_filter_params_omega = jnp.array([jnp.pi/2 - 2*omega,
        0.0])
    state = apply_bs(state, pol_filter_params_omega, (6,7))
    # Compute probabilities
    return det_threshold(state, det_modes)

# Vectorize over a grid of omega
num_points = 100  # resolution of sweep
omegas = jnp.linspace(0.0, jnp.pi, num_points)

probs_all = jax.vmap(lambda w: run_for_omega(state, w))(omegas
    )
```

The original result by Zeilinger group and the result of our experiment in EsQueranto are shown in Figure 2.10.

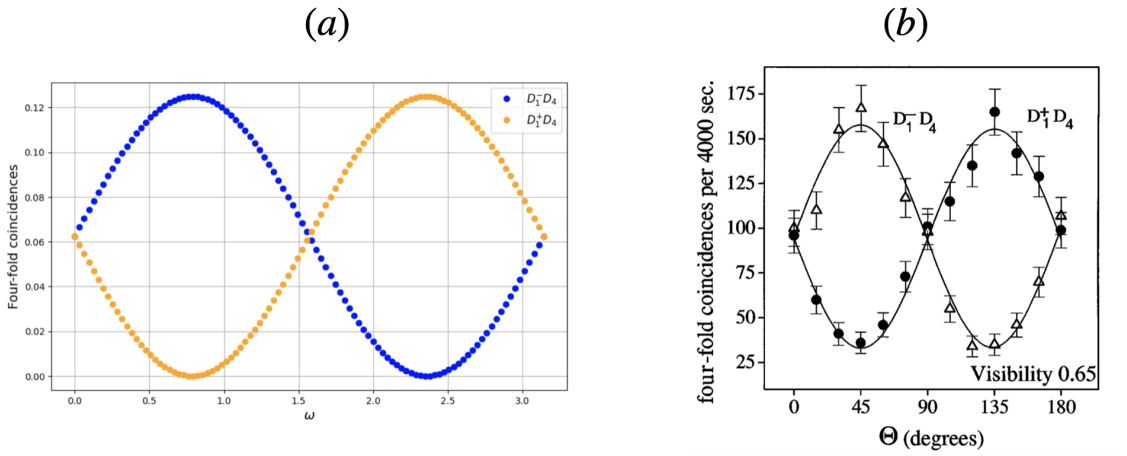


Figure 2.10: Comparison between the simulated experiment result in *EsQueranto* (a), and the original result by Zeilinger group (b). We perfectly recover the two sine curves shifted by a factor $\frac{\pi}{2}$.

Looking at this last piece of code it's interesting to notice the use of `vmap` instead of a more trivial `for` loop that one would try to implement in Python.

We decided to show these three examples as they are experimental simulations of phenomena that are particularly relevant in the context of quantum networks. However, they are not representative of all the simulation tools present in EsQueranto. In fact, in the original propsal [25], explicit use of squeezing effects (SPDC) is made to enhance quantum telescopes.

Chapter 3

Discovery with EsQueranto

3.1 Towards rediscovery

In the first two chapters, we provided an introduction to quantum networks, their significance, and their underlying components. We then discussed various AI models that have been used over the years in the simulation and discovery of new experimental apparatus, finally introducing our new simulator, *EsQueranto*, which we now want to apply to quantum networks. In this chapter, we will see how to conceive of the problem in the most general way possible, both from the point of view of the elements of the experimental apparatus and in terms of the strategy (i.e., the protocol) to be followed in order to use them. We will then describe how *EsQueranto* is used to address the discovery problem and finally present the results obtained, highlighting the strengths and current limitations.

3.1.1 Notable examples of ansatz

Let us begin by noting that producing extremely general ansatzes to discover experimental blueprints is an idea that has been around for some years, and in fields relatively distant from ours. We can cite, for example, the discovery of new designs for super-resolution microscopes by XLumina [44] or the numerous series of new gravitational wave detectors proposed in the Gravitational Wave Detector Zoo [29]. The ansatz of the former is shown in Figure 3.1, that of the latter in Figure 3.2. Without going into the physical details of these two approaches, what we are interested in noting is their structure based on fundamental constituent units that can then be repeated an arbitrary number of times, enlarging the complexity of the solution space at will. The salient feature of these diagrams is their ability to represent the experimental apparatus devised and already known to scientists (an example for the gravitational wave detector is shown in Figure 3.3). It is precisely this ability that we want to try to replicate in our case.

3.1.2 Our proposal for an ansatz

Let us begin by addressing the issue of the most general possible representation of the apparatus, understood as the set of optical instruments that operate on photons.

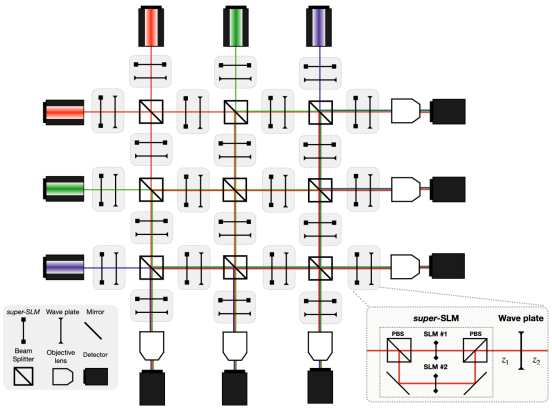


Figure 3.1: XLumina ansatz for the discovery of new-generation microscopes.

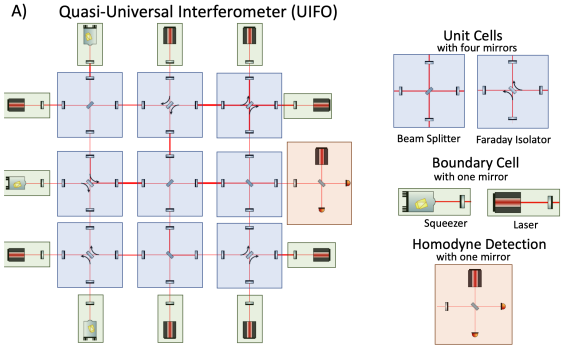


Figure 3.2: UIFO (universal interferometer) nsatz for the discovery of new gravitational waves detectors.

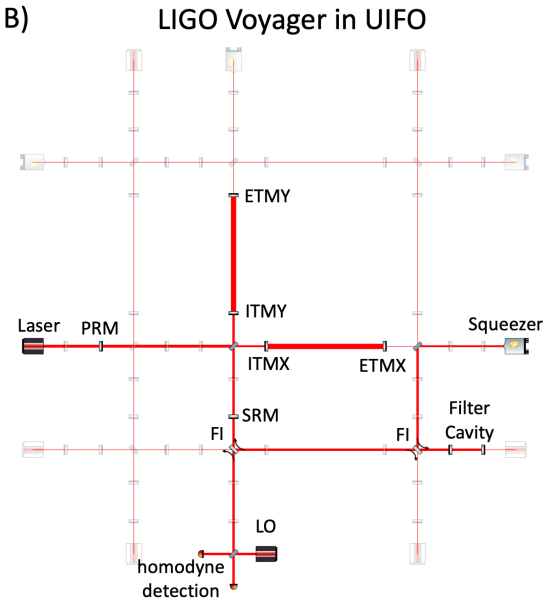


Figure 3.3: How to represent a known human solution (LIGO Voyager) in the UIFO ansatz.

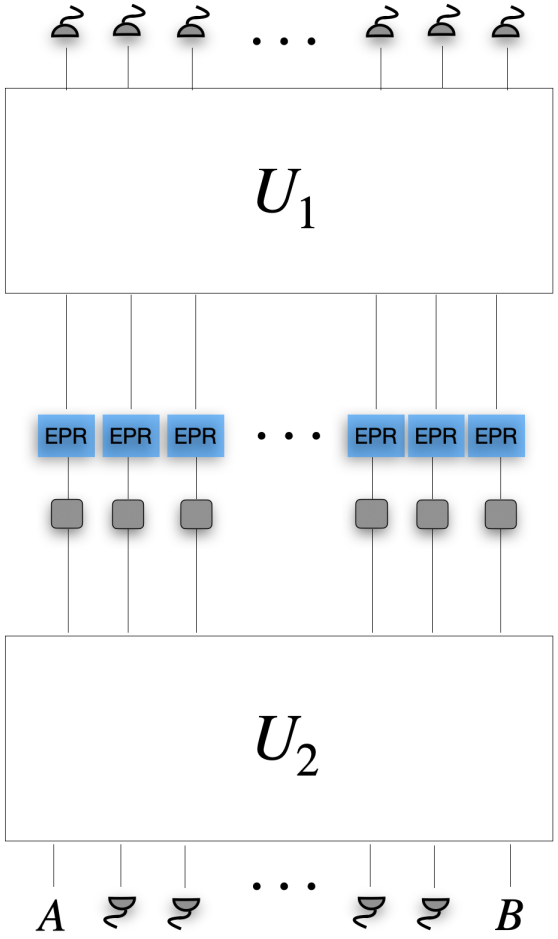


Figure 3.4: Our ansatz for representing all the repeater-like structures. We have in principle an arbitrary number of EPR sources, memories (in gray) and detectors.

Later on, we will address the question of an ansatz for the protocol. Recalling the structure of the quantum repeater seen in previous chapters, let us imagine our space as shown in Figure 3.4. Our goal is to allow photons, both those released immediately into the circuit and those released from quantum memories, to interact as fully as possible. To reduce complexity (for reasons that will be clarified later), we decide to start directly with EPR sources that generate states $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|HH\rangle + |VV\rangle$, and not to allow photons that will be written into quantum memories to interact before this happens. One could wonder why we choose to work with polarization encoding rather than Fock encoding, since the former is much more expensive in terms of number of modes we have to represent. The reason is closely linked to the nature of the system we are studying. In systems that involve loss, working with entangled states between a single photon and the vacuum makes it extremely difficult to recognise the loss correctly, because it converts one element of the basis ($|1\rangle$) into another ($|0\rangle$), effectively constituting a bit flip. Crucially, we are no longer able to determine whether the bit flip we observe is due to a deliberate operation in the circuit or whether it is due to the simple loss of the photon. This does not happen in dual-rail encoding, because the loss sends $|10\rangle, |01\rangle$ to $|00\rangle$, and we are immediately able to flag the loss and act accordingly.

We must now clarify the components of this ansatz one by one.

3.1.2.1 Universal multiport interferometer

In the ansatz shown in Figure 3.4, we do not specify the form that the unitary describing the interaction between photons should take in terms of optical tools. Fortunately for us, in physics it is well known that any unitary can be achieved through a set of beam splitters and phase shifters, as first proposed in [43] and then perfected in [13].

The concept underlying these constructions is that of a *universal multiport interferometer*. By this expression one refers to an optical network that can implement, in principle, *any* unitary transformation

$$U \in \mathbb{C}^{m\times m}, \qquad U^\dagger U = \mathbb{I}_m, \tag{3.1}$$

acting on m input modes of light. Since the evolution of non-interacting photons through a passive linear-optical circuit is fully characterized by such an $m \times m$ unitary, the ability to synthesize arbitrary U directly implies universality of linear optics.

The Reck decomposition [43] provides a constructive proof of this statement: starting from an arbitrary target unitary U, one systematically eliminates its off-diagonal entries by means of a sequence of two-mode operations, each corresponding to a tunable beam splitter followed by appropriate phase shifters. This procedure is mathematically equivalent to a QR decomposition [22] of U into a product of elementary unitaries acting only on two neighboring modes:

$$U = \prod_{k=1}^{m(m-1)/2} T_k, \tag{3.2}$$

where each T_k is a 2×2 transformation embedded in the m-dimensional Hilbert space. A beam splitter unitary we saw in Eq., and arbitrary phase shifters are used to set relative phases between modes. Hence, by chaining such building blocks one can reproduce the action of any U.

From a practical perspective, the Reck scheme arranges these beam splitters in a *triangular* geometry, in which mode 1 interacts sequentially with all other modes, followed by mode 2, and so on, as shown in Figure 3.5. While elegant, this geometry is not ideal, as it leads to long optical paths and unbalanced propagation losses [12, 49].

For this reason, Clements *et al.* [13] proposed an alternative decomposition that rearranges the same set of two-mode transformations into a *rectangular* geometry (see Figure 3.6). This guarantees equal optical depth across all paths, reducing circuit depth to $O(m)$ and improving stability against fabrication imperfections. In this architecture, each "layer" of the interferometer implements $m/2$ disjoint beam splitter operations in parallel, and layers alternate between odd-even and even-odd mode pairings. Both decompositions are universal and employ the same number of beam splitters, namely $m(m-1)/2$. However, they differ significantly in how errors propagate. In the triangular Reck scheme, an error in an early beam splitter affects all downstream operations, while in the Clements scheme errors are more evenly distributed, as Figure 3.7 illustrates.

