QUANDELA

Quantorum
Photonic QML with Perceval and MerLin

Alexia Salavrakos
5 March 2026

@alexiasalavrakos.bsky.social
www.quandela.com

About Quandela

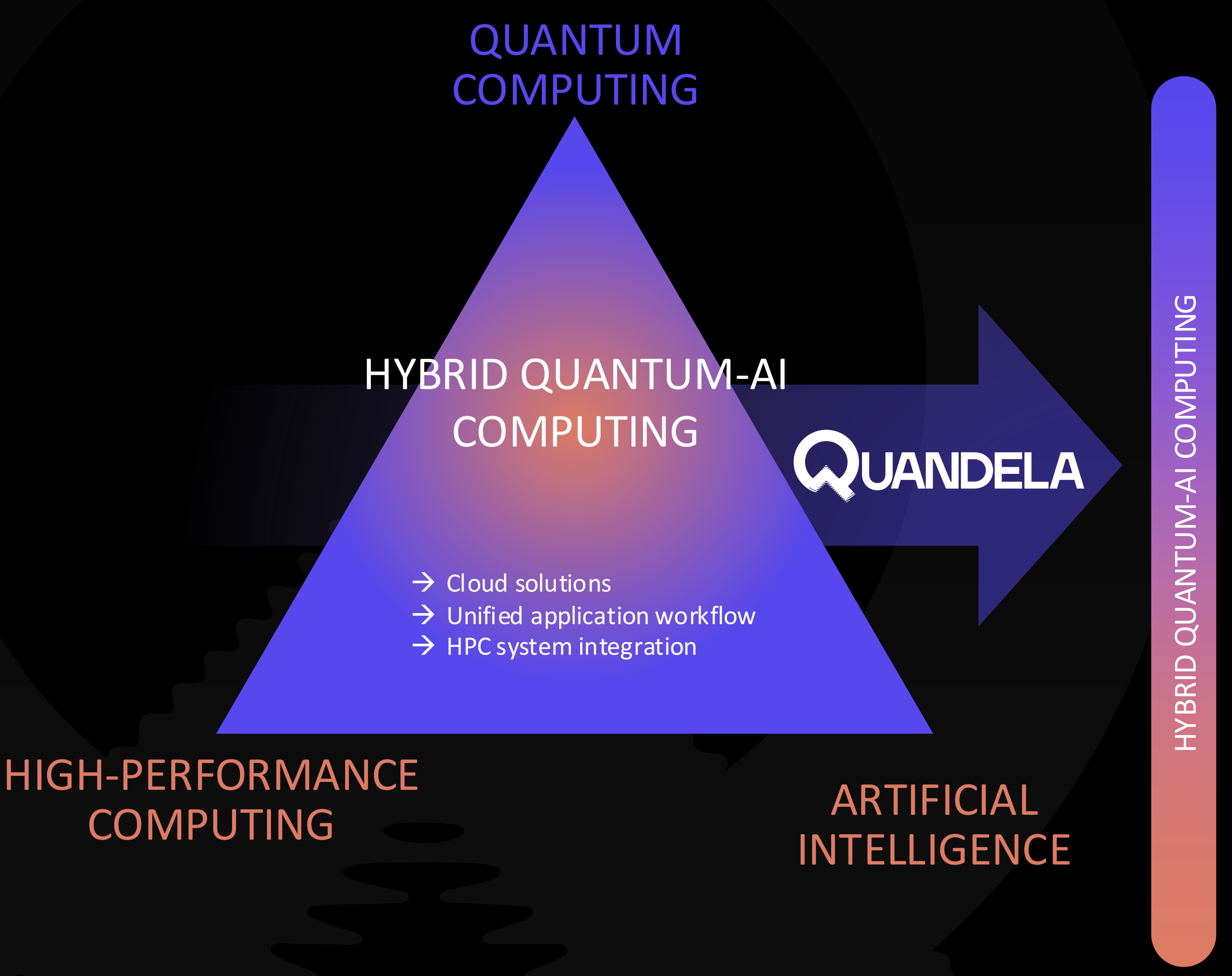
Founded in 2017
Spin off from Pascale Senellart's group at C2N
(CNRS & Paris-Saclay University)

Today:
140+ people, with >50 scientists and engineers
>40 scientific publications and patents

Quandela Scientific Advisory Board

R&D Centers		Production Centers	
			
C2N - Palaiseau	Massy	IPVF - Palaiseau	Massy

Where AI, HPC, and Quantum meet



ALREADY TESTED AND INTEGRATED BY MANY STAKEHOLDERS

End-users across numerous industries have experienced our solutions:

TotalEnergies + inCerebro · EDF + Dassault Systèmes · CA CIB · SNCF · Airbus · Orange · BMW Group

+ renewed contracts

Key partners have integrated our quantum solutions:

- Scaleway — ✓ Connected to their Cloud Platform
- Bull — ✓ Integrated to their Qaptiva Solution
- NVIDIA — ✓ CUDA QUANTUM + NVQ Link
- OVHcloud — On-going connexion to their Cloud Platform-> mid 2026

Leading HPC and datacentre operators have deployed our quantum systems:

- OVHcloud — ✓ Delivered system
- exaion EDF GROUP — ✓ Delivered system
- GENCI / EuroHPC Joint Undertaking — ✓ Delivered system

We are focused on delivering cutting-edge solutions

EXISTING CUSTOMERS

2023 — 2025

OVHcloud — 2 qubits ✓

exaion EDF GROUP — 6 qubits ✓

EuroHPC Joint Undertaking, cea, GENCI — 12 qubits ✓

✓ Delivered systems

3
days installation period

10
months delivery time

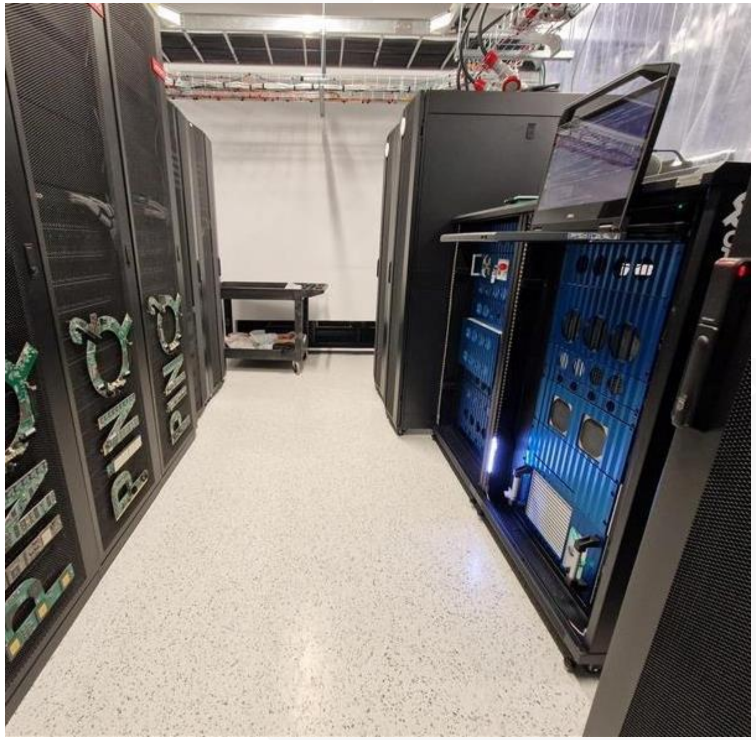
5
systems deployed over the past 20 months

Direct Integration
into HPC environment (resilient, low energy)

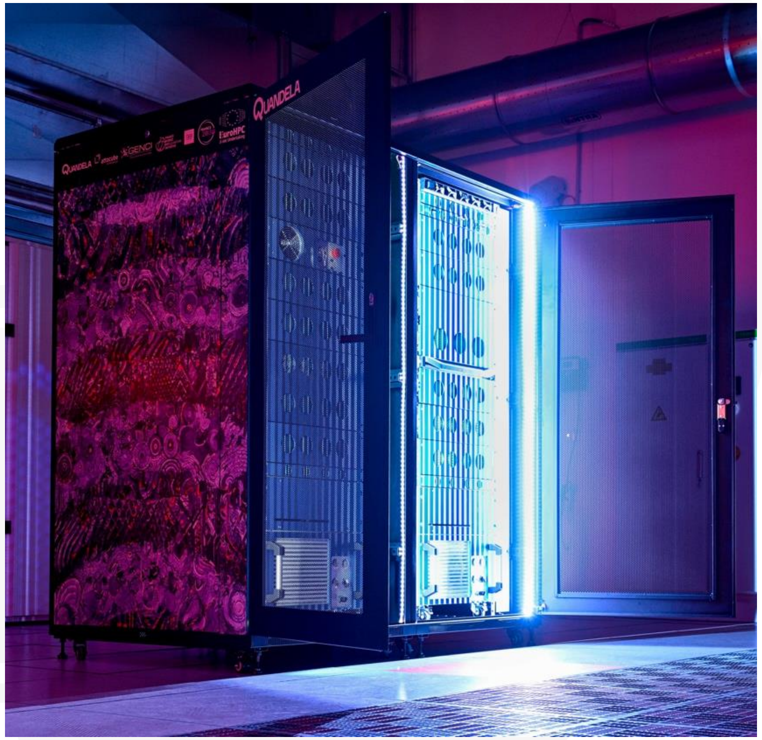
Deployed in October 2025 at TGCC (France) and operated by CEA, Lucy is the most powerful universal photonic quantum computer available in Europe.

We deliver HPC-ready Quantum Processor Units

Photograph of the Quantum Processor installed at OVH-Cloud data-center in 2023 (France)



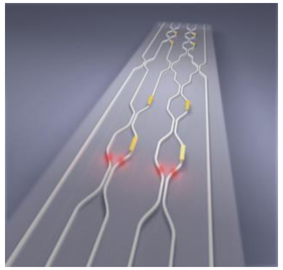
Quandela's Quantum Processor integrated at Exaion's datacenter in Sherbrooke (Canada) in 2025



Deployed in October 2025 at TGCC (France) and operated by CEA, Lucy is the most powerful universal photonic quantum computer available in Europe.

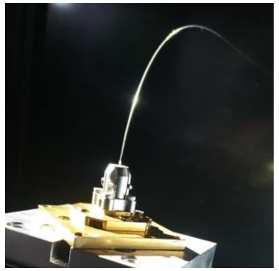
Photonic quantum computing companies

Psi-Quantum
USA - 2016



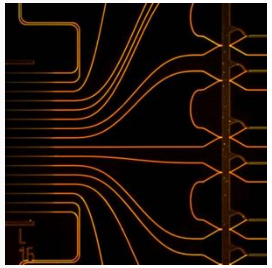
DV QC
SPDC sources

Quandela
France - 2017



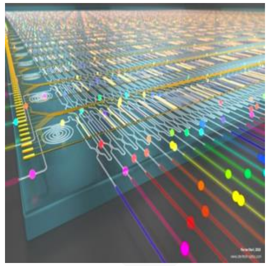
DV QC

Xanadu
Canada - 2018



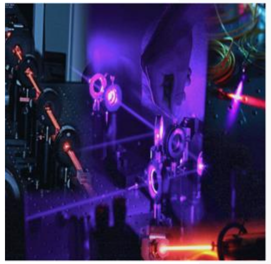
Continuous
variable QC

QuiX
Netherlands - 2019



SiN4 based
DV QC

ORCA
UK - 2020



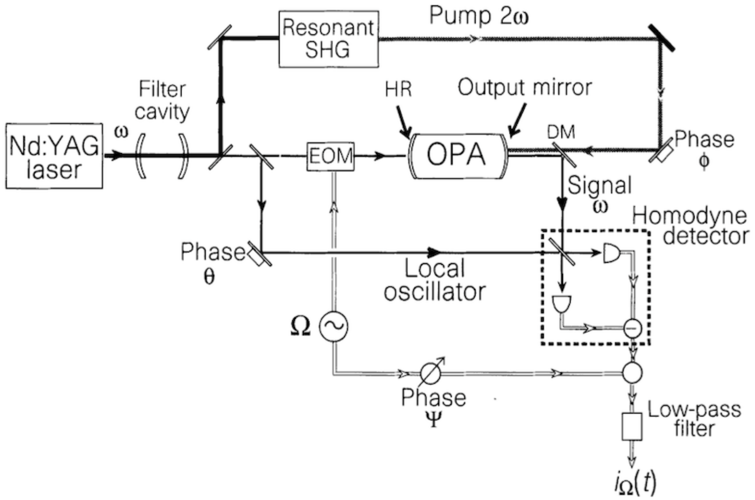
Memory-based
QC

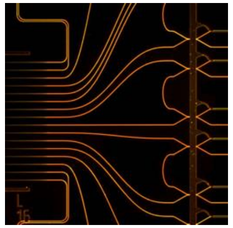
DV QC: discrete variable quantum computing
Information is encoded in single photons that can be in different modes (e.g. spatial or polarization)

Not covered here: continuous variable photonic quantum computing

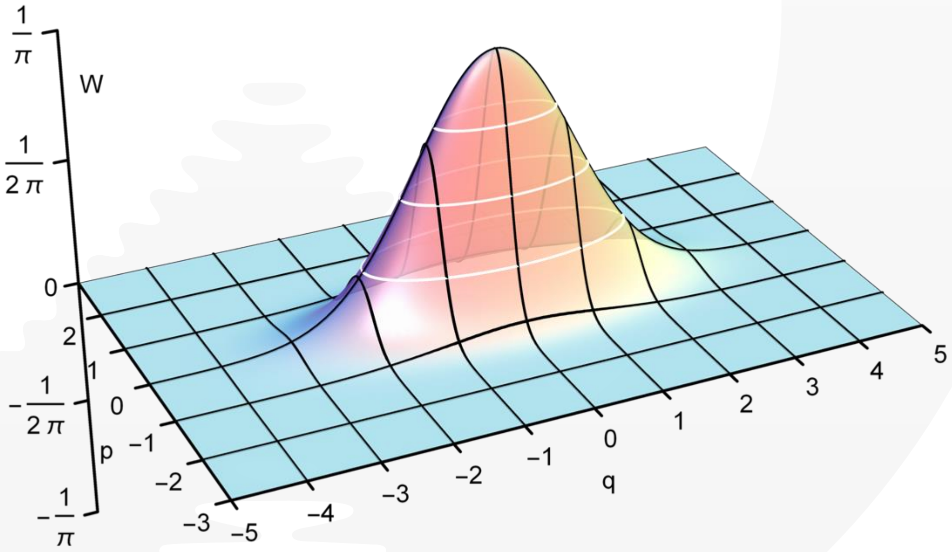
Qubit $|\phi\rangle = \phi_0 |0\rangle + \phi_1 |1\rangle$

Qumode $|\psi\rangle = \int dx\, \psi(x) |x\rangle$





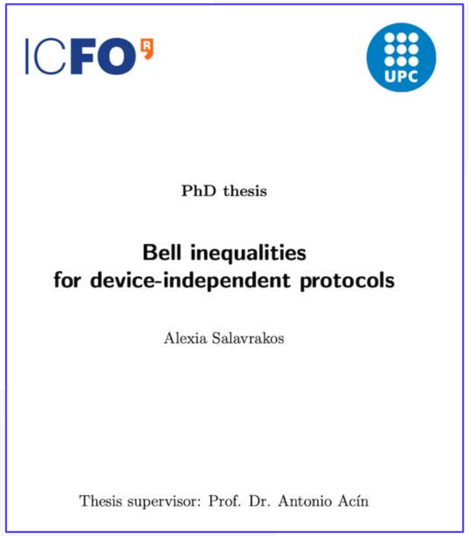
XANADU



G. Breitenbach et al. Nature 387 (1997)

Introducing myself

- Studied physics at Universite Libre de Bruxelles, Belgium
- PhD at Institute of Photonic Sciences (ICFO) in Barcelona, Spain, in quantum foundations
- Worked for a couple of years in data science and machine learning
- Joined Quandela in June 2022
- Research topics:
    - photonic quantum computing
    - quantum machine learning
    - machine learning for quantum


ICFO UPC

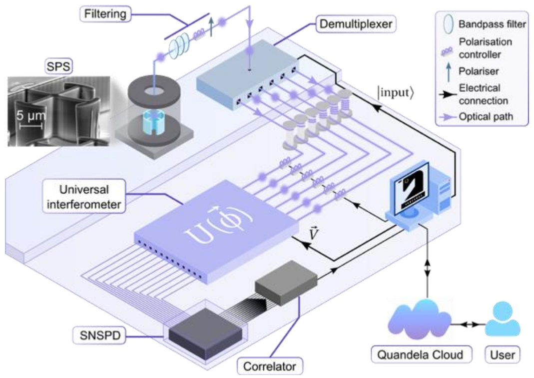
PhD thesis

Bell inequalities for device-independent protocols

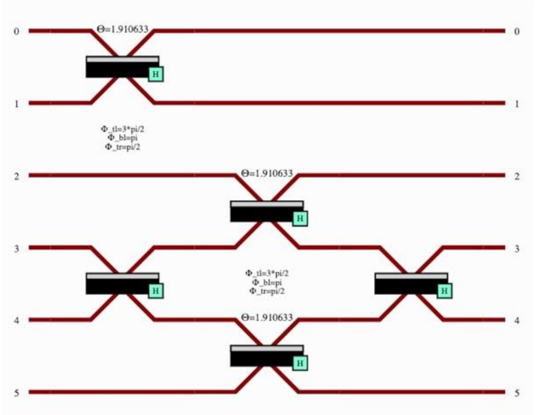
Alexia Salavrakos

Thesis supervisor: Prof. Dr. Antonio Acín

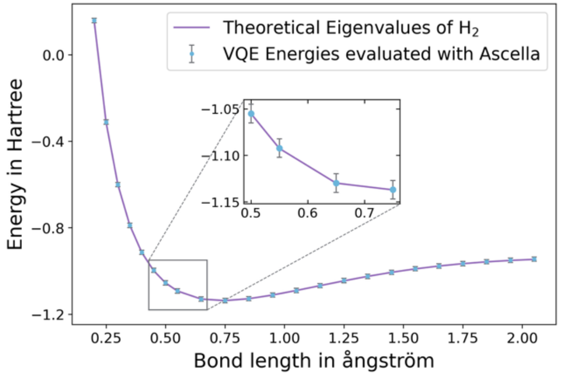
Structure of the presentation



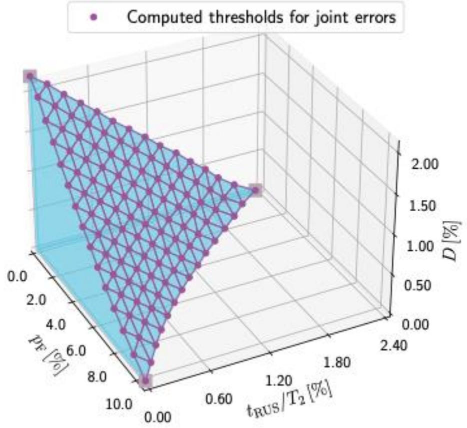
Experimental setup



Linear optical quantum computing



Algorithms examples



What next?

Perceval

Simulation close to the hardware
Noise models
Circuit design

MerLin

Photonic QML framework
PyTorch integration
Built for AI/ML practicioners

QUANDELA

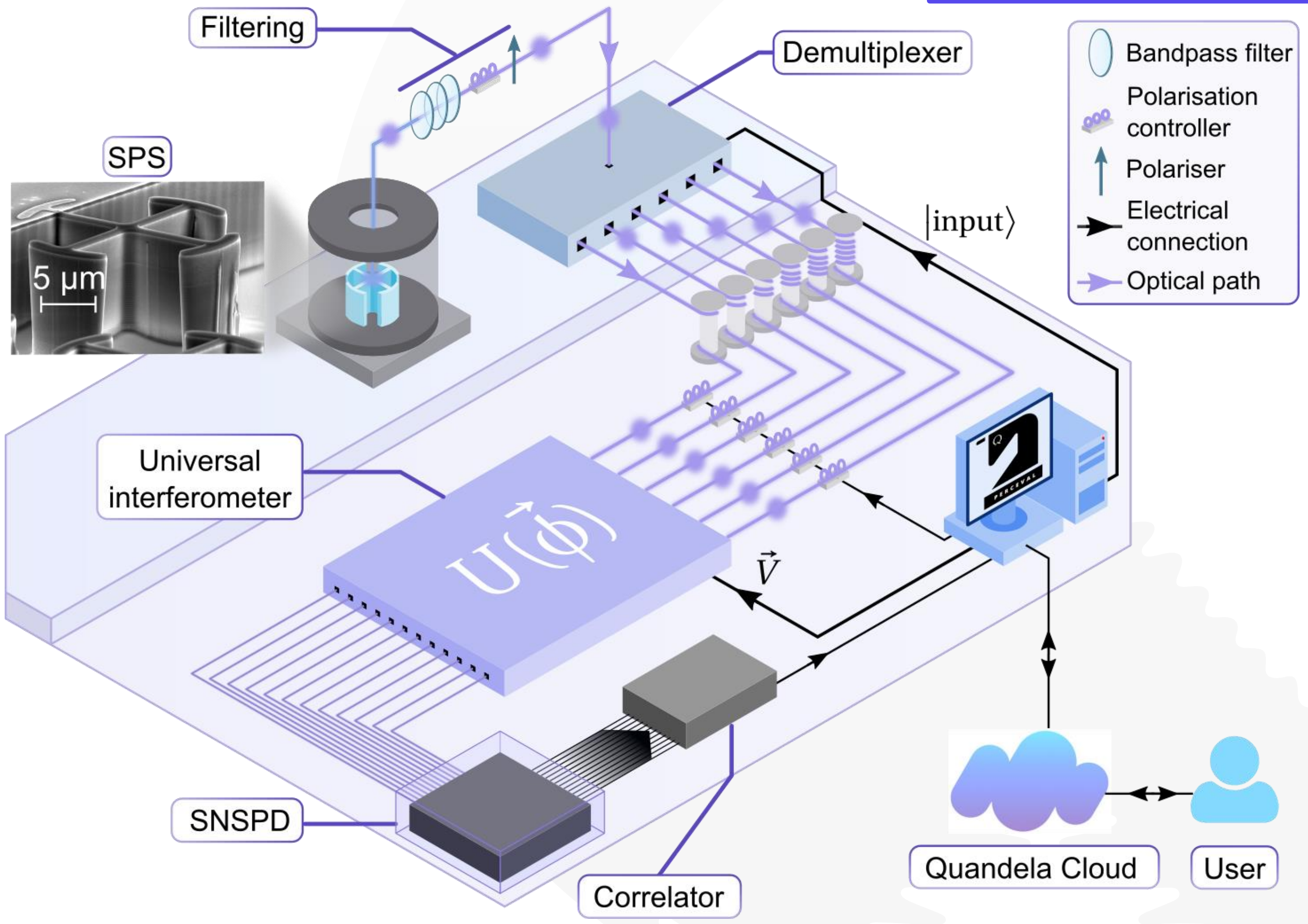
Experimental setup

EXPERIMENTAL SETUP

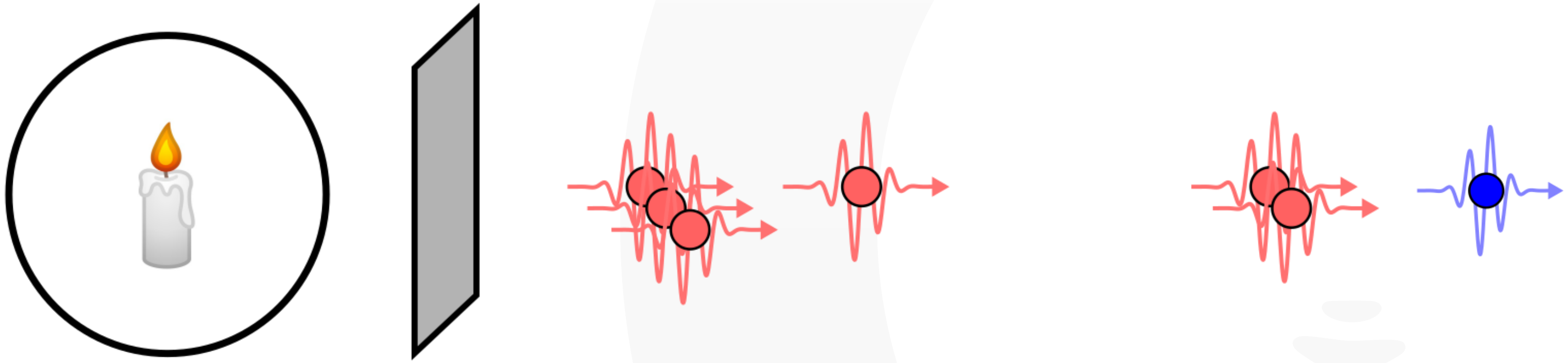
Near-term quantum processors

A versatile single-photon-based quantum computing platform

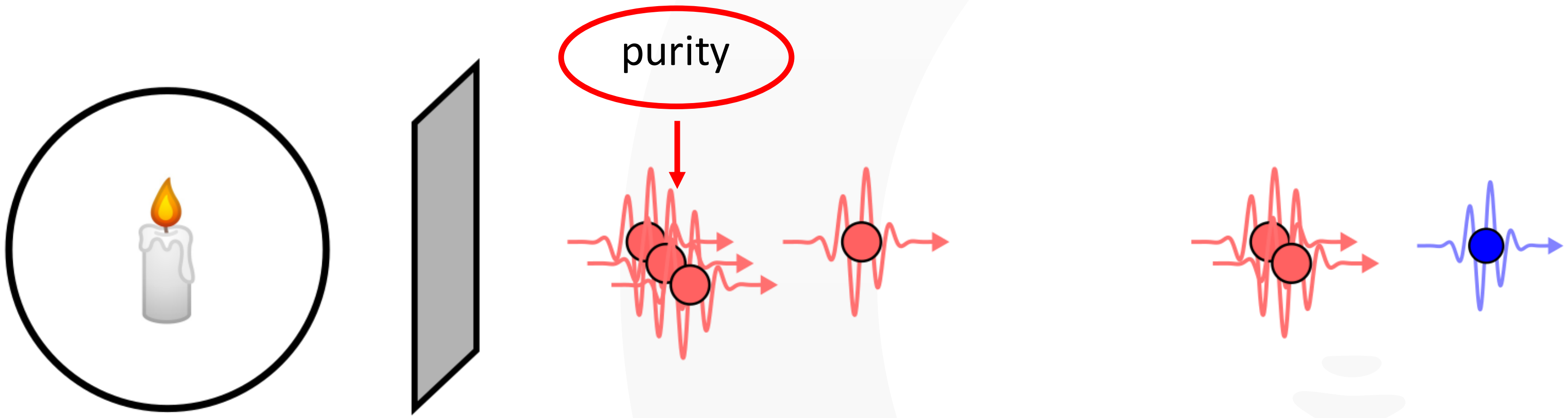
Nicolas Maring, Andreas Fyrillas, Mathias Pont, Edouard Ivanov, Petr Stepanov, Nico Margaria, William Hease, Anton Pishchagin, Aristide Lemaître, Isabelle Sagnes, Thi Huong Au, Sébastien Boissier, Eric Bertasi, Aurélien Baert, Mario Valdivia, Marie Billard, Ozan Acar, Alexandre Brieussel, Rawad Mezher, Stephen C. Wein, Alexia Salavrakos, Patrick Sinnott, Dario A. Fioretto, Pierre-Emmanuel Emeriau, Nadia Belabas, Shane Mansfield, Pascale Senellart, Jean Senellart & Niccolo Somaschi



