MerLin: A PyTorch-Integrated Framework for Scalable QML

FROM UTILITY TODAY TO QUANTUM ADVANTAGE TOMORROW

jean.senellart@quandela.com

HPC Quantum AI Conference → October 6st, 2025

About Quandela - We are focused in delivering cutting-edge solutions

→ We build full-stack Quantum Computers

2017
year of incorporation.
1st quantum computing startup in France

35 years
history of state-of-the art research

2018
first commercialized quantum device

140+
people

50+
people with PhD

55
research scientists

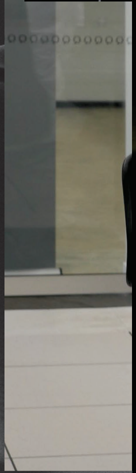
>40
patents & scientific articles

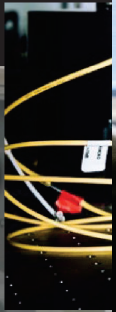
>30,000
citations for the lead scientists

2
Production facilities

4
main locations worldwide

ABOUT QUANDELA





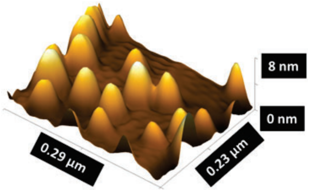
OVH cloud datacenter, Croix – France, 1st delivered QC system, November 2023.

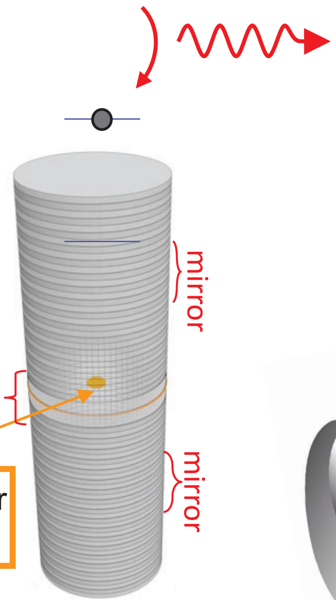
Anatomy of a photonic quantum computer

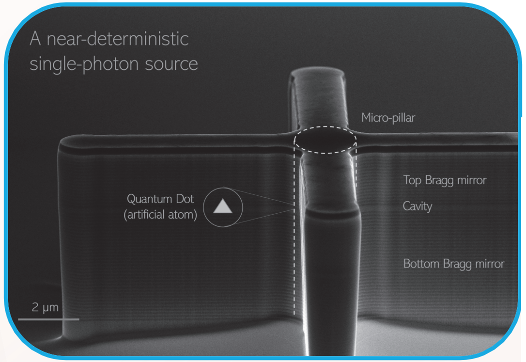
Turning Light into Qubits

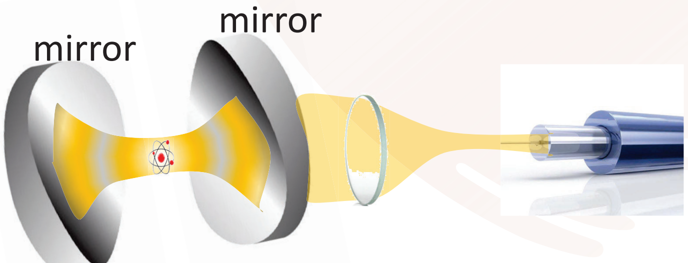
Quandela Quantum Dots

- Single photons are the building blocks of our quantum computer
- Now being fabricated in our semiconductor pilot line