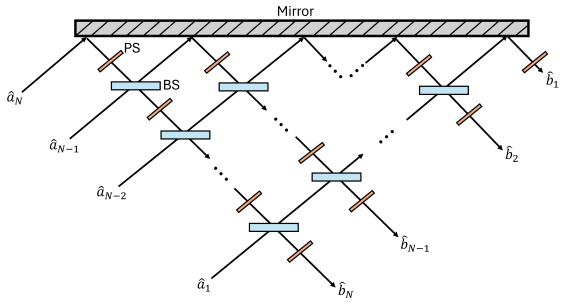


Figure 3.5: Reck-Zeilinger triangular scheme for the implementation of any unitary.

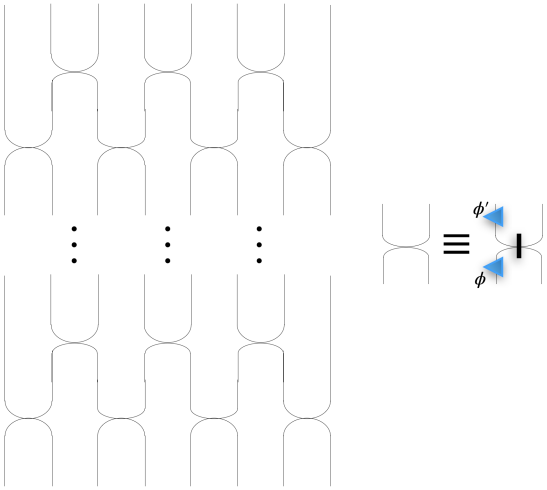


Figure 3.6: Clements rectangular scheme to implement any unitary. The basic component is a BS with phase shifters at input ports and output ports.

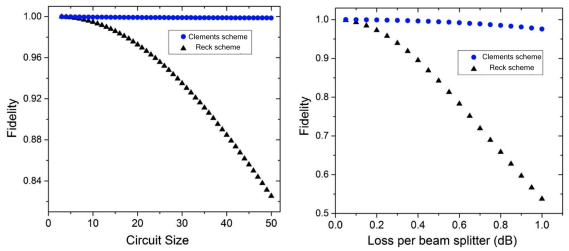


Figure 3.7: Left: average fidelity for an interferometer with a constant loss of 0.2 dB per beam splitter, for different interferometer sizes. Right: fidelity as a function of loss for interferometers implementing transformations. We see from these results that Clements design is much more loss-tolerant than the Reck design and maintains high fidelity with the target unitary even in the case of high loss.

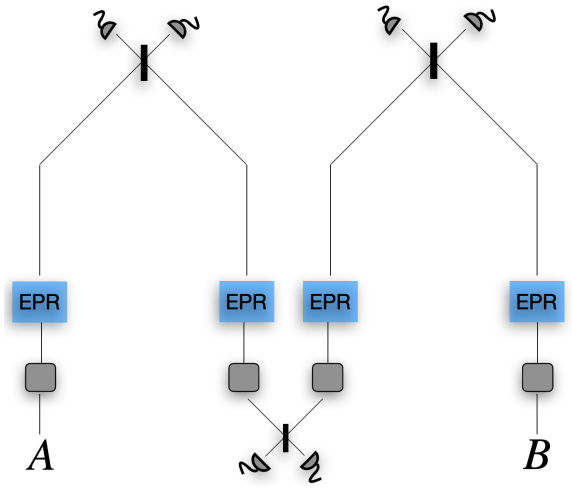


Figure 3.8: Representation of the quantum repeater setup. Here the two external sources are local with A and B respectively, while the repeater sources are in the middle, $\frac{L}{2}$. Measurements between the two external sources and the repeater happen at $\frac{L}{4}$ and $\frac{3L}{4}$ respectively.

3.1.2.2 Introducing loss

At this point, we might believe that using the diagram in Figure 3.6 to represent the unitaries in Figure 3.4 is sufficient to provide a fairly general description of the system. However, this ceases to be true if we decide to introduce photon loss. As already mentioned several times, we are considering photons travelling through optical fibres, and optical fibres can be schematised as pure-loss bosonic channels described by Eq. 1. Now, the most straightforward assumption we make is that each element of this scheme is connected to each of its neighbours via an optical fibre, which will introduce a loss if the two connected elements are not in the same position (i.e. there is a non-zero displacement). Limiting ourselves to a 1-D geometry, this means that by assigning each element of the scheme a position x_i, we can schematize the path of the photon-mode in the network by looking at the sequence of positions of the elements that mode goes through. This is easy to understand by looking at the example for the single repeater in Figure 3.8. If we tried to embed such a scheme in the general ansatz using the Reck-Clements representation, we would hit the contradiction of Figure 3.9. Therefore it is necessary to enlarge our ansatz, by adding some vacuum ancilla that allow us to control the geometry of the system without losing generality. A working ansatz is shown in Figure 3.10.

3.1.2.3 Making sense of the detectors

Now that we have a general approach, which is also capable of representing the geometry of the system, all that remains is to discuss the last optical element we use: the detector. Detection plays a fundamental role in this type of scheme, especially if we consider the objective of our quantum network, which is to share an entangled pair between two parties separated in space, so that they can use it for subsequent operations. This implies that the photons constituting the shared pair must not be

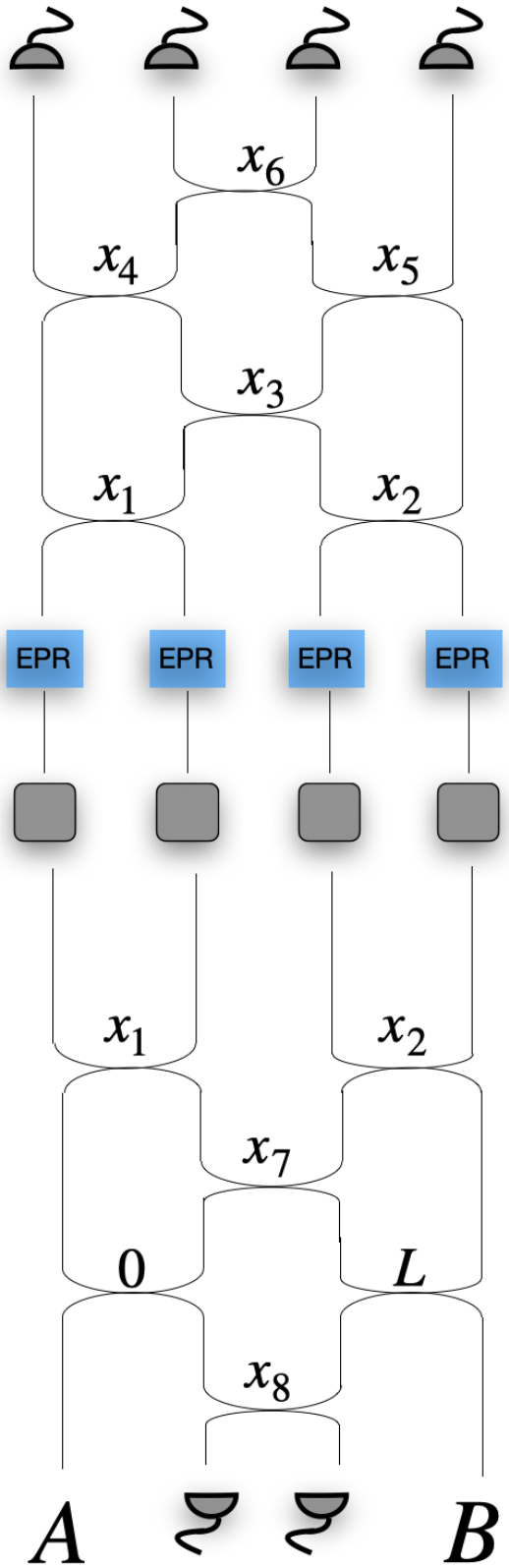


Figure 3.9: Minimal use of the Clements scheme as an ansatz. We consider just one mode per photon, and we fix the position of only A (0) and B (L). We realize that we cannot correctly reproduce the geoemtry of the quantum repeater, because the first two sources and the last two are forced to be local within each other (the first two have position x_1, the second two x_2. By the same logic, we cannot make the two BSM local, as one detector will be in x_4 or x_5 and the other in x_6, and vice-versa. The only possibility to have local BSM would be $x_4 = x_5 = x_6$, but this means tha we cannot materialize the repeater node.

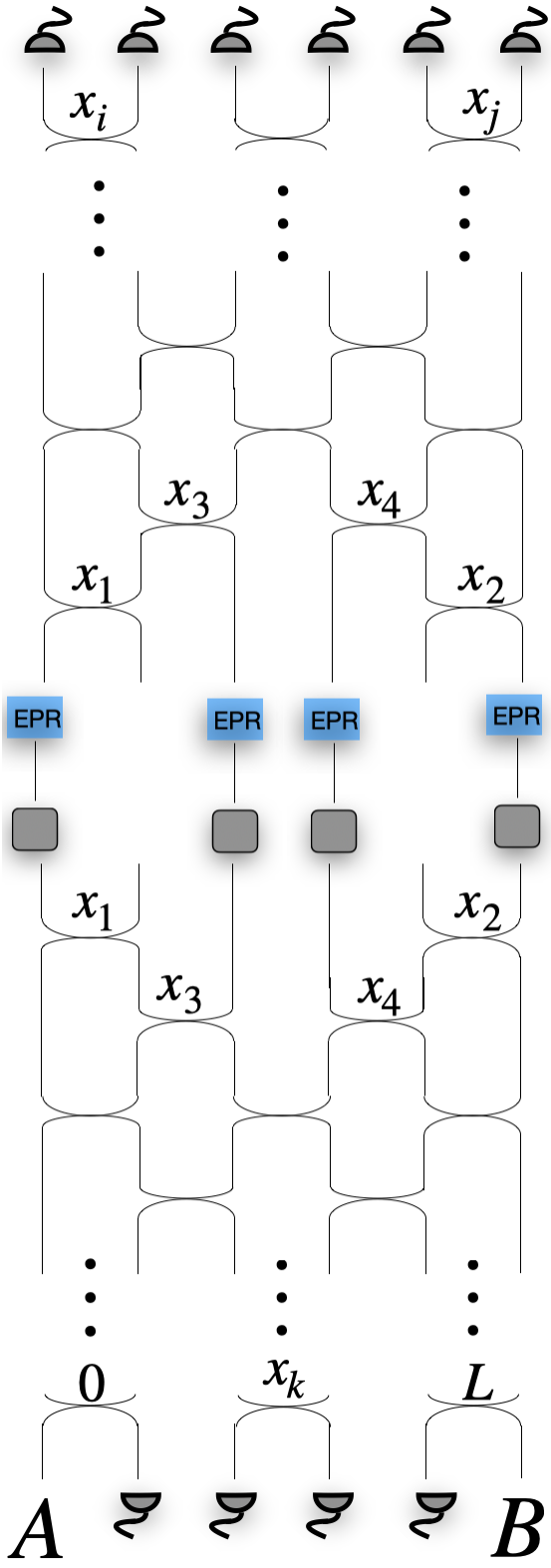


Figure 3.10: Minimum general and working ansatz if we want to make all the photons interact while allowing for non-trivial geometries. Now it's easy to see how all sources can safely occupy their own position, and also BSM can happen locally (x_i, x_j, x_k). We are at the same time allowing all orders of interactions between photons in the two circuits.

measured, since measuring them at the detectors would destroy them, thus preventing their further use. It is for this reason that we speak in these cases of heralding, as opposed to post-selection, in which all photons are measured at the detectors. The difference between the two methods is shown in the figure, and actually we can find an example of each already in this work; in fact, the generation of an entangled pair through BSM that occurs in quantum repeaters is a classic example of heralding, while the experiment used to demonstrate entanglement swapping itself [39] (and which we reproduced with EsQueranto) uses post-selection, measuring all the photons involved. In the latter, post-selection is required because it is a proof of principle experiment, but in general, quantum technologies must be based on heralding, otherwise we would not be able to use any of the states we generate.

Broadly speaking, there are two categories of detectors that we can consider: *threshold* and *photon-number-resolving* (PNR). Measurements with threshold detectors provide a binary outcome, which is usually the presence or absence of photons. PNR detectors, on the other hand, are capable of distinguishing and quantifying the number of photons measured at a given time. In the following we'll consider threshold detectors, in the spirit of what's typically done with dual-rail encoded photons. In the last chapter we'll however use PNR for some interesting applications.

3.1.3 The most general protocol

While for the works we mentioned earlier [44, 29], an experimental apparatus ansatz is essentially all that is needed to perform optimization aimed at discovering new setups, in our case there is a non-trivial level of decision-making that needs to be represented in some way. In fact, if we consider the quantum repeater, the quality of the scheme is not only measured in terms of the state we can generate through the use of all the resources involved, but is clearly characterized by the fact that, based on the detection pattern observed, we must make different choices. With reference to Figure 3.8, we can describe the quantum repeater protocol in these terms:

- All sources fire an EPR pair; one photon from each pair is written to a quantum memory, while the other is sent to a BSM station.