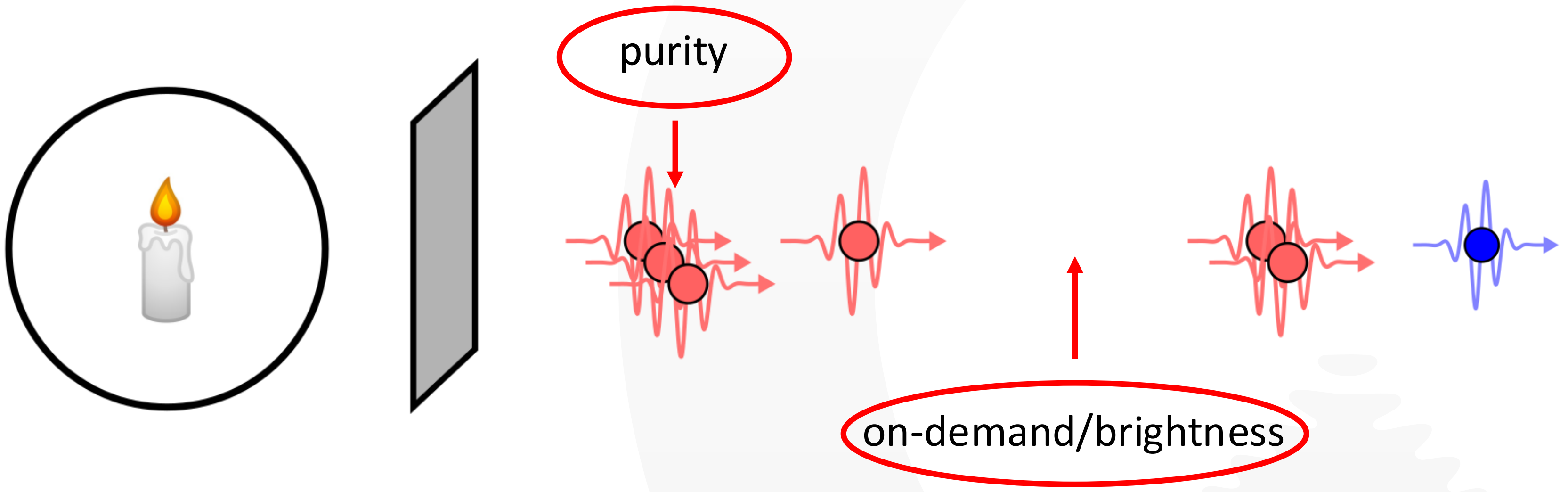
Desired properties for a single photon source



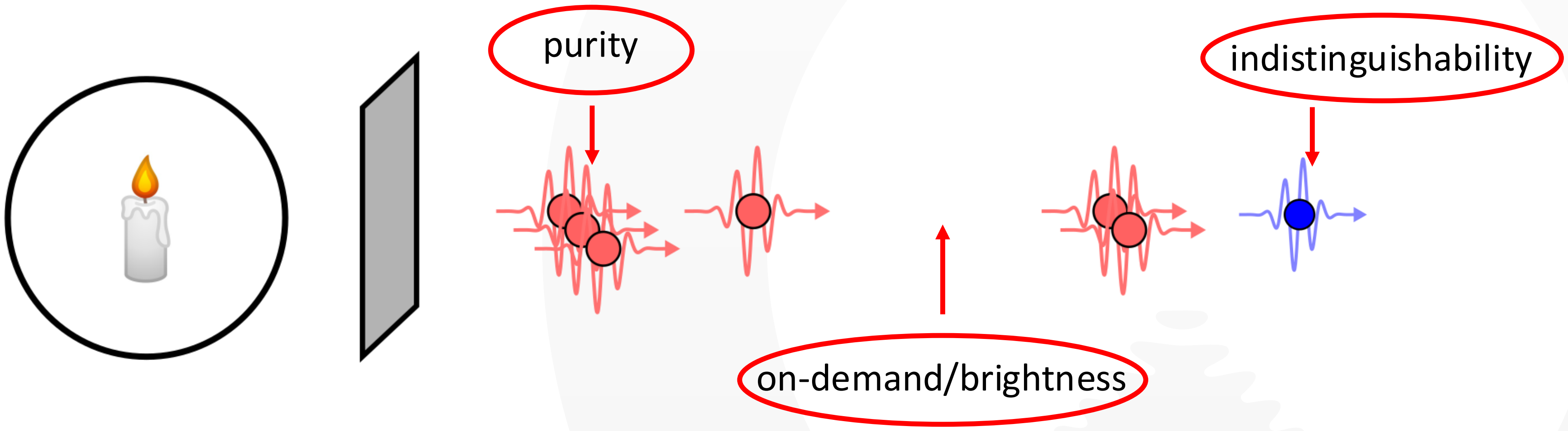
Desired properties for a single photon source



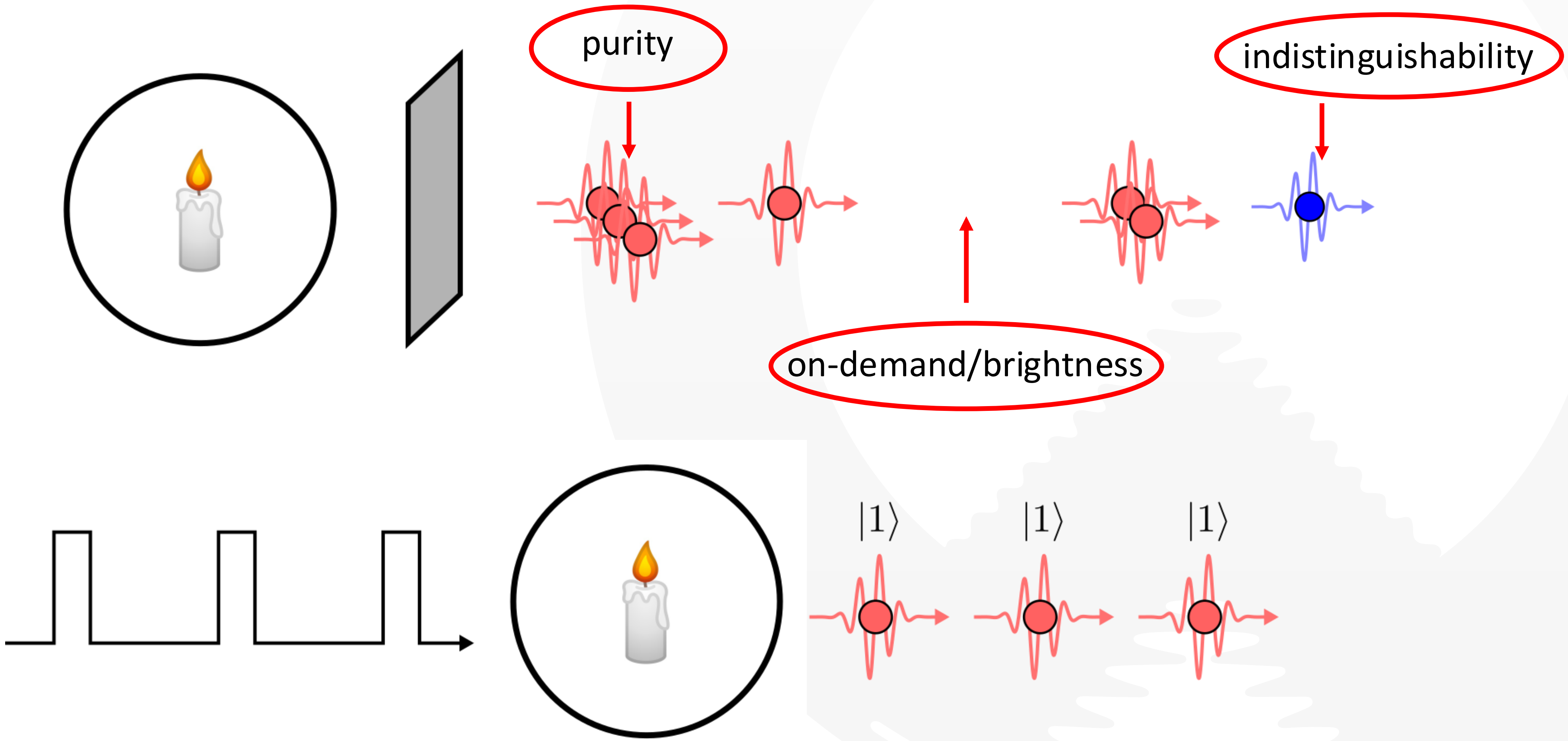
Desired properties for a single photon source



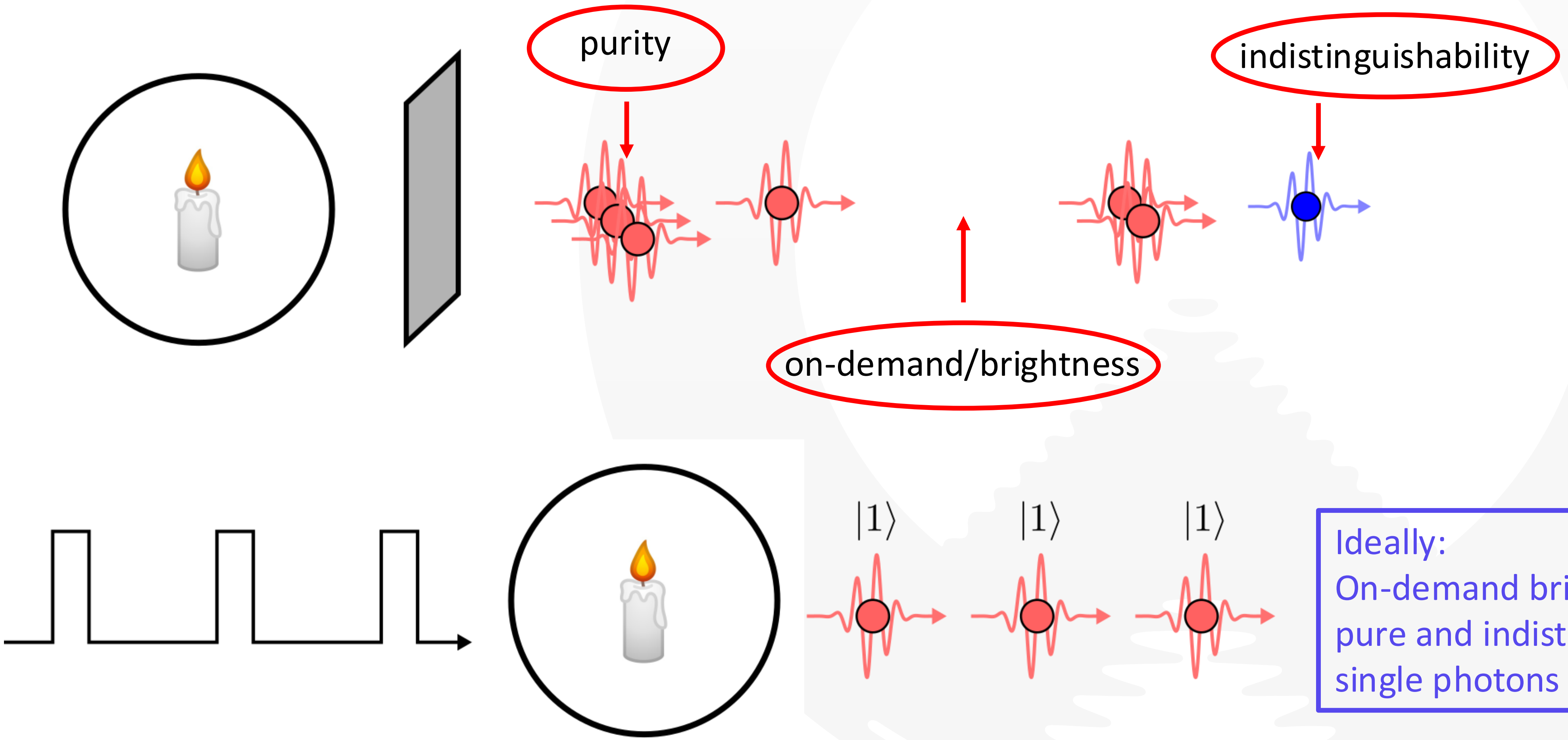
Desired properties for a single photon source



Desired properties for a single photon source



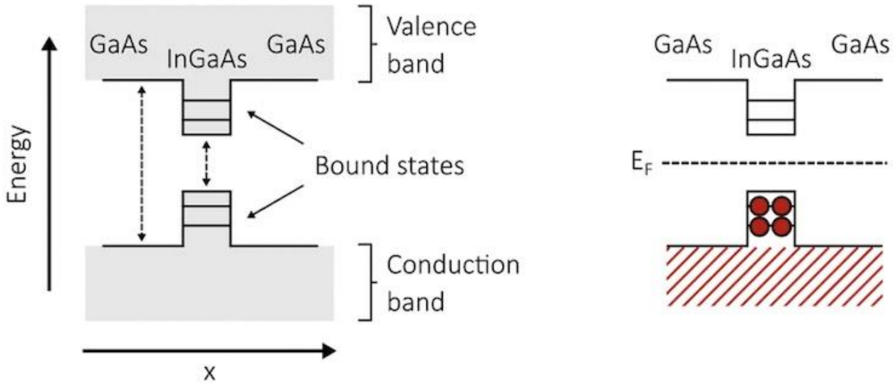
Desired properties for a single photon source



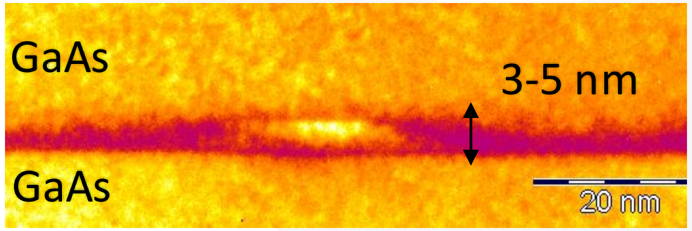
Ideally:
On-demand bright source of pure and indistinguishable single photons