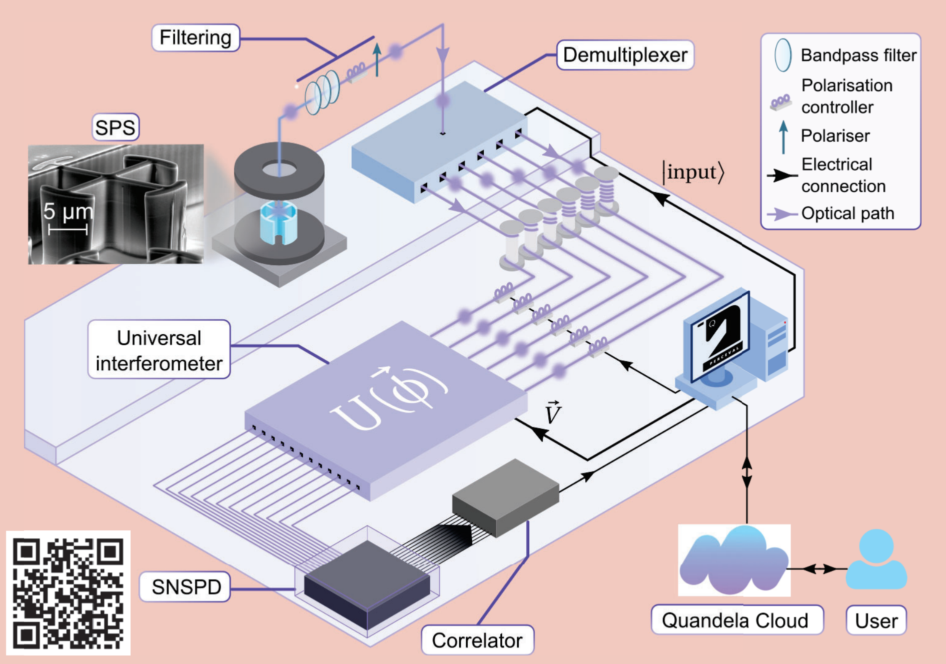


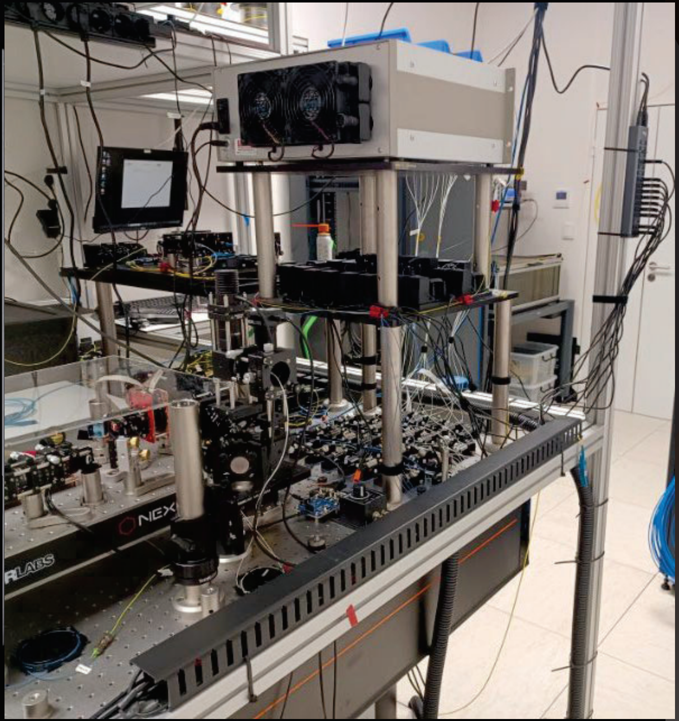






Ascella - the first photonic quantum processor available in the Quandela cloud



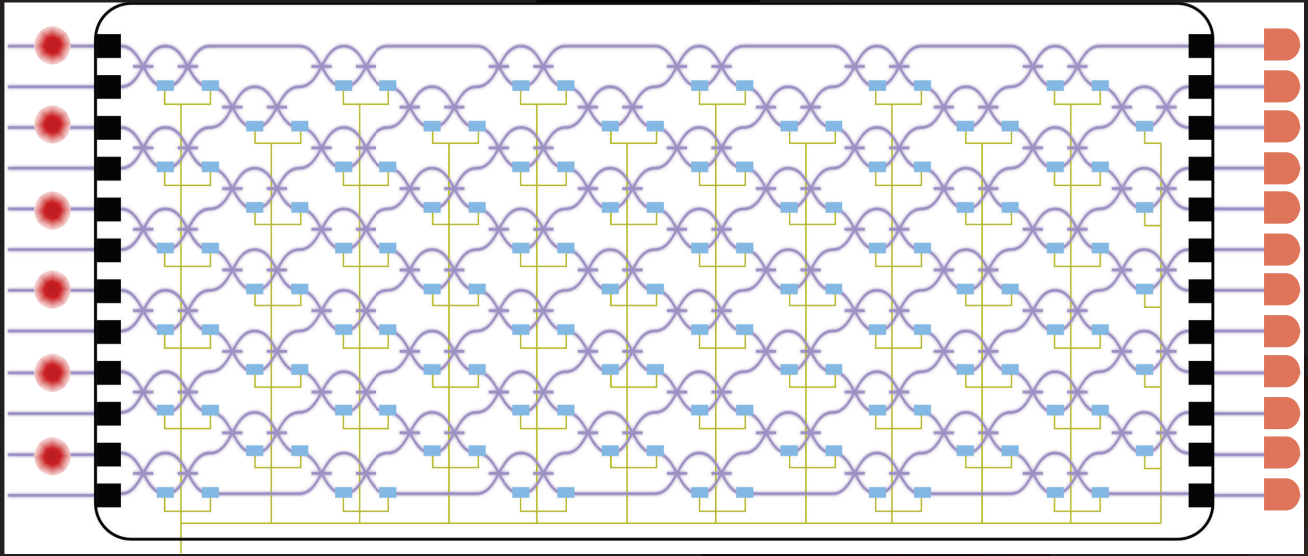


https://www.nature.com/articles/s41566-024-01403-4

Quantum Computing with photons

Starting with Boson Sampler

Generic interferometer representing a unitary transformation $U(x_i)$



Quantum Computing with photons

Starting with Boson Sampler

Time necessary to perform/simulate 1000 samples on n photons:

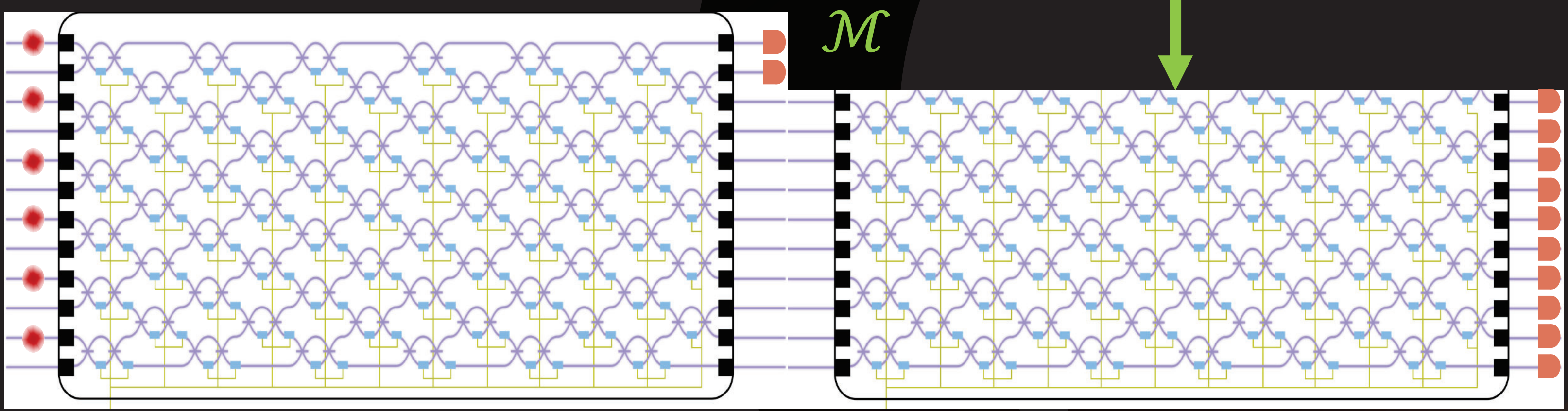
n	Number of operations per sample	High Performance Laptop	Jean Zay HPC #274 worldwide	1GHz QPU with 80% transmission	1GHz QPU with 90% transmission
4	64	**milliseconds**	**milliseconds**	**milliseconds**	**milliseconds**
10	10240	**milliseconds**	**milliseconds**	**milliseconds**	**milliseconds**
20	21M	**seconds**	**milliseconds**	**milliseconds**	**milliseconds**
30	32B	hour	**1s**	**milliseconds**	**milliseconds**
48	3.10^{15}	4 days	**100s**	**milliseconds**	**milliseconds**
80	10^{26}	-	95 years	1 hour	**1 second**

But competitive approximative algorithms !