- If we observe photons in all four detectors, this indicates that both BSMs were successful, and therefore all memories are in the correct state and can release the photons (we call this *complete success*).

- If, on the other hand, we observe only one pair of detectors clicking, while in the other only one detector clicks, or neither of them, this means that only one of the BSMs was successful, and therefore only two memories will be in the correct state (we call this *partial success*). This implies that a non-trivial choice must be made, namely to fix the result in one part of the scheme and reinitialize only the other section.

- If instead no pair of detectors clicks together, meaning no BSM was successful (we call this *failure*), we restart from scratch.

- When all the memories are ready we release them, performing another BSM with the middle photons; if we observe a double click (we call this *success*) we are done, otherwise, if we fail the BSM (again *failure*) we restart from scratch.

If we now look at the general scheme we set as our ansatz in Figure 3.4, we can outline **the most general protocol** based on all the discrete events that may occur during a round of experiment and the corresponding choices we could make. Let's enumerate the general procedure:

1. Initially all the sources are connected to the network and they fire in parallel, while the memories are empty;

2. We observe a certain detection pattern at the detectors of the first unitary, and based on that we can decide to start again from scratch, keep some memories fixed and restart the others, keep all the memories and move to the second unitary;

3. To start again from the beginning means that we throw away the information stored in all the memories and we fill them up again;

4. Keeping some memories fixed means that we have heralded the desired creation of states in those, but not in the others, so we have to repeat some processes until we herald the desired states also in the missing memories;

5. Immediately moving to the second unitary means that we heralded the creation of the desired states in all the memories, and we can release the information contained there in the rest of the network;

6. We release the information contained in the memories simultaneously (the first and second unitary are not active at the same time) and we herald through the detectors the production of the desired state between A and B. If we get the correct heralding the procedure is successful, otherwise we have to restart from scratch.

A summary diagram of such a general protocol is shown in Figure 3.11.

3.1.4 Engineering the loss function

We now have a sufficiently general scheme to represent a broad range of possible experimental setups (which also contains the known solutions), and a logical procedure that tells us what possibilities we have for each observed event. We now need to translate these tools into mathematical objects that can be easily used in an optimization problem. We are actually missing a loss function. It is quite clear that there are two fundamental elements to focus on: the first is network efficiency, i.e. the time required to perform the entire entangled pair sharing operation; the second is ensuring that the generated pair has a very high fidelity with respect to the Bell pair we want to generate. We can therefore imagine a loss function of the type

$$\mathcal{L} = E + \alpha \ \text{softplus}(F_{th} - F), \tag{3.3}$$

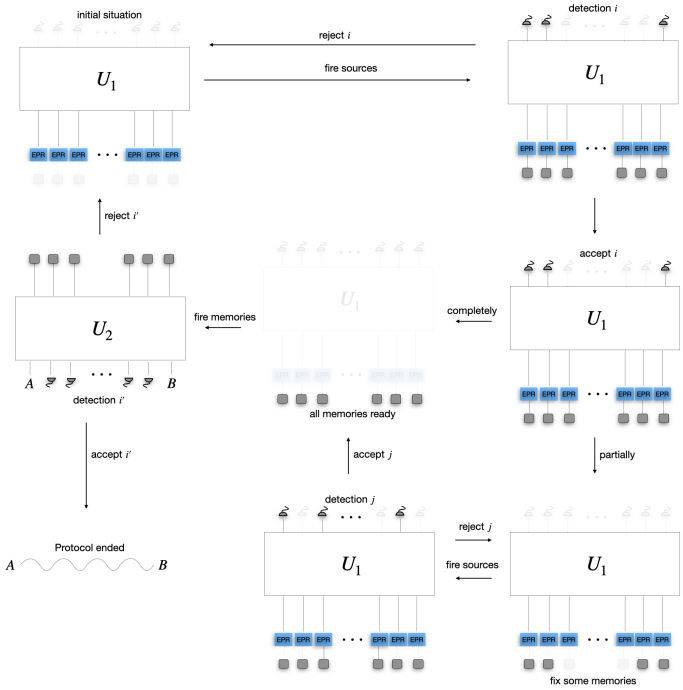


Figure 3.11: Visual description of the logic behind the most general protocol. At each step, active and non-active elements are individuated by colours contrast.

where E is once again the expected number of trials, i.e. the expected number of times we have to make measurements before complete success, F is the fidelity with respect to our target state, F_{th} is the fidelity threshold, i.e. the value above which we try to push our fidelity F and α is the weight of the fidelity penalization with respect to E. In fact, the softplus function [16] is thought exactly as a smooth penalization that forces the optimizer to go above the threshold, because it reads

$$\text{softplus}(x) = \ln(1 + e^x). \tag{3.4}$$

Minimize 3.3 means trying to minimize both terms of the sum, which consists in the minimization of E and the maximization of F. What's left to see is how we can make these quantities suitable for a continuous parameters optimization, while also representing the generality expressed by the general ansatz and the most general protocol.

3.1.4.1 Expected number of trials E

In order to reach a convincing expression for E, we can exploit the intrinsically discrete nature of our measurement-based protocol. If we forget for a moment about the physical interactions that occur between photons in unitary circuits, and limit ourselves to viewing them as black boxes, we realize that the information we obtain from each experiment is that of discrete events (detection patterns), each of which occurs with a certain probability.

Considering the first unitary, let's say that we have D detectors. We can consider the space of all the possibilities as composed by events $i \in [0, 2^D - 1]$, where i is a

detection pattern. With such a finite partition of the solution space we can write the **law of total expectation** in the form [52]:

$$E[X] = \sum_i p_i E[X|i], \tag{3.5}$$

where X is the random variable associated to the number of trials, p_i represents the probability of ending in the detection pattern i, and $E[X|i]$ is the expected number of trials to reach complete success once we observed configuration i.

We should now show how continuous parameters describing the setup and the protocol can be slipped inside this formulation. First of all, we anticipate that we'll use *EsQueranto* to simulate the ansatz, and each simulation will produce final states, each with a probability p_i. Therefore these probabilities are the parameters that implicitly represent all the experimentally tunable sets of parameters (θ, ϕ, x), which do not appear explicitly in the loss function, but do contribute in the value of the probability. Then, in order to account for the decision making process, it's convenient to introduce a series of weights parameters that describe our ability of accepting or rejecting an outcome. In particular, with reference to Figure 3.12 and to the enumerated rules of the most general protocol, we say that

1. When we observe a detection pattern i at the detectors of the first unitary, weights w_i ($0 \leq w_i \leq 1$) represents the free choice of the laboratory to accept configuration i (meaning that when we observe it we store all or part of the memories) or to reject it (meaning that we don't store anything);

2. If we accepted i, we can decide to keep all the memories or fix just part of them, and this is described by weights s_i ($0 \leq s_i \leq 1$);

3. To decide which memories to keep fixed and which not is indeed a non-trivial choice, therefore we say that the possibility of keeping fixed a subset $m \in [0, 2^M - 1]$ of memories among the total M after observing i is represented by the weight $a_{i,m}$ (they satisfy $0 \leq a_{i,m} \leq 1$ and also we must require $s_i + \sum_m a_{i,m} = 1, \forall i$ because they are all the possibilities after observing i);

4. After having observed i and decided to keep fixed m, we still have to fill the complementary $\bar{m}$, and we do that by repeating the experiment, but now accepting (or rejecting) in general a detection j happening with probability p_j; the free choice of the laboratory to accept that configuration is represented by $c_{i,m,j}$ ($0 \leq c_{i,m,j} \leq 1$);

5. Once all the memories are ready, we move to the second unitary and we observe a detection i' with probability $p_{i'}$, either accepting it and ending the procedure or rejecting it and restarting from scratch with free weights $w_{i'}$ ($0 \leq w_{i'} \leq 1$).

We finally know how to represent continuously all the decision making and setup, so let's make those parameters appear explicitly in Eq. 3.5. We can actually write

$$E[X|i] = w_i(1 + C_i) + (1 - w_i)(1 + E[X]), \tag{3.6}$$

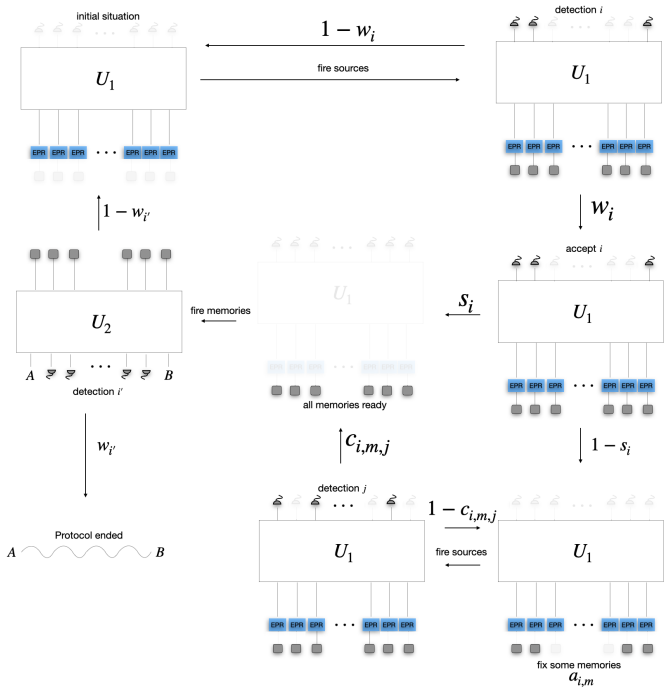


Figure 3.12: Updated version of the scheme in Figure 3.11, highlighting the role of the continuous parameters in the decision-making process.

where C_i is

$$C_i = (1 - s_i) \sum_j p_j \sum_m a_{i,m} \big[c_{i,m,j} + (1 - c_{i,m,j})(1 + C_i) \big]. \tag{3.7}$$

The term structure (1+...) indicates the trial spent observing a detection and storing in the memories. Eqs. 3.5, 3.7 describe the count of expected number of trials for the first unitary (i.e. the expected number of trials to correctly load the memories, that we from now on indicate with E_1). By a similar approach, we can write the law of total expectation for the second unitary (we call it from now on E_2), which result in a much simpler structure, not having to deal with memories:

$$E_2 = \sum_{i'} p_{i'} (1 - w_{i'})(E_1 + E_2). \tag{3.8}$$

Here we ignore the 1+ spent before because there is no control over the memories (in general we should say that there is no time spent to wait for a classical signal to tell us what the next move is). Of course now if we fail we have to restart from scratch, that's why the term $E_1 + E_2$. We can rewrite all the above equations in a more compact form (see Appendix):

$$C_i = \frac{(1 - s_i)^2}{1 - (1 - s_i) B_i}, \tag{3.9}$$

where $B_i = \sum_j p_j \sum_m a_{i,m}(1 - c_{i,m,j})$,

$$E_1 = \frac{1 + \sum_i p_i w_i C_i}{\sum_i p_i w_i}, \tag{3.10}$$

$$E_2 = \frac{1 - \sum_i p_i w_i}{\sum_i p_i w_i} E_1, \tag{3.11}$$

and we finally compute the global E as

$$E = E_1 + E_2. \tag{3.12}$$

This derivation can result quite heavy, but we show now that it's in fact quite intuitive to use.

As we mentioned in the first chapter, the expected number of trials for a quantum repeater is the one in Eq., and we said that different strategies are known to compute it explicitly. Here we show a novel computation of that result through our newly designed E.

Symbolical derivation of Eq. Let's again consider the scheme of Figure. We indicate a clicking detector with C and a non-clicking detector as N. As we said before, we can either herald two memories at a time, or even all four memories together, through successful BSM (considering 25% success probability the success is a coincidence clicking in detector pairs); this corresponds to choosing:

1. $w_{CCCC}, w_{CCNC}, w_{CCCN}, w_{NCCC}, w_{CNCC}, w_{CCNN}, w_{NNCC} = 1$, and all the other weights w_i equal zero.

2. With pattern $CCCC$ we have heralded all the memories, so we want to stop and immediately go to the second unitary. This implies choosing $s_{CCCC} = 1$, and therefore $C_{CCCC} = 0$.

3. With patterns $CCNC$,$CCCN$ and w_{CCNN} we have heralded the first two memories, but we cannot stop immediately, since we still have to herald the last two. Therefore we set $s_{CCNC},\ s_{CCCN},\ s_{CCNN} = 0$. We keep fixed the memories 1 and 2, and that means setting $a_{CCNC,m_{12}},\ a_{CCCN,m_{12}},\ a_{CCNN,m_{12}} = 1$.