Quantum dots as single photon sources

Quantum dot

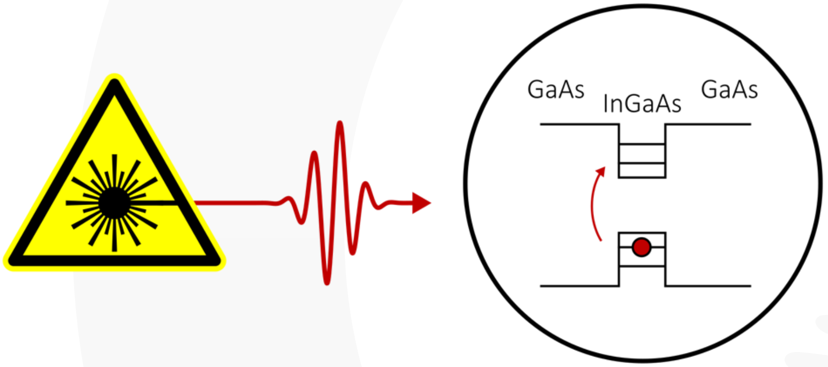


InGaAs

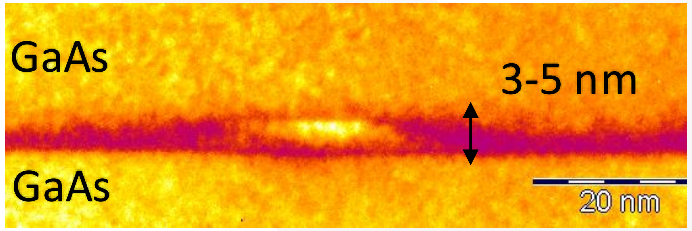


Quantum dots as single photon sources

Quantum dot

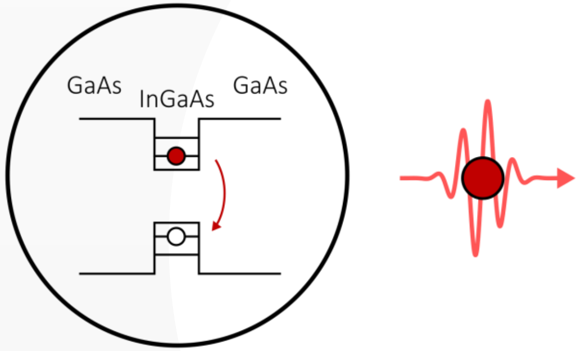


InGaAs

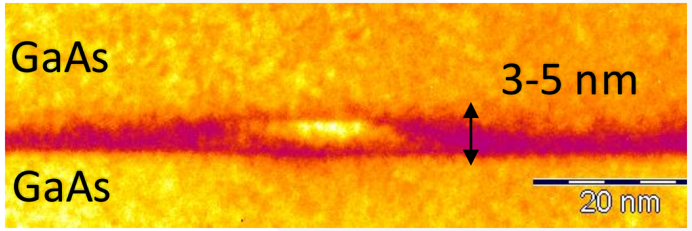


Quantum dots as single photon sources

Quantum dot

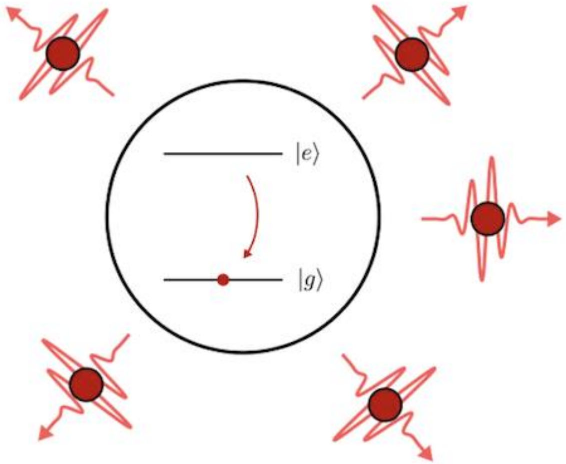


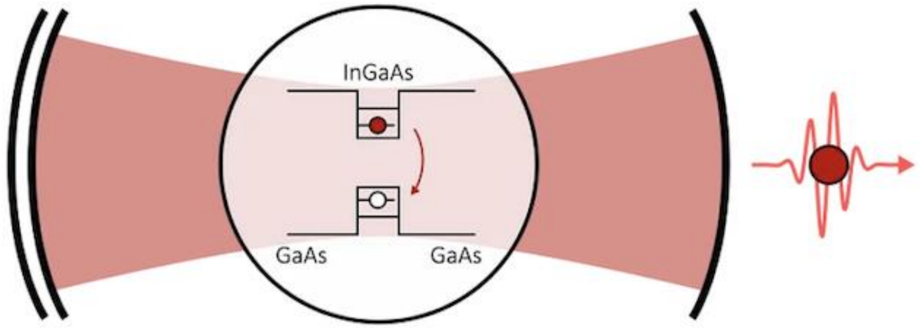
InGaAs



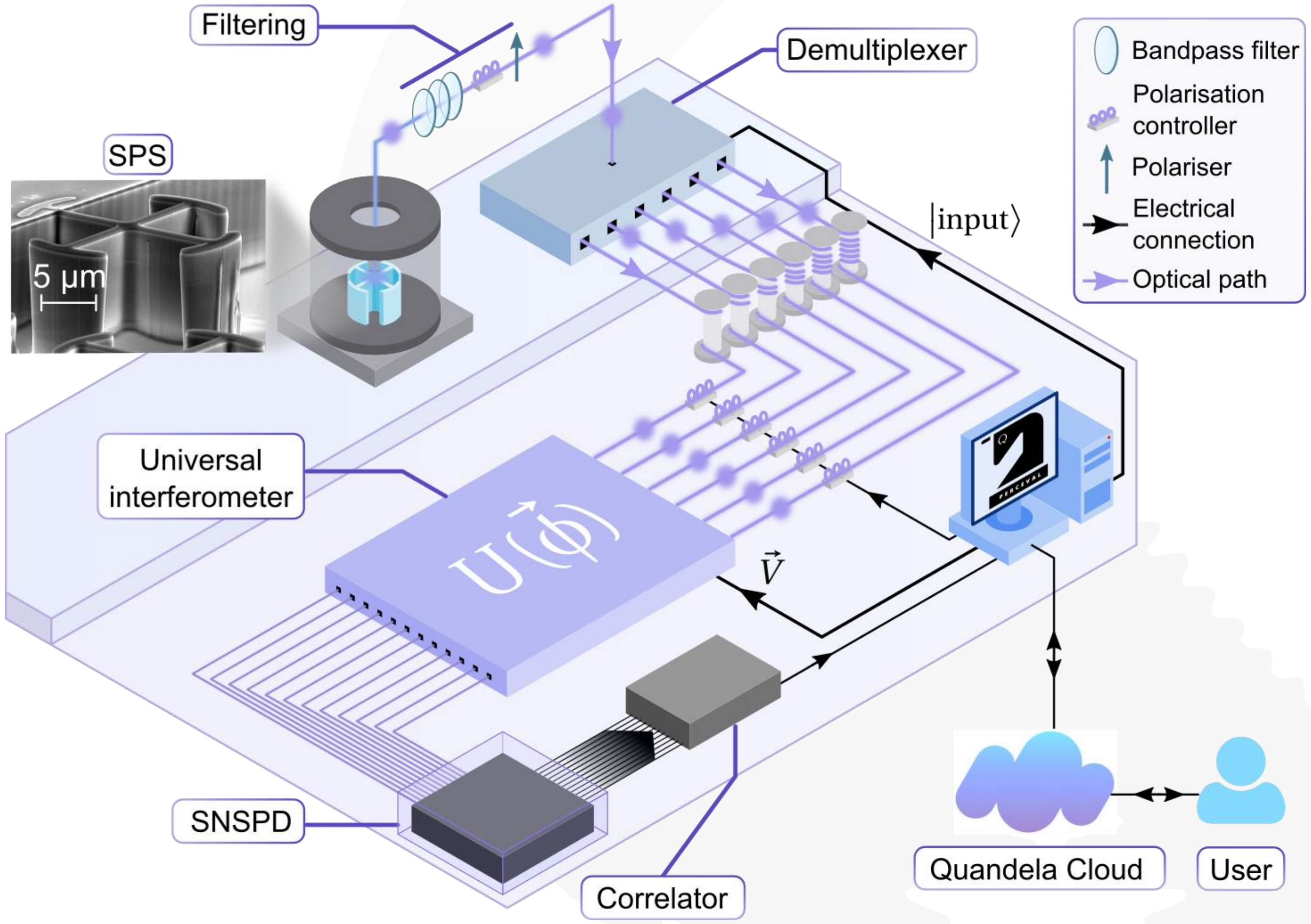
Quantum dots as single photon sources

Quantum dot in micropillar cavity

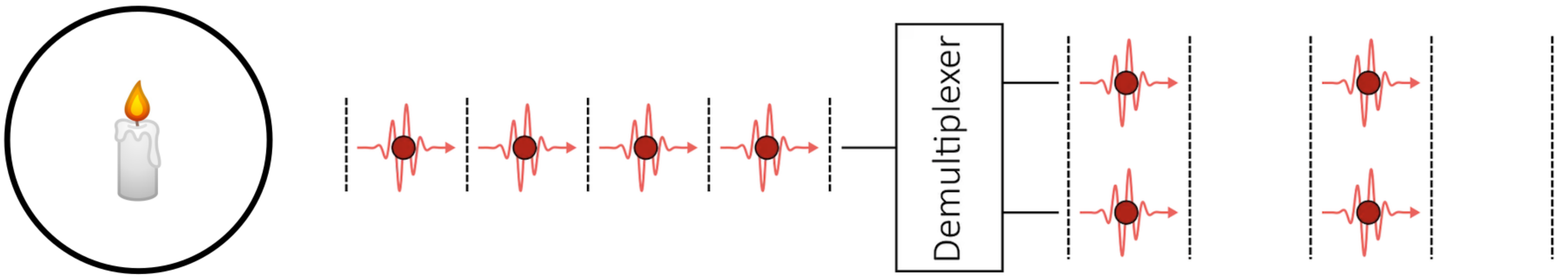




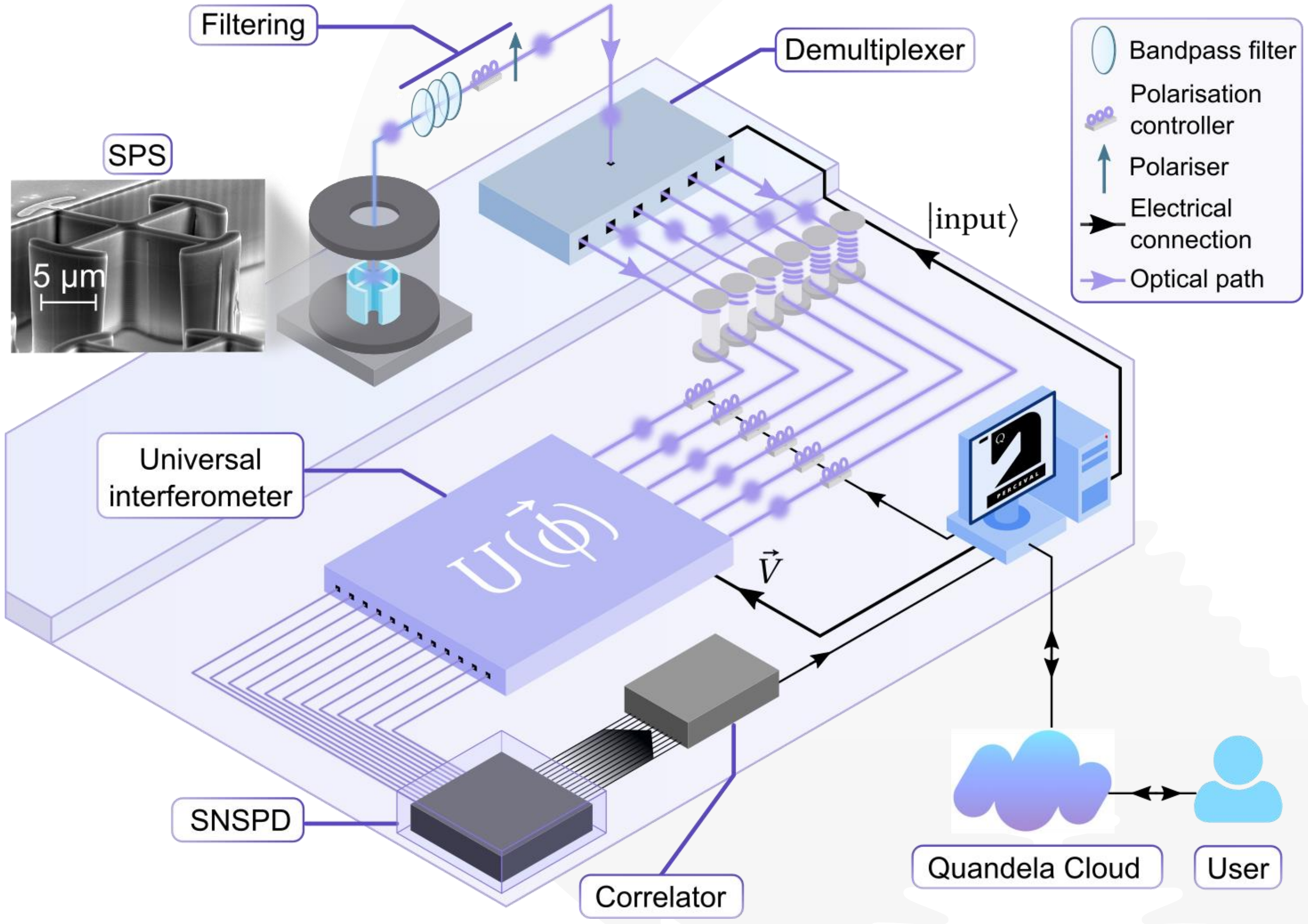
Near-term quantum processors



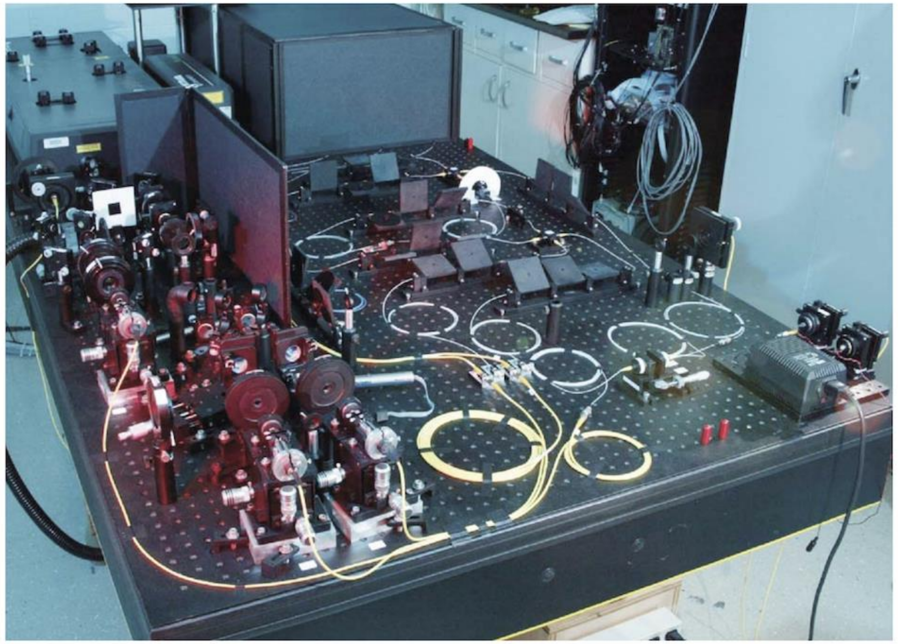
Demultiplexer for deterministic sources

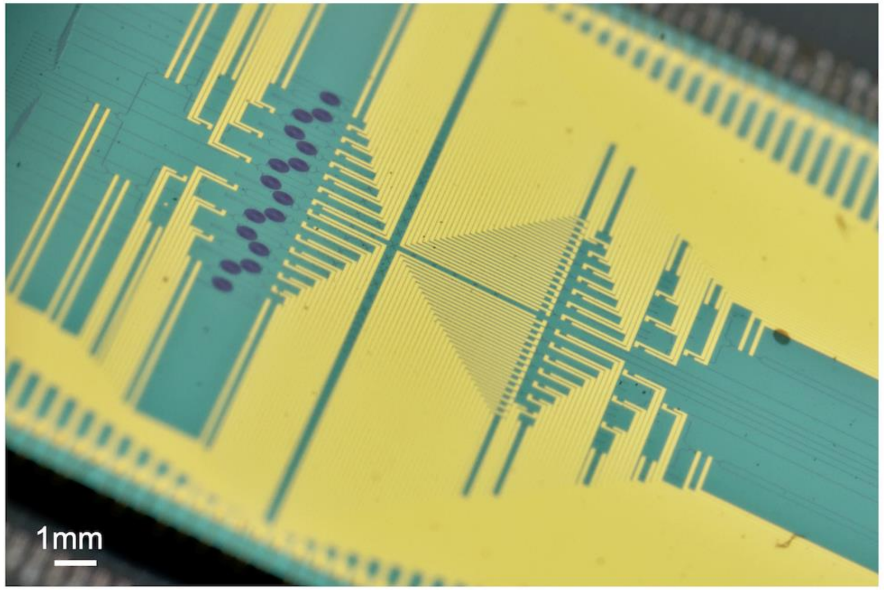


Near-term quantum processors



Photonic integrated chips

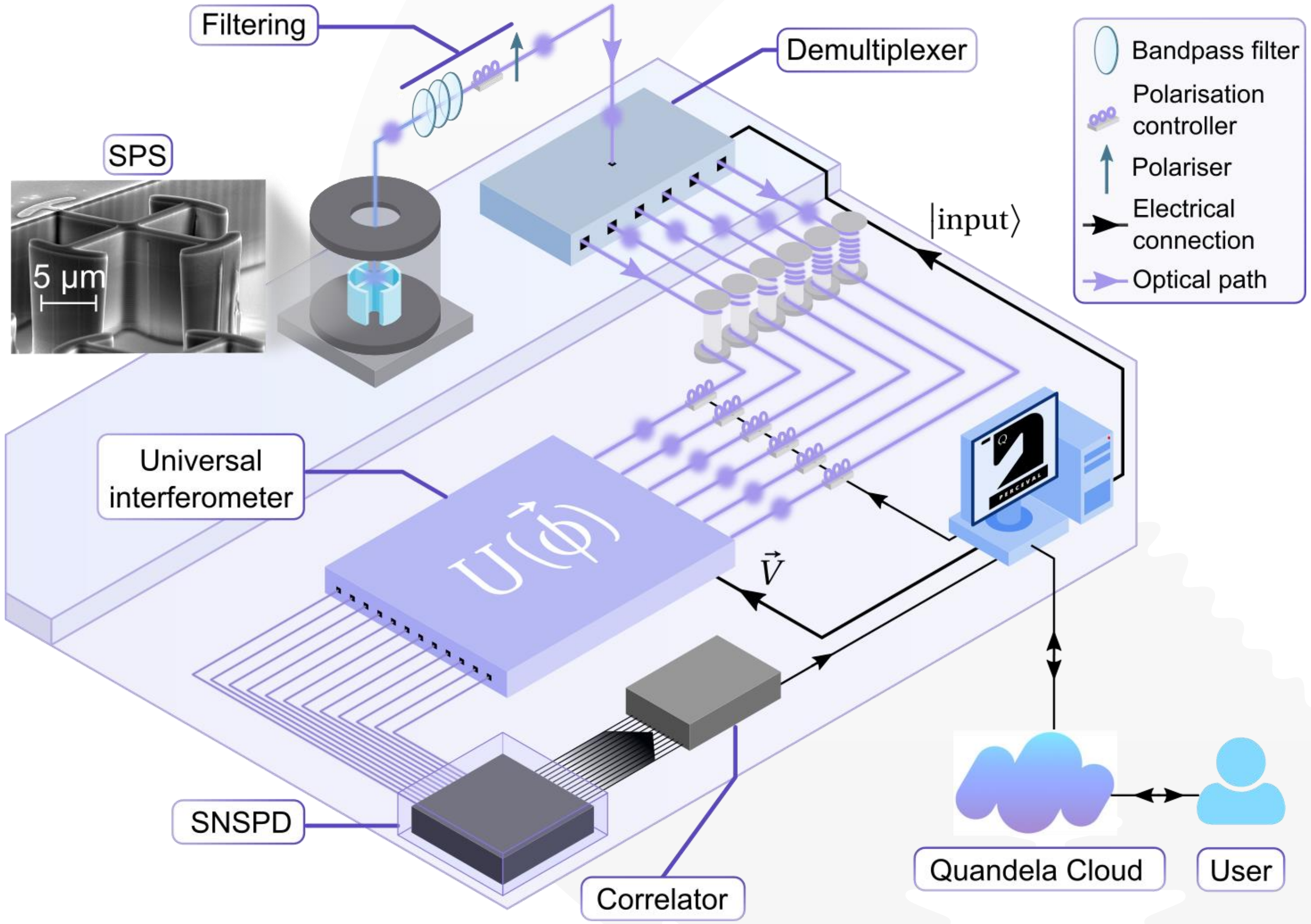




J. Wang et al. Science 360 (2018)

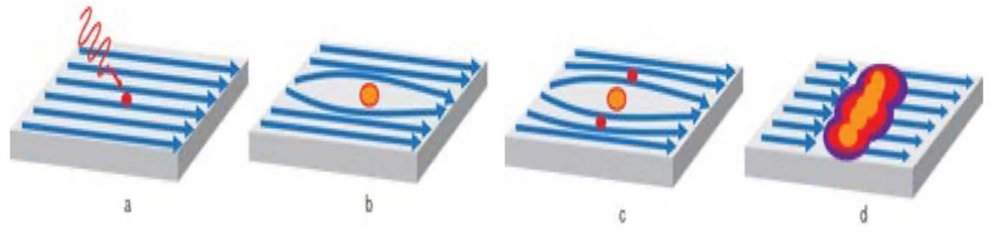
T. B. Pittman et al. Johns Hopkins APL Technical Digest 25 2 (2004)

Near-term quantum processors



Photon detectors

Superconducting Nanowire Single Photon Detector (SNSPD)



> 95% single photon detection

Photon number resolution (PNR)

Ideally: **PNR detectors**

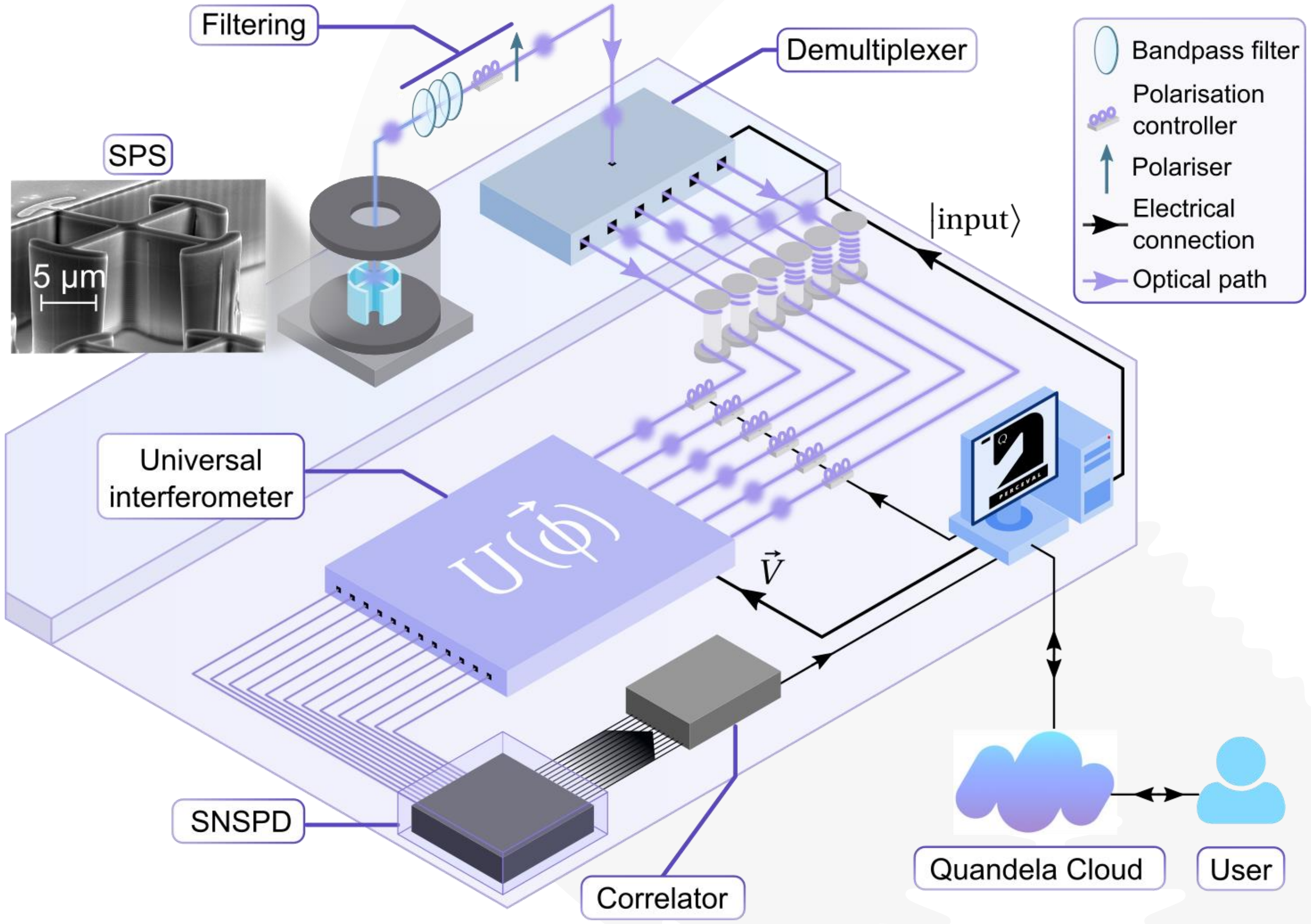
Output states such as $|0210301\rangle$

Current technology: **threshold detectors**

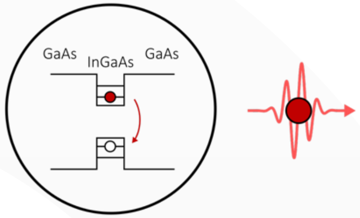
Indicates click or no click

Output states such as $|0110101\rangle$

Near-term quantum processors

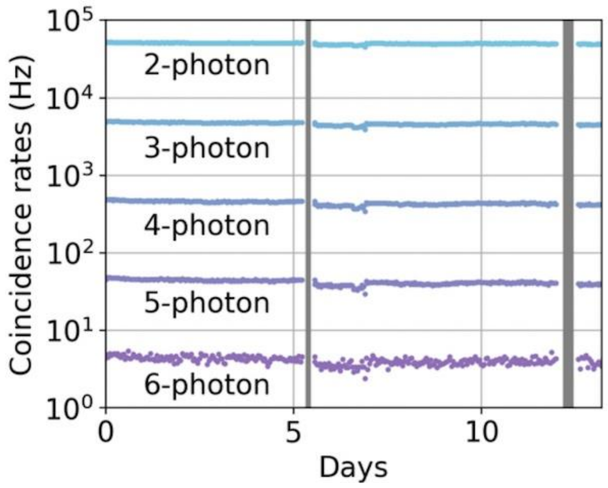


Sources of noise



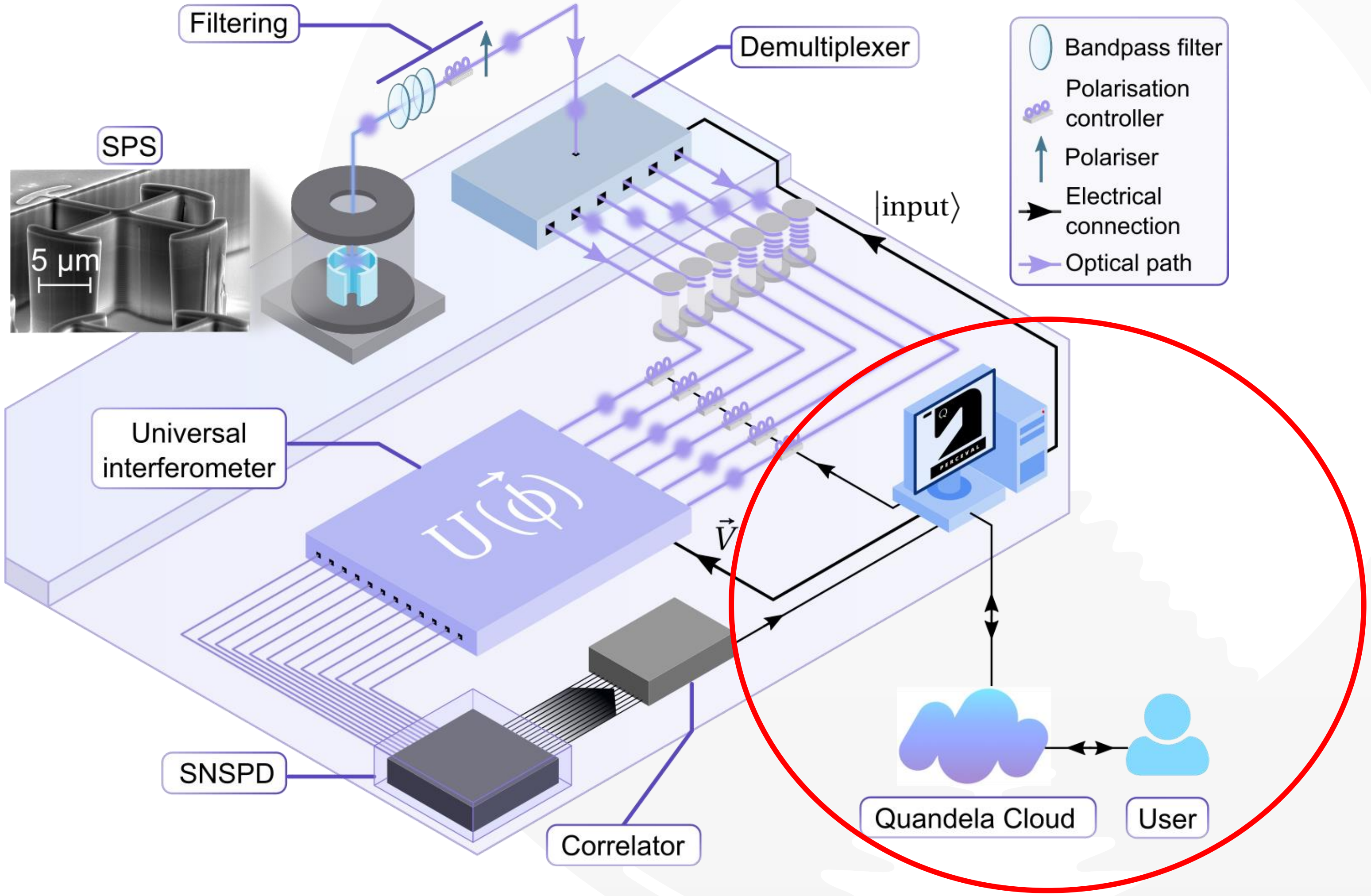
- Photon loss is the main source of noise
- Affects the whole circuit
- Exponential scaling with number of photons in an experiment

+ single photon purity
+ photon distinguishability

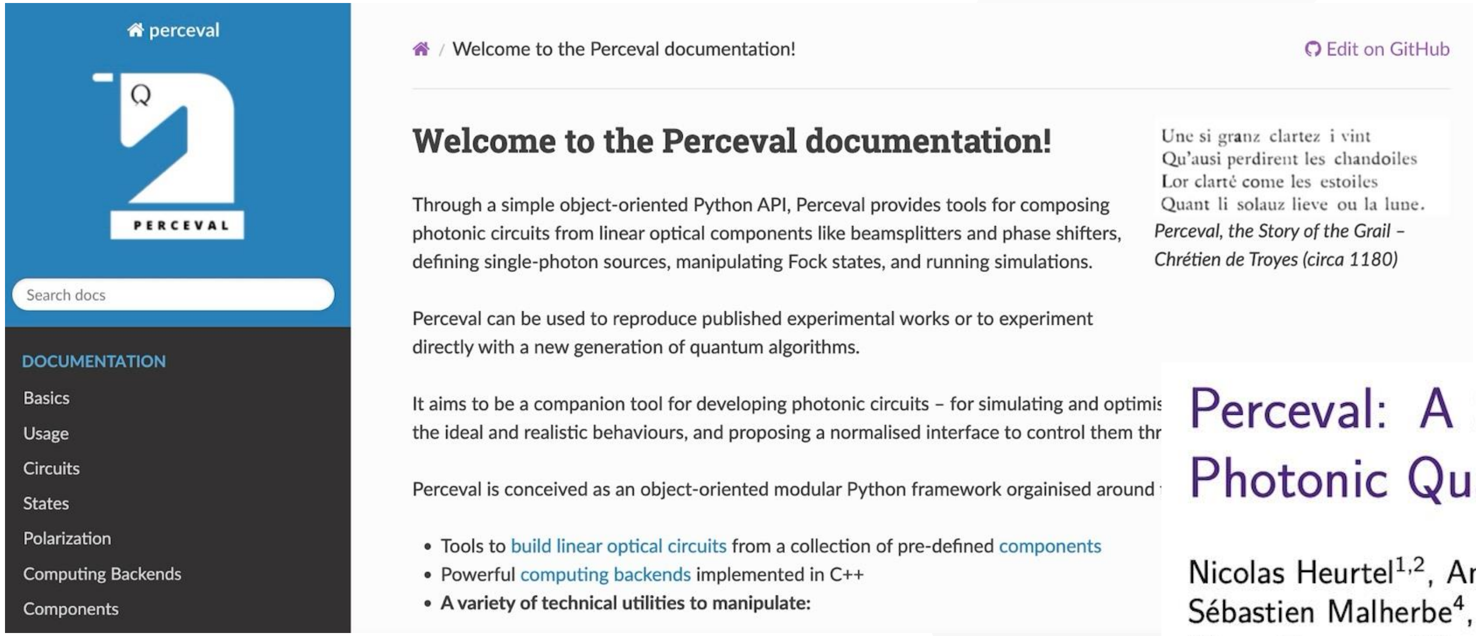


Module	Transmission/Efficiency	Near-term targets
First lens brightness	55 %	80% [69]
Single-mode fiber coupling	70 %	85% [70]
Spectral Filtering module	75 %	>82%[*]
Demultiplexer	70 %	>80%[*]
PIC insertion and transmission	45 %	70% [71]
SNSPDs	92 %	>95%[**]
Total	8.4 ± 0.2 %	27%
Pump laser repetition rate	80 MHz	320 MHz [72]
6-photon countrate	4 Hz	~35 kHz (computed)
12-photon countrate	200 nHz (computed)	~10 Hz (computed)

Near-term quantum processors



Perceval software



perceval

PERCEVAL

Search docs

DOCUMENTATION

- Basics
- Usage
- Circuits
- States
- Polarization
- Computing Backends
- Components

/ Welcome to the Perceval documentation!

Edit on GitHub

Welcome to the Perceval documentation!

Une si granz clartez i vint
Qu'ausi perdirent les chandoiles
Lor clarté come les estoiles
Quant li solauz lieve ou la lune.

Perceval, the Story of the Grail – Chrétien de Troyes (circa 1180)

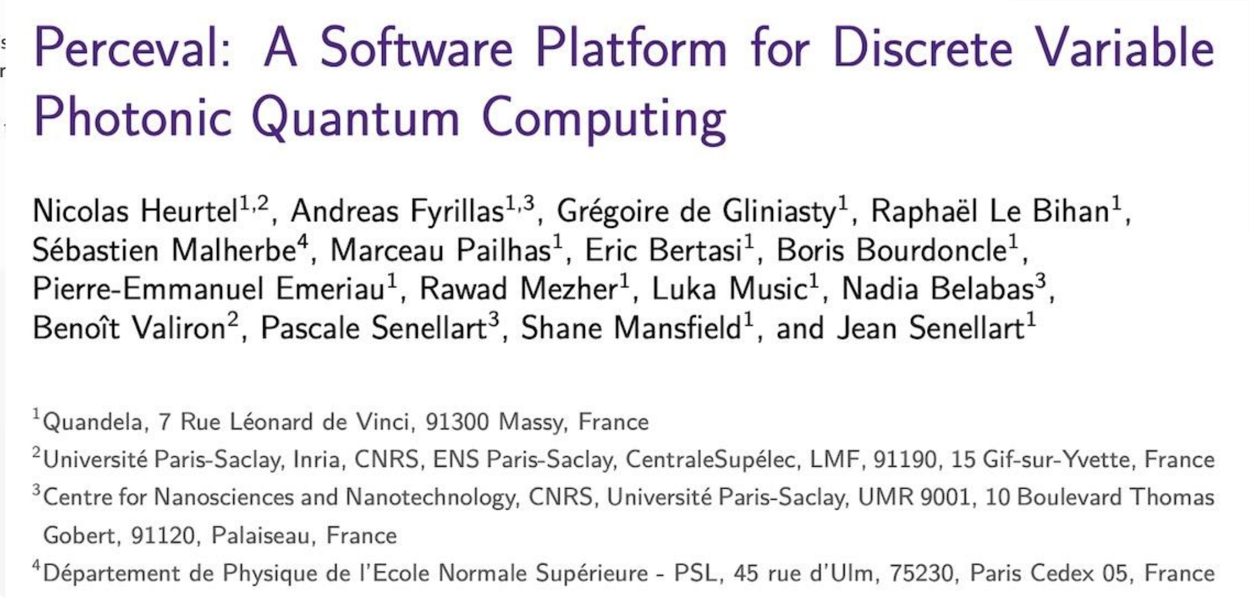
Through a simple object-oriented Python API, Perceval provides tools for composing photonic circuits from linear optical components like beamsplitters and phase shifters, defining single-photon sources, manipulating Fock states, and running simulations.

Perceval can be used to reproduce published experimental works or to experiment directly with a new generation of quantum algorithms.

It aims to be a companion tool for developing photonic circuits – for simulating and optimis the ideal and realistic behaviours, and proposing a normalised interface to control them thr

Perceval is conceived as an object-oriented modular Python framework orgainised around

- Tools to build linear optical circuits from a collection of pre-defined components
- Powerful computing backends implemented in C++
- **A variety of technical utilities to manipulate:**



Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing

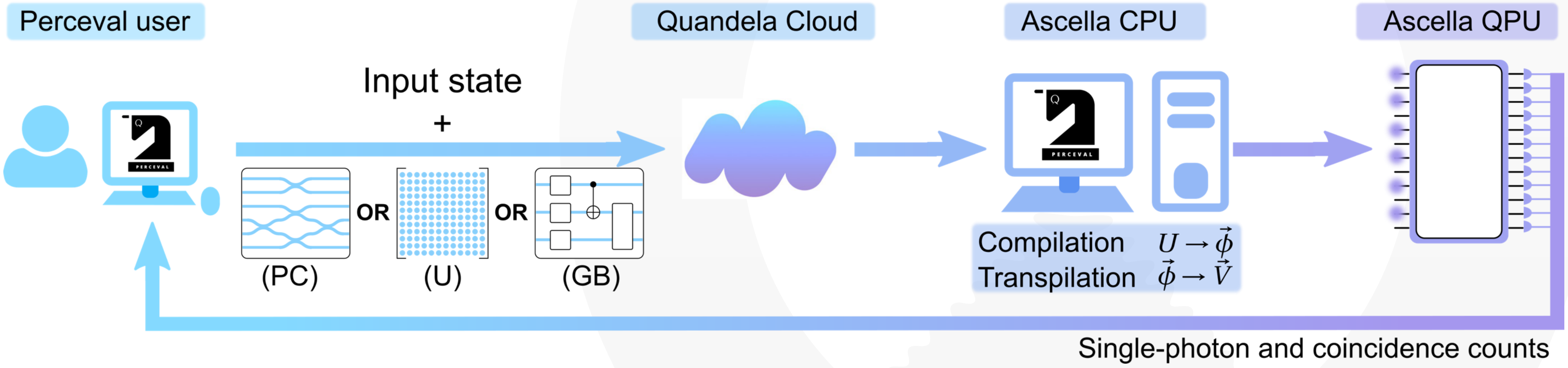
Nicolas Heurtel[1,2], Andreas Fyrillas[1,3], Grégoire de Gliniasty[1], Raphaël Le Bihan[1], Sébastien Malherbe[4], Marceau Pailhas[1], Eric Bertasi[1], Boris Bourdoncle[1], Pierre-Emmanuel Emeriau[1], Rawad Mezher[1], Luka Music[1], Nadia Belabas[3], Benoît Valiron[2], Pascale Senellart[3], Shane Mansfield[1], and Jean Senellart[1]

[1]Quandela, 7 Rue Léonard de Vinci, 91300 Massy, France
[2]Université Paris-Saclay, Inria, CNRS, ENS Paris-Saclay, CentraleSupélec, LMF, 91190, 15 Gif-sur-Yvette, France
[3]Centre for Nanosciences and Nanotechnology, CNRS, Université Paris-Saclay, UMR 9001, 10 Boulevard Thomas Gobert, 91120, Palaiseau, France
[4]Département de Physique de l'Ecole Normale Supérieure - PSL, 45 rue d'Ulm, 75230, Paris Cedex 05, France

```
pip install perceval-quandela
import perceval as pcvl
```

+ contains qiskit converter!

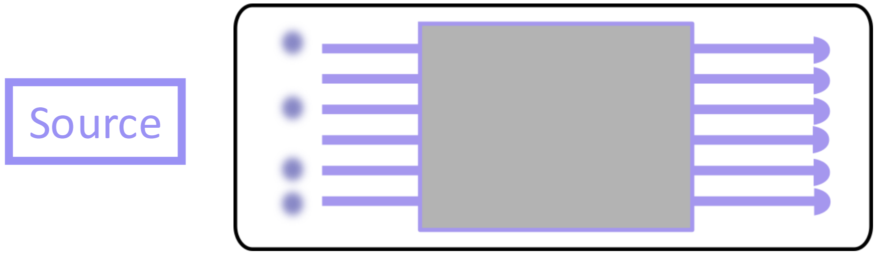
Cloud computing



Linear optical quantum computing

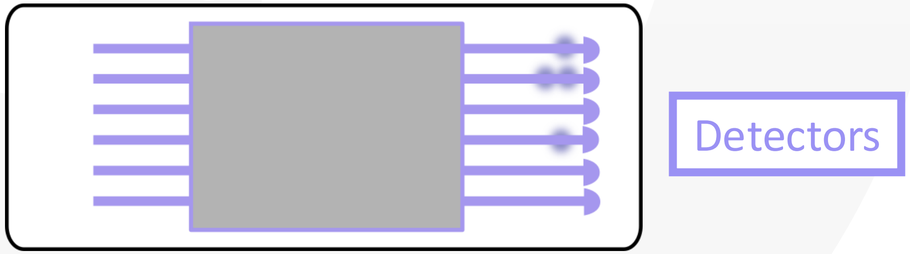
What do we mean by linear optics?