Quantum Computing with photons

Boson sampling with Feedforward

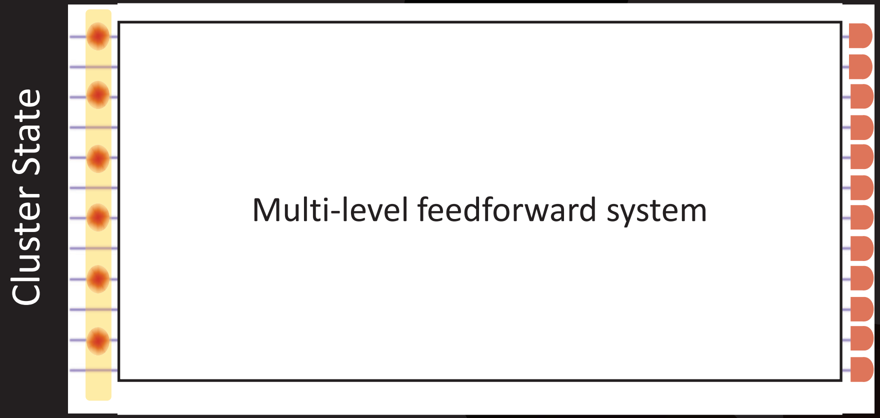
Dynamic Adaptation $U_2(x_i(\mathcal{M}))$



Provably harder to simulate

Quantum Computing with photons

Generalized Entangled Boson Sampler

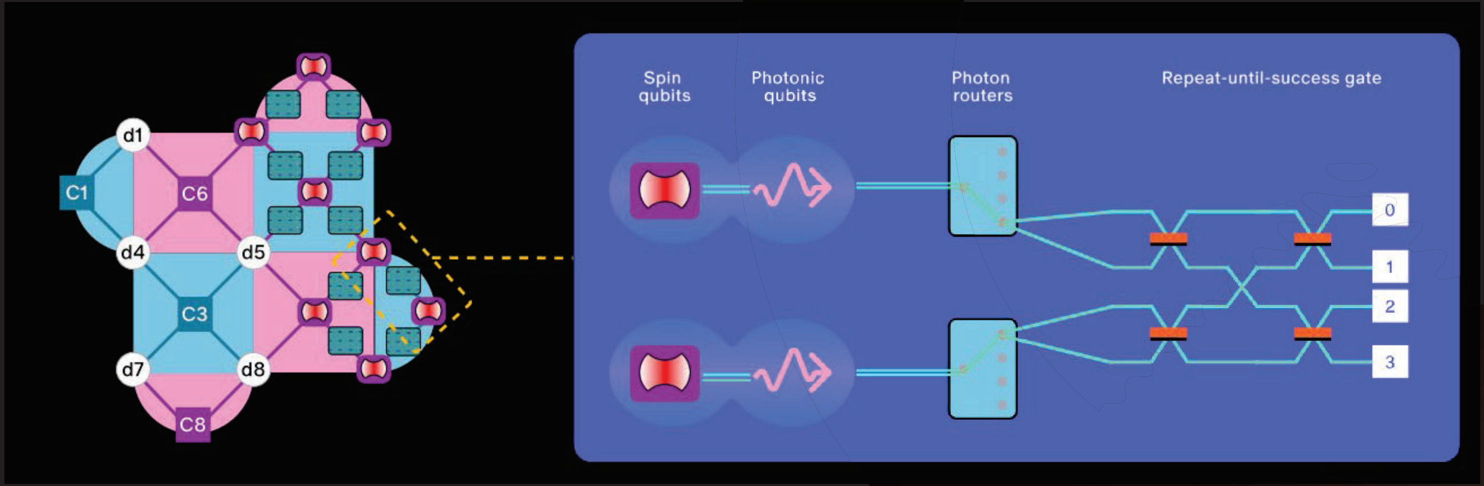


Provably even harder to simulate

Quantum Computing with photons

Spin Optical Quantum Computing

→ Hybrid approach using strengths of both photons and spin qubits



Universal Quantum Computing Scheme

From NISQ to Quantum Advantage
Today's Quantum Opportunity

- "If it's simulable, it has no utility."
- "Noise kills any chance of advantage."
- "Quantum is too slow to compete with GPUs."