4. Now, when we consider the heralding of the other two memories in any of the three situations, we know that the what we want is a double click in detectors 3 and 4, therefore we'll impose $c_{CCNC,m_{12},**CC},\ c_{CCCN,m_{12},**CC},\ c_{CCNN,m_{12},**CC} = 1$($**$ indicates that we are looking just at the last two detectors, so whatever combination of C and N on the first two is irrelevant). The same argument can be repeated for the patterns $NCCC$, $CNCC$ and $NNCC$ that herald the second two memories.

5. In the second unitary, the only weight which is non-zero is $w_{CC} = 1$, the condition to have a successful swapping.

With all the weights set correctly, we can rewrite 3.10 as:

$$E_1 = \frac{1+p_{CCNC}C_{CCNC}+p_{CCCN}C_{CCCN}+p_{CCNN}C_{CCNN}+p_{NCCC}C_{NCCC}+p_{CNCC}C_{CNCC}+p_{NNCC}C_{NNCC}}{p_{CCCC}+p_{CCNC}+p_{CCNN}+p_{CCCN}+p_{NCCC}+p_{CNCC}+p_{NNCC}},$$

and 3.7 as

$$\begin{aligned}
C_{CCNC} &= p_{**CC,m_{12}} + (1 - p_{**CC,m_{12}})(1 + C_{CCNC}),\\
C_{CCCN} &= p_{**CC,m_{12}} + (1 - p_{**CC,m_{12}})(1 + C_{CCCN}),\\
C_{CCNN} &= p_{**CC,m_{12}} + (1 - p_{**CC,m_{12}})(1 + C_{CCNN}),\\
C_{NCCC} &= p_{CC**,m_{34}} + (1 - p_{CC**,m_{34}})(1 + C_{NCCC}),\\
C_{CNCC} &= p_{CC**,m_{34}} + (1 - p_{CC**,m_{34}})(1 + C_{CNCC}),\\
C_{NNCC} &= p_{CC**,m_{34}} + (1 - p_{CC**,m_{34}})(1 + C_{NNCC}).
\end{aligned}$$

for all the relevant patterns. Considering that $p_{**CC,m_{12}},\ p_{CC**,m_{34}} = p$ (accordingly to Eq. we indicate with p the probability of establishing a Bell pair between memories on the same side), we easily find that

$$C_{CCNC} = C_{CCCN} = C_{CCNN} = C_{NCCC} = C_{NCCC} = C_{NNCC} = \frac{1}{p}.$$

Now we observe that $p_{CCNC} + p_{CCCN} + p_{CCNN} = p(1-p)$ and similarly $p_{NCCC} + p_{CNCC} + p_{NNCC} = p(1-p)$, because they correspond to the successful heralding of two memories, while the other two are not successfully heralded. With the same logic of course we have $p_{CCCC} = p^2$. If we plug all these terms in the above expression for E_1 we find

$$E_1 = \frac{1 + 2p(1-p)\frac{1}{p}}{p^2 + 2p(1-p)},$$

which equals to

$$E_1 = \frac{3 - 2p}{p(2-p)}.$$

Now we compute E_2, which is just

$$E_2 = (1 - p_{CC})(E_1 + E_2),$$

and we can rewrite it as

$$E_2 = \frac{1 - p_s}{p_s} E_1,$$

observing that p_{CC} corresponds to p_s(i.e. the probability of a successful final swapping). Finally, computing the total expectation value through 3.12

$$E = E_1 + E_2 = E_1 + \frac{1 - p_s}{p_s} E_1 = \frac{E_1}{p_s} = \frac{3 - 2p}{p(2-p)p_s},$$

we recover the theoretical expectation value of Eq. . We have therefore shown that our approach to the expected number of trials through the law of total expectation, designed to follow the rules we have listed to represent a general protocol, recovers the correct theoretical value known in the literature if we correctly set the weights such that they represent the quantum repeater protocol.

3.1.4.2 Characterization of the fidelity F in terms of protocol weights

Let's proceed again step by step following the rules for the general protocol to build the possible final state.

1. In the first unitary, firing all the sources we generate a state $|\psi\rangle$. Remembering that we can observe 2^D different pattern configurations i, the state $|\psi\rangle$ is collapsed in the states $|\psi\rangle_i$ (Figure 3.13). The information we are interested in is the one kept in the memories after the heralding measurement at the detectors, so we will trace out the heralding modes H and we'll keep the memory modes.

2. Now we can as always accept the heralding and immediately move to the second unitary, accept the heralding just for some memories, or reject it completely and restart; in the last case we are not saving any state. If we accept the heralding and immediately move to the second unitary (parameter s_i), the state in the memories will be $\rho_{i,\text{all}} = Tr_H|\psi\rangle_i\langle\psi|$, where the subscript "all" indicates that we are keeping all the memories. If we accept the heralding just for some memories instead, we will store them and restart the experiment to herald the memories we didn't immediately store; this means that the state stored will be $\rho_{i,m} = Tr_{H\overline{m}}|\psi\rangle_i\langle\psi|$, where $H\overline{m}$ indicates that we are tracing out the heralding modes and the modes of the memories we won't keep, leaving just the modes m of the kept memories.

3. When we perform a new experiment, the state produced will be the same $|\psi\rangle$, and as before also this one will be collapsed in all the possible pattern configurations j: $|\psi\rangle_j$. The final state we're interested to keep in this case is just the state over the modes $\overline{m}$ of the memories we didn't store earlier: $\rho_{j,\overline{m}} = Tr_{Hm}|\psi\rangle_j\langle\psi|$. According to the rules, once we observe a pattern i we can choose to fix a set of memories m, and then the choice of accepting a pattern j will be represented by $c_{i,m,j}$. Therefore the state we can produce once we observed i and chose m is

$$\rho_{i,\overline{m}} = \frac{\sum_j p_j \; c_{i,m,j} \; \rho_{j,\overline{m}}}{\sum_j p_j \; c_{i,m,j}}.$$

   We impose the normalization to still work with a density matrix.

4. The final state we can produce in the memories is therefore of two kinds: $\rho_{i,\text{all}}$, when we immediately stop and move to the second unitary with probability s_i, or $\rho_{i,m} \otimes \rho_{i,\overline{m}}$, when we herald just some memories through pattern detection i and then fix memories m with probability a_{im}. If we want to take into account all the weights of the protocol, we can write this final state at the memories ρ_{mem} as:

$$\rho_{\text{mem}} = \frac{\sum_i p_i w_i \left[s_i \rho_{i,\text{all}} + \sum_m a_{i,m} (\rho_{i,m} \otimes \rho_{i,\overline{m}}) \right]}{\sum_i p_i w_i}.$$

5. When we go to the second unitary, we make the state ρ_{mem} evolve under the optical transformations, leading to a state ρ'_{mem}. Now we can consider all the

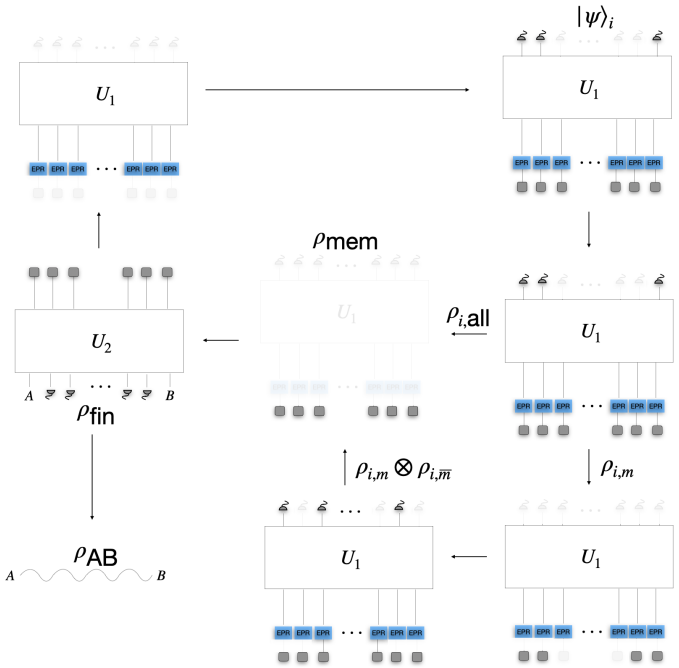


Figure 3.13: Same scheme of Figures 3.11, 3.12, now focusing on how the states evolve during the decision-making process.

detection patterns in the second unitary happening with probability $p_{i'}$ and weighted through $w_{i'}$. For each of these we'll consider the projection $\rho'_{\text{mem},i'}$ of the output state, from which we should trace out all the modes except for AB: $\rho_{\text{AB},i'} = Tr_{\overline{AB}}\rho'_{\text{mem},i'}$. The final state produced in the general protocol is:

$$\rho_{\text{fin}} = \frac{\sum_{i'} p_{i'} w_{i'} \rho_{\text{AB},i'}}{\sum_i p_{i'} w_{i'}}.$$

The fidelity F of ρ_{fin} with respect to the target state $|\Psi^-\rangle$ is:

$$F = \langle\Psi^-|\rho_{\text{fin}}|\Psi^-\rangle. \tag{3.13}$$

The expressions 3.12 and 3.13 are completely dependent on all the weights of the general protocol, and they will be optimized together with the experimental parameters when minimizing 3.3.

Numerical example for the quantum repeater As a last thing, let's take the weights we selected earlier in the proof of Eq. that correctly individuate the repeater protocol, and let's apply them to the computation of the fidelity. All the weights are zero except for:

$$w_{CCCC}, w_{CCNC}, w_{CCCN}, w_{NCCC}, w_{CNCC}, w_{CCNN}, w_{NNCC}$$
$$s_{CCCC}, a_{CCNC,m_{12}}, a_{CCCN,m_{12}}, a_{CCNN,m_{12}}, a_{NCCC,m_{34}}, a_{CNCC,m_{34}},$$
$$a_{NNCC,m_{34}}, c_{CCNC,m_{12},**CC}, c_{CCCN,m_{12},**CC}, c_{CCNN,m_{12},**CC},$$
$$c_{NCCC,m_{34},CC**}, c_{CNCC,m_{34},CC**}, c_{NNCC,m_{34},CC**}, w_{CC},$$

that are instead equal 1. Starting with the weights c, which are used in the computation of $\rho_{i,\overline{m}}$, we find:

$$\rho_{CCNC,\overline{m_{34}}} = |\Psi^-\rangle_{DB}\langle\Psi^-|,$$

$$\rho_{CCCN,\overline{m_{34}}} = |\Psi^-\rangle_{DB}\langle\Psi^-|,$$

$$\rho_{CCNN,\overline{m_{34}}} = |\Psi^-\rangle_{DB}\langle\Psi^-|,$$

$$\rho_{NCCC,\overline{m_{12}}} = |\Psi^-\rangle_{AC}\langle\Psi^-|,$$

$$\rho_{CNCC,\overline{m_{12}}} = |\Psi^-\rangle_{AC}\langle\Psi^-|.$$

$$\rho_{NNCC,\overline{m_{12}}} = |\Psi^-\rangle_{AC}\langle\Psi^-|.$$

For the state $\rho_{i,\text{all}}$ and the states $\rho_{i,m}$ we look at what weights w and a select for us:

$$\rho_{CCCC,\text{all}} = |\Psi^-\rangle|\Psi^-\rangle_{ACDB}\langle\Psi^-|\langle\Psi^-|,$$

$$\rho_{CCNC,m_{12}} = |\Psi^-\rangle_{AC}\langle\Psi^-|,$$

$$\rho_{CCCN,m_{12}} = |\Psi^-\rangle_{AC}\langle\Psi^-|,$$

$$\rho_{CCNN,m_{12}} = |\Psi^-\rangle_{AC}\langle\Psi^-|,$$

$$\rho_{NCCC,m_{34}} = |\Psi^-\rangle_{DB}\langle\Psi^-|,$$

$$\rho_{CNCC,m_{34}} = |\Psi^-\rangle_{DB}\langle\Psi^-|.$$

$$\rho_{NNCC,m_{34}} = |\Psi^-\rangle_{DB}\langle\Psi^-|.$$

In all these states, $ACDB$ represent the modes of the memories, A and B the external ones, and C and D the middle ones. If we put everything together to compute ρ_{mem} we find:

$$\rho_{\text{mem}} = \frac{p^2\rho_{CCCC,\text{all}} + 2p(1-p)\left[\rho_{CCNC,m_{12}}\otimes\rho_{CCNC,\overline{m_{34}}} + \rho_{CCCN,m_{12}}\otimes\rho_{CCCN,\overline{m_{34}}} + \rho_{CCNN,m_{12}}\otimes\rho_{CCNN,\overline{m_{34}}} + \rho_{NCCC,m_{34}}\otimes\rho_{NCCC,\overline{m_{12}}} + \rho_{CNCC,m_{34}}\otimes\rho_{CNCC,\overline{m_{12}}} + \rho_{NNCC,m_{34}}\otimes\rho_{NNCC,\overline{m_{12}}}\right]}{p^2 + 2p(1-p)}$$

noticing that all the tensor products are equivalent to $\rho_{CCCC,\text{all}} = |\Psi^-\rangle|\Psi^-\rangle_{ACDB}\langle\Psi^-|\langle\Psi^-|$, we conclude

$$\rho_{\text{mem}} = |\Psi^-\rangle|\Psi^-\rangle_{ACDB}\langle\Psi^-|\langle\Psi^-|.$$

Now this state, when undergoing the second layer, i.e. the last BSM with coincidence detection required ($w_{CC} = 1$), will end up in

$$\rho_{\text{fin}} = |\Psi^-\rangle_{AB}\langle\Psi^-|,$$

which gives us a fidelity F:

$$F = \langle\Psi^-|\rho_{\text{fin}}|\Psi^-\rangle = |\langle\Psi^-|\Psi^-\rangle|^2 = 1.$$

We conclude that by choosing the weights that recover the quantum repeater protocol, we effectively build a final state which is exactly the entangled pair $|\Psi^-\rangle_{AB}$.