- Discrete variable linear optical quantum computing (DVLOQC) uses beam splitters and phase shifters on an input of single photons to perform quantum computing.
- Fock state of n photons in a single mode : $|n\rangle$
- Fock state on m modes: $|n_1,\ldots,n_m\rangle$



$|1,0,1,0,1,1\rangle$

Input Fock state



$|1,2,0,1,0,0\rangle$

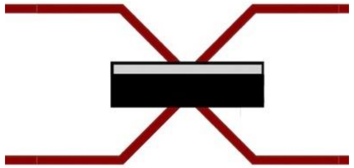
Output Fock state

What do we mean by linear optics?

- Linear optical transformation on m modes $U \in U(m)$.

- Linear optical transformations are made of beam splitters (BS) which are $U(2)$ transformations (phases) and phase shifters (PS) which are $U(1)$ transformations.

Beamsplitter



$$\begin{bmatrix} e^{i(\phi_{tl}+\phi_{tr})}\cos\left(\frac{\theta}{2}\right) & ie^{i(\phi_{bl}+\phi_{tr})}\sin\left(\frac{\theta}{2}\right) \\ ie^{i(\phi_{tl}+\phi_{br})}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi_{bl}+\phi_{br})}\cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

Phase shifter

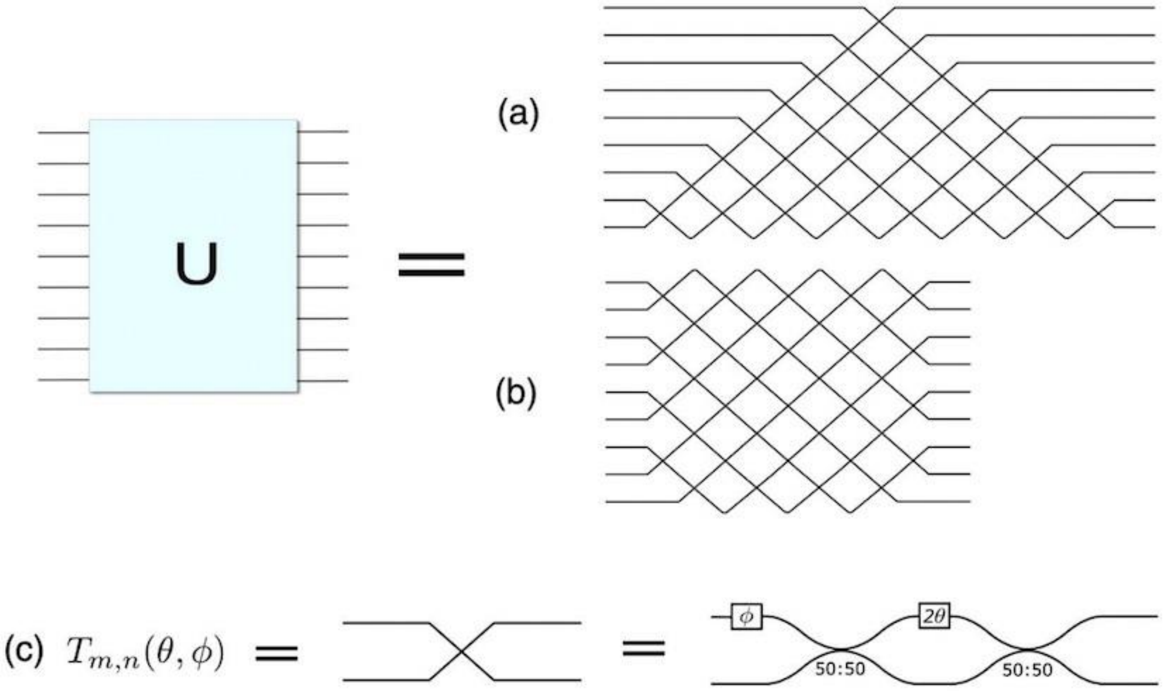
$$\begin{bmatrix} e^{i\phi} \end{bmatrix}$$

Implementing unitary transformations

Theorem by Reck et al. :

For any $U \in U(m)$, there exists an m-mode linear optical circuit implementing it

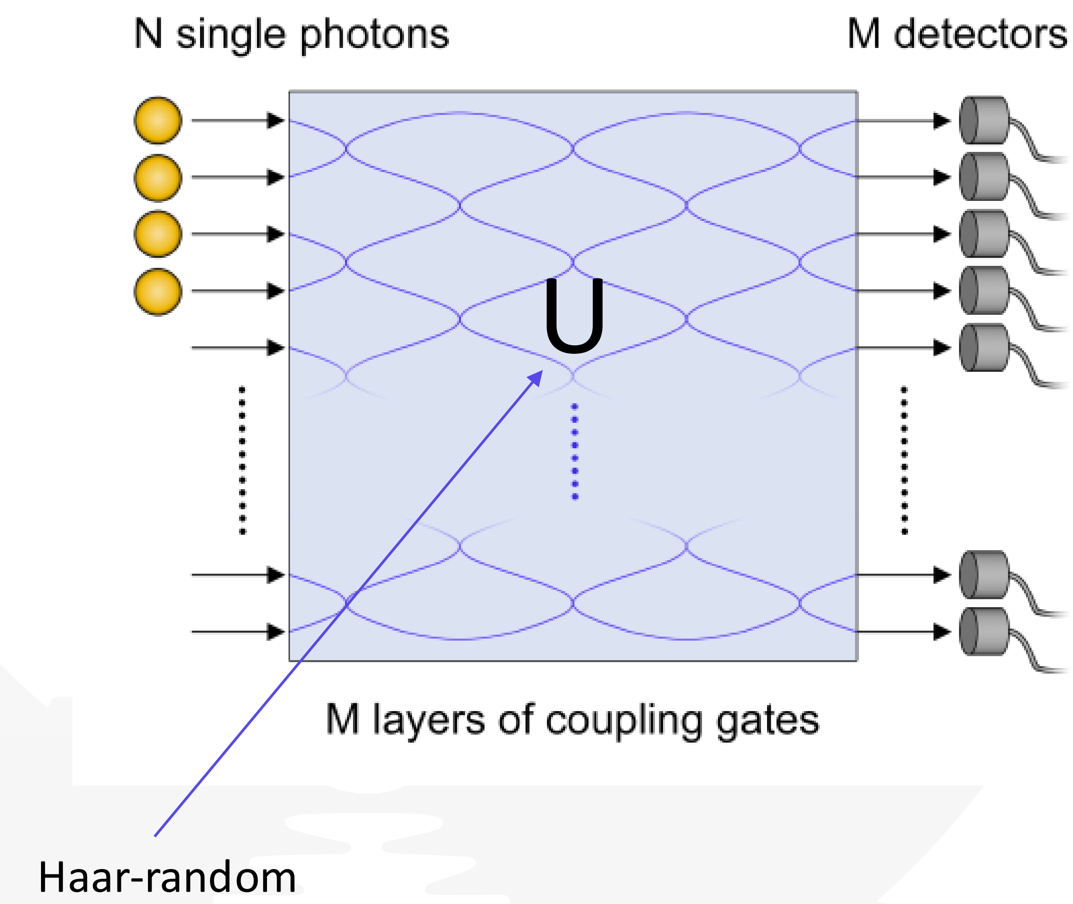
Scattering $m \times m$ unitary matrix implemented with $m(m-1)/2$ beam splitters



M. Reck et al. *Physical Review Letters* 73, 58 (1994)

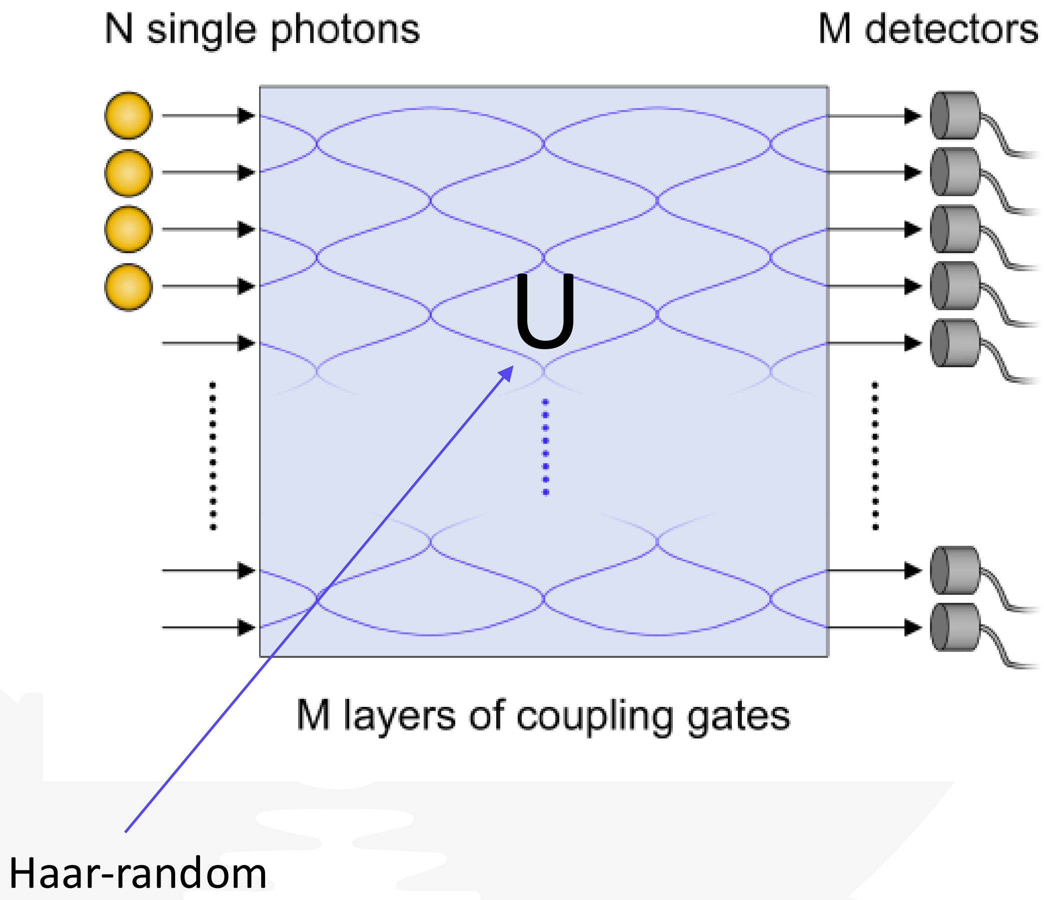
W. R. Clements et al. *Optica* 3, 12 (2016)

Boson sampling



- The probability to measure an output state $|s_1, \dots s_m\rangle$ is given by $|\alpha_s|^2 / s_1! \dots s_m!\, n_1! \dots n_m!$
- It can be shown that $|\alpha_s|^2 = |Per(U_{S,N})|^2$, $U_{S,N}$ submatrix of U determined by $S = (s_1, \dots., s_m)$ rows and N= $(n_1, \dots., n_m)$ column
- If A is an $\mathrm{n} \times n$ matrix, $Per(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} A_{i\sigma(i)}$
- Easy rule: like the determinant but with + signs everywhere
- The permanent, unlike the determinant, is hard to compute (best classical algorithms scale as O($n2^n$))

Boson sampling



- Boson sampling task: sample from the output probability distribution of a DVLOQC circuit
- More specifically, sample outputs S from P(S)
- With P(S) $\propto |Perm(U_{T,S})|^2$
- Hard to do classically, conditioned on some widely believed complexity theory conjectures

➢ Near term demonstration of quantum advantage
➢ Task may not be useful

Boson sampling

Gaussian Boson Sampling defined in continuous variable framework

RESEARCH

QUANTUM COMPUTING

Quantum computational advantage using photons

Han-Sen Zhong[1,2]*, Hui Wang[1,2]*, Yu-Hao Deng[1,2]*, Ming-Cheng Chen[1,2]*, Li-Chao Peng[1,2], Yi-Han Luo[1,2], Jian Qin[1,2], Dian Wu[1,2], Xing Ding[1,2], Yi Hu[1,2], Peng Hu[3], Xiao-Yan Yang[3], Wei-Jun Zhang[3], Hao Li[3], Yuxuan Li[4], Xiao Jiang[1,2], Lin Gan[4], Guangwen Yang[4], Lixing You[3], Zhen Wang[3], Li Li[1,2], Nai-Le Liu[1,2], Chao-Yang Lu[1,2]†, Jian-Wei Pan[1,2]†

Article

Quantum computational advantage with a programmable photonic processor

https://doi.org/10.1038/s41586-022-04725-x
Received: 12 November 2021
Accepted: 5 April 2022
Published online: 1 June 2022

Lars S. Madsen[1,3], Fabian Laudenbach[1,3], Mohsen Falamarzi. Askarani[1,3], Fabien Rortais[1], Trevor Vincent[1], Jacob F. F. Bulmer[1], Filippo M. Miatto[1], Leonhard Neuhaus[1], Lukas G. Helt[1], Matthew J. Collins[1], Adriana E. Lita[2], Thomas Gerrits[2], Sae Woo Nam[2], Varun D. Vaidya[1], Matteo Menotti[1], Ish Dhand[1], Zachary Vernon[1], Nicolás Quesada[1]✉ & Jonathan Lavoie[1]✉

Original Boson Sampling article

The Computational Complexity of Linear Optics

Scott Aaronson*   Alex Arkhipov†

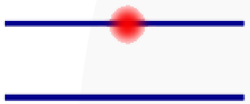
Qubit and logical gates with linear optics

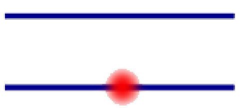
Choose an encoding

$|0\rangle_{qubit} := |1,0\rangle$

$|1\rangle_{qubit} := |0,1\rangle$

Dual rail





One qubit gates

$|0\rangle \rightarrow |0\rangle + |1\rangle$

Beamsplitter

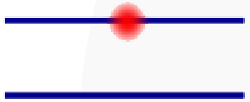
Qubit and logical gates with linear optics

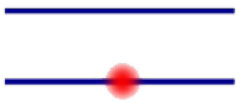
Choose an encoding

$|0\rangle_{qubit} := |1,0\rangle$

$|1\rangle_{qubit} := |0,1\rangle$

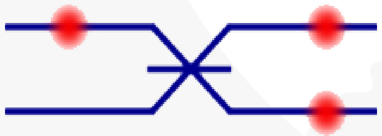
Dual rail





One qubit gates

$|0\rangle \rightarrow |0\rangle + |1\rangle$



Beamsplitter

Other encodings are possible, like polarization

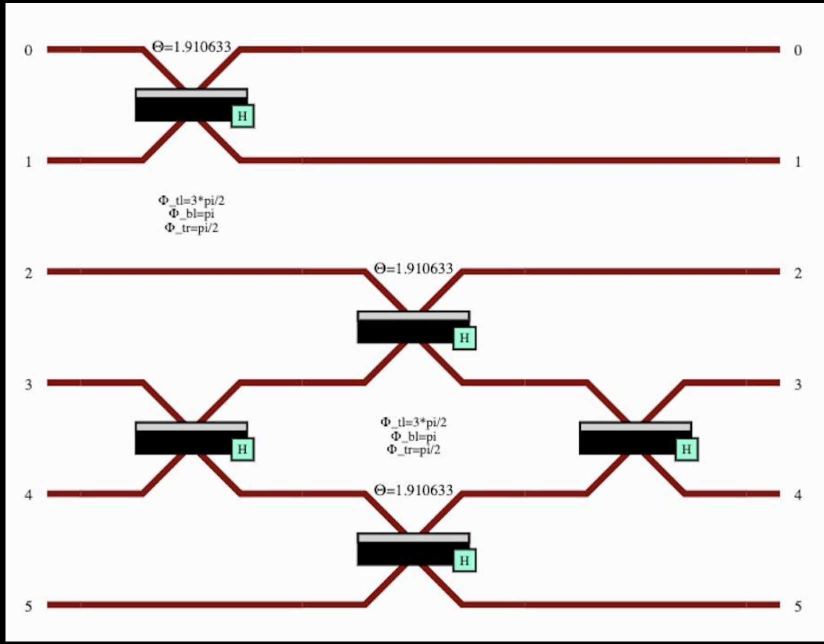
Qubit and logical gates with linear optics

However, some two-qubit gates cannot be achieved deterministically with passive linear optics

Options:

- Nonlinearities (materials unavailable)
- Post-selection (probabilistic)
- Heralding (probabilistic)
- Feedforward

Example: post-selected CNOT gate



Ralph, Timothy C., et al. *Physical Review A* 65.6 (2002): 062324.

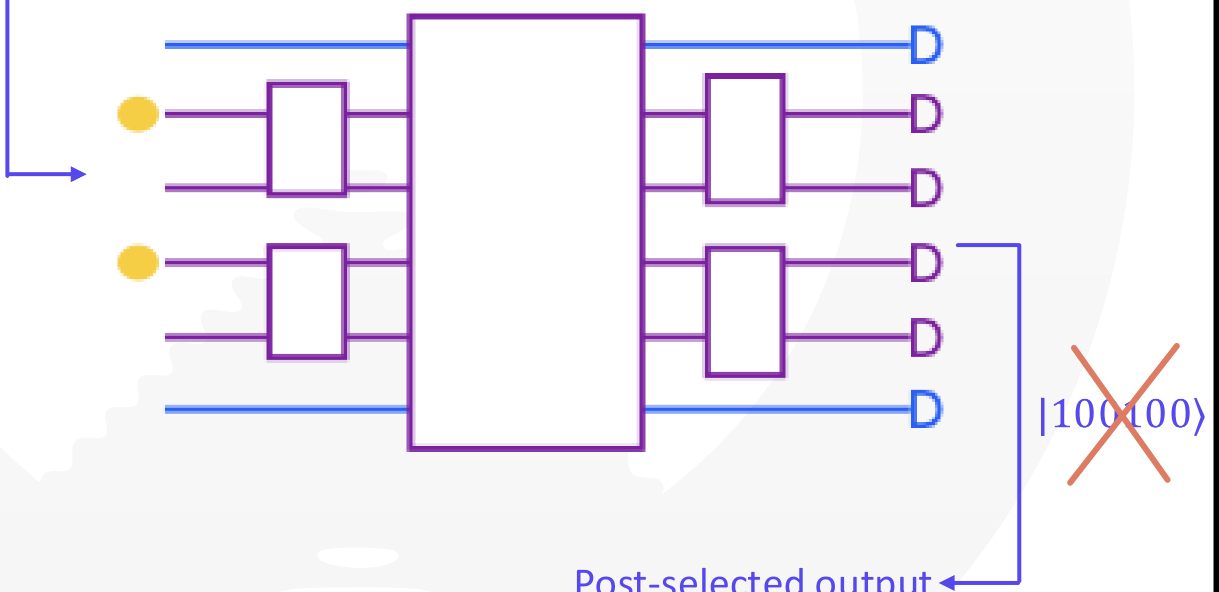
QUIZZ!

Algorithms design and examples

Two approaches to algorithm design on linear optics

Qubit-based

Dual rail encoding $|00\rangle = |1010\rangle$



Post-selected output

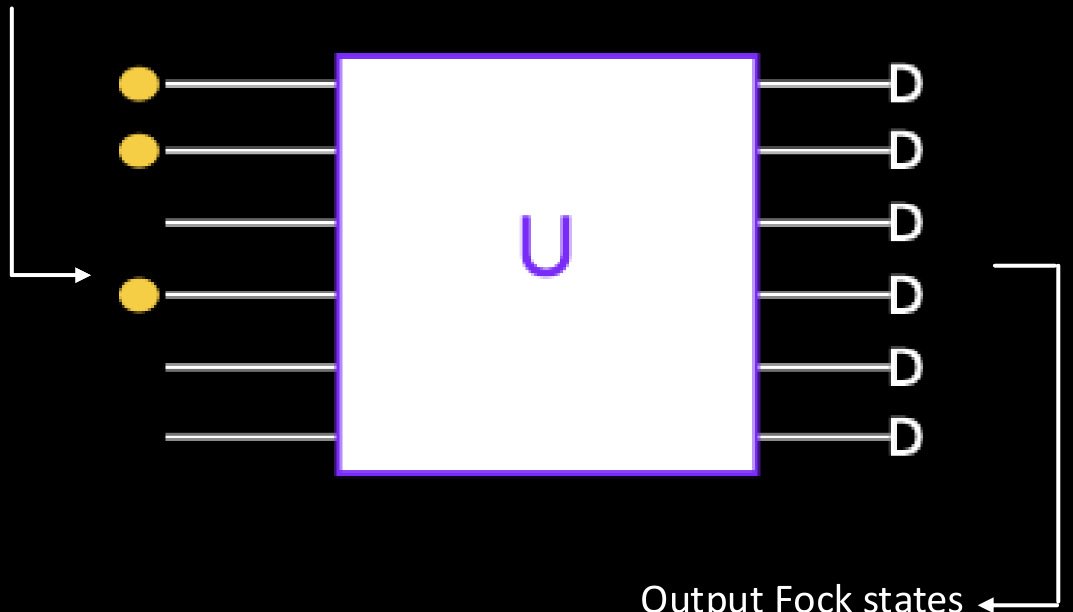
$|01\rangle = |1001\rangle$

$|11\rangle = |0101\rangle$

...

Photon-native

Input Fock state $|110100\rangle$



Output Fock states

$|111000\rangle$

$|001200\rangle$

...

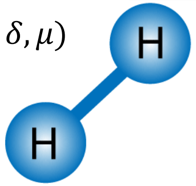
Qubit-based algorithms examples

Variational Quantum Eigensolver (VQE)

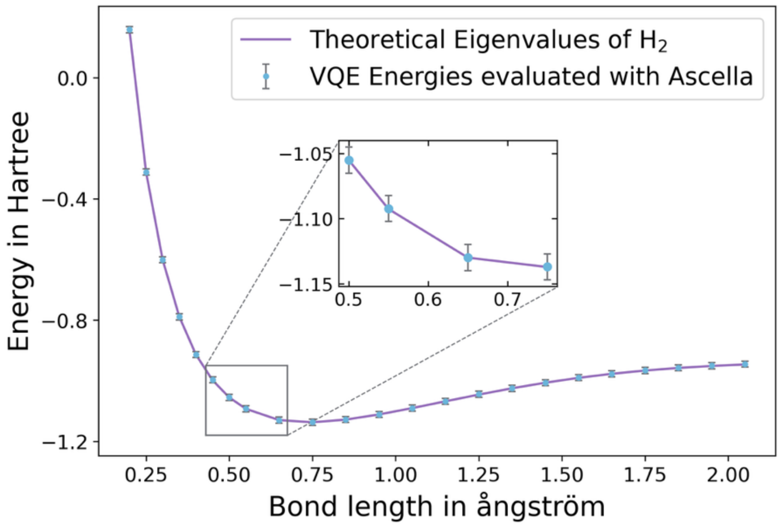
A versatile single-photon-based quantum computing platform

Nicolas Maring, Andreas Fyrillas, Mathias Pont, Edouard Ivanov, Petr Stepanov, Nico Margaria, William Hease, Anton Pishchagin, Aristide Lemaître, Isabelle Sagnes, Thi Huong Au, Sébastien Boissier, Eric Bertasi, Aurélien Baert, Mario Valdivia, Marie Billard, Ozan Acar, Alexandre Brieussel, Rawad Mezher, Stephen C. Wein, Alexia Salavrakos, Patrick Sinnott, Dario A. Fioretto, Pierre-Emmanuel Emeriau, Nadia Belabas, Shane Mansfield, Pascale Senellart, Jean Senellart & Niccolo Somaschi

$R = f(\alpha, \beta, \gamma, \delta, \mu)$



VQE for H2 molecule on QPU

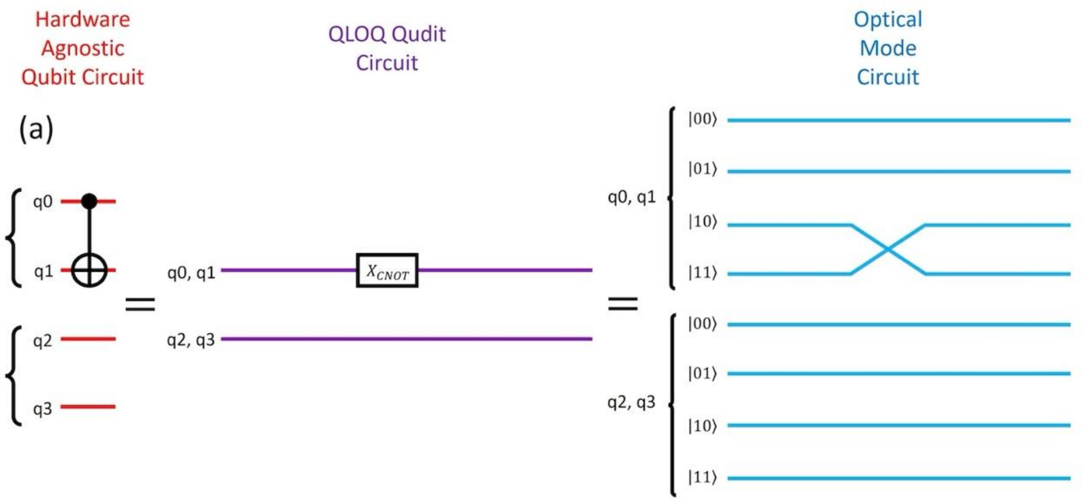


Qubit-based algorithms examples

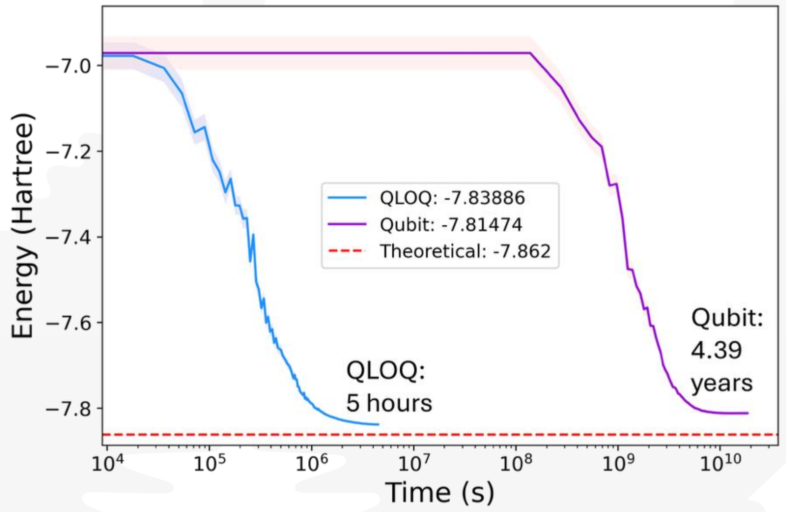
QLOQ encoding

Quantum circuit compression using qubit logic on qudits

Liam Lysaght, Timothée Goubault, Patrick Sinnott, Shane Mansfield, and Pierre-Emmanuel Emeriau