NISQ

Small number of physical qubits

Noisy system

Short-time decoherence

UTILITY

Identifying useful applications outperforming brute force

FAULT TOLERANT

Large number of logical qubits

Error correction scheme

PROOF OF CONCEPT

USEFUL APPLICATIONS

ENERGETIC ADVANTAGE

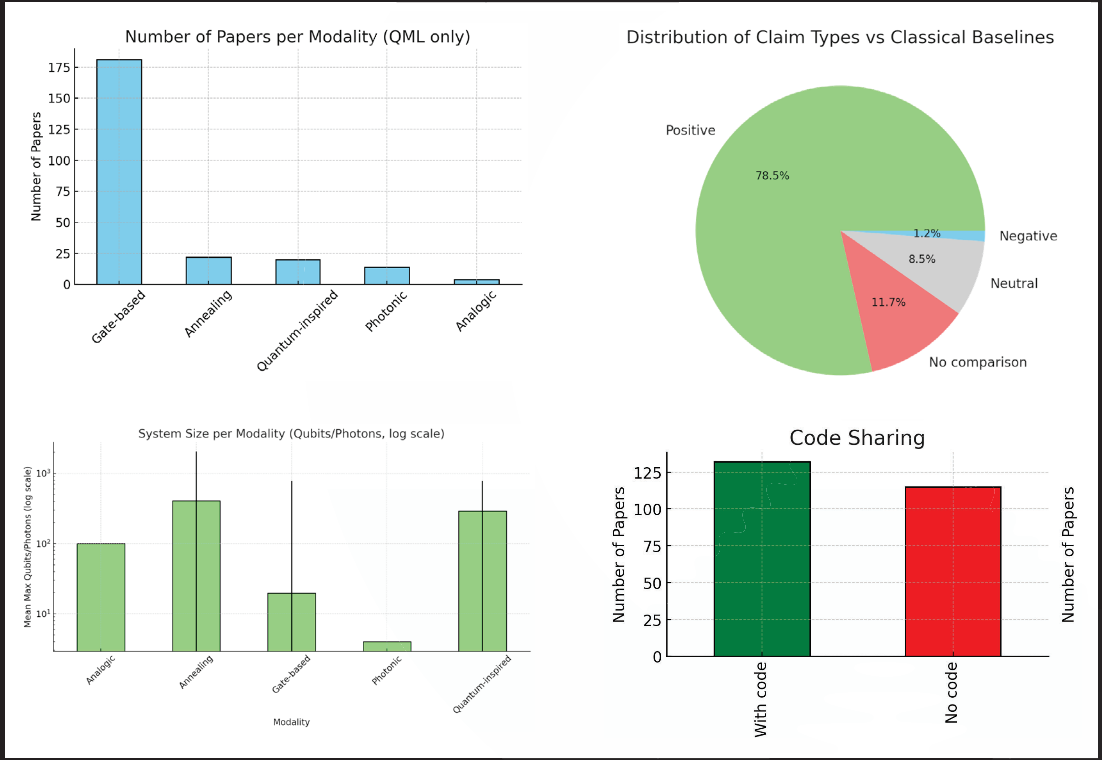
QUANTUM ADVANTAGE

First Generation of Quandela online computers

Intermediate computers

SPOQC Architecture

QML Landscape – over 10000 published papers



Thorough analysis on 250 papers (QML+MNIST)

Positive Claims

- **« Better performance »**
- **Comparable performance** *with better data/parameters budget.*

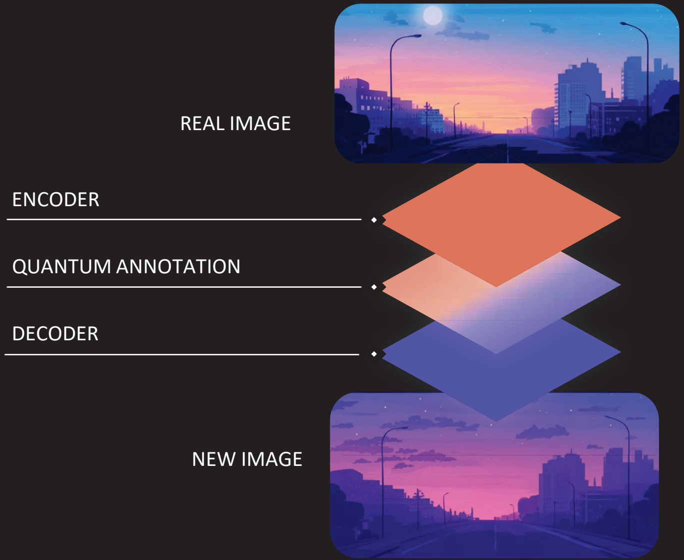
- **Qualitative Improvement:** *faster convergence, or qualitative benefits (e.g., interpretability).*