3.2 Rediscovery and current limitations

Having established a general approach for the experimental setup and the most general protocol, in addition to a loss function sensitive to all the parameters that characterise it, we are ready to test whether EsQueranto is capable of rediscovering the quantum repeater within the enormous space of solutions. Rediscovery means that the optimization should push all the parameters to values s.t. the experimental setup looks like the one in Figure 3.8, and the protocol parameters indicate a sequence of choices based on the detection patterns equivalent to the quantum repeater protocol. EsQueranto's general workflow is illustrated in Figure 3.14.

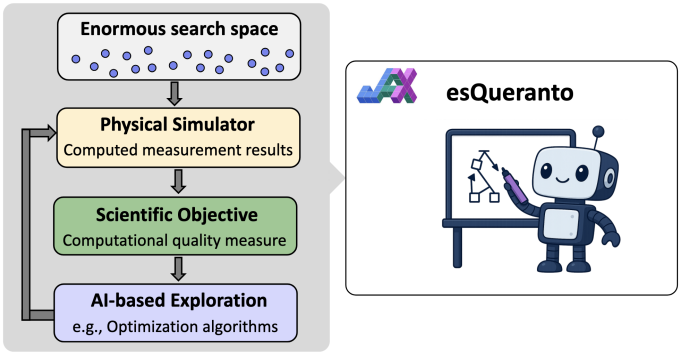


Figure 3.14: EsQueranto workflow for a generic discovery problem in quantum optics. Credits: [25]

.

In this work we will optimize through the stochastic gradient descent (SGD) Adam [1], already implemented in the `optax` library in JAX. Before showing the results, let's appreciate the power of the parallelization ability of JAX, that thanks to `jit` compilation and its computational graph keeps the execution time constant for an increasing number of parameters, just paying a little increase in compilation time of the graph, as shown in Figures 3.15, 3.16. Such a rapidity allowed us to evaluate the optimizer over many different runs for a reasonably high number of iterations as shown in Figure 3.17.
All these results correspond to rediscoveries of the quantum repeater setup and protocol for different fixed length L between A and B. All the found solutions match numerically the expected number of trials for a given L, but some of the layouts of the discovered setups are quite unnatural from our perspective, but still physically solid, as shown in Figure 3.18.

Now one could ask if *EsQueranto* was able to discover also some new experimental setups that can surpass in efficiency the quantum repeater. Unfortunately, at the present time, the GPUs we used for this work don't have enough memory to keep all the information about large ansatzes that we could use to surpass the quantum repeater. This is due to the fact that in our ket-based representation, a ket of M modes occupies at minimum $8 + M$ bytes. Now, with N photons in M modes we have a total of

$$\#_{kets} = \frac{(N + M - 1)!}{N!(M - 1)!}, \tag{3.14}$$

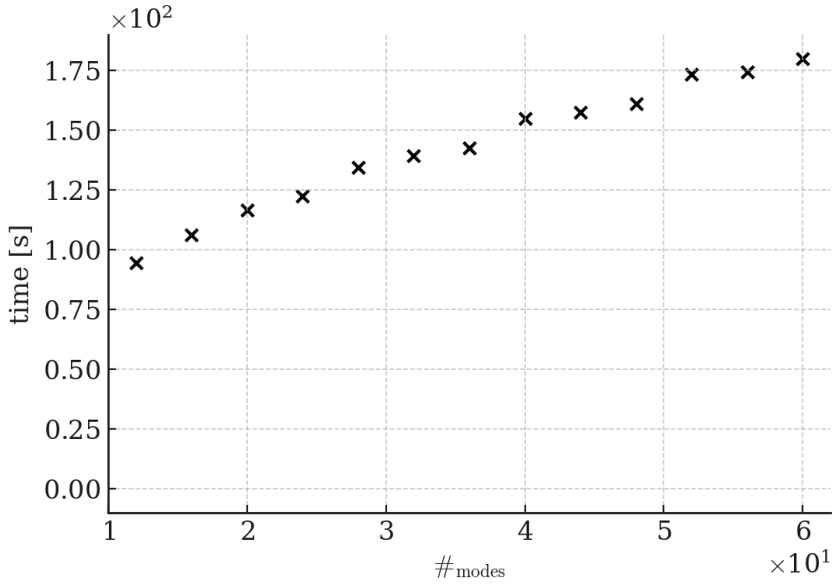


Figure 3.15: Compile time of the computational graph for an increasing number of optical modes in our ansatz. Due to memory issues (see below), the values obtained for the higher modes states do not consider the complete Hilbert space, but that shouldn't represent a game changer with respect to compile time.

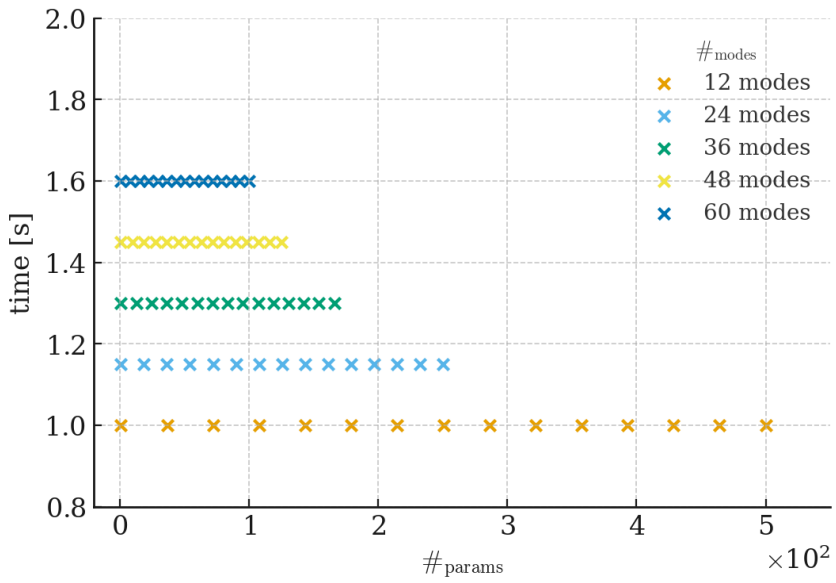


Figure 3.16: Run time of an experiment with circuits of the indicated number of modes. The impossibility of evaluating more continuous parameters with large modes is again due to memory issues. Still, in the hypothesis of larger storage memory, there's no reason why we shouldn't observe the same trend.

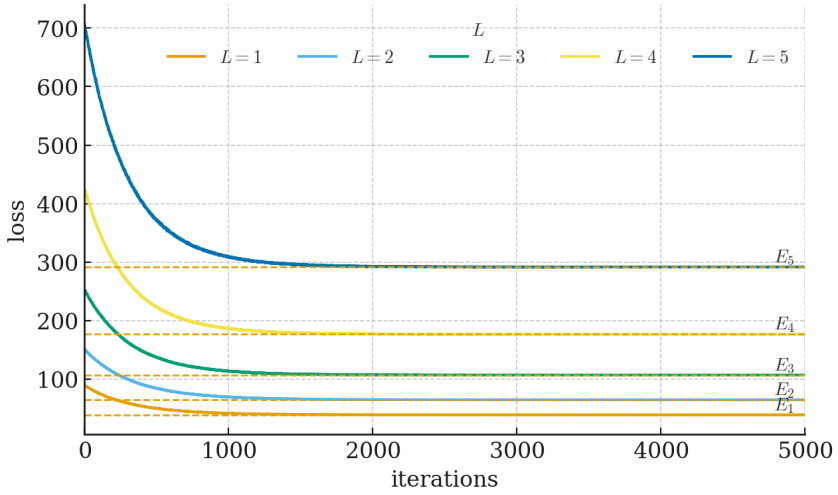


Figure 3.17: Average of 10000 converging loss functions over different L. We notice how, according to the expression of the loss in Eq. 3.3, each loss converges to the E value for the quantum repeater of that specific L. Optimization is done over a Nvidia Quadro RTX 6000.

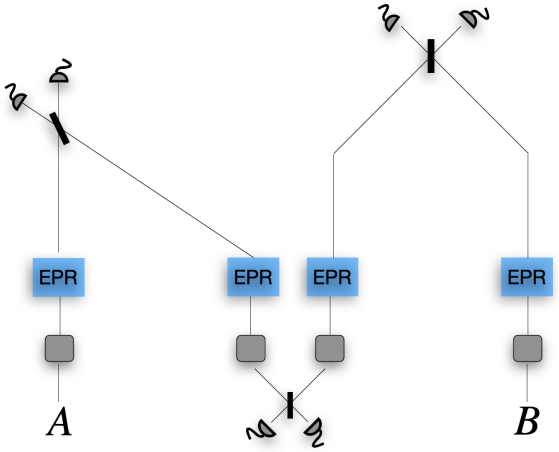


Figure 3.18: Some of the solutions found by EsQueranto shift from the symmetrical structures that we are used to look for as humans. Here for instance the repeater is quite asymmetric, as to generate entanglement on the left side the BSM is not performed in the middle, but instead one photon remains local and the other travels a distance $\frac{L}{2}$. This is of course numerically equivalent to making both photons travel for $\frac{L}{4}$.

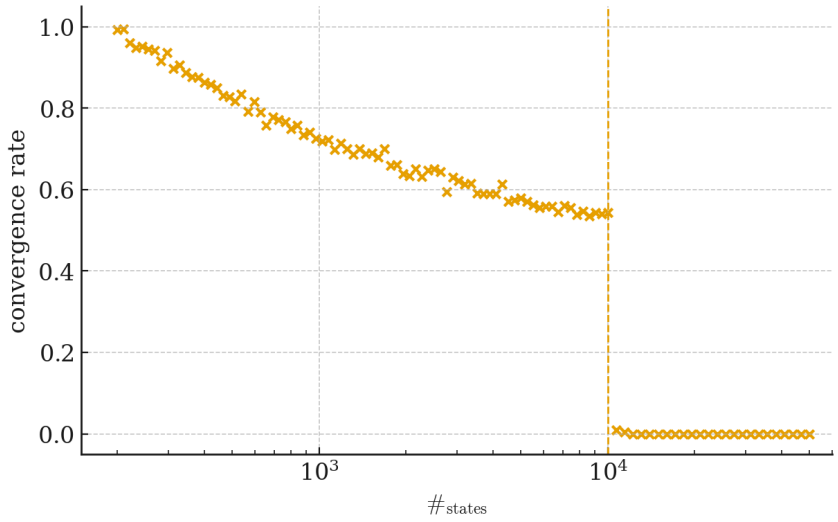


Figure 3.19: Convergence rate VS the size of the space, i.e. the number of states (kets) we let our system visit. The vertical line indicates the maximum number of states that we let our system see. For states with an actual number of states larger than our limit, the algorithm basically never converges.

and in our scheme, fixing 4 sources as EPR sources (which is already restricting), we produce $N = 8$ photons. Remembering the minimum general ansatz of Figure 3.10, we count $M = 24$ modes (remember the dual-rail encoding), but this is only on a single experiment! The final state as we have seen is in fact generally built over two experiments, and to keep all the information means duplicating again the number of modes, so $M = 48$, and also the number of kets becomes $\#^2_{kets}$ because of the tensor product between the two states produced by the two experiments. So to contain all the information needed to represent the minimum general ansatz we need already $\sim 3.5 \times 10^6$GB, which is clearly out of reach for every existing computer. The solution we tried has been to cap the maximum number of kets that our simulator can work with, but unfortunately this degrade totally the quality of the optimization, as shown in Figure 3.19. Basically we have never been able to work with the minimum general ansatz, and the biggest we worked with is the one in Figure 3.20. Although we have demonstrated that it is possible to automate the discovery of quantum networks through the use of AI, the challenge for the future remains to simplify the solution space and identify specific techniques that allow AI to work with only a portion of the available data and still achieve excellent results.