Application of QLOQ to VQE on LiH molecule



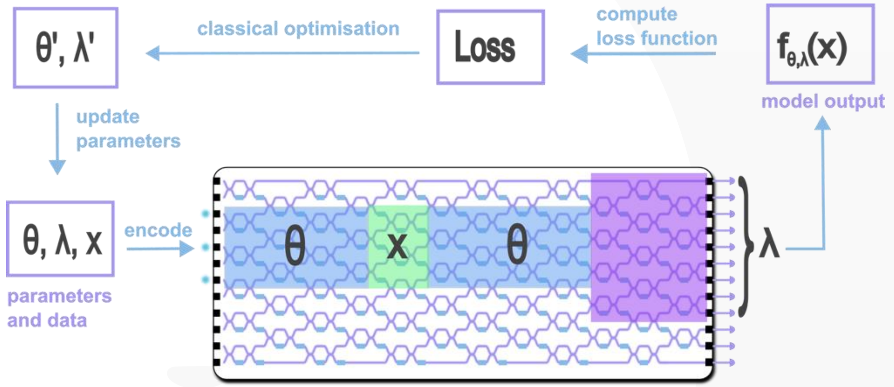
Photon-native algorithms examples

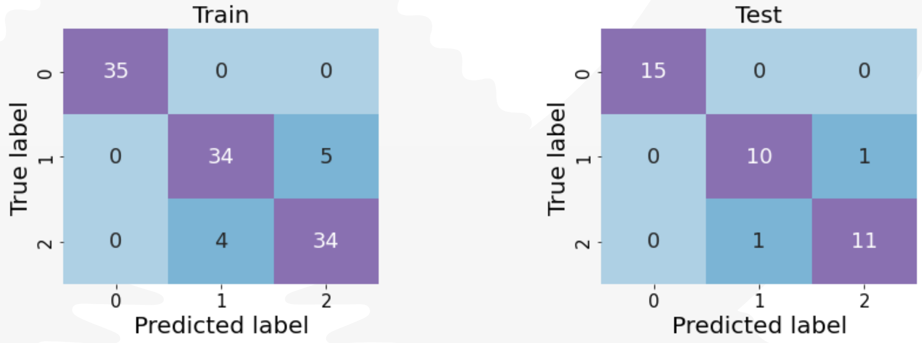
Variational quantum classifier

A versatile single-photon-based quantum computing platform

Nicolas Maring, Andreas Fyrillas, Mathias Pont, Edouard Ivanov, Petr Stepanov, Nico Margaria, William Hease, Anton Pishchagin, Aristide Lemaître, Isabelle Sagnes, Thi Huong Au, Sébastien Boissier, Eric Bertasi, Aurélien Baert, Mario Valdivia, Marie Billard, Ozan Acar, Alexandre Brieussel, Rawad Mezher, Stephen C. Wein, Alexia Salavrakos, Patrick Sinnott, Dario A. Fioretto, Pierre-Emmanuel Emeriau, Nadia Belabas, Shane Mansfield, Pascale Senellart, Jean Senellart & Niccolo Somaschi

Classifier: ansatz and optimisation





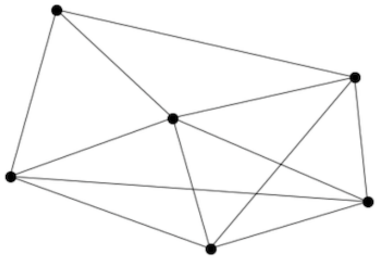
Photon-native algorithms examples

Solving graph problems with linear optics

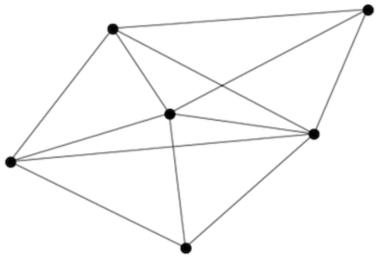
Solving graph problems with single photons and linear optics

Rawad Mezher, Ana Filipa Carvalho, and Shane Mansfield
Phys. Rev. A **108**, 032405 – Published 6 September 2023

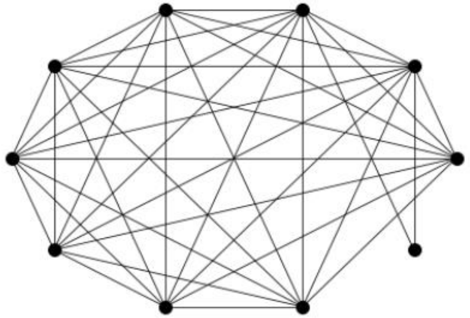
Graph isomorphism: compare permanents of adjacency matrices



(a)



(b)



$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The adjacency matrix of the graph is encoded into the linear optical circuit

Photon-native algorithms examples

Generative learning models

Photonic quantum generative adversarial networks for classical data

Tigran Sedrakyan[1,2] and Alexia Salavrakos[1]

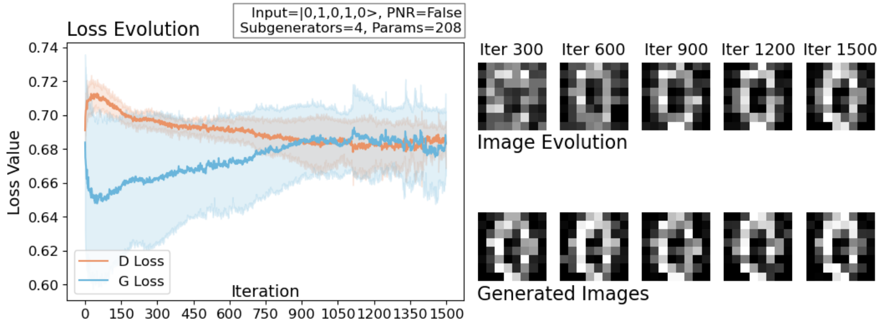
[1] *Quandela, 7 Rue Léonard de Vinci, 91300 Massy, France*
[2] *Sorbonne Université CNRS, LIP6, F-75005 Paris, France*

An error-mitigated photonic quantum circuit Born machine

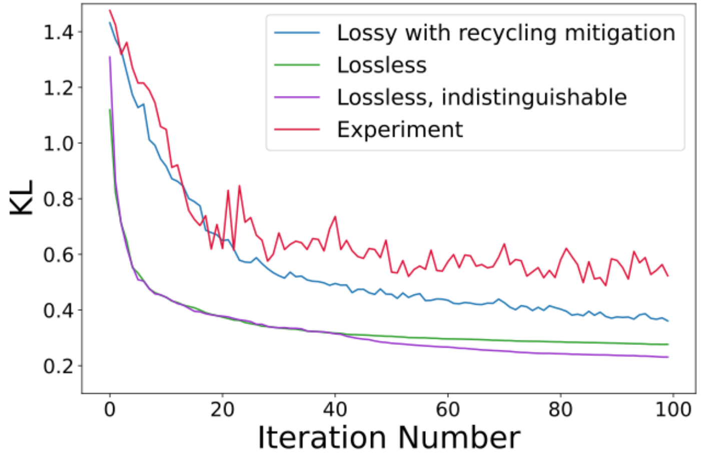
Alexia Salavrakos,[1,*] Tigran Sedrakyan,[1] James Mills,[1,2] Shane Mansfield,[1] and Rawad Mezher[1]

[1] *Quandela, 7 Rue Léonard de Vinci, 91300 Massy, France*
[2] *School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, Scotland*

Generating digits through simulations and on QPU



Error-mitigated experimental training of QCBM



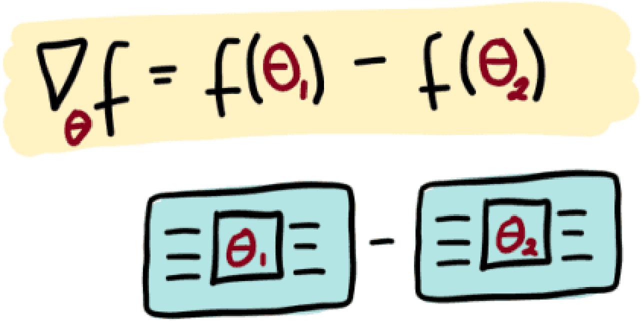
Photon-native algorithms examples

Photonic parameter shift rule

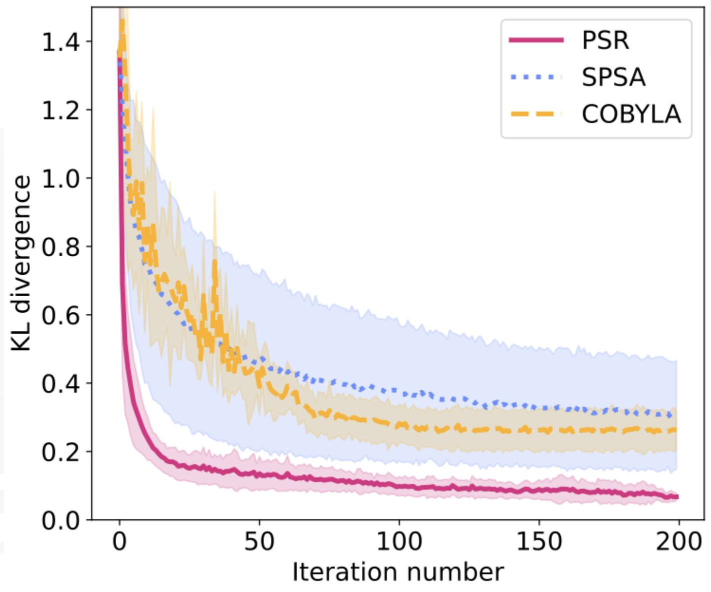
A Photonic Parameter-shift Rule: Enabling Gradient Computation for Photonic Quantum Computers

Axel Pappalardo[1], Pierre-Emmanuel Emeriau[1], Giovanni de Felice[2], Brian Ventura[1], Hugo Jaunin, Richie Yeung[2], Bob Coecke[2], and Shane Mansfield[1]

https://pennylane.ai/qml/glossary/parameter_shift



Application of PPSR to QCBM model



Photon-native algorithms examples

Reinforcement learning

Demonstration of quantum projective simulation on a single-photon-based quantum computer

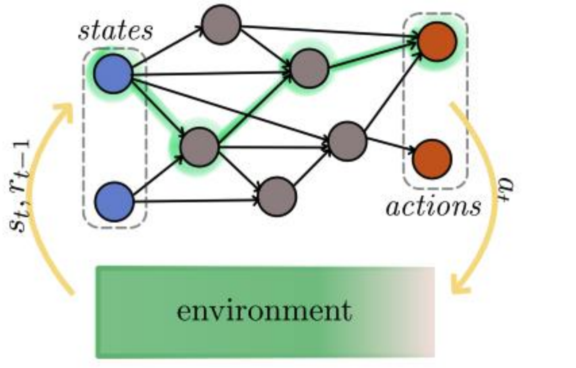
Giacomo Franceschetto[1,2,*] and Arno Ricou[1]

[1] *Quandela, 7 Rue Léonard de Vinci, 91300 Massy, France*

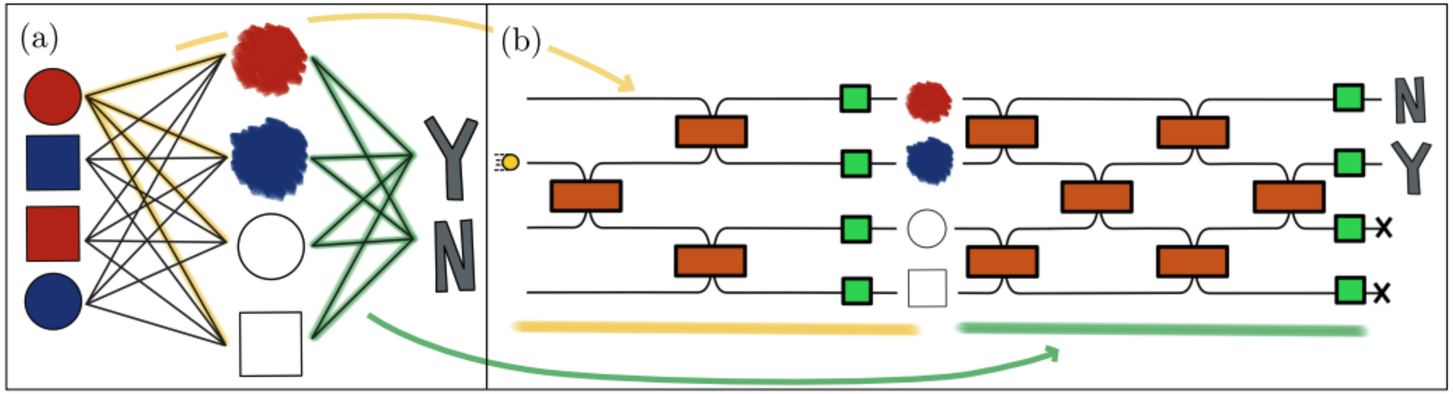
[2] *ICFO-Institut de Ciencies Fotoniques, The Barcelona Institute of Science and Technology, Av. Carl Friedrich Gauss 3, 08860 Castelldefels (Barcelona), Spain*

(Dated: April 22, 2024)

The agent is modelled by a graph



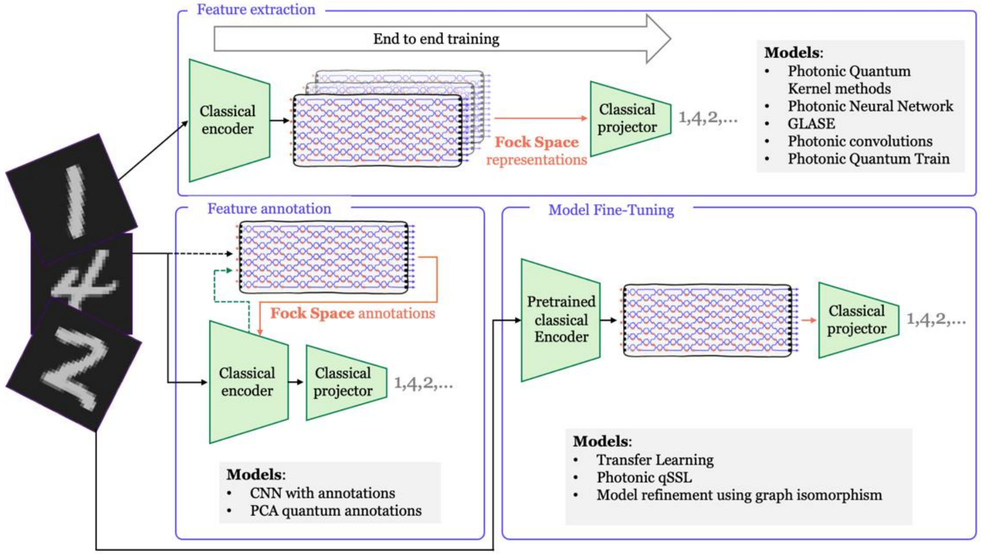
The agent is implemented as a linear optical circuit



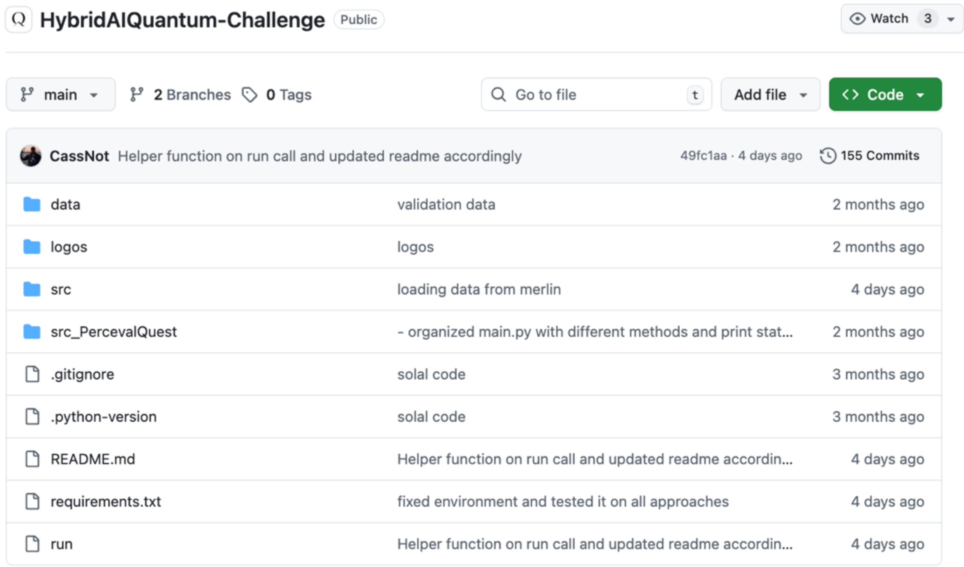
Perceval challenge

Establishing Baselines for Photonic Quantum Machine Learning: Insights from an Open, Collaborative Initiative

Cassandre Notton*[1], Vassilis Apostolou[2], Agathe Senellart[3], Anthony Walsh[2], Daphne Wang[2], Yichen Xie[12], Songqinghao Yang[7], Ilyass Mejdoub[5,6], Oussama Zouhry[4], Kuan-Cheng Chen[9,10], Chen-Yu Liu[11], Ankit Sharma[13], Edara Yaswanth Balaji[14], Soham Prithviraj Pawar[15], Ludovic Le Frioux[8], Valentin Macheret[8], Antoine Radet[8], Valentin Deumier[24], Ashesh Kumar Gupta[20], Gabriele Intoccia[22], Dimitri Jordan Kenne[19], Chiara Marullo[18], Giovanni Massafra[18], Nicolas Reinaldet[21], Vincenzo Schiano Di Cola[23], Danylo Kolesnyk[16,17], Yelyzaveta Vodovozova[16], Rawad Mezher[2], Pierre-Emmanuel Emeriau[2], Alexia Salavrakos[2], and Jean Senellart[2]



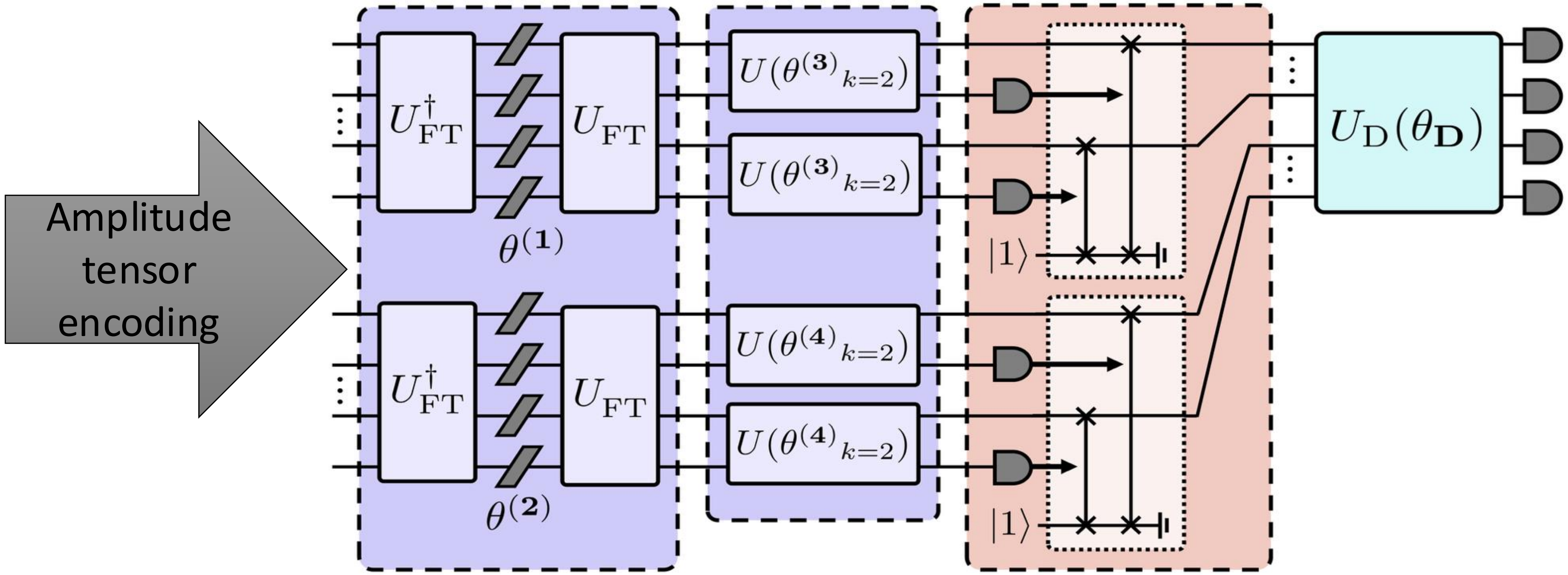
All code and methods available on Github



Upcoming work: QCNN

Quantum convolutional neural network (QCNN) – *D. Wang et al.*

Image input



Quantum Dense Layer

- Allows for interference of the photons
- Implemented using a universal interferometer

Quantum Circular Convolution

- Invariant under arbitrary translation
- Global operation

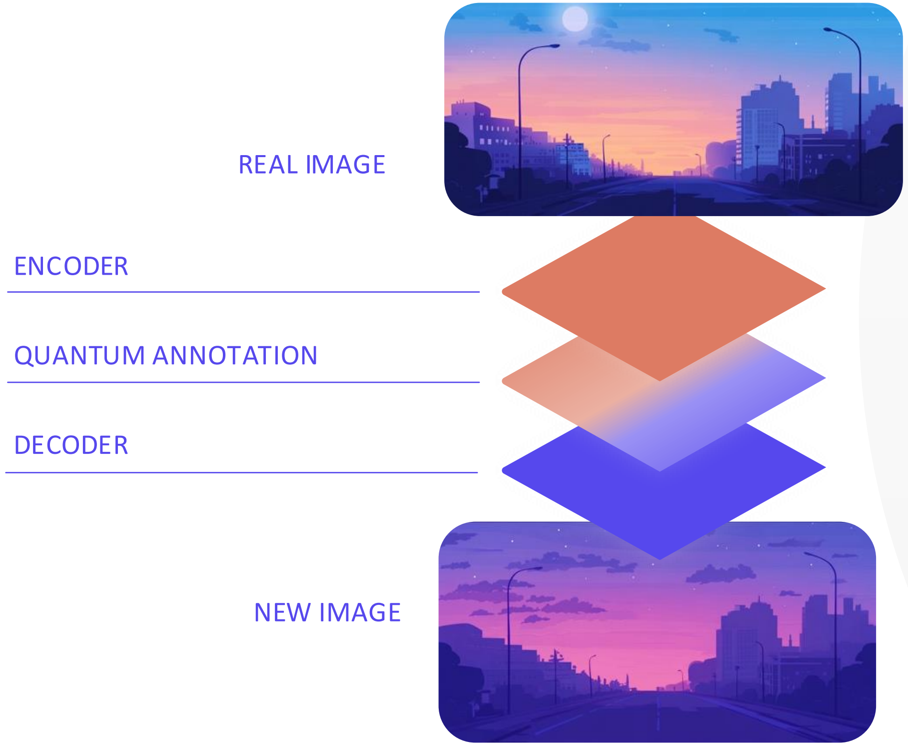
Quantum Convolution

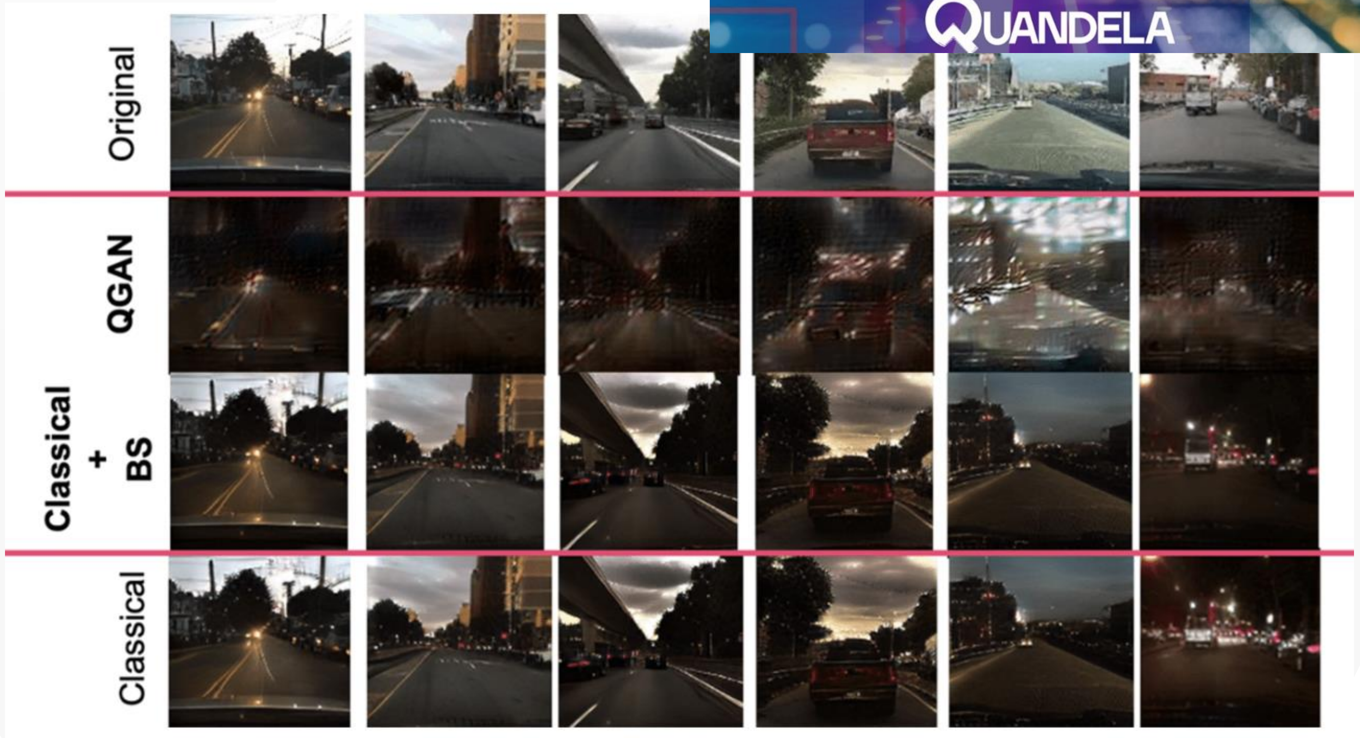
- Invariant under stride k translations
- Local operation

Quantum Pooling Layer

- Makes use of adaptive photon reinjection

Upcoming work: cycleGAN





01 Quantum transformation: no need for model training. It only requires source image annotation

02 Integration of Boson Sampling primitive improved translation and reduced hallucinations.

Other research groups also active in algorithms for photonic devices

Photonic Quantum Convolutional Neural Networks with Adaptive State Injection

Léo Monbroussou,[1,2] Beatrice Polacchi,[3] Verena Yacoub,[1] Eugenio Caruccio,[3] Giovanni Rodari,[3] Francesco Hoch,[3] Gonzalo Carvacho,[3] Nicolò Spagnolo,[3] Taira Giordani,[3] Mattia Bossi,[4,5] Abhiram Rajan,[4,5] Niki Di Giano,[4,5] Riccardo Albiero,[5] Francesco Ceccarelli,[5] Roberto Osellame,[5,*] Elham Kashefi,[1,6,†] and Fabio Sciarrino[3,‡]

Experimental quantum-enhanced kernels on a photonic processor

Zhenghao Yin,[1,2,*] Iris Agresti,[1,†] Giovanni de Felice,[3] Douglas Brown,[3] Alexis Toumi,[3] Ciro Pentangelo,[4,5] Simone Piacentini,[5] Andrea Crespi,[4,5] Francesco Ceccarelli,[5] Roberto Osellame,[5] Bob Coecke,[3] and Philip Walther[1,6,‡]

Experimental quantum speed-up in reinforcement learning agents

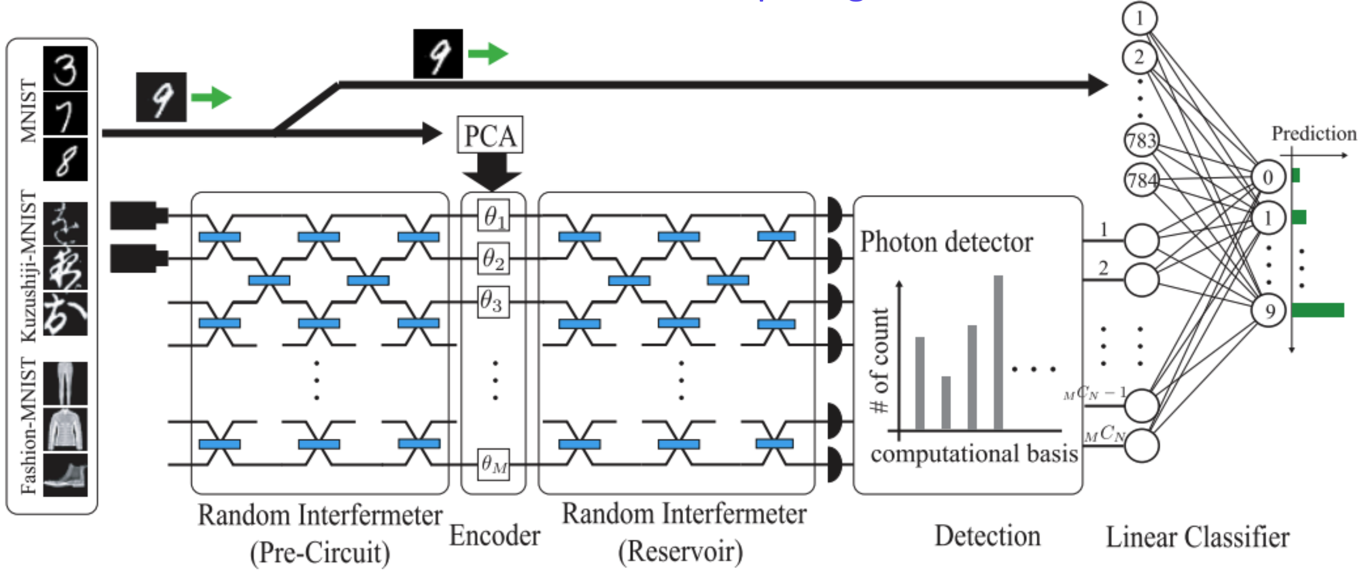
V. Saggio, B. E. Asenbeck, A. Hamann, T. Strömberg, P. Schiansky, V. Dunjko, N. Friis, N. C. Harris, M. Hochberg, D. Englund, S. Wölk, H. J. Briegel & P. Walther

Other research groups also active in algorithms for photonic devices

Quantum optical reservoir computing powered by boson sampling

Akitada Sakurai,[1,4] Aoi Hayashi,[1,2,3,5] William John Munro,[1,*] and Kae Nemoto[1,6]

Quantum reservoir computing



Results reproduced on MerLin platform

What next?
Our roadmap

A fault tolerant quantum computer

Threshold theorem:

A quantum computer with a physical error rate below a certain threshold can, through application of **quantum error correction schemes**, suppress the logical error rate to arbitrarily low levels.