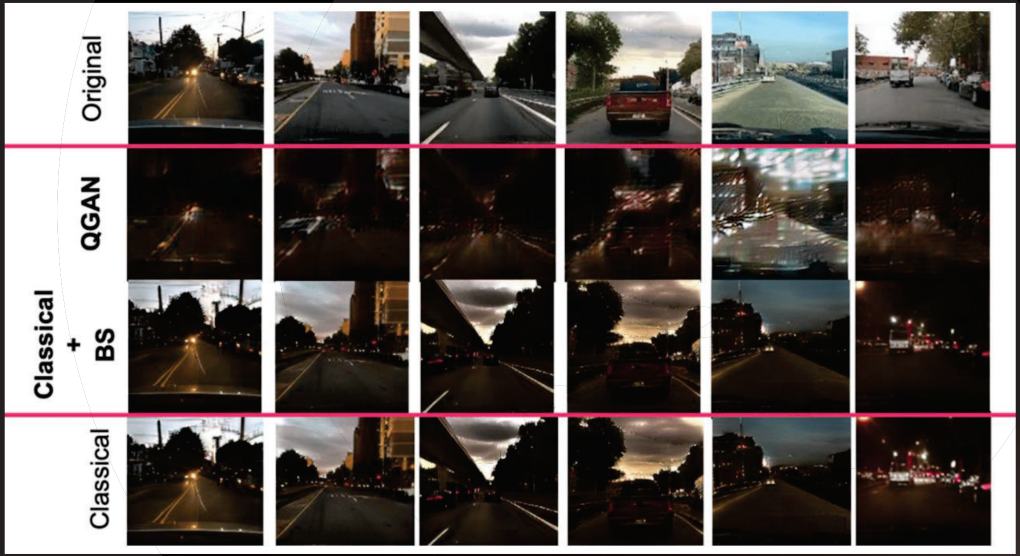
- **Robustness / stability**: Claims of more stable training, smoother learning curves, or improved resilience to noise.

Many claims, little reproducibility, hard to compare.

BMW GROUP AIRBUS

Proving the Value of Quantum ML – Airbus & BMW 2024 Challenge





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.

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

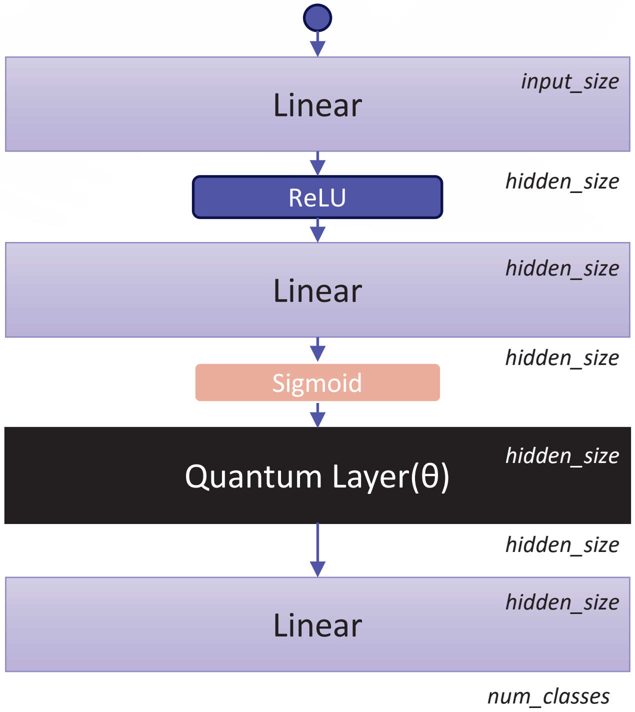
- Train **hybrid quantum–classical models** efficiently on GPUs
- Use familiar PyTorch APIs

3. Deploy on Hardware – QPU Ready

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

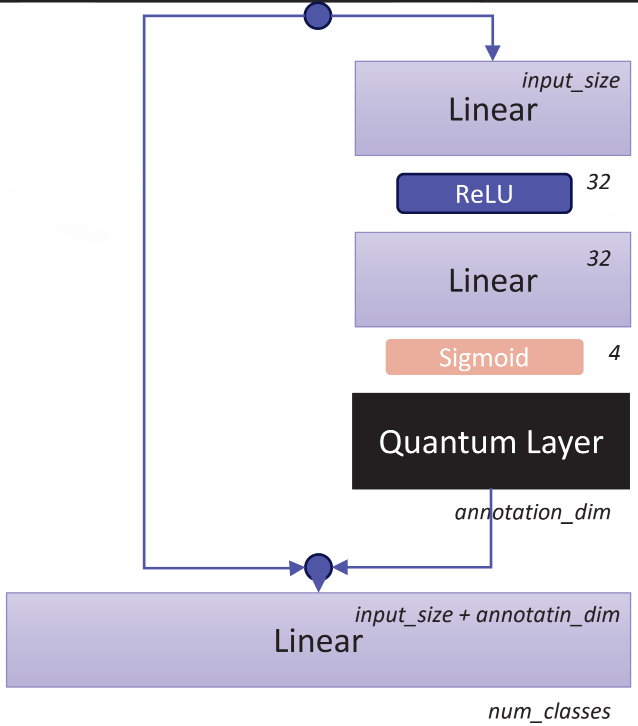
Example Hybrid Models
A hybrid classifier

```python
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
```