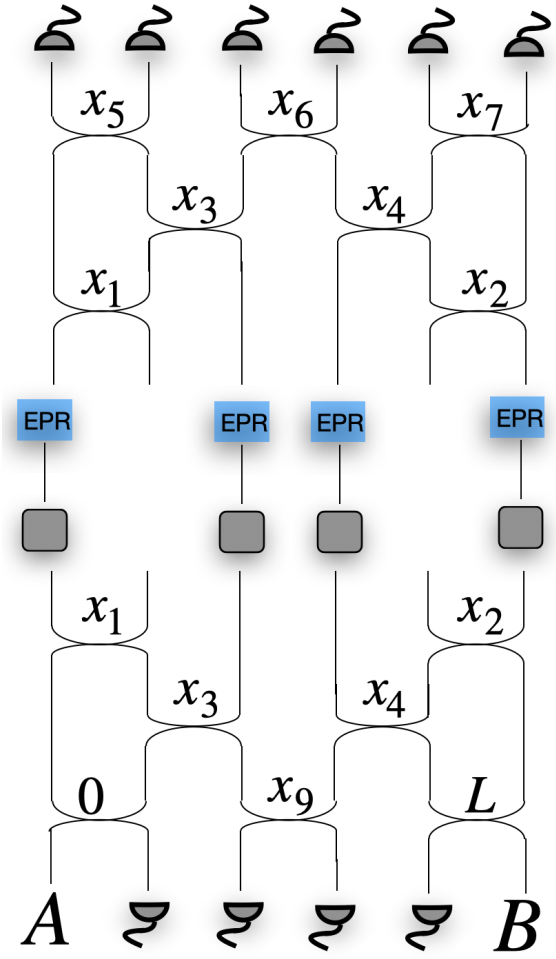


Figure 3.20: Larger ansatz (in terms of interference allowed) we've been able to use without having to limit the maximum number of states. Crucially we notice that, beside reducing the depth of the circuits, therefore limiting the interference, not all the detectors can be fed by all the sources, and this greately reduces the number of possible kets. In this case, the quantum repeater is embedded as in Figure 3.21.

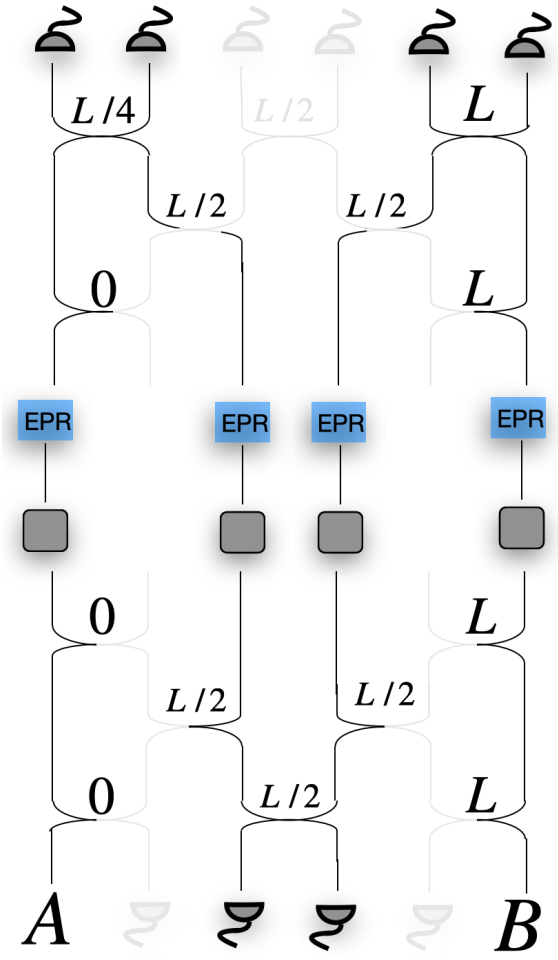


Figure 3.21: Active circuit and fixed positions in the largest ansatz of Figure 3.20 to retain the quantum repeater setup of Figure 3.8.

Chapter 4

Novel use of quantum memories

4.1 Generation of Bell pairs with single photons

In the last chapter we shall evaluate EsQueranto on another approach to generating entangled pairs that involves the use of single photons, so that it is not necessary to already start with entangled states (as is instead the case with quantum repeaters). Once again, the idea is to use heralding at the detectors to successfully validate the creation of an entangled pair, which can then be used as a resource for quantum communication protocols mentioned above, but also in quantum computing, quantum key distribution, quantum metrology, etc. In recent times, several schemes have been proposed, each with different uses of the number of photons, the number of modes, the number of optical instruments, and the type of detectors. The schemes we will refer to, and those we propose, will all have Bell pairs in dual-rail encoding as their target states, namely:

$$|\Phi^{\pm}\rangle = \frac{1}{\sqrt{2}}\big(|1010\rangle \pm |0101\rangle\big), \tag{4.1}$$

$$|\Psi^{\pm}\rangle = \frac{1}{\sqrt{2}}\big(|1001\rangle \pm |0110\rangle\big), \tag{4.2}$$

$$|\chi^{\pm}\rangle = \frac{1}{\sqrt{2}}\big(|1100\rangle \pm |0011\rangle\big), \tag{4.3}$$

where the latter can be converted to the usual encoding by switching modes. Many of these schemes are based on the logic of immediately releasing all photons into the circuit at the same time, but some instead envisage that only a portion of photons are initially released, and then, depending on the success or failure of heralding, the remaining photons interact in the circuit with the post-heralding state. An example of this latter type is presented in [18], and the corresponding scheme is shown in Figure 4.1.

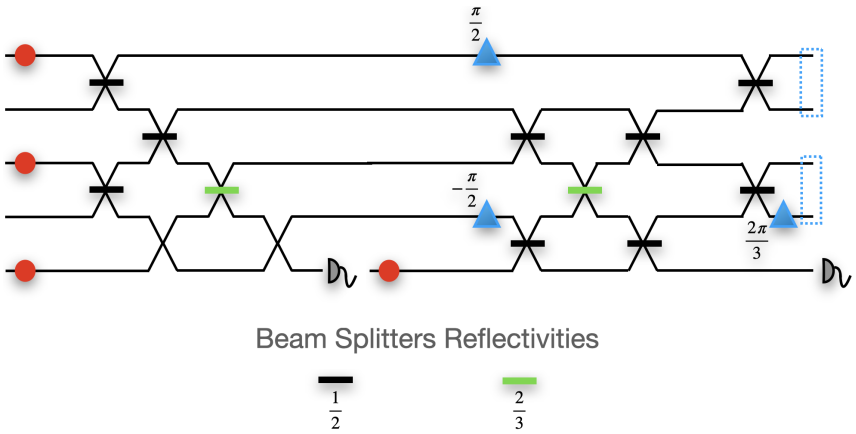


Figure 4.1: Feed-forward heralding scheme for the generation of the state $|\Phi^+\rangle$. The light blue triangles represent phase shifters with an associated value of the shift. The light blue dotted lines individuate the dual-rail modes over which the final state is obtained.

Three photons are initially fed into the first-sector circuit; if we detect a single photon in the first detector, a fresh photon is given in input together with the post-heralding state into the second-sector circuit. If a single photon is detected again, we have successfully heralded the generation of the Bell state $|\Phi^+\rangle$ with a success probability $p_{succ} = \frac{2}{27}$, which corresponds to an expected number of trials $E = \frac{1}{p} = \frac{27}{2} = 13.5$. It is natural to ask whether our early methods for autonomous discovery of setups and protocols can be equally applied to this new problem; the answer is of course positive, and moreover, the software we used to succeed in quantum repeater rediscovery makes it possible to introduce quantum memories into the picture, to study whether feed forward heralding schemes (and general heralding schemes) can benefit from the use of these tools.

4.2 Proposed ansatz with quantum memories

As we have learnt in the earlier chapters, what we want is a very general initial configuration that can in principle contain all, or at least many, different schemes and associated protocols, described by a fair amount of free parameters that will be optimized through stochastic gradient descent methods. For the specific problem of heralded generation of entanglement with single photons, wanting also to include quantum memories in the picture, we present the initial ansatz of Figure 4.2.

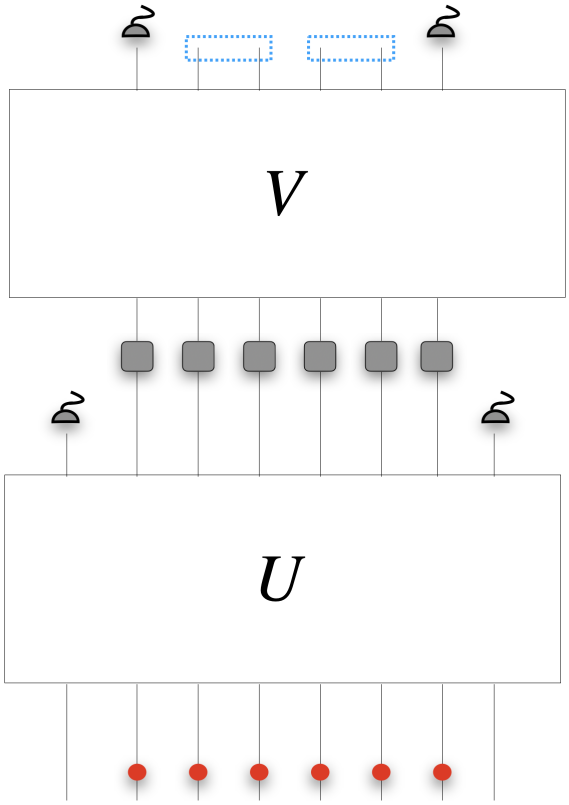


Figure 4.2: The ansatz we choose to represent the most general setup and protocol. We start with n photons in m modes, n memories on the undetected modes and two unitaries U and V. The choice of the number of photons should be based on the knowledge that we must at least work with 4 single input photons if we want to generate Bell pairs [47]. The parametrization of the unitary and the rules for the protocol are the same as the quantum repeater rediscovery.

In this ansatz we don't explicitly make use of single ancilla photons after the first sector, but this doesn't mean that we cannot rediscover the setup of Figure 4.1. In fact, one can always redraw such a feed-forward network making all the photons enter the circuit at once, as shown in Figure 4.3.

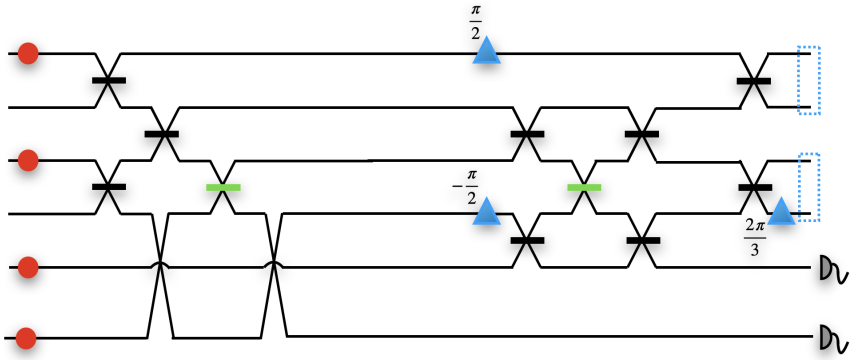
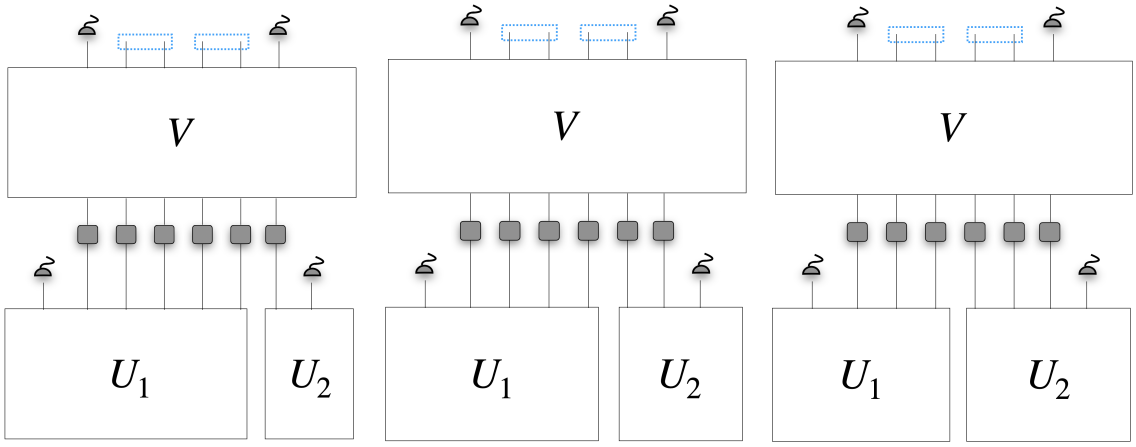


Figure 4.3: Non feed-forward scheme, by all means equivalent to the feed-forward one in Figure 4.1.