In other words, quantum computers can be made fault-tolerant.

FAULT-TOLERANT QUANTUM COMPUTATION WITH CONSTANT ERROR RATE *

DORIT AHARONOV[†] AND MICHAEL BEN-OR[‡]

Resilient Quantum Computation: Error Models and Thresholds

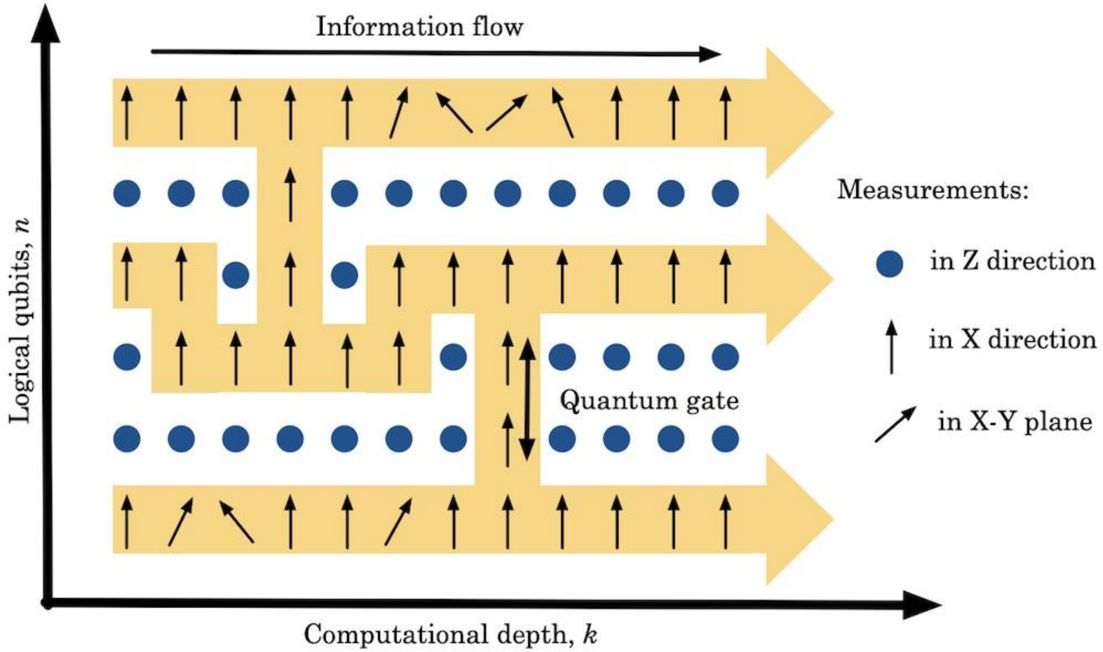
Emanuel Knill[1*], Raymond Laflamme[2†], Wojciech H. Zurek[2‡]

Fault-tolerant quantum computation by anyons

A. Yu. Kitaev

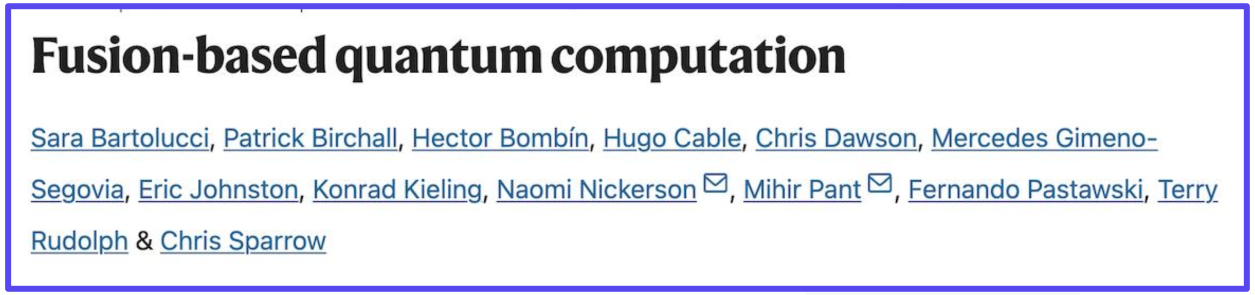
Photonic platforms – scaling proposals

Measurement based quantum computing (MBQC)



Proposed by R. Raussendorf and H. J. Briegel in *A One-Way Quantum Computer* (2001)

Fusion based quantum computing



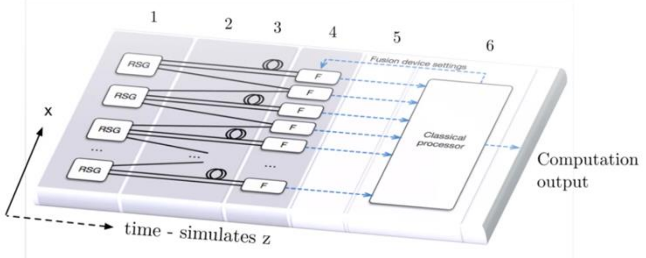
Fusion-based quantum computation

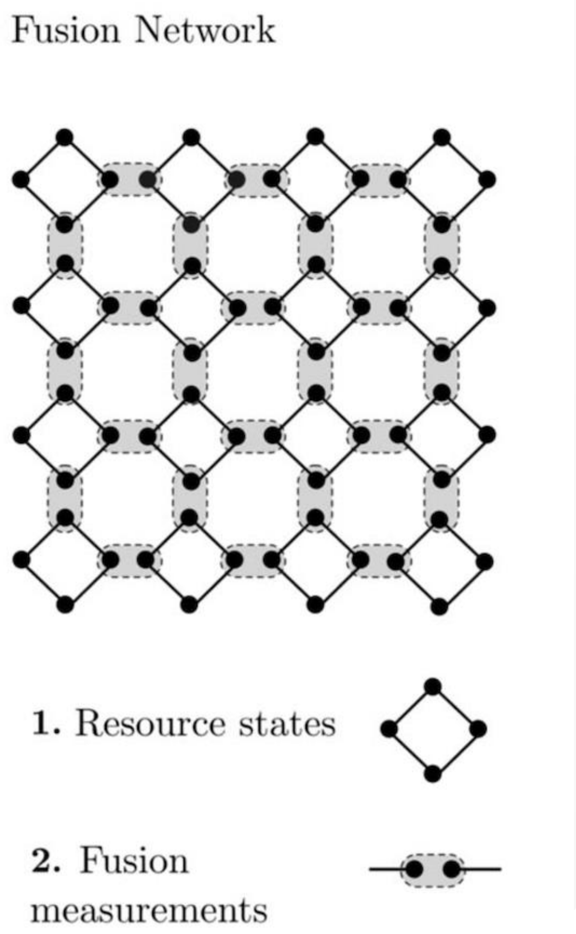
Sara Bartolucci, Patrick Birchall, Hector Bombín, Hugo Cable, Chris Dawson, Mercedes Gimeno-Segovia, Eric Johnston, Konrad Kieling, Naomi Nickerson, Mihir Pant, Fernando Pastawski, Terry Rudolph & Chris Sparrow

Proposal by PsiQuantum team

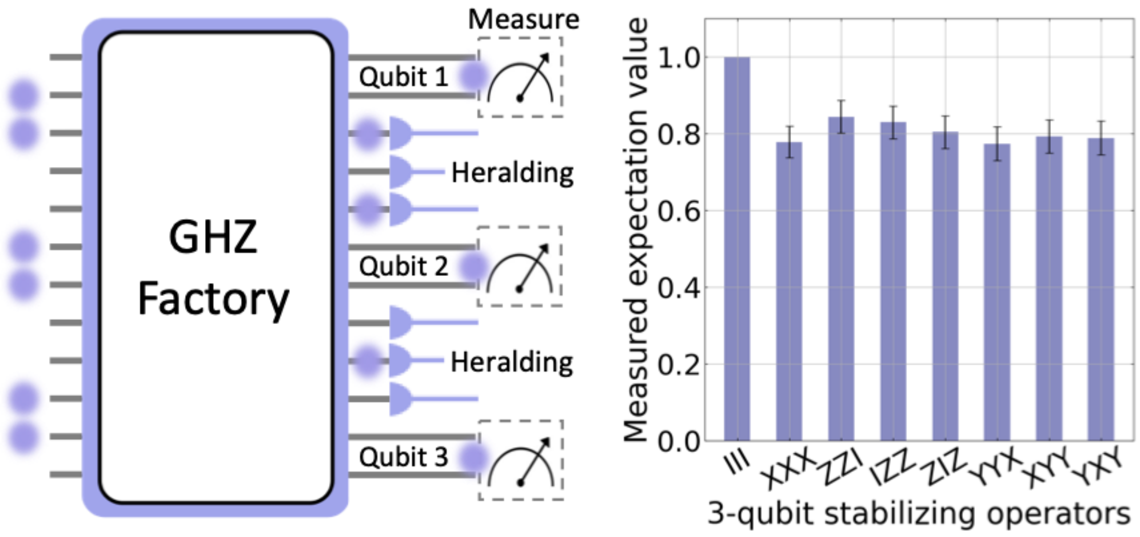
Compared to MBQC it also integrates fault tolerance

Proposal comes with photonic hardware as well





Path for graph state generation: GHZ states



Heralded generation of 3-photon GHZ states. Measured expectation values of the stabilizing operators of the heralded 3-photon GHZ state $|\mathrm{GHZ}_3^+\rangle$ yielding a fidelity of $F_{\mathrm{GHZ}_3^+} = 0.82 \pm 0.04$.

Generate small GHZ states
Add fusion operations

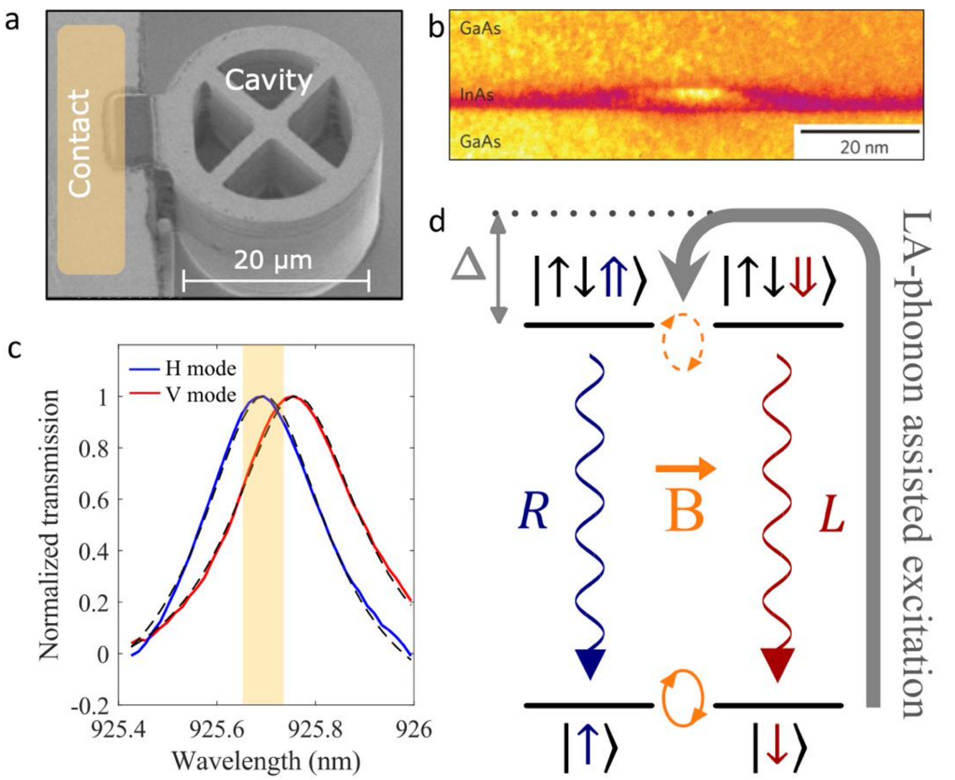
Path for graph state generation: from the source

High-rate entanglement between a semiconductor spin and indistinguishable photons

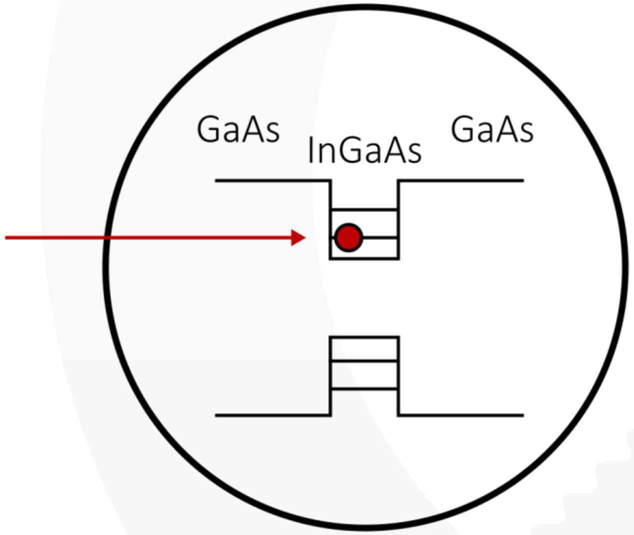
N. Coste, D. A. Fioretto, N. Belabas, S. C. Wein, P. Hilaire, R. Frantzeskakis, M. Gundin, B. Goes, N. Somaschi, M. Morassi, A. Lemaître, I. Sagnes, A. Harouri, S. E. Economou, A. Auffeves, O. Krebs, L. Lanco & P. Senellart

Proposal for Pulsed On-Demand Sources of Photonic Cluster State Strings

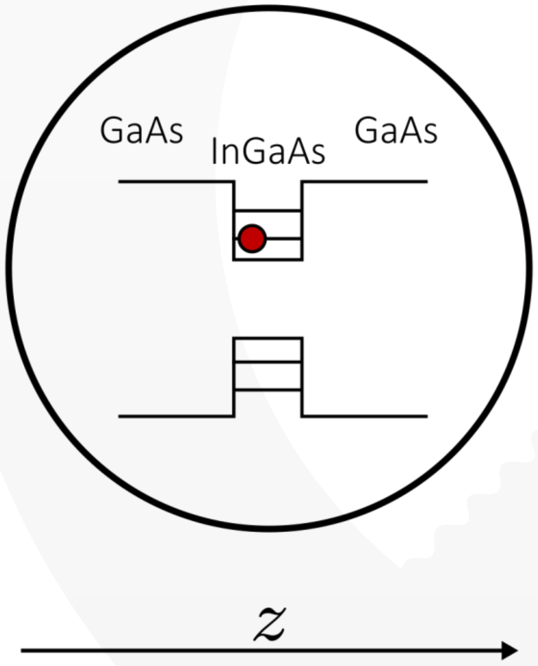
Netanel H. Lindner and Terry Rudolph
Phys. Rev. Lett. **103**, 113602 – Published 8 September 2009



Path for graph state generation: from the source

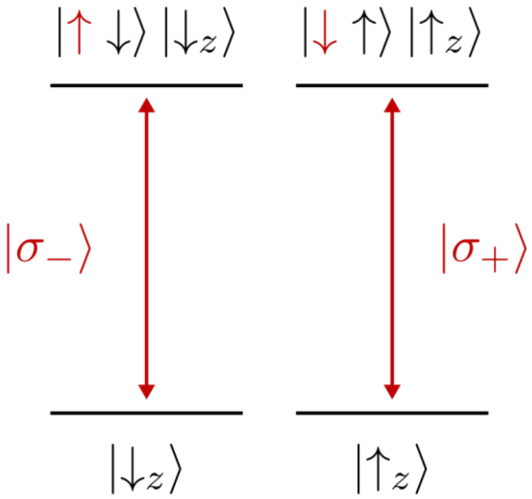


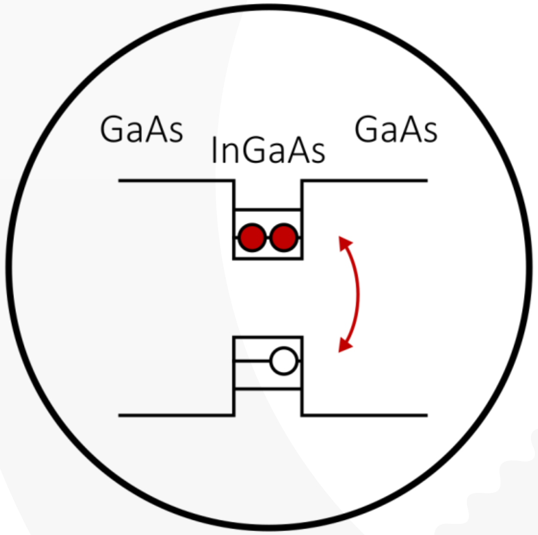
Path for graph state generation: from the source



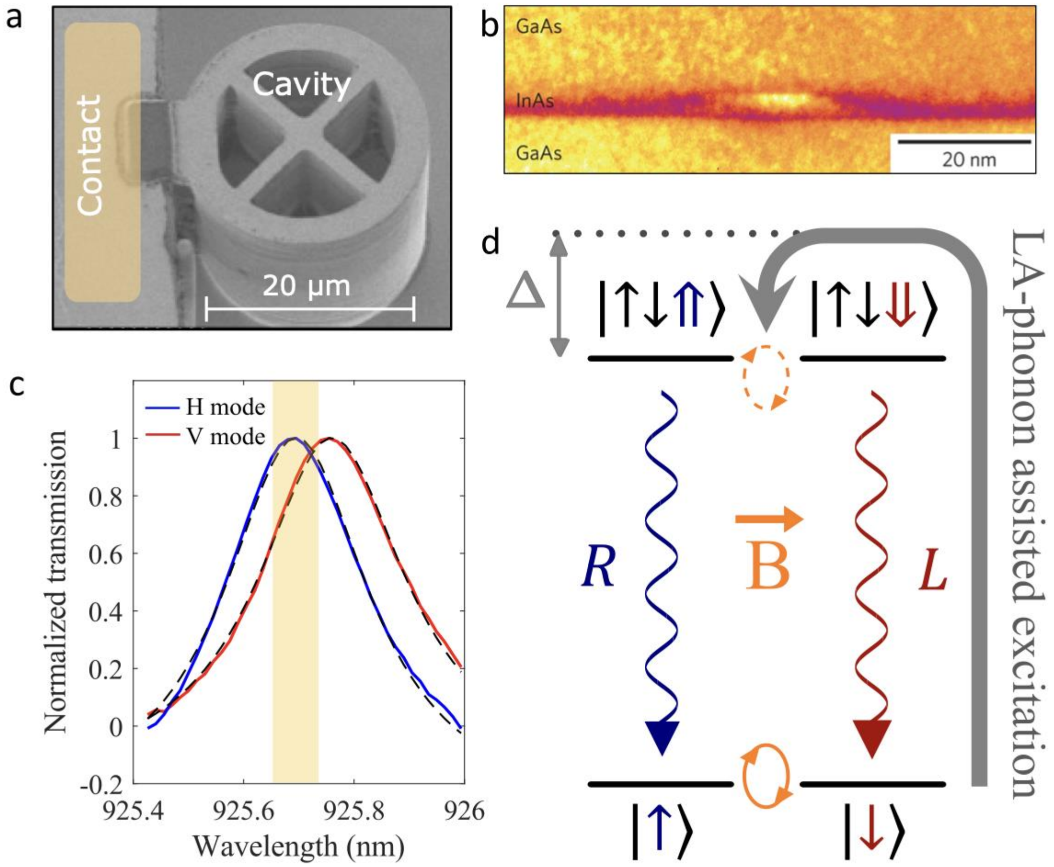
spin basis: $|\downarrow_z\rangle, |\uparrow_z\rangle$

Path for graph state generation: from the source





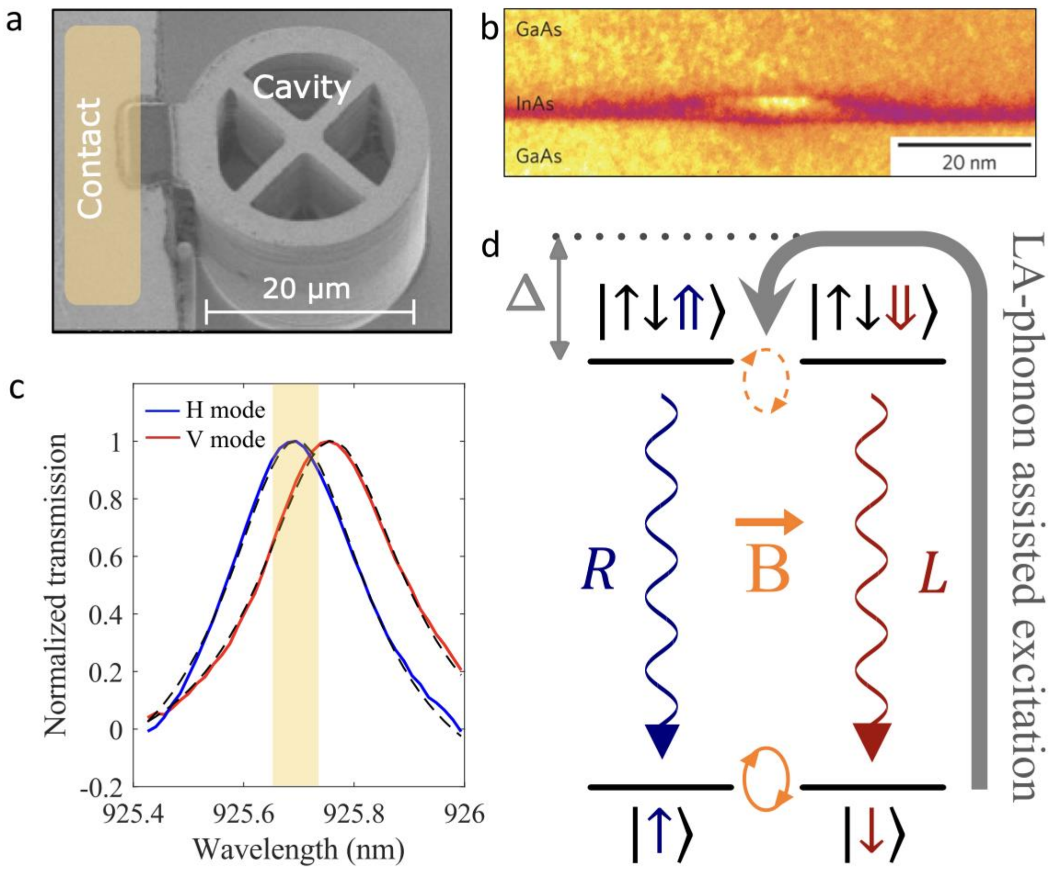
Path for graph state generation: from the source



$|\uparrow\rangle|R\rangle + |\downarrow\rangle|L\rangle$

Spin-photon entanglement

Path for graph state generation: from the source

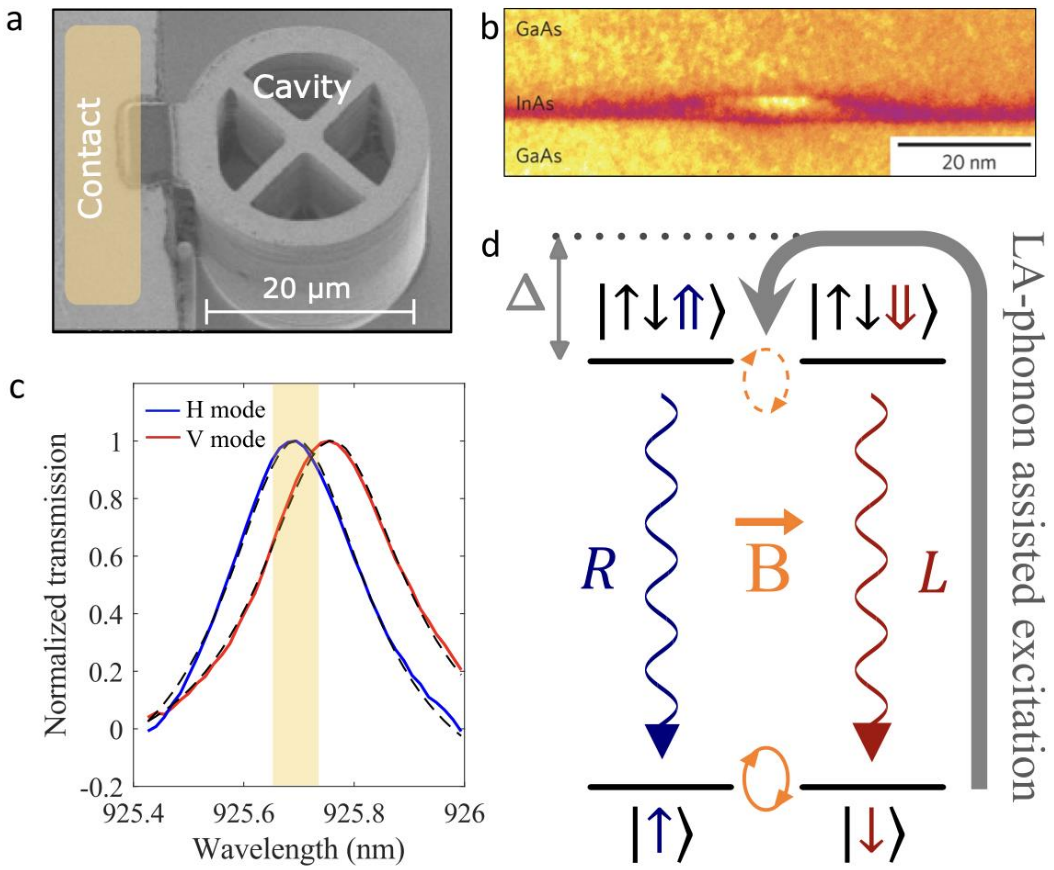


$$|\uparrow\rangle|R\rangle + |\downarrow\rangle|L\rangle$$

$\pi/2$ spin-gate

$$\frac{|\uparrow\rangle + |\downarrow\rangle}{\sqrt{2}}|R\rangle + \frac{|\uparrow\rangle - |\downarrow\rangle}{\sqrt{2}}|L\rangle$$