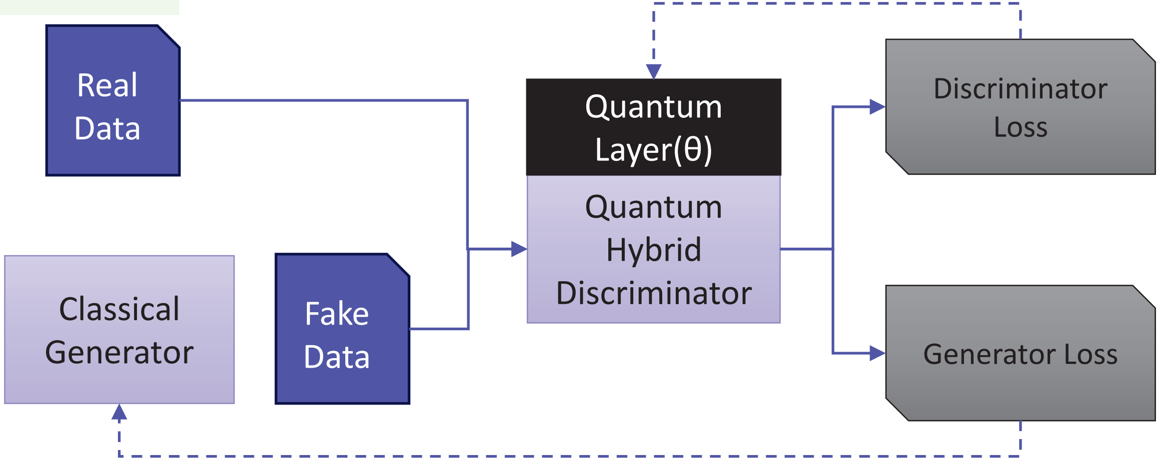
Example Hybrid Models
Quantum Annotation in a classical classifier

```python
# Initial feature compression
self.feature_compressor = nn.Sequential(
    nn.Linear(input_dim, 32),
    nn.ReLU(),
    nn.Linear(32, 4),
    nn.Sigmoid()  # Scale to [0, 1] for quantum input
)

# Quantum annotation layer
self.quantum_annotator = QuantumLayer(
    input_size=4,
    output_size=annotation_dim,
    circuit=circuit,
    input_parameters=["x"],
    input_state=input_state,
    output_mapping_strategy=OutputMappingStrategy.LINEAR
)

# Original path - processes raw input
self.original_path = nn.Sequential(
    nn.Linear(input_dim + annotation_dim, 64),
    nn.ReLU(),
    nn.Linear(64, num_classes)
)
```



Example Hybrid Models
GAN with a Quantum Discriminator

```
class QuantumGAN:
    def __init__(self, latent_dim=100, img_dim=28*28):
        self.latent_dim = latent_dim
        self.generator = ClassicalGenerator(latent_dim, img_dim)
        self.discriminator = QuantumDiscriminator(img_dim)

        # Setup optimizers
        self.g_optimizer = torch.optim.Adam(self.generator.parameters(), lr=0.0002)
        self.d_optimizer = torch.optim.Adam(self.discriminator.parameters(), lr=0.0002)
        self.criterion = nn.BCELoss()
```



Example Hybrid Models.
With Cross-Platform Quantum Layers

input_size

Linear

hidden_size

ReLU

Matter qubits

Gate-Based Quantum Linear

Quantum Bridge from Hilbert to Fock space

Photonic qubits

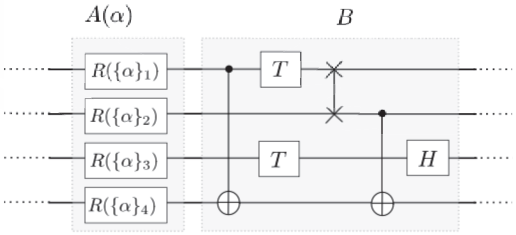
Photonic Quantum Layer(θ)