(a) (b) (c)

Figure 4.4: Preparing memories over two different rounds implies that the final state will be a separable state with respect to the two subset of memories we are considering. With six memories, we have either five-one, four-two or three-three. Notice that we are not restricting on the number of photons that enter a single U_i, it's just required that their sum is in the end n. Of course, in between a subset, memories can store entanglement.

4.3 Optimization results

We carry out the optimization through the same loss function as before

$$\mathcal{L} = E + \alpha \times \text{softplus}\big(F_{th} - F\big), \tag{4.4}$$

but this time the fidelity F allows for multiple targets t, so we have to slightly modify its definition:

$$F = \frac{\sum_i w_i \max_t F_{it}}{\sum_i w_i}, \tag{4.5}$$

where w_i are the protocol weights introduced in the earlier chapter and F_{it} is the fidelity obtained between the generated state $|\psi\rangle_i$ and the target t among $|\Phi^{\pm}\rangle, |\Psi^{\pm}\rangle, |\chi^{\pm}\rangle$. We end up with a surprisingly high amount of converging solutions, in the order of 75%. Of these solutions, some are just rediscoveries of already existing schemes, which are easy to spot because the protocol parameter s_i gets to 1 $\forall i$. The optimization anyway shows also another class of solutions, that explicitly rely on the use of memories, having the parameter s_i equal to 0 for some i. Since we consider that memories can be loaded only over two different rounds, all types of allowed solutions are easy to represent, and are shown in Figure 4.4. The immediate protocol we can associate to any of these schemes is the parallel *trial until success* for U_1 and U_2, success that happens with probability p_1 and p_2 respectively. Once all the memories are successfully loaded with the desired states, we release in V, where the successful outcome has probability q. The expected number of trials E reads in this case (explicit computation in appendix):

$$E = \left(\frac{\frac{p_1^2+p_2^2}{p_1 p_2} - p_1 - p_2 + 1}{p_1 + p_2 - p_1 p_2} \right) \frac{1}{q}. \tag{4.6}$$

Our model is able to push this value below 13.5, hitting specifically 10.53 in the best solutions. This result is encouraging, since it already beats the known result for the feed forward scheme, but on the other hand it does so by employing more resources, namely photons, modes and of course quantum memories. One could ask if it's worth to complicate the scheme this much. The answer is yes, because as it turns out, much more can be obtained if we focus on a particular subclass of solutions, shown in Figure 4.5 (In the appendix we will have the explicit setup of U_1 and V).

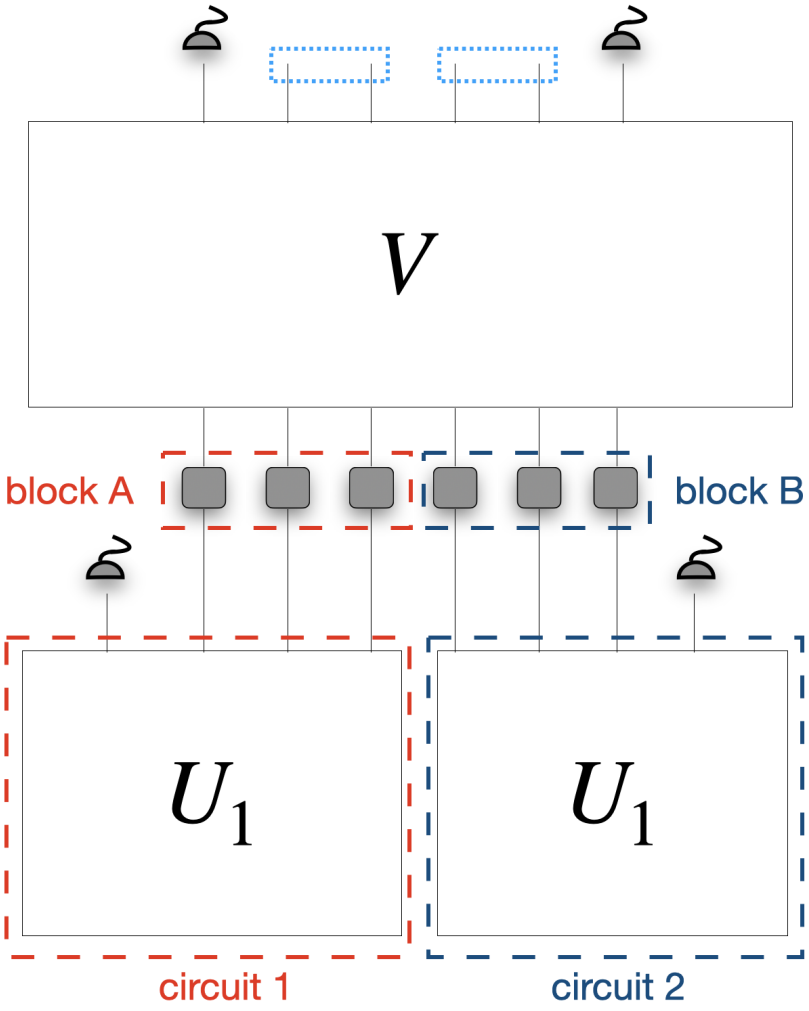
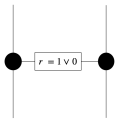


Figure 4.5: A particular set of solutions we found consists of not just an equal split of memories, but also by the exact same unitary acting on the same number of input photons ($n/2$ per part).

4.4 Introducing switches

The kind of solution of Figure 4.5 is the one of Figure 4.4c, characterized by $U_1 = U_2$ (which implies $p_1 = p_2 = p$), an highly symmetrical scheme, which seems interesting for a very simple reason: both blocks of memories (A and B in Figure 4.5) can now be equally fed by circuit 1 or circuit 2, given that we introduce another tool, the **optical switch**. An optical switch can be implemented between two modes as

a MZI (Mach-Zehnder interferometer)[6], that through an active phase shifter can implement either the identity or a BS with reflectivity r. In our schemes identity corresponds to $r = 1$, and we are just interested in switching between that and $r = 0$. We sketch the switch between two modes as



and we can define a *switching station* as a structure that implements switches between inputs of two different U_1 circuits, as shown in Figure 4.6.

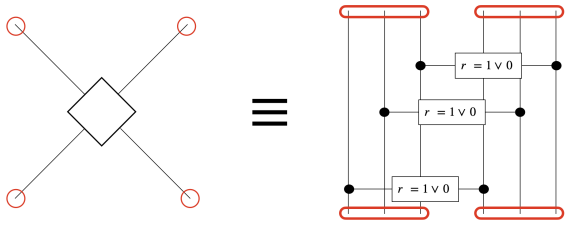


Figure 4.6: We introduce an handy representation for the application of the same switch over two equal groups of modes, outputs of circuit 1 and circuit 2 of Figure 4.5, respectively.

The switching station is the fundamental block that allows us to write the improved version of the scheme of Figure 4.5, that now becomes the one in Figure 4.7.

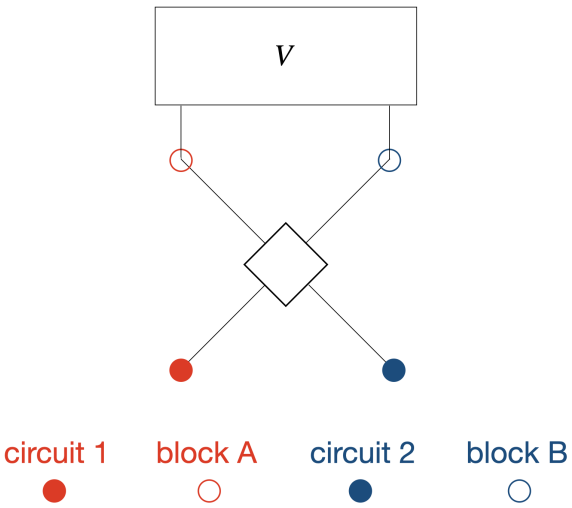


Figure 4.7: We exploit the fact that circuit V requires two copies of the exact same state in input, therefore both circuit 1 and 2 become good sources for both block A and B. The labels are exactly those of Figure 4.5.

The expected number of trials E reads in this case (explicit computation in the appendix)

$$E = \frac{4 - 4p + pq}{pq(4 - 4p + p^2 q)}, \tag{4.7}$$

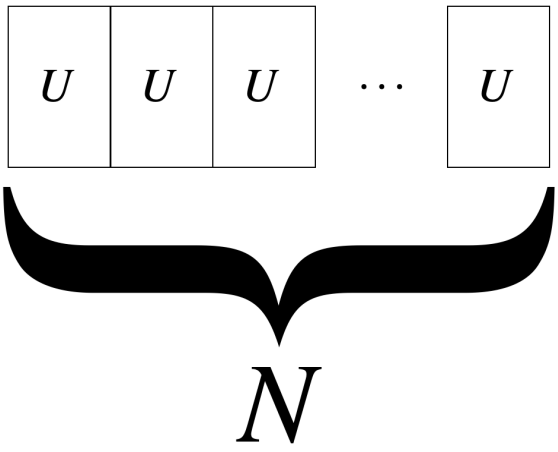


Figure 4.8: Multiplexing consists in the repetition of N copies of the same circuit U, and their exacution in parallel.

where p is the probability of observing a successful heralding, in either circuit 1 or 2. For $p = 0.27$ and $q = 0.5$ (these are the best values the optimization found), Eq. 4.7 is $E = 7.65$, a clear improvement over the old 10.53.

4.5 Multiplexing

Another surprising use that one can do of our newly discovered scheme together with switches is immediate when we consider multiplexing[5]. By multiplexing we mean using multiple copies of the same circuit U, that considered by itself has a success probability P, but repeated N times in parallel (see Figure 4.8), it grows as $P_{\text{multiplexing}} = 1 - (1 - P)^N$. The expected number of trials would be

$$E_{\text{multiplexing}} = \frac{1}{P_{\text{multiplexing}}} = \frac{1}{1 - (1 - P)^N}. \tag{4.8}$$

We claim that our newly discovered class of circuits can be used for a more advanced type of multiplexing, where the N copies are not completely independent like those in Figure 4.8, but are actually co-dependent, since we can combine the partial success of a cell with the partial success of another cell. In the following we'll analyze the situation where each cell is the scheme in Figure 4.7 (so 2 circuits per cell), but in principle the results can be extended for an arbitrary cell that uses even more circuits.

Exploiting the switching properties, we can combine whichever of the $2N$ circuits U_1 produces a good state, attempting all the available V among the total N in parallel. The unconsumed good states generated by a U_1 but not immediately used by a V are stored inside memories. A sketch of the diagram can be seen in Figure 4.9. Applying the known law of total expectation at this specific case (explicit calculations in the appendix), the expected number of trials reads:

$$E_{\text{ new multiplexing}} = \frac{2 - A - B + \frac{A-B}{g}}{2(1 - A)(1 - B)}, \tag{4.9}$$

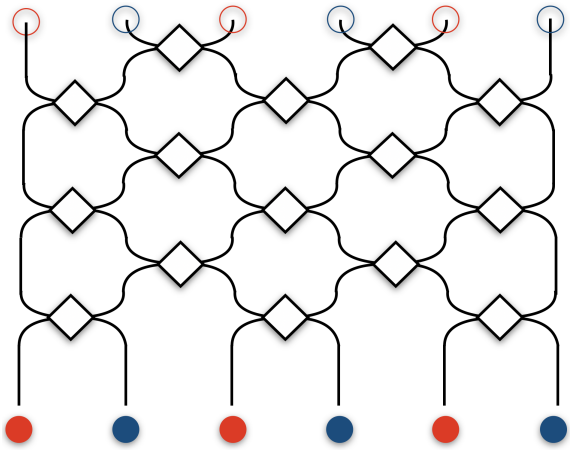


Figure 4.9: Multiplexing of our new scheme in the case of $N = 3$, trying to combine the greatest amount of partial success we have in a single trial.

where $g = \sqrt{1-q}$, $A = a_+^{2N}$, $B = a_-^{2N}$ with $a_\pm = 1-p\pm pg$. To understand the value of this result, let's see what happens initially if we have two circuits 1 and 2 that generate Bell pairs without using quantum memories, one with an expected number of trials E_1 and the other with an expected number of trials E_2. Their multiplexing is the independent one shown in Figure 4.8. In this case the expected number of trials is computed through 4.8. We ask, based on the sign of $E_1 - E_2$, after which N the multiplexing of circuit with expected number of trials E_1 becomes more efficient than the one with expected number of trials E_2. The solution is trivial and shown in Figure 4.10.