Path for graph state generation: from the source

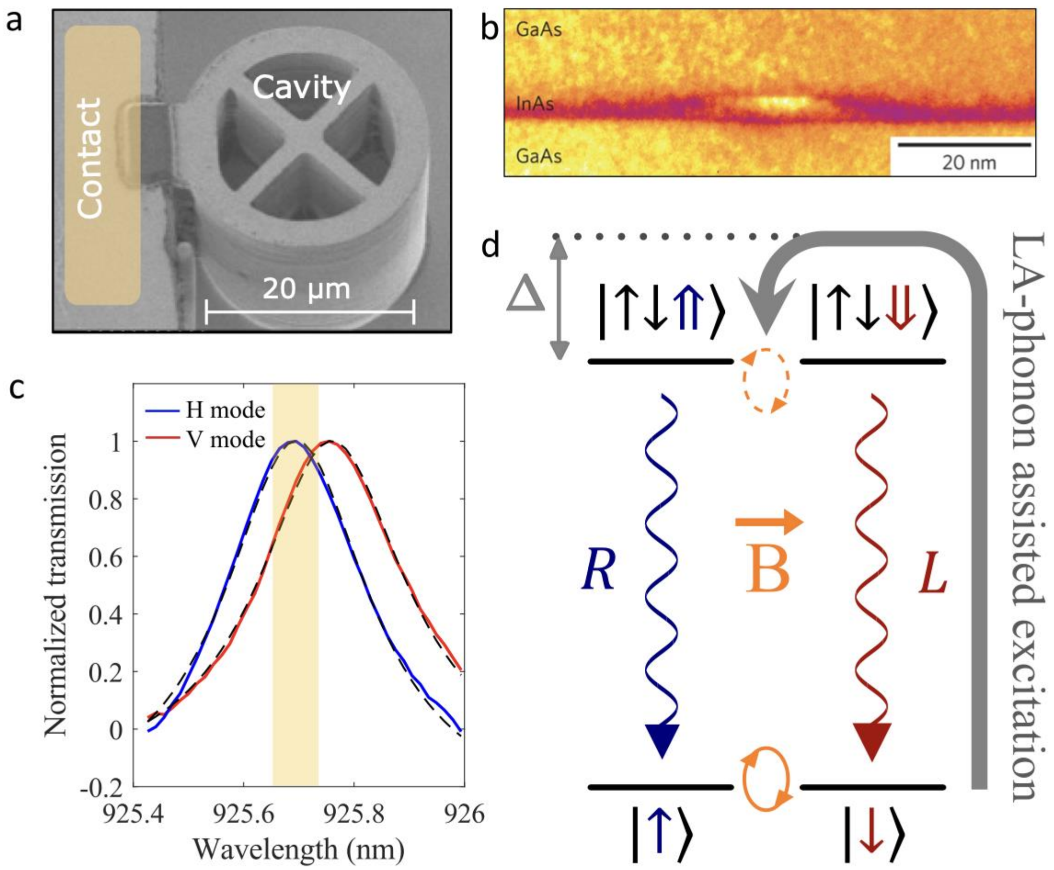


$$|\uparrow\rangle|R\rangle + |\downarrow\rangle|L\rangle$$

$^{\pi}/_{2}$ spin-gate

$$-i|\uparrow\rangle|V\rangle + |\downarrow\rangle|H\rangle$$

Path for graph state generation: from the source



$$|\uparrow\rangle|R\rangle + |\downarrow\rangle|L\rangle$$

$^{\pi}/_{2}$ spin-gate

$$-i|\uparrow\rangle|V\rangle + |\downarrow\rangle|H\rangle$$

Excitation H

$$-i|\uparrow\rangle|V\rangle|R\rangle + |\downarrow\rangle|H\rangle|L\rangle$$

…

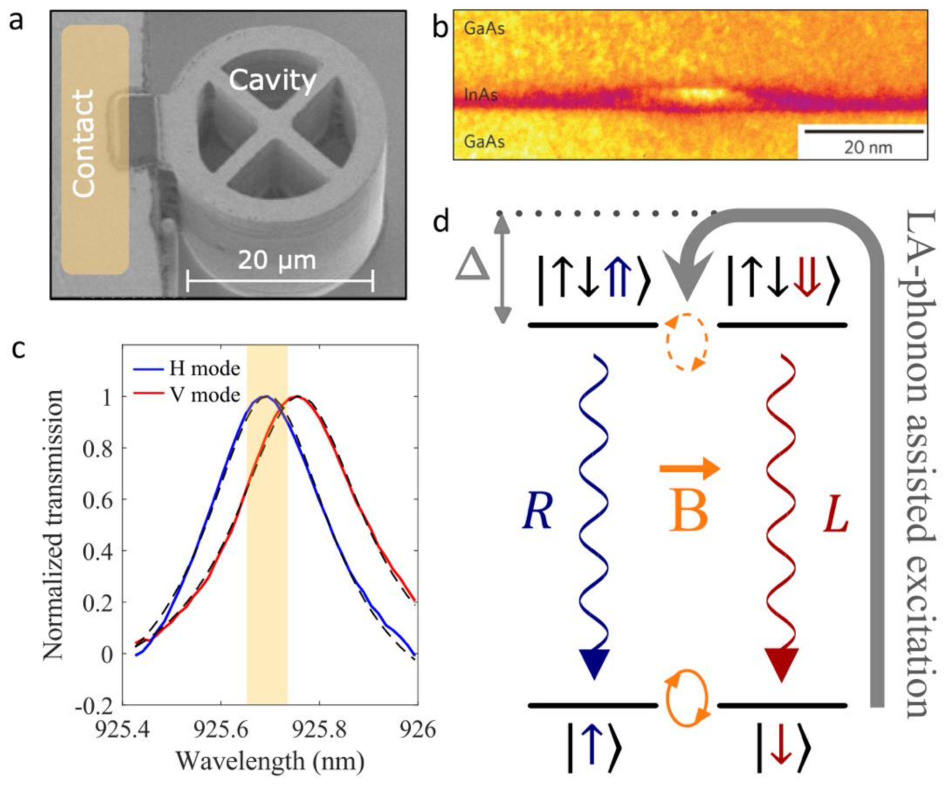
Path for graph state generation: from the source

High-rate entanglement between a semiconductor spin and indistinguishable photons

N. Coste, D. A. Fioretto, N. Belabas, S. C. Wein, P. Hilaire, R. Frantzeskakis, M. Gundin, B. Goes, N. Somaschi, M. Morassi, A. Lemaître, I. Sagnes, A. Harouri, S. E. Economou, A. Auffeves, O. Krebs, L. Lanco & P. Senellart

Proposal for Pulsed On-Demand Sources of Photonic Cluster State Strings

Netanel H. Lindner and Terry Rudolph
Phys. Rev. Lett. **103**, 113602 – Published 8 September 2009



Proposal for fault tolerant architecture: SPOQC

A Spin-Optical Quantum Computing Architecture

Grégoire de Gliniasty[1,2], Paul Hilaire[1], Pierre-Emmanuel Emeriau[1], Stephen C. Wein[1], Alexia Salavrakos[1], and Shane Mansfield[1]

General intuition:

- Strategies like FBQC can be achieved with many quantum dots
- Why not leverage those dots as carriers of quantum information?

Proposal for fault tolerant architecture: SPOQC

A Spin-Optical Quantum Computing Architecture

Grégoire de Gliniasty[1,2], Paul Hilaire[1], Pierre-Emmanuel Emeriau[1], Stephen C. Wein[1], Alexia Salavrakos[1], and Shane Mansfield[1]

General intuition:

- Strategies like FBQC can be achieved with many quantum dots
- Why not leverage those dots as carriers of quantum information?

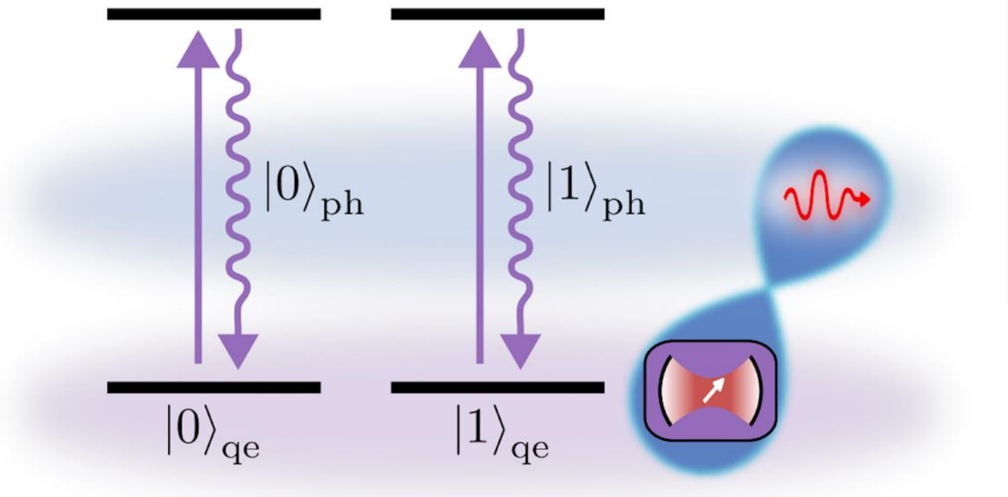
➢ Using spins of quantum dots as qubits
➢ Using spin-entangled photon to perform 2-qubit gates

Proposal for fault tolerant architecture: SPOQC

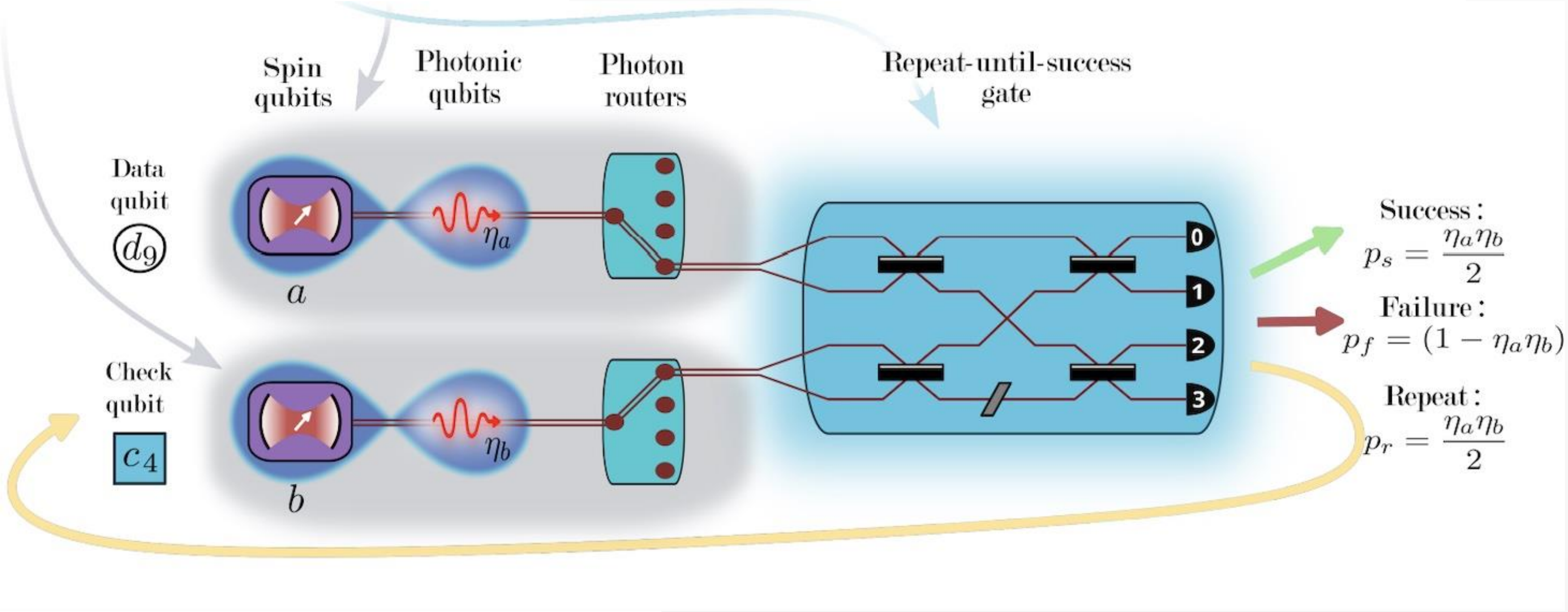
Emission process operator:

$$E_{\mathrm{qe,ph}} = |0\rangle_{\mathrm{qe}} |0\rangle_{\mathrm{ph}} \langle 0|_{\mathrm{qe}} + |1\rangle_{\mathrm{qe}} |1\rangle_{\mathrm{ph}} \langle 1|_{\mathrm{qe}}$$

$$E_{\mathrm{qe,ph}} |+\rangle_{\mathrm{qe}} = |0\rangle_{\mathrm{qe}} |0\rangle_{\mathrm{ph}} + |1\rangle_{\mathrm{qe}} |1\rangle_{\mathrm{ph}}$$

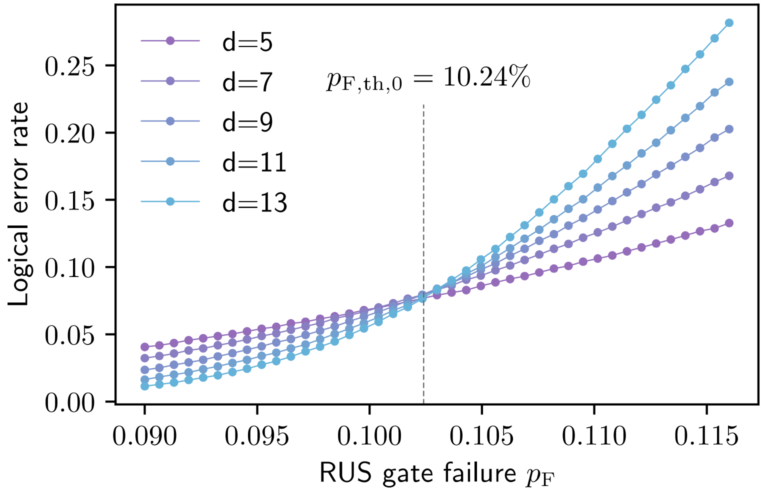


Proposal for fault tolerant architecture: SPOQC

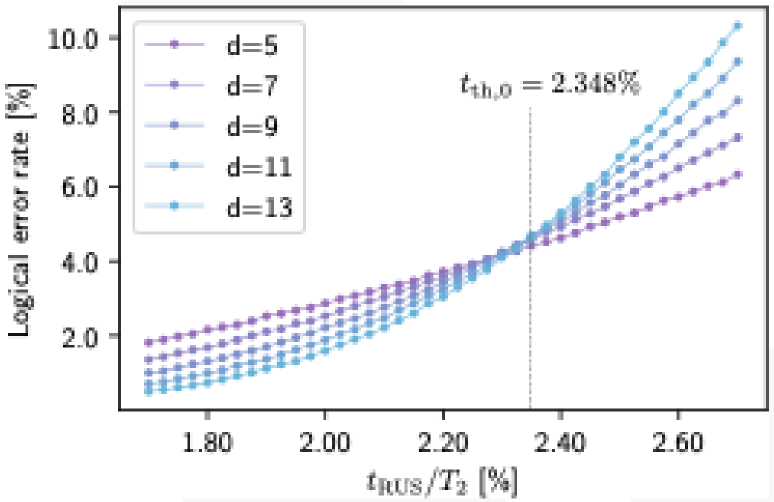


Proposal for fault tolerant architecture: SPOQC

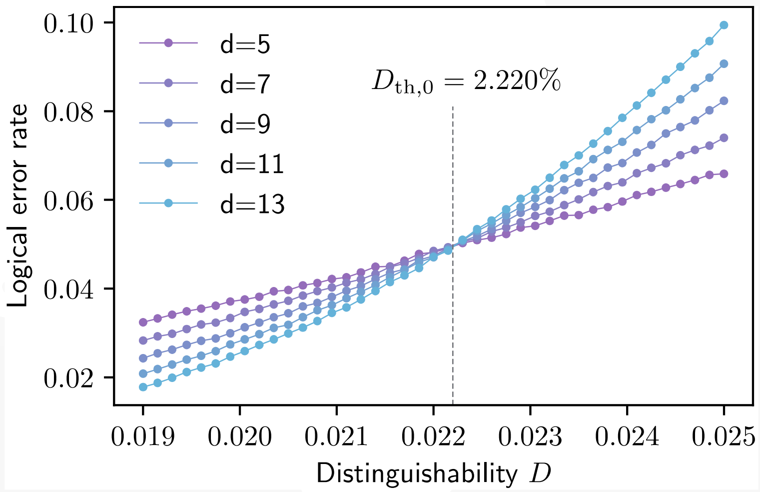
RUS gate failure



Spin coherence time

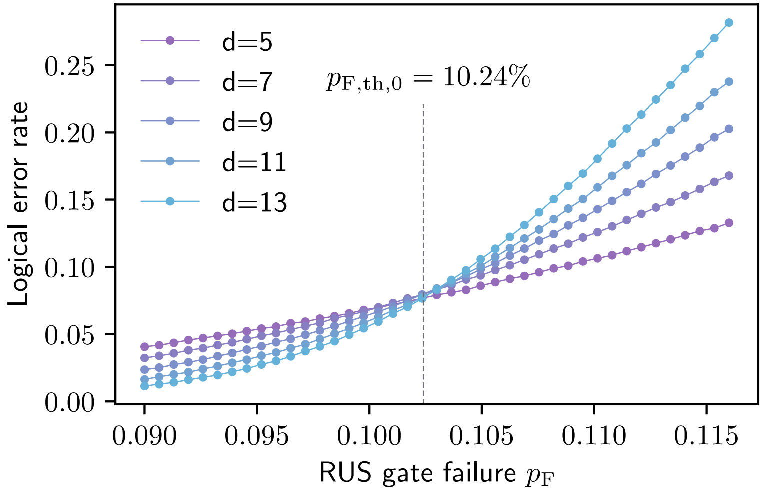


Distinguishability



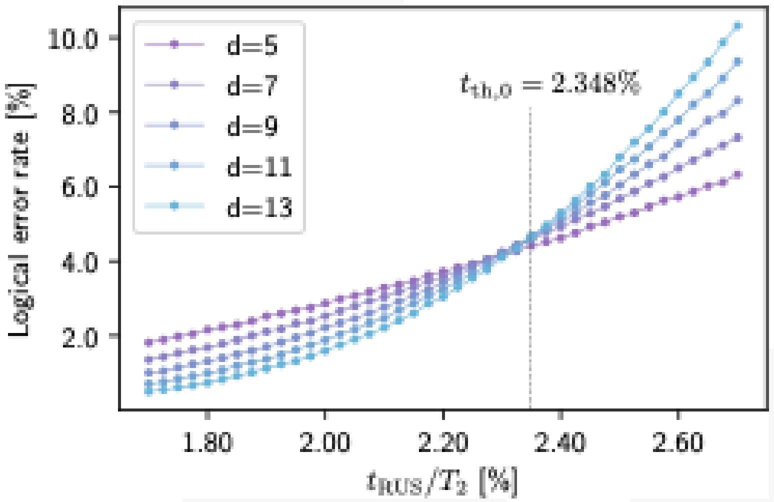
Proposal for fault tolerant architecture: SPOQC

RUS gate failure



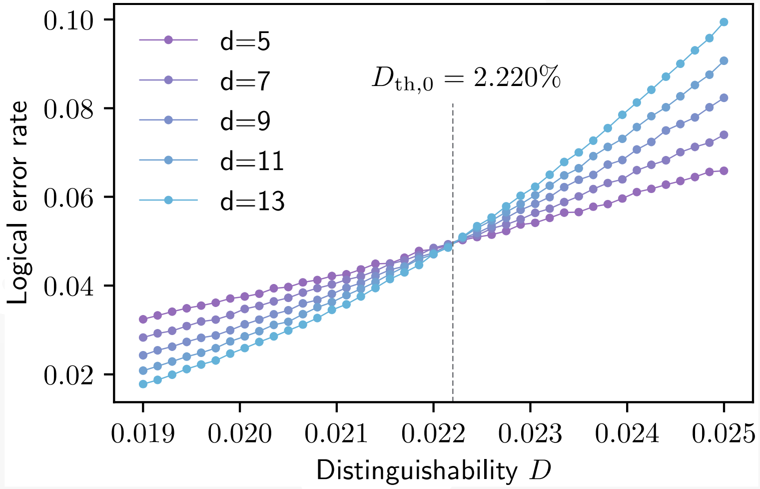
Depends on photon loss

Spin coherence time



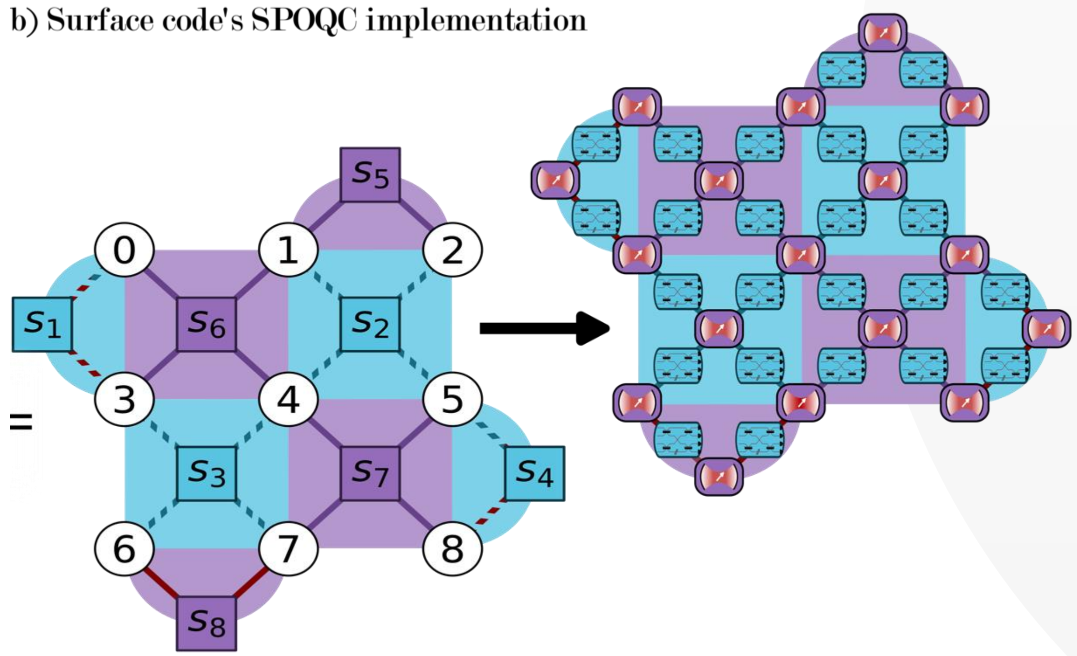
Express decoherence time T_2 relative to time required for RUS gate t_{RUS}

Distinguishability

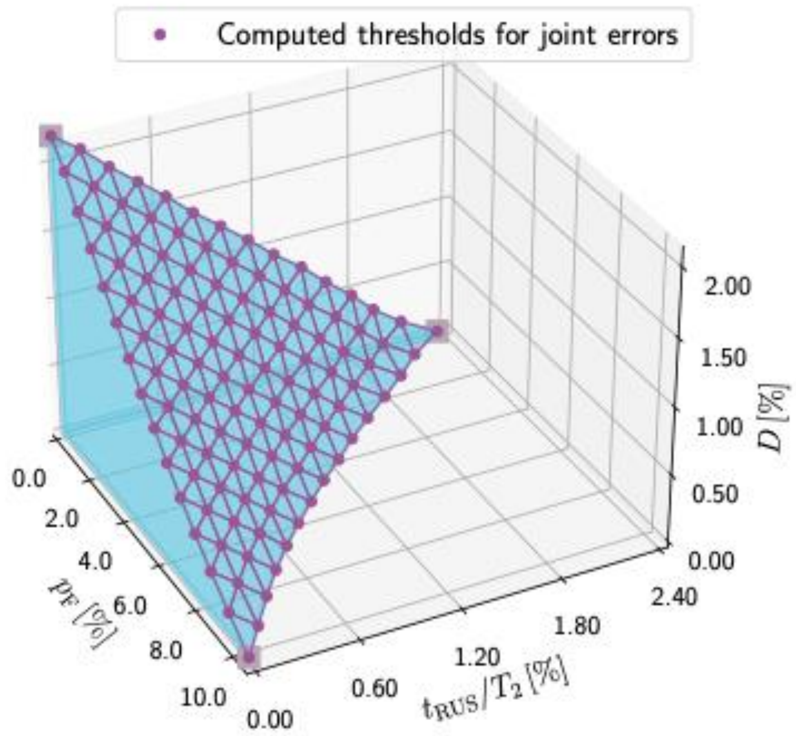


Proposal for fault tolerant architecture: SPOQC

Integrating quantum error correction with surface code implementation



Thresholds for distinguishability, spin coherence time and RUS gate failure



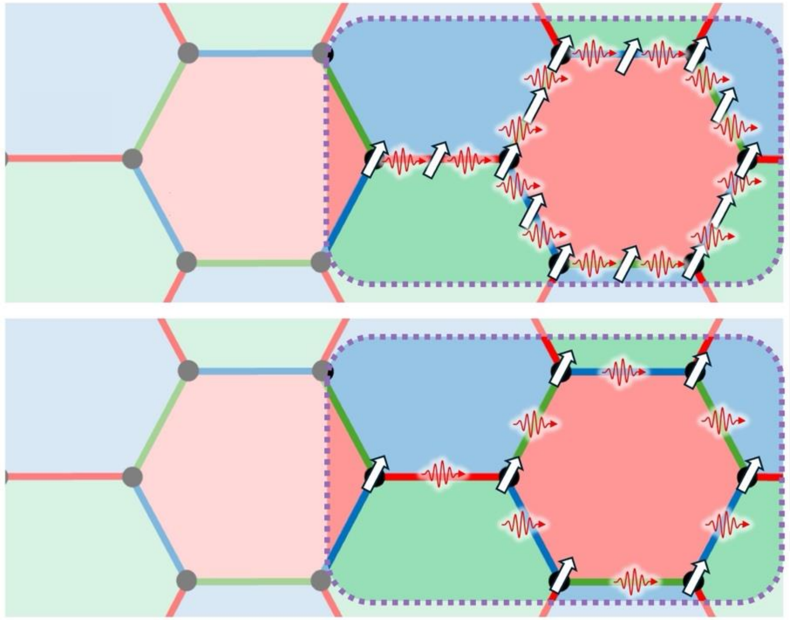
Proposal for fault tolerant architecture: SPOQC

- Alternative to all-photonic and all-matter-based architectures
- Similar loss tolerance threshold to fusion-based photonic proposal
- Arbitrary physical connectivity, which implies compatibility with important classes of QEC codes
- Adaptable to other platforms (no need to stick to quantum-dot emitters)

Proposal for fault tolerant architecture: SPOQC

Enhanced Fault-tolerance in Photonic Quantum Computing: Floquet Code Outperforms Surface Code in Tailored Architecture

Paul Hilaire,[1,*] Théo Dessertaine,[1] Boris Bourdoncle,[1] Aurélie Denys,[1]
Grégoire de Gliniasty,[1,2] Gerard Valentí-Rojas,[3,4] and Shane Mansfield[1]



Conclusions

- Near term:
  - Linear optics
  - Boson sampling
  - Variational algorithms and error mitigation techniques
- Long term:
  - SPOQC proposal
- Mid term:
  - Beyond passive linear optics: entangled input states, feedforward, …
  - Early fault-tolerant models with SPOQC building blocks

Research directions

- More error mitigation techniques tailored to photonics
- Improve compilation and optimization of algorithms
- Utility and search for applications and use cases
- QML exploration and heuristics with MerLin
- Scaling and experimental teams working towards SPOQC

QUIZZ!

QUANDELA

Hands-on programming with Perceval and MerLin

Install Perceval and MerLin

Install Perceval: pip install perceval-quandela
Install MerLin: pip install merlinquantum

Try:
import perceval as pcvl
import merlin as ML

If you wish to run the notebooks with jupyter – if not, feel free to use any IDE:
pip install notebook
jupyter notebook

Find notebooks here:
https://github.com/alexiasalavrakos/merlin_tutorials

Perceval software

perceval

Search docs

DOCUMENTATION

- Basics
- Usage
- Circuits
- States
- Polarization
- Computing Backends
- Components

/ Welcome to the Perceval documentation!