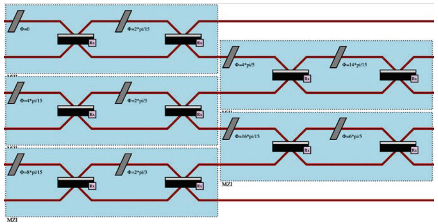
hidden_size

hidden_size

Linear

num_classes

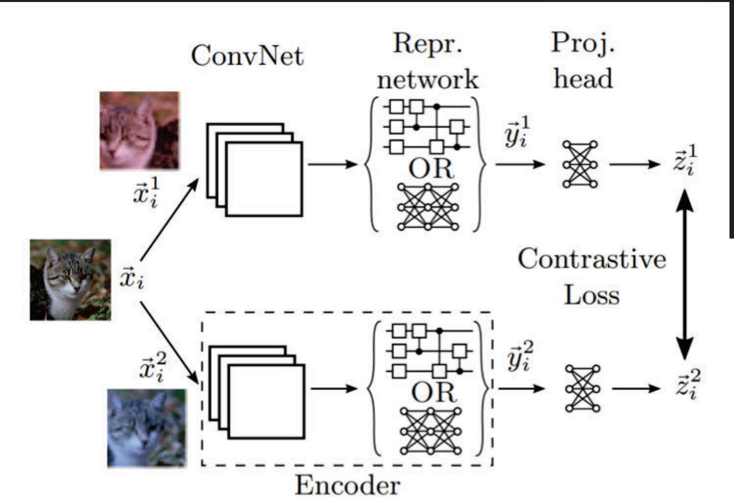


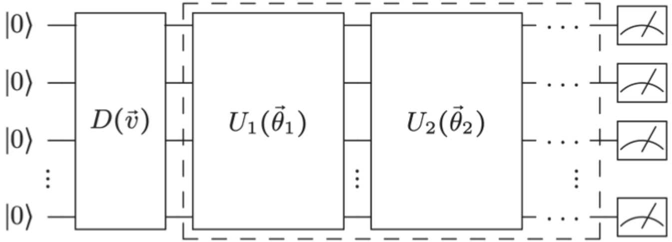


Example of Reproduced Papers

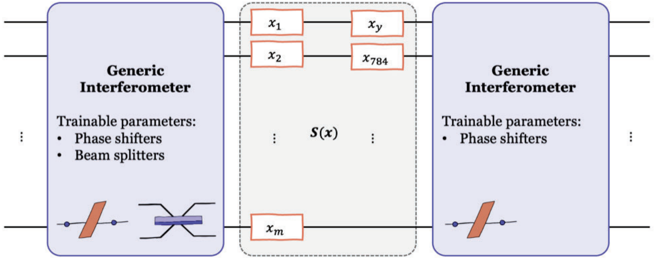
Quantum Self-Supervised Learning

B. Jaderberg,[1,*] L. W. Anderson,[1,*] W. Xie,[2] S. Albanie,[3] M. Kiffner,[1,4] and D. Jaksch[1,4,5]
[1] *Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, United Kingdom*
[2] *Visual Geometry Group, Department of Engineering Science, University of Oxford*
[3] *Department of Engineering, University of Cambridge*
[4] *Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, Singapore 117543*
[5] *Institut für Laserphysik, Universität Hamburg, 22761 Hamburg, Germany*





Gate-based Ansatz



Photonic Ansatz

https://github.com/merlinquantum/reproduced_papers/tree/main/qSSL

Quantum + AI + HPC: Unlocking the Next Steps

Quantum Today

- New **unconventional** quantum primitives **are available**
- The challenge: **discovering workflows where they bring real value**

Need of high-level, easy to integrate framework

- ❑ Experiment, benchmark, and refine *models*
- ❑ Simulator-first, **GPU training on HPC**, QPU testing
- ❑ Open to multiple quantum technologies to leverage complementary strengths

And AI-like dynamics

- ❑ **Benchmark-driven** progress, **open science**, **active ecosystem**
- ❑ Integration into **heterogeneous HPC** infrastructures

- ❑ Define what is "value"

QUANDELA

→ https://cloud.quandela.com

→ https://merlinquantum.ai

Thank you!