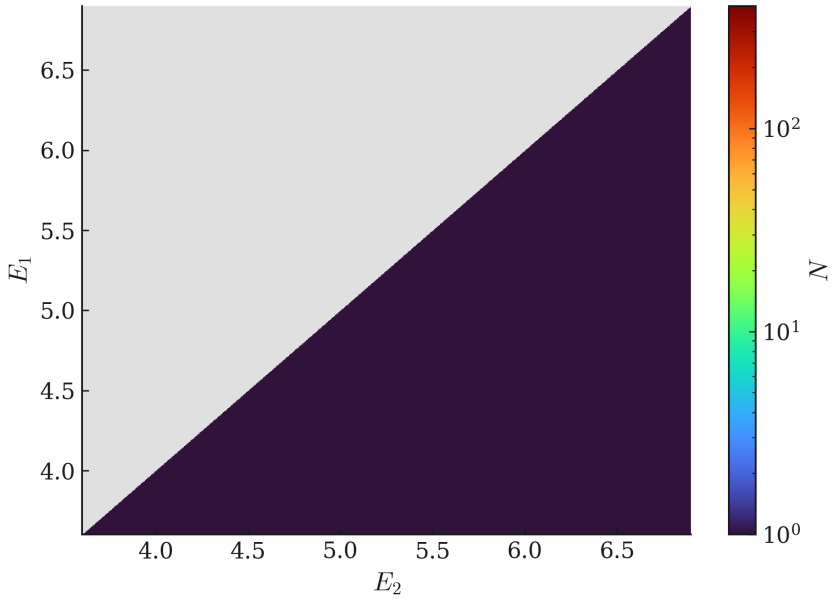


Figure 4.10: Heatmap showing after which value of N the multiplexing of a circuit with expected number of trials E_1 becomes more efficient that the multiplexing of a circuit with expected number of trials E_2. All the cells are independent.

Without memories, the multiplexing of a circuit with expected number of trials $E_1 > E_2$ will never surpass in efficiency the second circuit, with the same number N of cells. If $E_1 < E_2$ instead, it's always more efficient (for any $N \geq 1$). On the other hand, if we consider that the circuit 1 is instead composed by memories, we will say that the expected number of trials E_1 is computed through 4.7, and the multiplexed version is the one in Figure 4.9, described by 4.9. If we look again for values N after which using circuit 1 is more efficient than using circuit 2, where now 1 uses memories and 2 doesn't, we get the results in Figure 4.11.

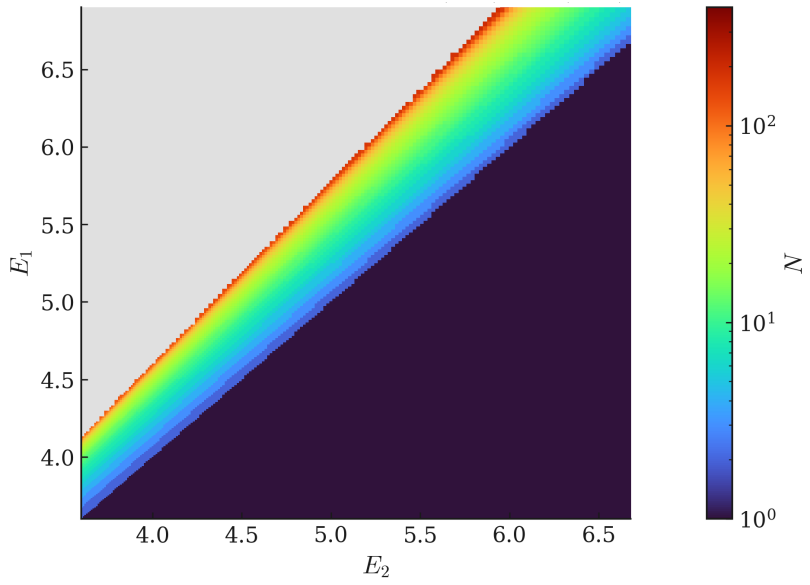


Figure 4.11: Heatmap showing after which value of N the multiplexing of a circuit with expected number of trials E_1 becomes more efficient that the multiplexing of a circuit with expected number of trials E_2. Cells of circuit 1 are dependent, according to the scheme of Figure 4.9. Here we capped the evaluation at $N = 400$.

Comparison between Figure 4.10 and Figure 4.11 shows quite clearly that heralding schemes that make explicit use of quantum memories lead to a new approach to multiplexing, shifting from an independent to a co-dependent logic, that will eventually beat the first one, even if the single circuit 1 is less efficient than single circuit 2.

Conclusion

Bibliography

[1] Kingma DP Ba J Adam et al. "A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* 1412.6 (2014).

[2] K. Azuma, A. Mizutani, and H.-K. Lo. "All-photonic quantum repeaters". In: *Nature Communications* 7 (2016), p. 13523.

[3] Koji Azuma et al. "Tools for quantum network design". In: *AVS Quantum Science* 3.1 (2021).

[4] Koji Azuma et al. "Quantum repeaters: From quantum networks to the quantum internet". In: *Reviews of Modern Physics* 95.4 (2023), p. 045006.

[5] S Bartolucci et al. "Switch networks for photonic fusion-based quantum computing (2021)". In: *arXiv preprint arXiv:2109.13760* ().

[6] Sara Bartolucci et al. "Creation of entangled photonic states using linear optics". In: *arXiv preprint arXiv:2106.13825* (2021).

[7] C. H. Bennett et al. "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels". In: *Physical Review Letters* 70.13 (1993), pp. 1895–1899. DOI: 10.1103/PhysRevLett.70.1895.

[8] Charles H Bennett and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing". In: *Theoretical computer science* 560 (2014), pp. 7–11.

[9] Dik Bouwmeester et al. "Experimental quantum teleportation". In: *Nature* 390.6660 (1997), pp. 575–579.

[10] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. http://github.com/google/jax. 2018.

[11] R. A. Campos, Bahaa E. A. Saleh, and Malvin C. Teich. "Quantum-mechanical lossless beam splitter: SU(2) symmetry and photon statistics". In: *Physical Review A* 40.3 (1989), pp. 1371–1384. DOI: 10.1103/PhysRevA.40.1371.

[12] Jacques Carolan et al. "Universal linear optics". In: *Science* 349.6249 (2015), pp. 711–716.

[13] William R. Clements et al. In: *Optica* 12 (), pp. 1460–1465.

[14] Jonathan P. Dowling. "Quantum optical metrology – the lowdown on high-NOON states". In: *Contemporary Physics* 49.2 (2008), pp. 125–143. DOI: 10.1080/00107510802091298.

[15] L-M Duan et al. "Long-distance quantum communication with atomic ensembles and linear optics". In: *Nature* 414.6862 (2001), pp. 413–418.

[16] Charles Dugas et al. "Incorporating Second-Order Functional Knowledge for Better Option Pricing". In: *Proceedings of the 13th International Conference on Neural Information Processing Systems (NIPS 2000)*. Denver, CO, USA: MIT Press, 2000, pp. 451–457. URL: https://proceedings.neurips.cc/paper/2000/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

[17] A. K. Ekert. "Quantum cryptography based on Bell's theorem". In: *Physical Review Letters* 67.6 (1991), pp. 661–663. DOI: 10.1103/PhysRevLett.67.661.

[18] Suren A. Fldzhyan, Mikhail Yu. Saygin, and Sergei P. Kulik. "Compact linear optical scheme for Bell state generation". In: *Phys. Rev. Res.* 3 (4 Oct. 2021), p. 043031. DOI: 10.1103/PhysRevResearch.3.043031. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.3.043031.

[19] Imogen Forbes et al. "Heralded generation of entanglement with photons". In: *Reports on Progress in Physics* 88.8 (2025), p. 086002.

[20] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. "Quantum Metrology". In: *Physical Review Letters* 96.1 (2006), p. 010401. DOI: 10.1103/PhysRevLett.96.010401.

[21] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. "Advances in quantum metrology". In: *Nature Photonics* 5.4 (2011), pp. 222–229. DOI: 10.1038/nphoton.2011.35.

[22] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. 4th ed. Johns Hopkins University Press, 2013. ISBN: 9781421407944.

[23] Google. *XLA: Accelerated Linear Algebra*. https://www.tensorflow.org/xla. 2018.

[24] D. Gottesman, T. Jennewein, and S. Croke. "Longer-baseline telescopes using quantum repeaters". In: *Physical Review Letters* 109.7 (2012), p. 070503.

[25] Xuemei Gu. "Artificial Scientific Discovery for New Quantum Experiments". In: *Proceedings of the Fifth MODE Workshop on Differentiable Programming for Experiment Design*. OAC conference center / MODE Workshop. Kolymbari, Crete, Greece, 2025. URL: https://indico.cern.ch/event/1481852/contributions/6464734/.

[26] C. K. Hong, Z. Y. Ou, and Leonard Mandel. "Measurement of subpicosecond time intervals between two photons by interference". In: *Physical Review Letters* 59.18 (1987), pp. 2044–2046. DOI: 10.1103/PhysRevLett.59.2044.

[27] Pieter Kok et al. "Linear optical quantum computing with photonic qubits". In: *Reviews of Modern Physics* 79.1 (2007), pp. 135–174.

[28] P. Kómár et al. "A quantum network of clocks". In: *Nature Physics* 10 (2014), pp. 582–587.

[29] Mario Krenn, Yehonathan Drori, and Rana X Adhikari. "Digital discovery of interferometric gravitational wave detectors". In: *Physical Review X* 15.2 (2025), p. 021012.

[30] Mario Krenn, Manuel Erhard, and Anton Zeilinger. "Computer-inspired quantum experiments". In: *Nature Reviews Physics* 2.11 (2020), pp. 649–661.

[31] Mario Krenn et al. "Automated search for new quantum experiments". In: *Physical review letters* 116.9 (2016), p. 090405.

[32] Mario Krenn et al. "Conceptual understanding through efficient automated design of quantum optical experiments". In: *Physical Review X* 11.3 (2021), p. 031044.

[33] Mario Krenn et al. "On scientific understanding with artificial intelligence". In: *Nature Reviews Physics* 4.12 (2022), pp. 761–769.

[34] Mario Krenn et al. "Artificial intelligence and machine learning for quantum technologies". In: *Physical Review A* 107.1 (2023), p. 010101.

[35] Rodney Loudon. *The Quantum Theory of Light*. 3rd. Oxford, UK: Oxford University Press, 2000. ISBN: 9780198501763.

[36] Alexander I Lvovsky, Barry C Sanders, and Wolfgang Tittel. "Optical quantum memory". In: *Nature photonics* 3.12 (2009), pp. 706–714.

[37] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge, UK: Cambridge University Press, 2000. ISBN: 0-521-66068-5.

[38] Julien Niset, Jaromír Fiurášek, and Nicolas J. Cerf. "No-Go theorem for Gaussian quantum error correction". In: *Physical Review Letters* 102.12 (2009), p. 120501.

[39] Jian-Wei Pan et al. "Experimental entanglement swapping: entangling photons that never interacted". In: *Physical review letters* 80.18 (1998), p. 3891.

[40] S. Pirandola. *Tools for quantum network design*. arXiv e-print `arXiv:1601.00966`. 2016.

[41] S. Pirandola. "Advances in quantum teleportation". In: *Communications Physics* 2 (2019), p. 51.

[42] S. Pirandola et al. "Fundamental limits of repeaterless quantum communications". In: *Nature Communications* 8 (2017), p. 15043.

[43] Michael Reck et al. "Experimental realization of any discrete unitary operator". In: *Physical review letters* 73.1 (1994), p. 58.

[44] Carla Rodríguez et al. "Automated discovery of experimental designs in super-resolution microscopy with XLuminA". In: *Nature Communications* 15.1 (2024), p. 10658.

[45] Carlos Ruiz-Gonzalez et al. "Digital discovery of 100 diverse quantum experiments with PyTheus". In: *Quantum* 7 (2023), p. 1204.

[46] Nicolas Sangouard et al. "Quantum repeaters based on atomic ensembles and linear optics". In: *Reviews of Modern Physics* 83.1 (2011), pp. 33–80.

[47] Stasja Stanisic et al. "Generating entanglement with linear optics". In: *Physical Review A* 96.4 (2017), p. 043861.

[48] The JAX Authors. *JAX Experimental Sparse (BCOO) module*. `https://jax.readthedocs.io/en/latest/jax.experimental.sparse.html`. Accessed: 2025-10-01; experimental batched-coordinate sparse array support in JAX. 2025.

[49] Jianwei Wang et al. "Integrated photonic quantum technologies". In: *Nature Photonics* 14.5 (2020), pp. 273–284. DOI: 10.1038/s41566-019-0532-1.

[50] Kai Wang et al. "Entangling independent particles by path identity". In: *Physical Review Letters* 133.23 (2024), p. 233601.

[51] S. Wehner, D. Elkouss, and R. Hanson. "The quantum internet: A vision for the road ahead". In: *Science* 362 (2018), eaam9288.

[52] Neil A. Weiss, Paul T. Holmes, and Michael Hardy. *A Course in Probability*. Boston: Pearson, 2014. ISBN: 978-0-321-78373-0.

[53] Mark M. Wilde. *Quantum Information Theory*. Cambridge, UK: Cambridge University Press, 2013. ISBN: 978-1-107-03425-9.