Edit on GitHub

Welcome to the Perceval documentation!

Une si granz clartez i vint
Qu'ausi perdirent les chandoiles
Lor clarté come les estoiles
Quant li solauz lieve ou la lune.

Perceval, the Story of the Grail – Chrétien de Troyes (circa 1180)

Through a simple object-oriented Python API, Perceval provides tools for composing photonic circuits from linear optical components like beamsplitters and phase shifters, defining single-photon sources, manipulating Fock states, and running simulations.

Perceval can be used to reproduce published experimental works or to experiment directly with a new generation of quantum algorithms.

It aims to be a companion tool for developing photonic circuits – for simulating and optimis the ideal and realistic behaviours, and proposing a normalised interface to control them thr

Perceval is conceived as an object-oriented modular Python framework organised around

- Tools to build linear optical circuits from a collection of pre-defined components
- Powerful computing backends implemented in C++
- **A variety of technical utilities to manipulate:**

Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing

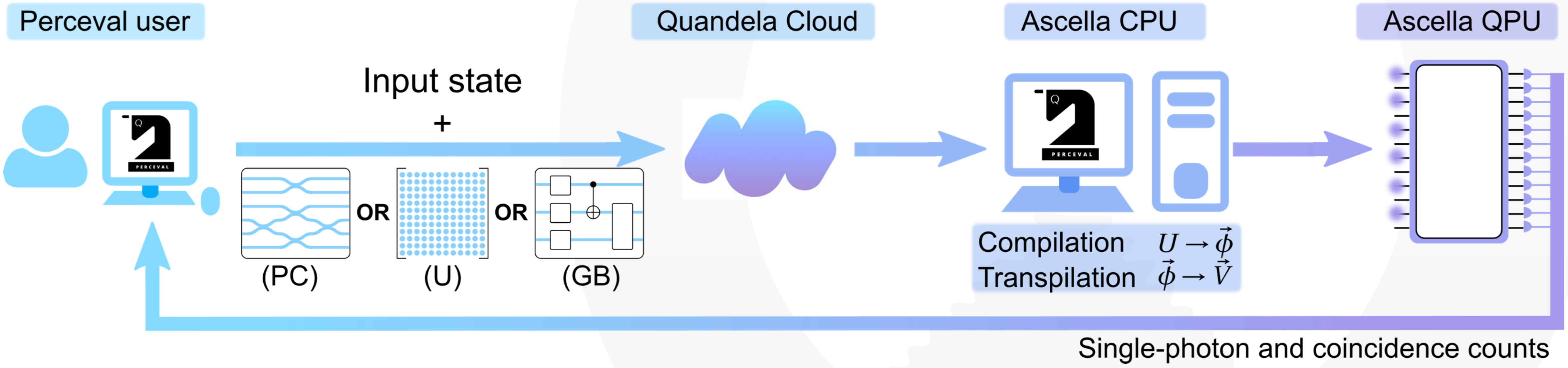
Nicolas Heurtel[1,2], Andreas Fyrillas[1,3], Grégoire de Gliniasty[1], Raphaël Le Bihan[1], Sébastien Malherbe[4], Marceau Pailhas[1], Eric Bertasi[1], Boris Bourdoncle[1], Pierre-Emmanuel Emeriau[1], Rawad Mezher[1], Luka Music[1], Nadia Belabas[3], Benoît Valiron[2], Pascale Senellart[3], Shane Mansfield[1], and Jean Senellart[1]

[1]Quandela, 7 Rue Léonard de Vinci, 91300 Massy, France
[2]Université Paris-Saclay, Inria, CNRS, ENS Paris-Saclay, CentraleSupélec, LMF, 91190, 15 Gif-sur-Yvette, France
[3]Centre for Nanosciences and Nanotechnology, CNRS, Université Paris-Saclay, UMR 9001, 10 Boulevard Thomas Gobert, 91120, Palaiseau, France
[4]Département de Physique de l'Ecole Normale Supérieure - PSL, 45 rue d'Ulm, 75230, Paris Cedex 05, France

```
pip install perceval-quandela
import perceval as pcvl
```

+ contains qiskit converter!

Cloud computing



MerLin — our first step toward ML frameworks for hybrid AI+Quantum.

Photonic focus, open design

https://merlinquantum.ai

1. Start Anywhere – Simulator First

- Develop and test quantum-enhanced ML models without hardware dependency
- Run everything locally or in the cloud
- **Focus on cross-modality paper reproduction**

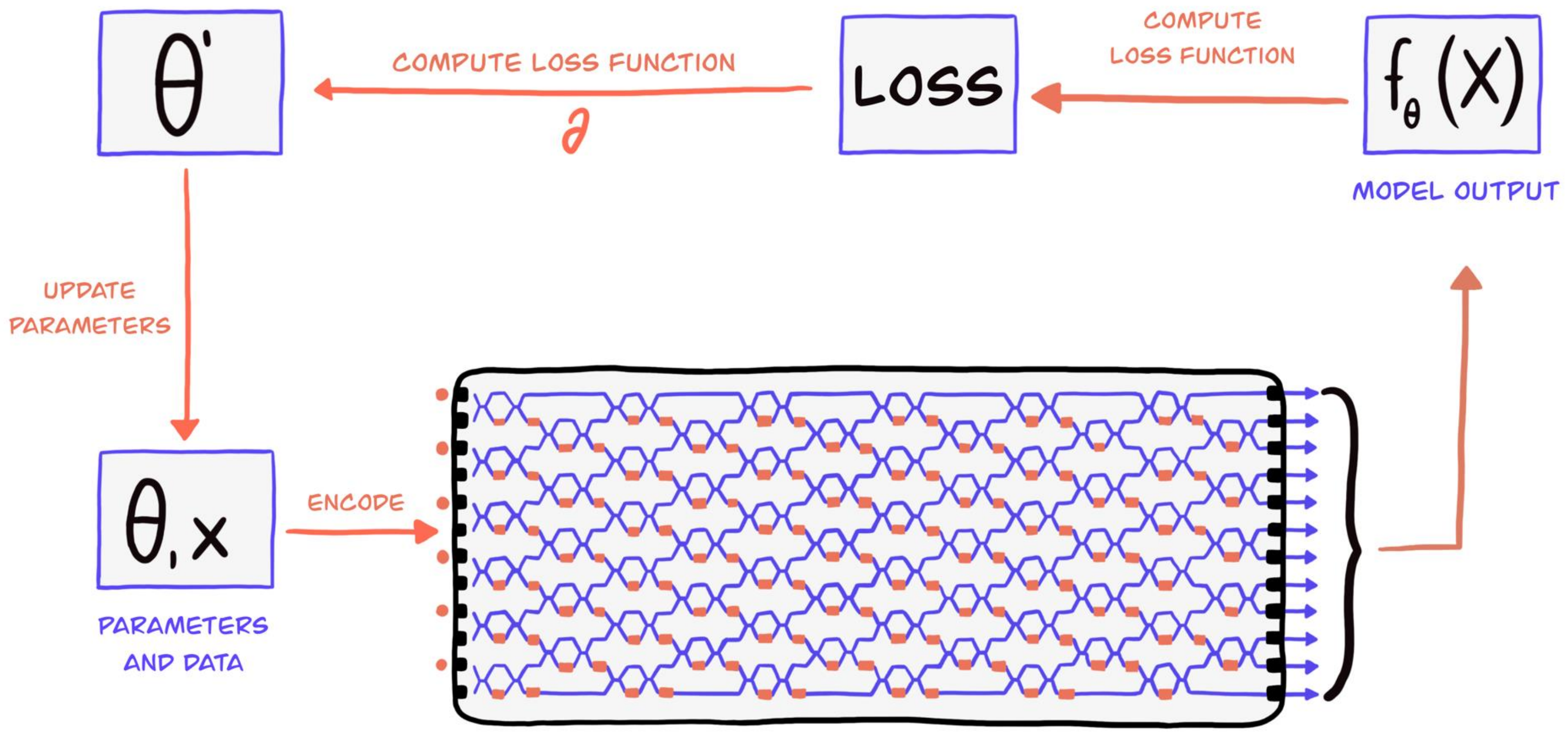
2. Train at Scale – GPU Acceleration on HPC

- Train **hybrid quantum–classical models** efficiently on GPUs
- Use familiar PyTorch APIs

3. Deploy on Hardware – QPU Ready

- Fine-tune and execute on Quandela's photonic QPUs
- Framework evolves with new features (feedforward, entangled sources, SPOQC)

Differentiable Quantum Layer
… in simulation

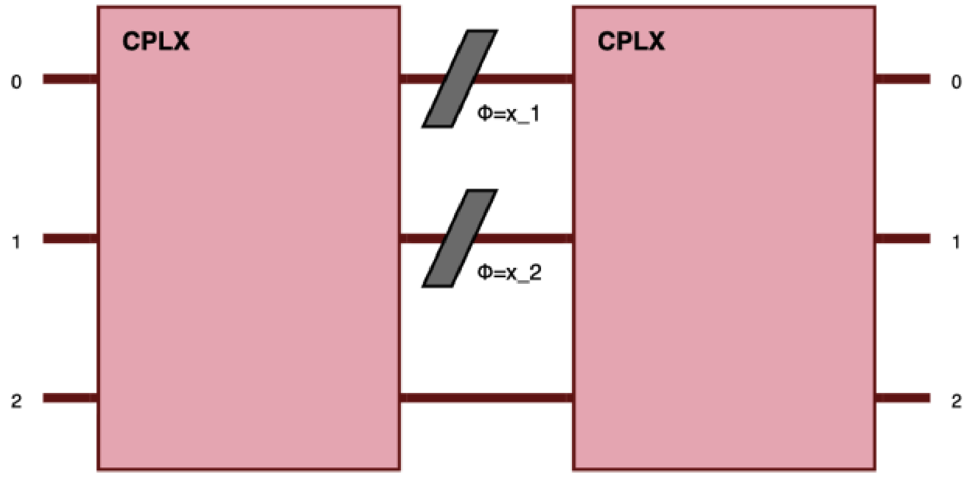


PyTorch

Quantum Layer essentials

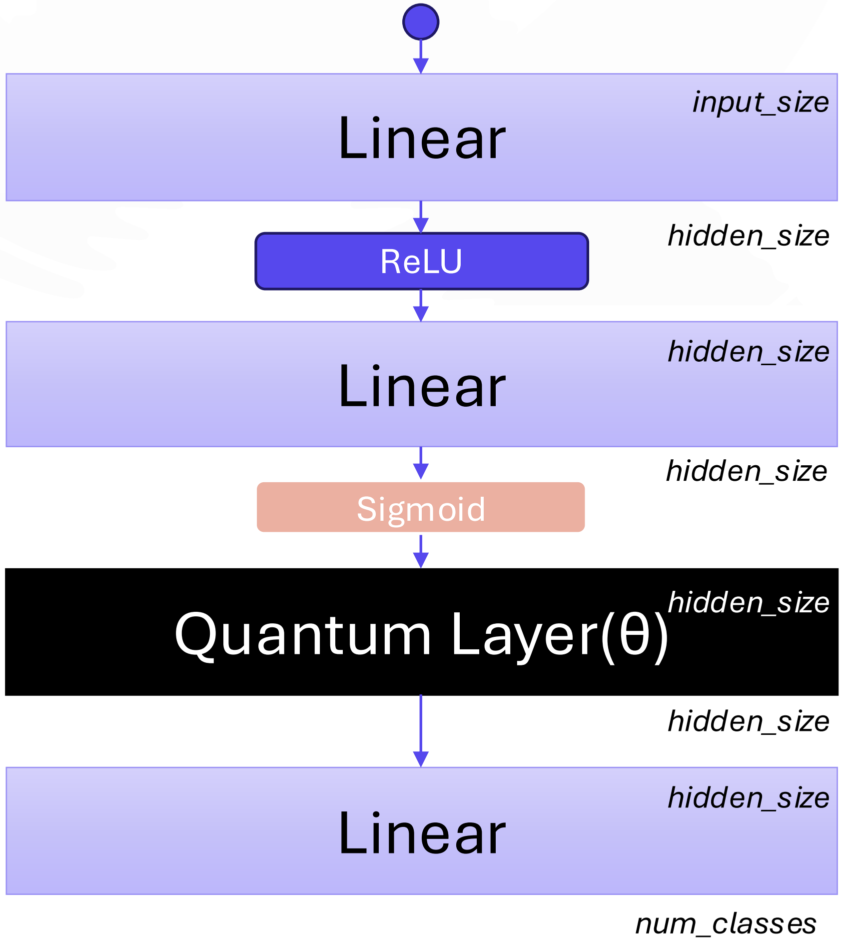
- Data encoding
  - Angle encoding
  - Amplitude encoding
- Measurement strategy
  - Full probability distribution as output
  - Expectation per mode
  - Complex amplitudes
- Computation space
  - Full Fock space
  - Encoded schemes

→ Allows us to choose between qubit-based or photon-native approaches



MerLin examples
A hybrid classifier

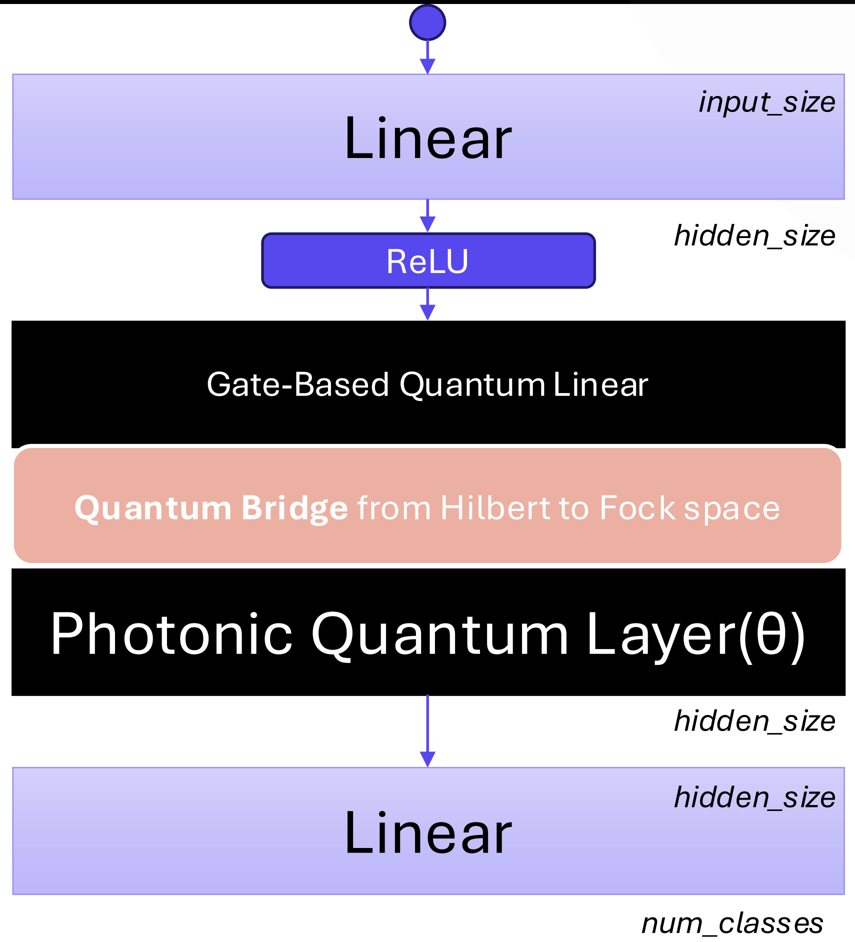
```
def create_quantum_classifier(input_size=10, hidden_size=16, num_classes=2):
    # Create a quantum circuit
    n_modes = 4
    circuit = pcvl.Circuit(n_modes)
    wl = pcvl.GenericInterferometer(n_modes, lambda i: pcvl.BS() // pcvl.PS(pcvl.P(f"theta{i}")))
    circuit.add(0, wl, merge=True)

    # Create the model with a quantum layer in the middle
    model = nn.Sequential(
        nn.Linear(input_size, hidden_size),
        nn.ReLU(),
        nn.Linear(hidden_size, 2),  # Compress to 2 features for quantum input
        nn.Sigmoid(),  # Scale to [0, 1] range
        QuantumLayer(
            input_size=2,
            output_size=hidden_size,
            circuit=circuit,
            trainable_parameters=["theta"],
            input_parameters=["x"],
            input_state=[1, 0, 1, 0],  # 2 photons in 4 modes,
            output_mapping_strategy=OutputMappingStrategy.LINEAR
        ),
        nn.Linear(hidden_size, num_classes)
    )
    return model
```

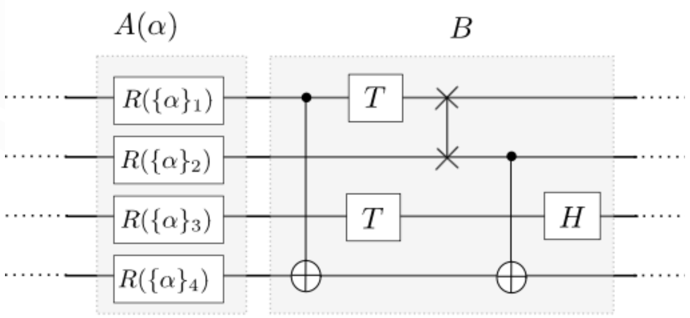


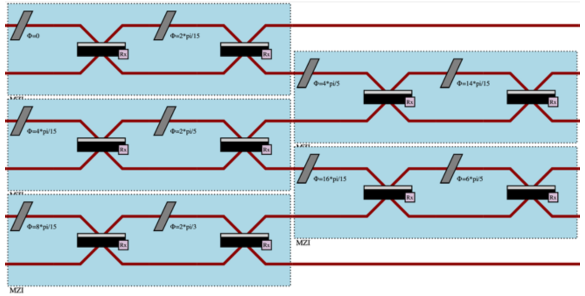
MerLin examples
With Cross-Platform Quantum Layers

Matter qubits

Photonic qubits







MerLin is also a benchmarking and paper reproduction platform

MerLin: A Discovery Engine for Photonic and Hybrid Quantum Machine Learning

Cassandre Notton*‖, Benjamin Stott†‖, Philippe Schoeb*‡, Anthony Walsh†, Grégoire Leboucher†§, Vincent Espitalier†, Vassilis Apostolou†, Louis-Félix Vigneux*¶, Alexia Salavrakos†, Jean Senellart†

Reproduced Papers

MerLin enables researchers to reproduce and build upon published quantum machine learning research. This section provides implementations of key papers in the quantum ML field, complete with working code, analysis, and extensions.

Overview

Each reproduction may include:

- **Original paper implementation** - Faithful recreation of the paper's methodology
- **Reproduction status** - Indicating whether the reproduction is partial or complete
- **Jupyter notebooks** - Interactive exploration of results and concepts
- **Full code** - Available on GitHub for easy access and modification
- **Performance analysis** - Comparison with paper results
- **Extension opportunities** - Ideas for building upon the work

All reproductions are implemented using MerLin's high-level API, making them accessible to ML practitioners without deep quantum expertise.

README

MerLin Reproduced Papers

About this repository

This repository contains implementations and resources for reproducing key quantum machine learning papers, with a focus on photonic and optical quantum computing.

It is part of the main MerLin project: https://github.com/merlinquantum/merlin and complements the online documentation available at:

https://merlinquantum.ai/research/reproduced_papers.html

Each paper reproduction is designed to be accessible, well-documented, and easy to extend. Contributions are welcome!

Papers reproduced:

Paper	Reproduction
Quantum Optical Reservoir Computing. Sources: munro_2024, rambach_2025, sakurai_simple_2025, lau_2025, sakurai_2025	- Scalability: increasing number of modes leads to better performance for the reservoir on MNIST. - In the original work and in our reproduction, we see a quantum boost with modest resources. - Our reproduction is QPU compliant.
Computational Advantage in Hybrid Quantum NeuralNetworks: Myth or Reality?. Source: kashif_2024	The original paper and our reproduction display that an HQNN model requires less parameters than a classical NN to achieve at least 90% accuracy on the noisy spiral dataset used that has a variable number of features (between 5 and 60).

MerLin is also a benchmarking and paper reproduction platform

MerLin: A Discovery Engine f... Hybrid Quantum Machi...

Cassandre Notton[*||], Benjamin Stott[†||], Philippe Schoeb[*‡], An...
Vincent Espitalier[†], Vassilis Apostolou[†], Louis-Félix Vigneux[*¶]...

You can do your own pull request and our team will review it!

Reproduced Papers

MerLin enables researchers to reproduce and build upon published quantum machine learning rese... papers in the quantum ML field, complete with working code, analysis, and extensions.

Overview

Each reproduction may include:

- **Original paper implementation** - Faithful recreation of the paper's methodology
- **Reproduction status** - Indicating whether the reproduction is partial or complete
- **Jupyter notebooks** - Interactive exploration of results and concepts
- **Full code** - Available on GitHub for easy access and modification
- **Performance analysis** - Comparison with paper results
- **Extension opportunities** - Ideas for building upon the work

All reproductions are implemented using MerLin's high-level API, making them accessible to ML practitioners without deep quantum expertise.

README

...ced Papers

...entations and resources for reproducing key quantum machine learning papers, ...tical quantum computing.

...ct: https://github.com/merlinquantum/merlin and complements the online

...h/reproduced_papers.html

...gned to be accessible, well-documented, and easy to extend. Contributions are

Papers reproduced:

Paper	Reproduction
Quantum Optical Reservoir Computing. Sources: munro_2024, rambach_2025, sakurai_simple_2025, lau_2025, sakurai_2025	- Scalability: increasing number of modes leads to better performance for the reservoir on MNIST. - In the original work and in our reproduction, we see a quantum boost with modest resources. - Our reproduction is QPU compliant.
Computational Advantage in Hybrid Quantum NeuralNetworks: Myth or Reality?. Source: kashif_2024	The original paper and our reproduction display that an HQNN model requires less parameters than a classical NN to achieve at least 90% accuracy on the noisy spiral dataset used that has a variable number of features (between 5 and 60).

Perceval

Simulation close to the hardware
Noise models
Circuit design

MerLin

Photonic QML framework
PyTorch integration
Built for AI/ML practicioners

Install Perceval and MerLin

Install Perceval: pip install perceval-quandela
Install MerLin: pip install merlinquantum

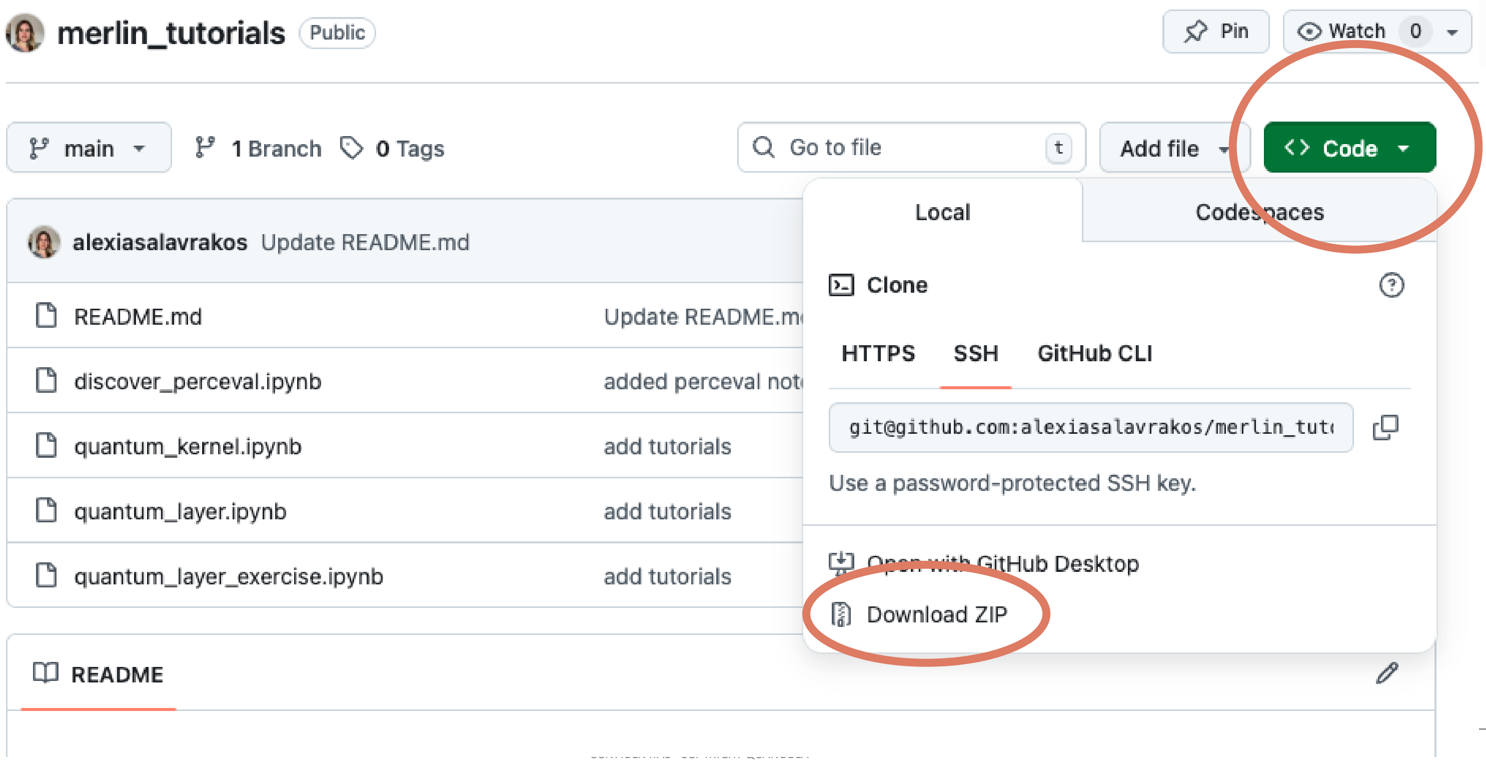
Try:
import perceval as pcvl
import merlin as ML

If you wish to run the notebooks with jupyter – if not, feel free to use any IDE:
pip install notebook
jupyter notebook

Find notebooks here:
https://github.com/alexiasalavrakos/merlin_tutorials

Install Perceval and MerLin

Find notebooks here:
https://github.com/alexiasalavrakos/merlin_tutorials



If you have trouble with the installation…

https://colab.research.google.com/drive/1SROabiDq_k9cNOQn1QfB5wsdKC8T1pcY?usp=sharing

Colab file → Save a copy to Drive

Perceval: https://tinyurl.com/2zdrusj4

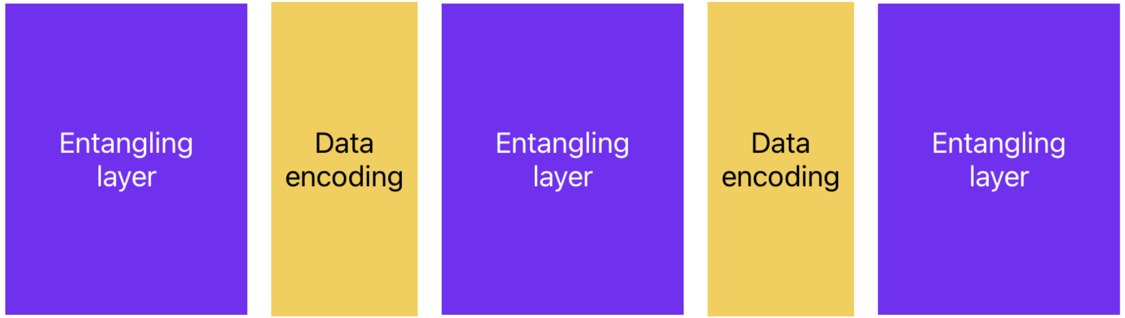
MerLin: https://tinyurl.com/46c9yhex

Coding exercise

- Now try to classify the make_moons dataset!

- What would you change in the model definition given this dataset?

- Can you implement a "data reuploading" scheme? Does it improve the performance?
  Hint: use np.hstack to reformat the data



- Feel free to play around with model structure, data encoding, measurement strategy, etc, to further improve performance